

Cell-centered indirect Arbitrary Lagrangian-Eulerian numerical strategy for solving 3D gas dynamics equations

S. Guisset^{1,2}, G. Damour¹, J. Breil¹

Abstract

Solving the Euler equations under the Lagrangian formalism enables to simulate various complex engineering applications. However, the use of this formalism can lead to significant mesh deformations as the mesh follows the fluid velocity. The mesh quality may be considerably deteriorated requiring a regularization procedure. In the present document, it is shown that the ideas presented by *J. Yao (2013)* [68] and *J. Yao and D. Stillman (2016)* [30] may be used efficiently in a 3D hydrodynamics code to perform reliable regularization steps. The flexibility of the methodology in addition to its simplicity, since it only relies on trivial geometrical considerations, opens the way for various extensions. The remapping step then considered, uses the geometrical splitting procedure of the Lagrangian phase to perform effective projections. This coherence ensures the compactness of the overall algorithm. At last but not least, a 3D Flux-Corrected-Remapping method is presented. This yields a particularly robust remapping algorithm while also leading the way for higher order projection extensions.

Key words: Arbitrary Lagrangian-Eulerian hydrodynamics, mesh regularization, Flux-Corrected-Remapping, multi-dimensional.

1 Introduction

The arbitrary Lagrangian-Eulerian (ALE) method was first proposed in [26] to solve time-dependent fluid dynamics problems. While in the Lagrangian

¹CEA/CESTA, 15 Avenue des Sablières CS 60001, 33116 Le Barp cedex, France

²Corresponding author: sebastien.guisset@cea.fr

framework the grid follows the flow, the ALE approach enables the use of a grid which follows a given velocity field. This yields robust and accurate numerical simulations. ALE algorithms are now widely used for the simulation of multi-material fluid flows especially in the presence of large mesh deformation [2, 3, 48, 27, 5, 19, 29]. Standard ALE methods can be organized in two categories. The first one, often referred as “direct ALE” in the literature consists in solving directly the fluid equation attached to the moving mesh [49, 25]. Notice here that all convective terms are integrated into the numerical resolution so the “unsplit” terminology is also used. We refer to [62, 63] for investigation of the subject. The second category, often referred as “Lagrangian plus Remapping” or indirect ALE methods, is the one of interests in the present study. It consists in three steps. First of all, an explicit Lagrangian phase is considered. In this phase, the computational grid follows the fluid velocity. Here the mesh regularity may be deteriorated bringing the need for a regularization phase (rezoning phase). In this second step, the positions of the nodes of the Lagrangian grid are modified in order to improve the mesh regularity. Finally, the numerical solutions is remapped onto the new regular grid. This remapping phase is sensitive and should be also carefully addressed [43].

Over the years, several families of numerical schemes have been designed to solve the Lagrangian hydrodynamic equations. First of all, “staggered schemes” consider the kinematic variables (nodes velocity and nodes position) at the nodes while the usual thermodynamics quantities are evaluated at the center of the cells. This scheme follows the pioneer work of [61, 65], have been largely extended [10, 36, 47, 58, 9, 14, 64, 8] and are used for many engineer applications. The second family consist of cell-centered finite volume schemes. The schemes are inspired from the work of Godunov [23] and consider cell-centered quantities while the nodes velocities are obtained with a nodal solver. The 1D version based on an underlying approximate Riemann solver (HLLC type solver), is conservative and coherent with the second law of thermodynamics. Multi-directional schemes have then been proposed [15, 1, 40] with various accuracy enhancement methods [7, 11, 28, 38, 41]. We point out here that finite element methods have also been investigated for solving Lagrangian hydrodynamics equations. Here, high-order extensions are more straightforward since high order interpolation functions may be used [16, 17, 56, 55].

This study starts from the cell-centered 3D Lagrangian scheme presented in [22]. In this work, the gas dynamic equations are solved under the La-

grangian formalism to simulate compressible flows. A 3D symmetric geometrical decomposition of polyhedral cells was proposed. More precisely a supplementary point is added on each face whose kinematics is solved by choosing a barycenter formula of the node velocities. This geometrical splitting is interesting since it enables to prove the geometric conservation law (GCL). Second order extensions have also been proposed [22]. This achievement constitutes the first step the present work relies on. As a matter of fact, this numerical scheme is used to solve the Lagrangian phase of our 3D ALE algorithm.

The regularization procedure is a challenging key step in the indirect ALE strategy. While the first goal consists in improving the mesh quality, one also wishes to remain close to the Lagrangian mesh since it contains attractive features of the flow [33]. Here, one understands the need for a balance between these two “sometimes opposite” objectives. In the present work, we do not change the connectivity and simply change the coordinates of existing nodes. In this simplified framework (fixed topology), regularization step often consists in solving optimization problems in which a functional needs to be minimized. This functional may be made more or less complex (so the associated minimization problem) if strong requirements on the mesh are to be enforced (orthogonality, complex geometries...). Early ideas proposed by Winslow [66, 67] are still popular today. Various extensions have been proposed over the years as for example [34, 32]. Among standard mesh regularization methods are Laplacian-type algorithms [37]. In this case the mesh velocity is directly modified. We also mention force-based methods [45, 69] where the (equilibrium) mesh coordinates and the associated node velocities are recomputed. It should be noticed here, that many mesh regularization techniques are largely inspired by the mesh generation community since many encountered issues are similar.

Concerning the remapping step in 3D, different approaches are found in the literature. We mention for example [24] in which a geometrical intersection strategy is presented for polyhedral meshes. In [21], a linearity and bound preserving method for polyhedral meshes is proposed. This algorithm is based on reconstruction, approximate integration and conservative redistribution technics. In [35] a conservative scheme for remapping high-order discontinuous Galerkin fields on polytopal meshes with curved faces is studied. This strategy uses a virtual element function to define the remapping velocity. In the present study a 3D sweeping technique is used extensively. This method, presented in [44], is restricted to grids with the same connec-

tivity and is particularly popular since it does not necessitate to compute expensive intersections between two grids. This approach may be seen as a compatibility exercise in which the volume integral over a new cell is obtained as the integral of the old cell plus the contribution of the region swept by the displacement of the cell faces (starting from the old to the new positions). This approach is preferred to the cell-intersection based method which is particularly expensive in this 3D setting. We refer to [60] for a discussion on this subject. The parallelization procedure, which is mandatory for this 3D context, has been performed with MPI but is not detailed in this paper.

We believe the originality of the present work comes from various reasons. First of all, it is shown that ideas presented in [30, 68], may be used and extended, inside a 3D hydrodynamics code, to perform reliable regularization steps. We point out here that the literature dealing with ALE regularization methods in 3D is “relatively” recent. Here we mention the code ALEGRA [54] developed at Sandia National Laboratories. In [46] a 3D point-centered hydrodynamics algorithm was used on tetrahedral meshes with interesting rates of convergence. In [51] different algorithms are considered to compute subzonal corner masses of a staggered compatible Lagrangian discretization. In [13] a 3D finite element staggered method is studied. Here we mention [12], in which a 3D cell-centered numerical scheme is used in combination to a Flux-Corrected-Transport (FCT) remap for the simulation of multiphase flows. We also mention [18] where in the context of multi-material applications, tools for high-order mesh optimization are used. The philosophy presented here is strongly different with the ones traditionally used and previously mentioned. The strengths of this approach are numerous since the method only relies on trivial geometrical considerations. Its simplicity, efficiency and suitability for parallel computing and complex geometries makes this regularization process particularly attractive. In our opinion, this regularization technique also opens the way for various extensions. It should be stated here, that, for the time being, this method is limited to block-structured meshes. This constitutes the main limitation of the present method. Extending it to fully unstructured meshes is challenging and this aspect will not be discussed here. This point should be investigated properly in a different study. However, we believe that most engineer applications can be handled by working with block-structured meshes and it will be shown that the present geometrical method may already be used efficiently as regularization tool in a large 3D ALE code. This constitutes one of the main goals of this study. The remapping step presented in this article is also original

since it uses the geometrical splitting procedure of the Lagrangian phase to perform efficient projections. This coherence between the Lagrangian and remapping phases ensures the compactness of the overall algorithm. At last but not least, a 3D Flux-Corrected-Remapping method is presented. This yields a particularly robust remapping algorithm while also leading the way for higher order projection extensions.

The document is organized as follows. Firstly, the governing equations with the overall numerical strategy are detailed. In particular, the general ALE philosophy considered is recalled. It is based on an indirect strategy in which the Lagrangian phase is followed by a regularization step plus a remapping procedure. In a second part, the 3D Lagrangian phase which is taken from [22] is recalled. Indeed, the geometrical splitting chosen will be broadly used for the remapping step. In a third part, the regularization method is presented. This method extent the ideas proposed in [30, 68] into an ALE context. In particular, it is shown that this regularization methodology which is purely geometrical enables the regularization of strongly deformed meshes. In this part, some regularization examples are displayed to show the capability of the method. In this case no hydrodynamics motions occurs but this point out the flexibility of the method which handles the regularization of very strongly deformed meshes. In a fourth section, the remapping step is explained. More precisely, the principles of the 3D sweeping strategy are detailed. Then, in order to improve the robustness of the method while keeping a good level of accuracy the 3D Flux-Corrected-Remapping (FCR) method is presented. This method strongly relies of the fact that low order remapped conservative quantities (first order remapping here) satisfies some robustness conditions. Finally, numerous 3D test cases are carried out to assess the behavior of the overall ALE numerical method. Numerical comparisons are presented with the Lagrangian code (in this case there is no regularization involved) and an Eulerian one (in this case the remapping step is performed at the end of each time iteration on the initial grid). Among the different test cases presented, some are particularly difficult for the Lagrangian code (which sometimes crashes) and demonstrate the interest and robustness of the proposed ALE strategy.

2 Governing equations and overall numerical strategy

Let $\omega(t) \in \mathbb{R}^3$ be a region filled with a fluid and $\partial\omega(t)$ denotes its boundary. For a fluid of density ρ , velocity \mathbf{V} , pressure P and total energy E , consider the unsteady compressible Euler equations for an arbitrary moving control volume moving with velocity \mathbf{V}^{ALE}

$$\begin{aligned}
 \frac{d}{dt} \int_{\omega(t)} \rho \, dv + \int_{\partial\omega(t)} \rho (\mathbf{V} - \mathbf{V}^{ALE}) \cdot \mathbf{n} \, ds &= 0, \\
 \frac{d}{dt} \int_{\omega(t)} \rho \mathbf{V} \, dv + \int_{\partial\omega(t)} (\rho (\mathbf{V} - \mathbf{V}^{ALE}) \cdot \mathbf{n} \mathbf{V} + P \mathbf{n}) \, ds &= \mathbf{0}, \\
 \frac{d}{dt} \int_{\omega(t)} \rho E \, dv + \int_{\partial\omega(t)} (\rho E (\mathbf{V} - \mathbf{V}^{ALE}) + P \mathbf{V}) \cdot \mathbf{n} \, ds &= 0, \\
 \frac{d}{dt} \int_{\omega(t)} dv - \int_{\partial\omega(t)} \mathbf{V}^{ALE} \cdot \mathbf{n} \, ds &= 0,
 \end{aligned} \tag{1}$$

where \mathbf{n} denotes the unit outward normal of the volume $\partial\omega(t)$. These set of equations is often referred as the Arbitrary Lagrangian-Eulerian (ALE) integral form of the Euler equations [26, 20] and respectively corresponds to the mass conservation, the momentum conservation and the total energy conservation. The last equation expresses a volume conservation and is equivalent to the following trajectory relation

$$\begin{cases} \frac{d}{dt} \mathbf{x}^{ALE} = \mathbf{V}^{ALE}, \\ \mathbf{x}^{ALE}(t=0) = \mathbf{x}_0^{ALE}, \end{cases}$$

where \mathbf{x}^{ALE} defines the position of a node on the control volume surface while \mathbf{x}_0^{ALE} stands for its initial position. This governing set of equations needs to be closed by using an equation of state. In the following, a closure under the form $P = P(\rho, \varepsilon)$ will be considered, where the internal energy ε is defined by $\varepsilon = E - \frac{1}{2} \mathbf{V}^2$. Remark here that in the peculiar case $\mathbf{V}^{ALE} = \mathbf{V}$, a standard Lagrangian formalism (the control region follows the fluid velocity) is recovered while for $\mathbf{V}^{ALE} = \mathbf{0}$, the usual Euler description is obtained. In the following the numerical resolution of the set of equations (1) proposed is based on a indirect ALE strategy. More precisely, the first ingredient consists in solving the set of equations (1) in the Lagrangian framework (in this

case $\mathbf{V}^{ALE} = \mathbf{V}$). During this process the mesh follows the fluid velocity. Despite the great properties obtained using this Lagrangian approach, the mesh quality may be strongly distorted requiring a regularization procedure. Here starts a rezoning step in which, from the nodes moved during the Lagrangian phase, a new regularized grid is built up. Finally, a remapping procedure is applied to project all the conservative quantities onto the new regularized one. The overall strategy indirect ALE strategy is displayed in Figure 1. During the Lagrangian step the mesh quality may be considerably deteriorated. The regularization step however, intends to improve it while keeping some interesting features of the Lagrangian phase.

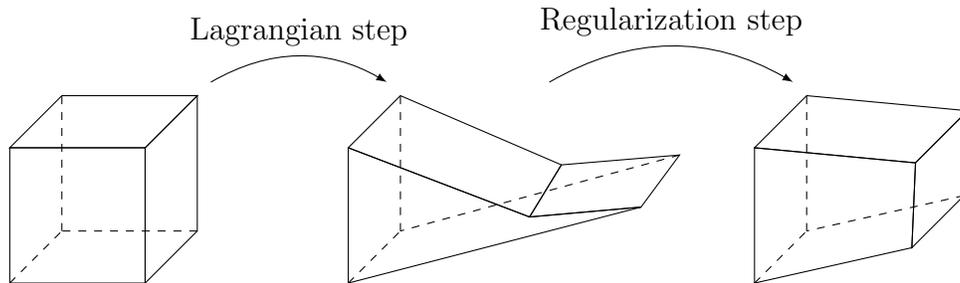


Figure 1: Example of a possible geometrical cell deformation occurring during the Lagrangian phase and its regularization.

3 3D Lagrangian step

3.1 Mesh notations and geometrical splitting

In order to explain in details the overall numerical procedure some mesh notations are now introduced. Here we follow the notations introduced in [22]. The spatial domain of interest denoted $\omega(t)$ is paved with non-overlapping polyhedrons denoted ω_c such that $\omega(t) = \bigcup_c \omega_c$. A polyhedron is defined as a volume bounded by polygonal faces. In this 3D context, the faces may not be planar, therefore the definitions of outward normal and area is not straightforward. In [22], the cells are subdivided by adding one point p_f^* for each face. This choice is displayed in Figure 2 and is significant for the derivation of the ALE strategy. First of all, as pointed out in [22], it enables to prove the GCL property for the 3D Lagrangian phase (discrete compatibility relation

between mass conservation and mesh geometry). Furthermore, this choice enforces a correct discrete symmetry preservation of the flow. In addition, as it will be explained in the subsequent sections, this face splitting is particularly convenient for deriving the remapping procedure in 3D configurations. Concerning the notations introduced in [22], the index c stands for a cell ω_c , f refers to a face and p to a node. The triangles created by the the face splitting are denoted t_r . For completeness, we recall here all the notations used

- $\mathcal{P}(c)$ is the set of nodes p of cell c without the nodes p_f^* ,
- $\mathcal{P}(f)$ is the set of nodes p of face f without the node p_f^* ,
- $\mathcal{F}(c, p)$ is the set of faces f of cell c and sharing point p ,
- $\mathcal{C}(p)$ is the set of cells c' sharing node p ,
- $\mathcal{T}(c)$ is the set of all the triangles t_r resulting from the splitting of the faces of cell c ,
- $\mathcal{T}(c, f)$ is the set of triangles t_r resulting from the splitting of the face f of cell c ,
- $\mathcal{T}(c, f, p)$ is the set of triangles t_r resulting from the splitting of the face f of cell c and sharing point p .

3.2 3D Lagrangian scheme

The Lagrangian step considered is directly taken from [22]. Recall here that in a Lagrangian scheme, the mass conservation equation imposes the cell mass m_c to be constant. In this framework the semi-discrete momentum and total energy conservation equations writes

$$\begin{aligned}
 m_c \frac{d\mathbf{V}_c}{dt} + \sum_{p \in \mathcal{P}(c)} \sum_{f \in \mathcal{F}(c, p)} S_{pf} P_{c,fp} \mathbf{n}_{pf} &= \mathbf{0}, \\
 m_c \frac{dE_c}{dt} + \sum_{p \in \mathcal{P}(c)} \sum_{f \in \mathcal{F}(c, p)} S_{pf} P_{c,fp} \mathbf{V}_p \cdot \mathbf{n}_{pf} &= 0,
 \end{aligned} \tag{2}$$

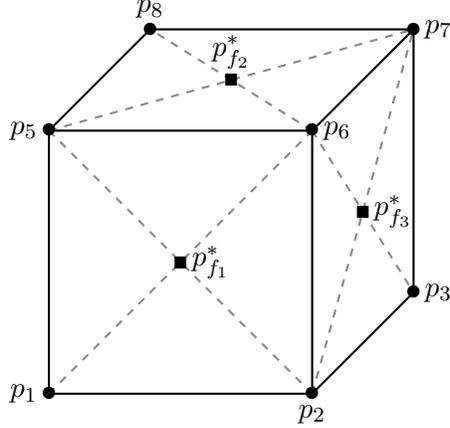


Figure 2: Taken from [22]. Result of the splitting of the cell faces into triangles using the face barycenter p_f^* . Simple case of a hexahedral cell with square faces.

where the index c denotes that the quantity has been mass averaged over the considered cell. In this Lagrangian step, the semi-discrete trajectory equation simplifies into

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{V}_p. \quad (3)$$

The nodal fluxes P_{cfp} and \mathbf{V}_p are the remaining unknowns to be determined. Following [39] in order to ensure a positive entropy production, the pressure jumps may be written in terms of the velocity jump as follow

$$P_{cfp} - P_c = Z_c (\mathbf{V}_c - \mathbf{V}_p) \cdot \mathbf{n}_{pf}, \quad (4)$$

where $Z_c = (\rho a)_c$ defines the acoustic impedance inside cell c and $a = \sqrt{(dP/d\rho)_\eta}$ the speed of sound [40]. Considering the total momentum and energy conservation [42], the nodal velocity \mathbf{V}_p may be computed by imposing a momentum balance around the node p . In this case, the node velocity \mathbf{V}_p is defined by

$$\mathbb{M}_p \mathbf{V}_p = \mathbf{B}, \quad (5)$$

where

$$\begin{aligned}\mathbb{M}_p &= \sum_{c \in \mathcal{C}(p)} \sum_{f \in \mathcal{F}(c,p)} S_{pf} Z_c (\mathbf{n}_{pf} \otimes \mathbf{n}_{pf}), \\ \mathbf{B} &= \sum_{c \in \mathcal{C}(p)} \sum_{f \in \mathcal{F}(c,p)} S_{pf} \left[P_c \mathbf{n}_{pf} + Z_c (\mathbf{n}_{pf} \otimes \mathbf{n}_{pf}) \mathbf{V}_c \right].\end{aligned}$$

Recall here that the matrix \mathbb{M}_p is positive definite thus invertible therefore the nodal velocity is easily computed. To compute the evolution of the remaining node p_f^* its velocity $\mathbf{V}_{p_f^*}$ is required. In practice, assuming linear velocity fields (with respect to \mathbf{x}) over the face leads to define $\mathbf{V}_{p_f^*}$ as the barycenter of the face vertices velocity

$$\mathbf{V}_{p_f^*} = \frac{1}{N_{p,f}} \sum_{q \in \mathcal{P}(f)} \mathbf{V}_q,$$

where $N_{p,f}$ is the number of nodes on the face f (without p_f^*). For completeness, we recall the definition of the face area vector $S_{pf} \mathbf{n}_{pf}$

$$S_{pf} \mathbf{n}_{pf} = \frac{1}{3} \left(\sum_{t_r \in \mathcal{T}(c,f,p)} S_{t_r} \mathbf{n}_{t_r} + \sum_{t_r \in \mathcal{T}(c,f)} \frac{1}{N_{p,f}} S_{t_r} \mathbf{n}_{t_r} \right), \quad (6)$$

and refers to [22] for its importance in enforcing the GCL condition. Finally, the $S_{pf} \mathbf{n}_{pf}$ term is called face area vector may be seen as the contribution of face f to the corner area vector [15] where the corner area vector writes $\mathbf{n}_p = \sum_{f \in \mathcal{F}(c,p)} S_{pf} \mathbf{n}_{pf}$.

4 Regularization step

In this section, the regularization method considered in the present document is presented in details. This approach is largely inspired by the Line-Sweeping regularization method described in [68]. The Line-Sweeping regularization method consists in a simple local iterative geometrical process. More precisely, each node is moved according to the position of its neighbors. It should be stated here, that this method is limited to block-structured meshes. Despite its great qualities (such as simplicity, efficiency and suitable for parallel computing and complex geometries), this is the main limitation of the present

method. Extending it to fully unstructured meshes is challenging and this aspect will not be discussed here. This point should be investigated properly in a different study. However, many engineering applications simulations may be performed by working with block-structured meshes and it will be shown in the next sections that the present geometrical method may already be used efficiently as regularization tool in a large 3D ALE code. This constitutes the main goal of this study. Now, since the work is devoted to 3D block-structured meshes, the indexes (i, j, k) are introduced to represent the three dimensions of space. A 2D stencil is a set of nodes obtained when one dimension index is i, j or k fixed. A 1D stencil is a set of nodes obtained if two dimensions indexes (i, j) , (i, k) or (j, k) are fixed. This philosophy is followed to extend the 1D method to multi-dimensional stencils. The multi-dimensional method may then be seen as a succession of one-dimensional stencil regularization steps.

4.1 3D line sweeping regularization

For a given 1D stencil, the regularized position (new point) computed at iteration $m + 1$ of a given node \mathbf{x}_i^m (displayed in blue in Figure 3), it is sufficient to fix the new node \mathbf{x}_i^{m+1} (displayed in red) at equal distance from the neighboring nodes \mathbf{x}_{i-1}^m and \mathbf{x}_{i+1}^m while following the geometry of the initial stencil. More precisely, after computing the length l^m of the branch (here the cell-index i is omitted in the length notations for more clarity)

$$l^m = l_1^m + l_2^m, \quad l_1^m = |\mathbf{x}_i^m - \mathbf{x}_{i-1}^m|, \quad l_2^m = |\mathbf{x}_i^m - \mathbf{x}_{i+1}^m|,$$

the regularized point \mathbf{x}_i^{m+1} (new point) named “equal-space-point” is then defined as follows

$$\mathbf{x}_i^{m+1} = \left(\mathbf{x}_{i-1}^m + \frac{l^m}{2} \mathbf{e}_1 \right) \delta_{\{\frac{l^m}{2} \leq l_1^m\}} + \left(\mathbf{x}_{i+1}^m + \frac{l^m}{2} \mathbf{e}_2 \right) \delta_{\{\frac{l^m}{2} < l_2^m\}},$$

where

$$\mathbf{e}_1 = \frac{\mathbf{x}_i^m - \mathbf{x}_{i-1}^m}{|\mathbf{x}_i^m - \mathbf{x}_{i-1}^m|}, \quad \mathbf{e}_2 = \frac{\mathbf{x}_i^m - \mathbf{x}_{i+1}^m}{|\mathbf{x}_i^m - \mathbf{x}_{i+1}^m|},$$

and

$$\delta_{\{\frac{l^m}{2} \leq l_1\}} = \begin{cases} 1 & \text{if } \frac{l^m}{2} \leq l_1^m \\ 0 & \text{otherwise} \end{cases}, \quad \delta_{\{\frac{l^m}{2} < l_2\}} = \begin{cases} 1 & \text{if } \frac{l^m}{2} < l_2^m \\ 0 & \text{otherwise} \end{cases}.$$

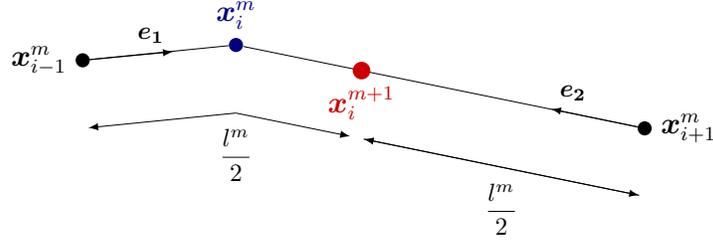


Figure 3: Regularization of a 1D stencil (1D equal-space-point method). The new node \mathbf{x}_i^{m+1} (red) is fixed at equal distance from the neighboring nodes \mathbf{x}_{i-1}^m and \mathbf{x}_{i+1}^m while following the geometry of the initial stencil.

We emphasize that the new computed point has the advantage to be on the initial geometry of the stencil. It is not the case when working with a simple arithmetic average of the nodes \mathbf{x}_{i-1}^m and \mathbf{x}_{i+1}^m . This simple but key idea enables to work with complex geometries (cylindrical geometry for example). Here, we choose to drop the regularization iteration index m for clarity. In the case of a 2D stencil, each triplet of points forms a 1D stencil and may be used to compute a new point fixed at equal distance from the neighboring nodes. This new point is called the equal-space-point of the 1D stencil. Thus, in both directions of the 2D stencil six points can now be computed. They are respectively denoted \mathbf{x}_{j-1} , \mathbf{x}_j , \mathbf{x}_{j+1} in one direction and \mathbf{x}_{k-1} , \mathbf{x}_k , \mathbf{x}_{k+1} in the other. These new points form two 1D stencils denoted $\{\mathbf{x}_{j-1}, \mathbf{x}_j, \mathbf{x}_{j+1}\}$ and $\{\mathbf{x}_{k-1}, \mathbf{x}_k, \mathbf{x}_{k+1}\}$. They are displayed in Figure 4. Again, two new points $\tilde{\mathbf{x}}_j$ and $\tilde{\mathbf{x}}_k$ are computed as the equal-space-point

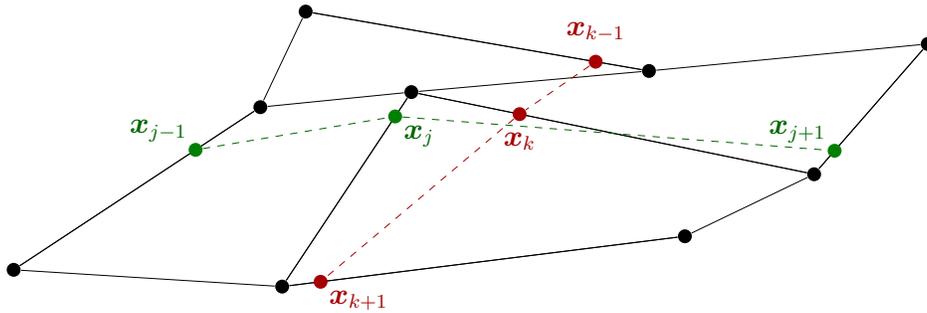


Figure 4: Regularization of a 2D stencil. Step 1: green and red points correspond to 1D equal-space-points in both directions. Two 1D stencils (one per direction) are then created.

of the stencils $\{\mathbf{x}_{j-1}, \mathbf{x}_j, \mathbf{x}_{j+1}\}$ and $\{\mathbf{x}_{k-1}, \mathbf{x}_k, \mathbf{x}_{k+1}\}$. The final equal-space-point \mathbf{x} in a 2D stencil is simply computed as the arithmetic average of $\tilde{\mathbf{x}}_j$ and $\tilde{\mathbf{x}}_k$ as displayed in Figure 5. This method naturally extends in 3D since each

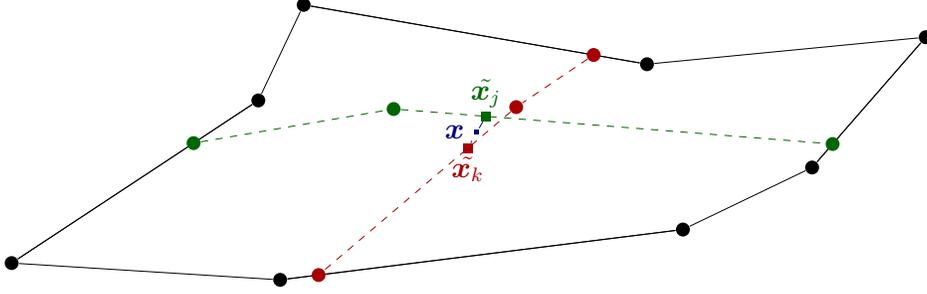


Figure 5: Regularization of a 2D stencil. Step 2: green and red points respectively denoted $\tilde{\mathbf{x}}_j$ and $\tilde{\mathbf{x}}_k$ correspond to each 1D equal-space-point associated to each 1D stencil. The final regularization point \mathbf{x} is computed with an arithmetic mean.

direction owns three 2D stencils as displayed in Figure 6. For each direction, three 2D-equal-space-points method are applied. This leads to a 1D stencil in each direction. These 1D stencils are respectively denoted $\{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \mathbf{x}_{i,3}\}$, $\{\mathbf{x}_{j,1}, \mathbf{x}_{j,2}, \mathbf{x}_{j,3}\}$ and $\{\mathbf{x}_{k,1}, \mathbf{x}_{k,2}, \mathbf{x}_{k,3}\}$ for the directions i, j, k . They are displayed in Figure 7. Now, the equal-space-points $\tilde{\mathbf{x}}_i$, $\tilde{\mathbf{x}}_j$ and $\tilde{\mathbf{x}}_k$ of these 1D stencils are computed. The final regularization node denoted \mathbf{x} , is displayed in Figure 8, and is fixed as the arithmetic mean of $\tilde{\mathbf{x}}_i$, $\tilde{\mathbf{x}}_j$ and $\tilde{\mathbf{x}}_k$.

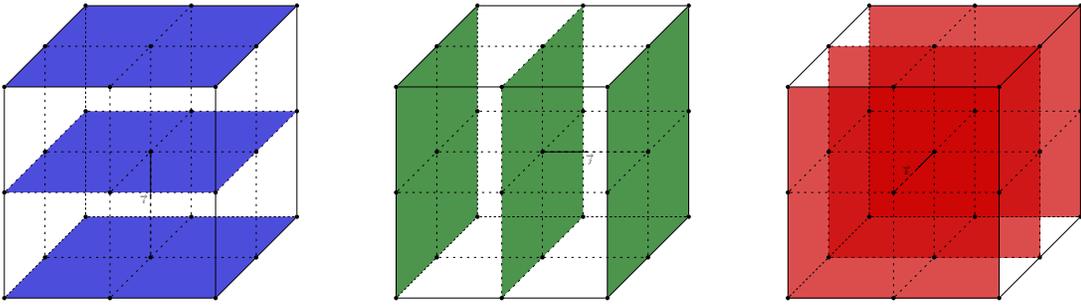


Figure 6: Representation of the 2D stencils in each direction used for the 3D regularization step.

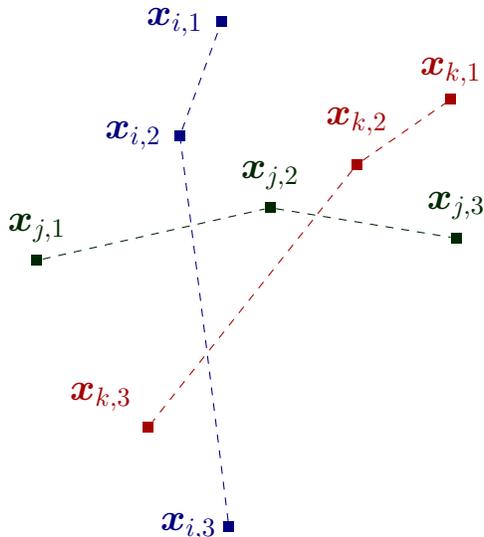


Figure 7: Regularization of a 3D stencil. Step 1: blue, green and red points correspond to the 2D equal-space-points obtained for each 2D stencil in each direction. Three 1D stencils (one per direction) are then created.

Remark. Yao and Stillman (2016) [30] also proposed an alternative to the geometric averaging method explained here. In this strategy, the spatial intersections of polygons are computed. In the present work the geometrical averaging has been favored over the intersection method. It should be noticed that the intersection-based method has interesting convergence property. However, computing the intersection of polygons in 3D can be costly and not straightforward. Here, the simplicity of the geometrical method is preferred over the efficiency of the intersection-based one. The main idea followed in this document is to show that the geometrical averaging method presented by Yao and Stillman (2016) [30], which relies on very simple geometrical considerations, can be useful to design robust and efficient regularisation steps inside an indirect ALE code. Of course, this geometrical averaging method may be considered too simplistic and could be improved if needed.

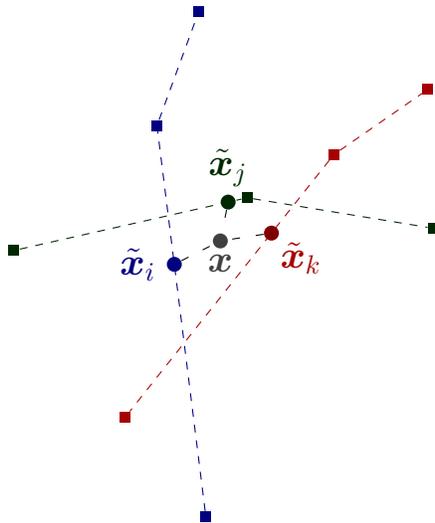


Figure 8: Regularization of a 3D stencil. Step 2: blue, green and red points respectively denoted $\tilde{\mathbf{x}}_i$, $\tilde{\mathbf{x}}_j$, and $\tilde{\mathbf{x}}_k$ correspond to each 1D equal-space-points associated to each 1D stencil. The final regularization point \mathbf{x} , displayed in black, is computed with an arithmetic mean.

4.2 Application on usual meshes

The effectiveness of the regularization method is now assessed by applying the regularization method starting from a perturbed mesh. Perturbed meshes are generated by introducing random deformations. The iterative method is then applied and the behavior of the method is studied. The first results presented starts from a perturbed mesh of a cubic domain as illustrated in Figure 9 (left). After a few iterations, the nodes are evenly distributed as desired (right figure). It is observed that the mesh quality has been greatly improved. The second regularization result presented consists in a perturbed cylindrical domain mesh as illustrated in Figure 10 (left). After a few iterations, the nodes are evenly distributed as desired (right figure). The proposed method produces elements of equal sizes even if concave boundaries are present. This must be emphasized since standard equi-potential relaxation based method [67] may provide a grid attraction phenomenon when working with curvilinear meshes leading to a poor mesh quality near a concave boundary. We also point out that this regulation method is able to handle strongly distorted grids with tangled meshes. Here we refer, to appendix A where impressive

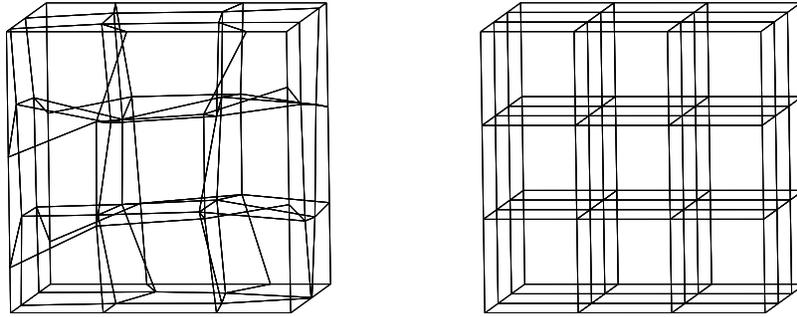


Figure 9: Regularization of a perturbed cubic domain mesh: starting from a perturbed cubic domain mesh (left) a regular mesh is recovered (right) after 10 iterations.

regularization results are obtained.

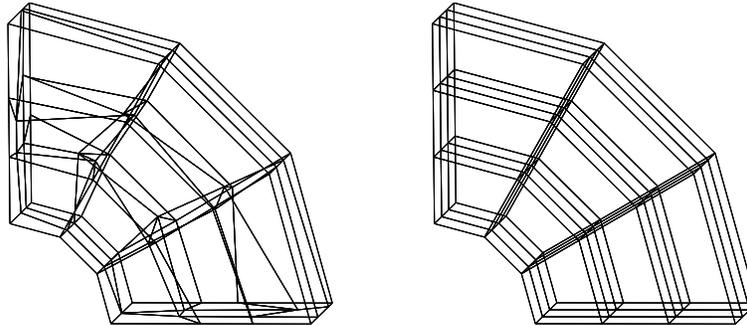


Figure 10: Regularization of a perturbed cylindrical domain mesh: starting from a perturbed cylindrical domain mesh (left) a regular mesh is recovered (right) after 10 iterations. We note the cylindrical geometry is correctly preserved.

5 Remapping step

In this section a detailed presentation of the remapping step is provided. In particular it is shown that the face splitting presented in the previous section (Lagrangian section) is extensively used to perform effective projections. This

numerical strategy relies on a 3D sweeping procedure. Finally, in order to ensure the robustness while keeping the accuracy of the method, a Flux-Corrected-Remap (FCR) [52, 53, 20] strategy is described.

5.1 3D face sweeping strategy

The methodology presented in [44] is now extended to the 3D context. This can be done thanks to the splitting of the cell faces into triangles using the face barycenter presented in the previous section. The notations introduced in [20] are now used. Consider ψ a given conservative variable (here obtained at the end of the Lagrangian step) defined on a collection of non overlapping polygons $\{c\}$, the main idea consists in computing this variable on a new collections of cells denoted \tilde{c} . In practice, given the variable ψ on the collection $\{c\}$ we aim to compute

$$\psi_{\tilde{c}} = \frac{1}{V_{\tilde{c}}} \int_{\tilde{c}} \psi d\mathbf{x}.$$

To determine the volume $V_{\tilde{c}}$ of the new cell \tilde{c} it is necessary to compute the volume of the cell V_c of the old cell c plus the summation of the volumes (which are signed quantities) of the regions swept during the displacement of each triangle associated with a face (the triangles are obtained by the splitting of the cell faces). The swept region, starting from a cell $\{c\}$ whose three constitutive points are denoted p, p^+, p^{++} , to the regularized one $\{\tilde{c}\}$, constitutes a prism and is denoted $\mathcal{P}_{\{p, p^+, p^{++}\}}$. An example of the swept region is displayed in Figure 11. The volume associated with this region is denoted as $V(\mathcal{P}_{\{p, p^+, p^{++}\}})$. Consequently the new volume $V(\tilde{c})$ can be expressed as follows

$$V_{\tilde{c}} = V_c + \sum_{\{p, p^+, p^{++}\} \in \mathcal{T}(c)} V(\mathcal{P}_{\{p, p^+, p^{++}\}}).$$

Similarly the computation of the new variable $\psi_{\tilde{c}}$ (where ψ is the conservative quantity to be projected in this remapping phase) is performed as follows

$$\psi_{\tilde{c}} = \frac{1}{V_{\tilde{c}}} \left(\int_c \psi d\mathbf{x} + \sum_{\{p, p^+, p^{++}\} \in \mathcal{T}(c)} \int_{\mathcal{P}_{\{p, p^+, p^{++}\}}} \psi d\mathbf{x} \right), \quad (7)$$

where the last integral in the right hand side of the previous equation corresponds to the contribution to the flux of the swept region. These integrals

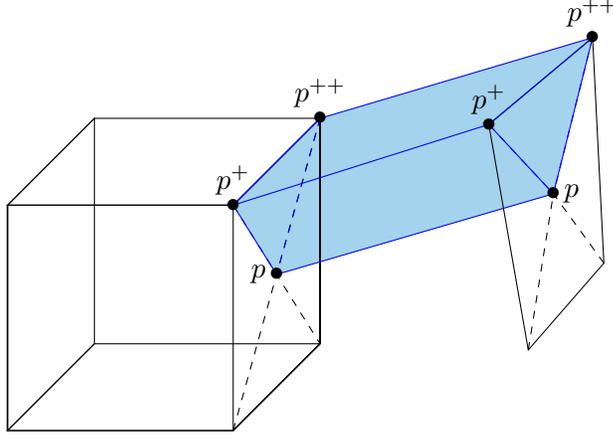


Figure 11: Representation of the swept region in blue, starting from an old triangle (formed by the points $\{p, p^+, p^{++}\}$) to the new one.

are computed using piecewise linear reconstructions and a standard upwind strategy as follows

$$\int_{\mathcal{P}_{\{p, p^+, p^{++}\}}} \psi d\mathbf{x} = \begin{cases} \int_{\mathcal{P}_{\{p, p^+, p^{++}\}}} \psi_{c^+}(\mathbf{x}) d\mathbf{x} & \text{if } V(\mathcal{P}_{\{p, p^+, p^{++}\}}) > 0, \\ \int_{\mathcal{P}_{\{p, p^+, p^{++}\}}} \psi_c(\mathbf{x}) d\mathbf{x} & \text{if } V(\mathcal{P}_{\{p, p^+, p^{++}\}}) < 0, \end{cases}$$

where c and c^+ represent the two neighboring Lagrangian cells that share the triangle $\{p, p^+, p^{++}\}$ and $\psi_c(\mathbf{x})$ and $\psi_{c^+}(\mathbf{x})$ are their respective piecewise linear reconstructions of ψ . It is worth mentioning here that the strategy presented in [59] could be used to improve the accuracy of the remapping step by computing the terms $\psi_c(\mathbf{x})$ and $\psi_{c^+}(\mathbf{x})$ using higher order reconstructions. The piecewise linear reconstructions of the conservative variables on the Lagrangian grid is a standard procedure and has been used in [20, 22]. It involves modeling a discrete reconstructions under the form

$$\psi_c(\mathbf{x}) = \psi_c + (\nabla\psi)_c \cdot (\mathbf{x} - \mathbf{x}_c),$$

where ψ_c represents the known mean value in the cell and $(\nabla\psi)_c$ a gradient of the quantity ψ in cell c . This gradient is computed using a least square procedure (see [22] for the details). A standard Barth-Jespersen slope limiter approach is used to enforce the monotonicity of the reconstructed solution [4].

5.2 3D Flux-Corrected-Remapping method (FCR)

In this section, the 3D Flux-Corrected-Remapping method used to improve the robustness of the overall numerical strategy is detailed. Applying the overall ALE methodology presented here to stiff configurations such as the one encountered with the well-known Noh test case [50], a lack of robustness of the method has been sometimes observed. This is expected since higher order methods may oscillate and require to be stabilized. For this particular test case, a negative internal energy may be observed after the remapping step when using large time step and applying the remapping procedure after large mesh deformations. To address this issue, a 3D Flux-Corrected-Remapping (FCR) strategy is now introduced. The idea of using the flux corrected transport method for the remapping step in a ALE-type numerical strategy has been presented and used in [59]. We apply it to this 3D framework to derive a particularly robust numerical scheme. The main point of this numerical method is to consider a convex combination of a low order numerical flux (here a first order method) which preserves the bounds of the numerical solution with a higher order flux (here second order which does not necessarily preserves the bounds). More precisely, the projected conservative quantity in cell c is written under the form

$$V_{\bar{c}} \psi_{\bar{c}} = V_c \psi_c + \sum_{c' \in \mathcal{C}(c)} (\mathbf{F}_{cc'}^L + C_{cc'} d\mathbf{F}_{cc'}) = V_c \psi_c^L + \sum_{c' \in \mathcal{C}(c)} C_{cc'} d\mathbf{F}_{cc'},$$

where $\mathcal{C}(c)$ is the set of cells in contact with cell c and we have introduced the notation

$$d\mathbf{F}_{cc'} = \mathbf{F}_{cc'}^H - \mathbf{F}_{cc'}^L,$$

in which $\mathbf{F}_{cc'}^H$ and $\mathbf{F}_{cc'}^L$ respectively denotes a high and low order numerical flux and $\psi_c = (\rho_c, \rho_c \mathbf{V}_c, \rho_c E_c)$. The numerical fluxes are computed from the quantities ψ_c and $\psi_{c'}$ which are available at the end of Lagrangian step. Here $\psi_{\bar{c}}$ denotes the remapped (second order) quantity while ψ_c^L is the remapped first order one. The choice of the correction factor is critical and will be set to enforce the preservation of the solution bounds. In this study, we focus on enforcing the density and internal energy bounds. In practice, it has been observed that these conditions are sufficient to ensure the robustness of the remapping procedure even for challenging cases like the Noh test case. Velocity bounds may also be added, as discussed in [59]. It should be pointed out, that the numerical strategy followed here relies on the fact that the low

order method ensures the positiveness of the internal energy. To explain why this statement holds true, we consider a simplified case where only a triangle on a face has moved during the Lagrangian face. This idea may be extended in a more general case. For clarity and coherence with the previous section, this particular triangle is denoted by its constitutive points p, p^+, p^{++} . We recall here that c and c^+ represent the two neighboring Lagrangian cells that share the triangle $\{p, p^+, p^{++}\}$. In this simplified setting equation (7) writes

$$V_{\bar{c}} \psi_{\bar{c}} = V_c \psi_c + \int_{\mathcal{P}_{\{p, p^+, p^{++}\}}} \psi d\mathbf{X},$$

where

$$V_{\bar{c}} = V_c + V(\mathcal{P}_{\{p, p^+, p^{++}\}}).$$

In the first order case this last equation also reads

$$(V_c + V(\mathcal{P}_{\{p, p^+, p^{++}\}})) \psi_{\bar{c}} = (V_c + V^-(\mathcal{P}_{\{p, p^+, p^{++}\}})) \psi_c + V^+(\mathcal{P}_{\{p, p^+, p^{++}\}}) \psi_{c^+},$$

where we have introduced the notation $V^\pm = \frac{1}{2}(V \pm |V|)$. It is important to note that the first order projection simplifies to a straightforward linear interpolation between two states

$$\psi_{\bar{c}} = \theta \psi_c + (1 - \theta) \psi_{c^+}, \quad \theta = \frac{V_c + V^-(\mathcal{P}_{\{p, p^+, p^{++}\}})}{V_c + V^+(\mathcal{P}_{\{p, p^+, p^{++}\}}) + V^-(\mathcal{P}_{\{p, p^+, p^{++}\}})}.$$

Consequently, since $\rho_{\bar{c}} = m_{\bar{c}}/V_{\bar{c}}$ the computation of the new specific internal energy (computed from the remapped quantities) can be rearranged as follows

$$\varepsilon_{\bar{c}} = E_{\bar{c}} - \frac{\mathbf{V}_{\bar{c}}^2}{2} = \frac{m_{\bar{c}} E_{\bar{c}}}{m_{\bar{c}}} - \frac{(m \mathbf{V})_{\bar{c}}^2}{2m_{\bar{c}}^2} = \frac{\rho_{\bar{c}} E_{\bar{c}}}{\rho_{\bar{c}}} - \frac{(\rho \mathbf{V})_{\bar{c}}^2}{2\rho_{\bar{c}}^2}, \quad (8)$$

Consequently, one may write equation (8) as follows

$$\begin{aligned} \rho_{\bar{c}} \varepsilon_{\bar{c}} &= \theta (\rho E)_c + (1 - \theta) (\rho E)_{c^+} - \frac{1}{2\rho_{\bar{c}}} (\theta (\rho \mathbf{V})_c + (1 - \theta) (\rho \mathbf{V})_{c^+})^2 \\ &= \theta (\rho \varepsilon)_c + (1 - \theta) (\rho \varepsilon)_{c^+} + \theta \frac{\rho_c \mathbf{V}_c^2}{2} + (1 - \theta) \frac{\rho_{c^+} \mathbf{V}_{c^+}^2}{2} \\ &\quad - \frac{1}{2\rho_{\bar{c}}} (\theta^2 (\rho \mathbf{V})_c^2 + 2\theta(1 - \theta) (\rho \mathbf{V})_c \cdot (\rho \mathbf{V})_{c^+} + (1 - \theta)^2 (\rho \mathbf{V})_{c^+}^2) \\ &= \theta (\rho \varepsilon)_c + (1 - \theta) (\rho \varepsilon)_{c^+} + \frac{\theta (\rho_{\bar{c}} - \theta \rho_c)}{2\rho_c \rho_{\bar{c}}} (\rho \mathbf{V})_c^2 \\ &\quad - \theta(1 - \theta) \frac{(\rho \mathbf{V})_c \cdot (\rho \mathbf{V})_{c^+}}{\rho_{\bar{c}}} + (1 - \theta) \frac{(\rho_{\bar{c}} - (1 - \theta) \rho_{c^+})}{2\rho_{c^+} \rho_{\bar{c}}} (\rho \mathbf{V})_{c^+}^2. \end{aligned}$$

Now by using the definition of $\rho_{\bar{c}}$ one may finally simplify the previous relations to finally reach

$$\rho_{\bar{c}}\varepsilon_{\bar{c}} = \theta (\rho\varepsilon)_c + (1 - \theta) (\rho\varepsilon)_{c^+} + \theta(1 - \theta) \frac{\rho_c\rho_{c^+}}{2\rho_{\bar{c}}} (\mathbf{V}_c - \mathbf{V}_{c^+})^2. \quad (9)$$

This last equation (9) demonstrates the positiveness of the remapped internal energy. In this study, for the sake of simplicity and clarity, the same correction parameter $C_{cc'}$ is considered for the density and internal energy flux limitations. In practice, they could be different. Here, we refer to [53] for more complex and efficient definitions of the correction factor. Obviously, this simple choice may lead to an increase of numerical diffusion. This drawback may be largely mitigated by the use of higher order reconstruction when computing the numerical fluxes. Here we only restrict ourself to the linear reconstruction procedure presented above. The practical implementation of the FCR method is now specified by providing the correction factor. As discussed in [59] the method preserves the bounds of the remapped density. Concerning the internal energy however, only a minimum bound may be enforced. We point out that this minimum bound turns out to be critical when running the challenging Noh test case. A maximum bound for the internal energy may not be fixed. This may be easily understood when looking at the last equation above since even the low order method does not keep the maximum bound. In the present study only a minimum bound for the internal energy has been enforced with the FCR method. Despite this choice, in practice it has been observed that the resulting numerical method is sufficiently robust to deal with stiff numerical test cases. The 3D Flux-Corrected-Remapping method considered here then satisfy the following properties

$$\begin{aligned} \rho_c^{\min} \leq \rho_{\bar{c}} \leq \rho_c^{\max}, \quad \rho_c^{\min/\max} &= \min_{c' \in \mathcal{C}(c)} / \max_{c' \in \mathcal{C}(c)} (\rho_{c'}), \\ \varepsilon_c^{\min} \leq \varepsilon_{\bar{c}}, \quad \varepsilon_c^{\min} &= \min_{c' \in \mathcal{C}(c)} (\varepsilon_{c'}), \end{aligned}$$

providing the following conditions

$$\begin{aligned} C_{cc'} \leq D_c^1 &= \frac{\rho_c^{\min} V_{\bar{c}} - \rho_{\bar{c}}^L V_{\bar{c}}}{\sum_{c' \in \mathcal{C}(c)} \min(0, dF_{cc'}^m)}, \quad C_{cc'} \leq D_c^2 = \frac{\rho_c^{\max} V_{\bar{c}} - \rho_{\bar{c}}^L V_{\bar{c}}}{\sum_{c' \in \mathcal{C}(c)} \max(0, dF_{cc'}^m)}, \\ C_{cc'} \leq D_c^3 &= \frac{\varepsilon_c^{\min} (\rho_{\bar{c}}^L V_{\bar{c}})^2 - E_{\bar{c}}^L (\rho_{\bar{c}}^L V_{\bar{c}})^2 + \frac{1}{2} (\rho_{\bar{c}}^L V_{\bar{c}} \mathbf{V}_{\bar{c}}^L)^2}{\sum_{c' \in \mathcal{C}(c)} \max \left(0, dF_{cc'}^{de, \min} + \sum_{c'' \in \mathcal{C}(c')} \min \left(0, dF_{cc'c''}^{ds, \min} \right) \right)}, \end{aligned}$$

where dF^m is the first component of $d\mathbf{F}$ (mass component). Finally we simply set $C_{cc'} = \min_{j=\{1;2;3\}} (D_c^j, D_{c'}^j)$. The demonstration to derive these conditions is very similar to the one given in [59] and is not rewritten here. We simply recall the definitions of $dF_{cc'}^{de,\min}$ and $dF_{cc'c''}^{ds,\min}$

$$dF_{cc'}^{de,\min} = dF_{cc'}^E m_{\tilde{c}}^L - d\mathbf{F}_{cc'}^{m\mathbf{V}} \cdot m_{\tilde{c}}^L \mathbf{V}_{\tilde{c}}^L + dF_{cc'}^m (m_{\tilde{c}}^L E_{\tilde{c}}^L - 2\varepsilon_c^{\min} m_{\tilde{c}}^L),$$

and

$$dF_{cc'c''}^{ds,\min} = dF_{cc'}^E dF_{cc''}^m - \frac{1}{2} d\mathbf{F}_{cc'}^{m\mathbf{V}} \cdot d\mathbf{F}_{cc''}^{m\mathbf{V}} - \varepsilon_c^{\min} dF_{cc'}^m dF_{cc''}^m.$$

where $d\mathbf{F}^{m\mathbf{V}}$ and dF^E are respectively the velocity (momentum) and energy components of $d\mathbf{F}$. The numerical test cases presented in the next section are performed using this 3D FCR methodology.

6 Numerical test cases

In the test cases presented here, the fluid is assumed to be a perfect gas so that the fluid pressure writes

$$p = (\gamma - 1)\rho\varepsilon,$$

where γ , known as the polytropic index of the gas takes a value of 5/3 in the monoatomic case and 7/5 in the diatomic case. The sound speed denoted a , is then defined as follows

$$a = \sqrt{\frac{\gamma p}{\rho}}.$$

6.1 Sod test case

6.1.1 Planar geometry

We start this numerical section with the well-known Sod test case [57]. Consider a spatial domain $x \in [0, 1]$ filled with a diatomic gas under the following initial conditions

$$\begin{cases} (\rho_l, p_l, \mathbf{V}_l) = (1.0, 1.0, \mathbf{0}), & \text{for } x \leq 0.5, \\ (\rho_r, p_r, \mathbf{V}_r) = (0.125, 0.1, \mathbf{0}), & \text{for } x \geq 0.5. \end{cases}$$

Even if this case is a simple 1D problem it is run by working with a 3D domain to assess the correct behavior of the presented numerical strategy. More

precisely, in Figure 12, the density profiles are displayed at time 0.6. Zoomed density profiles are shown on the right figure. The numerical results displayed in green correspond to the ones obtained using the ALE procedure while the ones obtained with the Lagrangian scheme are in red. For this simple test, one iteration of the rezoning procedure is applied after 10 Lagrangian iterations. This test may be seen as a sanity check and as expected, for such simple test case, the results obtained with the two approaches are very similar. In addition small diffusion may be observed at the contact discontinuity but none for the shock wave.

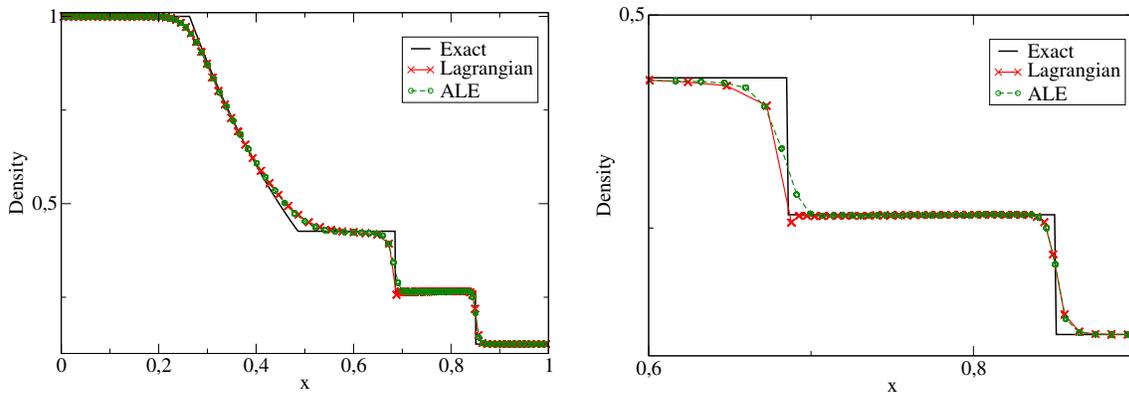


Figure 12: Density profiles with $100 \times 10 \times 10$ cells at time $t = 0.2$.

6.1.2 Cylindrical geometry

We continue this numerical section still considering a Sod problem but now working in a cylindrical geometry. More precisely, the geometry consists in a quarter of cylinder with an internal radius of 0.5 and external radius of 1.5. The depth (along the x-axis) is 0.25. Similarly to the previous test, the domain is filled with a diatomic gas under the following initial conditions

$$\begin{cases} (\rho_l, p_l, \mathbf{V}_l) = (1.0, 1.0, \mathbf{0}), & \text{for } r \leq 1, \\ (\rho_r, p_r, \mathbf{V}_r) = (0.125, 0.1, \mathbf{0}), & \text{for } r > 1. \end{cases}$$

where $r = \sqrt{y^2 + z^2}$. For this test, a $50 \times 20 \times 2$ mesh is used and the simulation is stopped at $t = 0.2$. Symmetry boundary conditions are used. This test is challenging for standard ALE regularisation method. Indeed, as

shown in [6], when working on a cylindrical geometry, the regularisation step often produces an accumulation of the cells near the origin (near the internal radius in the present case). In Figure 13, the density profiles are displayed at initial time and at time $t = 0.2$ working the Lagrangian method. More interestingly, in Figure 14, the density profiles obtained with the ALE method are displayed. For both pictures the regularisation process is applied after 10 Lagrangian steps but with 10 regularisation iterations (left picture) and 100 regularisation iterations (right picture). First of all, contrarily to what is observed in [6], the ALE regularisation process presented in this document does not induce a collapse of the cell layers to the origin. We believe this is an important achievement since the regularisation method is not especially designed for cylindrical geometries. Secondly, it is observed that if a large number of regularisation iterations is used (here after 10 Lagrangian steps) the mesh is very similar to the initial one. This test demonstrates the interest of the present ALE regularisation method when working with complex geometry.

6.2 Sedov test case

The second test we consider is the Sedov test case [31]. It consists in the propagation of a spherical shock wave. The space domain chosen is $[0; 1.2]^3$ filled with a diatomic gas with the following initial conditions

$$(\rho, p, \mathbf{V}) = (1, 10^{-6}, \mathbf{0}).$$

We follow the initial conditions given in [22] enforcing an initial pressure $p = (\gamma - 1)\varepsilon/V$ in the cells containing the space origin, with V the cell volume and $\varepsilon = 0.106384$ the initial specific internal energy. As recalled in [6], the maximum shock density reached at time $t = 1$ is 6 with a shock radius of 1. In Figure 15 the density profiles obtained at time $t = 1$ are displayed working with the Lagrangian strategy in red, Eulerian strategy (projection at each time step on the initial mesh) in blue and ALE strategy in green. For this ALE test, one iteration of the rezoning procedure is applied after 10 Lagrangian iterations. In addition to be a standard benchmark this test is relevant in this context since a part of the Lagrangian mesh is strongly distorted. The impact of the rezoning procedure is clearly observed. In Figure 15 (right picture) and Figure 16 corresponding meshes are displayed. First of all, as expected it is observed that the Eulerian strategy is particularly

diffusive and would require a strong mesh refinement to compete with the Lagrangian accuracy. The ALE results however are much closer to the Lagrangian. Indeed, a good agreement is observed between the Lagrangian and ALE methods for this problem despite the coarse nature of the mesh. This should be emphasized since the mesh representation clearly shows that the ALE mesh regularity is much better than the Lagrangian one. More precisely, while the cell in which the initial is strongly deformed with the Lagrangian strategy, the regularization procedure used within the ALE algorithm keeps a correct global mesh quality while preserving some interesting Lagrangian features.

6.3 Noh test case

The third test case we consider is the Noh problem [50]. It consists in an inwards moving shock wave at a constant speed $D = 1/3$. In our opinion, this test case is interesting to assess the robustness of the numerical method as it will stop with negative internal energy if the FCR method is not used. The initial conditions consist of a monoatomic gas defined as follows $(\rho, p, \mathbf{V}) = (1, 10^{-6}, \mathbf{e}_r)$ where \mathbf{e}_r is the radial vector. Symmetry conditions are enforced on the boundaries sharing the origin. Boundary conditions with $p = 10^{-6}$ are fixed on the other boundaries. On Figure 17 the density profiles are displayed at time $t = 0.6$ for the Lagrangian (in red), Eulerian (in blue) and ALE (in green) strategies. For this ALE test, one iteration of the rezoning procedure is applied after 10 Lagrangian iterations. As observed for the Sedov test case, the Eulerian approach is very diffusive while the ALE one remains closer to the Lagrangian one while being slightly more diffusive. Concerning the mesh regularity in Figures 17 and 18, for this test case the ALE mesh remains close to the Lagrangian one (while not being identical).

6.4 2D Triple point problem

We consider now the triple point problem in planar geometry [6]. This numerical test case is particularly interesting in order to assess the robustness of the ALE method. The computational domain is rectangular and composed of three regions whose 2D dimensions are depicted in Figure 19 with the associated initial conditions. The length of the domain used in the last dimension (not given in Figure 19) is 1.5. All the boundary conditions are set to wall. The computation the ALE algorithm is made on a grid initially

paved with $280 \times 60 \times 120$ cells until a final time $t = 5$. For this simulation, comparison with a full Lagrangian computation can not be performed since it suffers from severe mesh tangling. More precisely, the Lagrangian code breaks down after only few time iterations. As a matter of fact, the strong mesh deformation makes the Lagrangian approach particularly not adapted for this kind of problem. For this ALE test, one iteration of the rezoning procedure is applied after each Lagrangian iterations. In Figure 20 and 21 the density and internal energy profiles are displayed at time $t = 5$ for a $280 \times 60 \times 120$ mesh using the ALE strategy. This large mesh requires an efficient MPI parallelism and a robust regularization procedure.

6.5 3D triple point problem

In this section a 3D version of the previous 2D triple point problem is studied. While this test case seems similar to the previous one, it is a real 3D configuration (the previous could be run with a 2D code). The only difference comes from the initial geometry of the third material which now occupy half of the domain in the depth direction which is now three. In Figures 22 and 23 the density and internal energy profiles are displayed at time $t = 5$ for a $140 \times 60 \times 60$ mesh using the ALE strategy. Here also, this large mesh requires an efficient MPI parallelism and a robust regularization. While being a 3D problem, the regularization procedure enables an efficient computation of this test case.

6.6 Taylor-Green vortex

In this section the Taylor-Green vortex [17] test case is presented. It is a standard benchmark often used to assess the order of convergence of a numerical method. It consists in the simulation of a 2D stationary vortex flow on a domain $(x, y, z) \in [0, 1]^3$ initially filled with a perfect diatomic gas. Symmetry boundary conditions are applied to all boundaries. The flow is modelled by a divergence free velocity field \mathbf{V}^0 defined as follows

$$\mathbf{V}^0(x, y) = C_1 \begin{pmatrix} \sin(\pi x) \cos(\pi y) \\ -\cos(\pi x) \sin(\pi y) \end{pmatrix},$$

where C_1 is a constant. The associated pressure field writes

$$P^0(x, y) = \frac{1}{4} \rho^0 C_1^2 [\cos(2\pi x) + \cos(2\pi y)] + C_2,$$

where C_2 is an integration constant. In addition a source is added to the total energy evolution equation at each time step. It is defined as follows

$$S(x, y) = \frac{\pi \rho^0 C_1^3}{4 \gamma - 1} [\cos(3\pi x) \cos(\pi y) - \cos(3\pi y) \cos(\pi x)].$$

The constants are chosen such that $(\rho^0, C_1, C_2) = (1, 1, 1)$. The order of the Lagrangian and ALE methods can be computed since the solution is regular and an exact solution is available. It should be noticed that this test case is difficult for Lagrangian numerical methods since the mesh is greatly distorted. In Figure 24 and 25, the pressure fields obtained with the Lagrangian and ALE methods are displays at initial time, at time $t = 0.5$ and $t = 0.7$. It is observed here that the Lagrangian mesh is strongly distorted while the regularity of the ALE one is much better. To obtain these results, one step of the regularisation procedure is applied at the end of each Lagrangian step. Following [22] the order of convergence is computed on the pressure field at time $t = 0.7$ using an unlimited second order procedure (for the Lagrangian step). This is only possible since the flow of the Taylor-Green Vortex is smooth and no oscillations arises from a possible discontinuity. The results are displayed in Table 27. In Table 27, the meshes used are denoted $n \times n$ even if in practise a $n \times n \times 1$ have been used. In addition, \mathcal{E}_{L_p} and \mathcal{O}_{L_p} , $p \in \{1; 2; \infty\}$ denotes respectively the global error of convergence in L_p norm. The errors are computed as follows

$$\mathcal{E}_{L_1} = \frac{1}{v_\omega} \sum_c v_c |\Delta P_c|, \quad \mathcal{E}_{L_2} = \sqrt{\frac{1}{v_\omega} \sum_c v_c |\Delta P_c|^2}, \quad \mathcal{E}_{L_\infty} = \max_c |\Delta P_c|,$$

where $v_\omega = \sum_c v_c$ corresponds to the total volume of the domain while v_c is the volume of cell ω_c . In addition, $\Delta P_c = P_c^{num} - P_c^{exact}$ is the difference in cell c between the pressure obtained by the numerical approximation and the exact value. The order of convergence \mathcal{O}_{L_p} is computed with

$$\mathcal{O}_{L_p} = \frac{\log(\mathcal{E}_{L_p}^a) - \log(\mathcal{E}_{L_p}^b)}{\log(\Delta x^a) - \log(\Delta x^b)},$$

with $\mathcal{E}_{L_p}^a$ and $\mathcal{E}_{L_p}^b$ are respectively the L_p errors computed on two meshes of characteristic length Δx^a and Δx^b . Since the meshes considered here are cartesian we have $\Delta x^a = \Delta y^a$. It is observed here that even if a regularisation step is performed at the end of each Lagrangian step, the second order

accuracy of the numerical method is preserved.

7 Conclusion

Solving the Euler equations under the Lagrangian formalism may require a regularization procedure since the mesh quality can be considerably deteriorated. In this document it has been shown that the ideas presented in [30, 68] based on simple geometrical considerations can be successfully adapted to perform reliable regularization steps in a 3D ALE hydrodynamics code. The strength of the method lies in its simplicity, efficiency and suitability for parallel computing and complex geometries. On the other hand, for the time being it is restricted to block structured meshes. The extension to fully non-structured meshes should be investigated. In addition, it has also been shown that the geometrical splitting procedure used in the Lagrangian phase also enables to perform effective projections. This coherence between Lagrangian phase and remapping phase yields the compactness of the overall algorithm. Finally, a Flux-Corrected-Remapping method has been presented. It has been observed that the FCR approach is particularly useful (and necessary) for challenging numerical test cases (Noh test case). This demonstrates the robustness of the numerical strategy and opens exciting perspectives. For example, this FCR philosophy will be applied to design higher order projection extensions. In the future, extensions to hyperelastic material models will also be investigated.

References

- [1] R. Abgrall, J. Breil, P.-H. Maire, and J. Ovadia. Un schéma centré pour l'hydrodynamique Lagrange bidimensionnelle. *Rapport interne Mathématiques Appliquées de Bordeaux LRC-04-10*, 2004.
- [2] H.R. Anbarlooei and K. Mazaheri. Moment of fluid interface reconstruction method in multi-material arbitrary Lagrangian-Eulerian (MMALE) algorithms. *Comput. Methods Appl. Mech. Eng.*, 198:3782–3794, 2009.
- [3] A.J. Barlow. A compatible finite element multi-material ALE hydrodynamics algorithm. *Int. J. Numer. Meth. Fluids*, 56 (8):953–964, 2007.

- [4] T.J. Barth and D.C. Jespersen. The design and application of upwind schemes on unstructured meshes. *AIAA Paper, 27th Aerospace Sciences Meeting, Reno, Nevada*, 89-0366, 1989.
- [5] D.J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Comput. Methods Appl. Mech. Eng.*, 99:235–394, 1992.
- [6] Jérôme Breil. *Numerical methods for Lagrangian and Arbitrary-Lagrangian-Eulerian Hydrodynamic Contribution to the simulation of High-Energy-Density-Physics Problems*. Habilitation à diriger des recherches, Bordeaux University, 2016.
- [7] D.E. Burton, N.R. Morgan, T.C. Carney, and M.A. Kenamond. Reduction of dissipation in Lagrange cell-centered hydrodynamics (CCH) through corner gradient reconstruction (CGR). *J. Comput. Phys.*, 299:229–280, 2015.
- [8] E. Caramana and M. Shashkov. Elimination of artificial grid distortion and hourglass-type motions by means of Lagrangian subzonal masses and pressures. *J. Comput. Phys.*, 142:521–561, 1998.
- [9] E.J. Caramana, D.E. Burton, M.J. Shashkov, and P.P. Whalen. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *J. Comput. Phys.*, 146:227–262, 1998.
- [10] E.J. Caramana, C.L. Rousculp, and D.E. Burton. A compatible, energy and symmetry preserving Lagrangian Hydrodynamics algorithm in three-dimensional Cartesian geometry. *J. Comput. Phys.*, 157:89–119, 2000.
- [11] G. Carré, S. Del Pino, B. Després, and E. Labourasse. A cell-centered Lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. *J. Comput. Phys.*, 228:5160–5183, 2009.
- [12] V. Chiravalle, A. Barlow, and N.R. Morgan. 3D Cell-centered hydrodynamics with subscale closure model and multi-material remap. *Comput. and Fluids*, 207:104592, 2020.
- [13] V. P. Chiravalle and N. R. Morgan. A 3D finite element ALE method using an approximate Riemann solution. *Int. J. Numer. Meth. Fluids*, 83(8):642–663, 2017.

- [14] B. Després and E. Labourasse. Stabilization of cell-centered compressible Lagrangian methods using subzonal entropy. *J. Comput. Phys.*, 231:6559–6595, 2012.
- [15] B. Després and C. Mazeran. Lagrangian gas dynamics in dimension two and Lagrangian systems. *Arch. Ration. Mech. Anal.*, 178:327–372, 2005.
- [16] V.A. Dobrev, T.E. Ellis, T.V. Kolev, and R.N. Rieben. High-order curvilinear finite elements for axisymmetric Lagrangian hydrodynamics. *Comput. and Fluids*, 83:58–69, 2013.
- [17] V.A. Dobrev, T.V. Kolev, and R.N. Rieben. High-order curvilinear finite element methods for Lagrangian hydrodynamics. *SIAM J. Sci. Comput.*, 34:606–641, 2012.
- [18] Veselin Dobrev, Patrick Knupp, Tzanio Kolev, Ketan Mittal, Robert Rieben, and Vladimir Tomov. Simulation-driven optimization of high-order meshes in ALE hydrodynamics. *Comput. and Fluids*, 208:104602, 2020.
- [19] J.K. Dukowicz, M.C. Cline, and F.S. Addressio. A general topology method. *J. Comput. Phys.*, 82:29–63, 1989.
- [20] S. Galera, P.-H. Maire, and J. Breil. A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction. *J. Comput. Phys.*, 229:5755–5787, 2010.
- [21] Rao Garimella, Milan Kucharik, and Mikhail Shashkov. An efficient linearity and bound preserving conservative interpolation (remapping) on polyhedral meshes. *Comput. and Fluids*, 36(2):224–237, 2007.
- [22] G. Georges, J. Breil, and P.-H. Maire. A 3D GCL compatible cell-centered Lagrangian scheme for solving gas dynamics equations. *J. Comput. Phys.*, 305:921–941, 2016.
- [23] S.K. Godunov, A. Zabrodine, M. Ivanov, A. Kraiko, and G. Prokopov. Résolution numérique des problèmes multi-dimensionnels de la dynamique des gaz. *Éditions Mir Moscou*, 1979.
- [24] Jeffrey Grandy. Conservative Remapping and Region Overlays by Intersecting Arbitrary Polyhedra. *J. Comput. Phys.*, 148(2):433–466, 1999.

- [25] H. Luo and J.D. Baum and R. Löhner. On the computation of multi-material flows using ALE formulation. *J. Comput. Phys.*, 194:304–328, 2004.
- [26] C.W Hirt, A.A Amsden, and J.L Cook. An Arbitrary Lagrangian-Eulerian computing method for all flow speeds. *J. Comput. Phys.*, 14(3):227–253, 1974.
- [27] P. Hoch. An Arbitrary Lagrangian-Eulerian strategy to solve compressible fluid flows. *HAL: hal-00366858, version 1, 2009. Available at: [http:// hal.archives-ouvertes.fr/docs/00/36/68/58/PDF/ale2d.pdf](http://hal.archives-ouvertes.fr/docs/00/36/68/58/PDF/ale2d.pdf)*, 2009.
- [28] P. Hoch and E. Labourasse. A frame invariant and maximum principle enforcing second-order extension for cell-centered ALE schemes based on local convex hull preservation. *Int. J. Numer. Meth. Fluids*, 76:1043–1063, 2014.
- [29] A. Huerta J. Donea, J. Ph. Ponthot, and A. Rodriguez-Ferran. Encyclopedia of Computational Mechanics, volume 1: Fundamentals, Chapter 14: Arbitrary Lagrangian-Eulerian Methods,. *John Wiley and Sons*, 2004.
- [30] Jin Yao and Douglas Stillman. An equal-space algorithm for block-mesh improvement. *Procedia Engineering*, 163:199–211, 2016.
- [31] J.R. Kamm and F.X. Timmes. On efficient generation of numerically robust Sedov solutions. *Technical report, Los Alamos National Laboratory*, LA-UR-07-2849, 2007.
- [32] P. Knupp. Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I - A framework for surface mesh optimization. *Int. J. Numer. Methods Eng.*, 48 (3):401–420, 2000.
- [33] P. Knupp, L.G. Margolin, and M.J. Shashkov. Reference Jacobian optimization-based rezone strategies for Arbitrary Lagrangian-Eulerian methods. *J. Comput. Phys.*, 176:93–128, 2002.
- [34] P. Knupp and S. Steinberg. Fundamentals of Grid Generation. *CRC Press, Boca Raton*, 1993.

- [35] Konstantin Lipnikov and Nathaniel Morgan. Conservative high-order discontinuous Galerkin remap scheme on curvilinear polyhedral meshes. *J. Comput. Phys.*, 420:109712, 2020.
- [36] R. Loubère, P.-H. Maire, and P. Váchal. 3D staggered Lagrangian hydrodynamics scheme with cell-centered Riemann solver-based artificial viscosity. *Int. J. Numer. Meth. Fluids*, 72:22–42, 2013.
- [37] Löhner R and Yang C. Improved ALE mesh velocities for moving bodies. *Numer. Methods Eng.*, 12:599–608, 1996.
- [38] P.-H. Maire. A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. *J. Comput. Phys.*, 228:2391–2425, 2009.
- [39] P.-H. Maire. *Contribution to the numerical modeling of Inertial Confinement Fusion*. Habilitation à diriger des recherches, Bordeaux University, 2011.
- [40] P.-H. Maire, R. Abgrall, J. Breil, and J. Ovardia. A cell-centered Lagrangian scheme for two-dimensional compressible flows problems. *SIAM J. Sci. Comput.*, 29:1781–1824, 2007.
- [41] P.-H. Maire and J. Breil. A second-order cell-centered Lagrangian scheme for two-dimensional compressible flow problems. *Int. J. Numer. Meth. Fluids*, 56:1417–1423, 2007.
- [42] P.-H. Maire and B. Nkonga. Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics. *J. Comput. Phys.*, 228:799–821, 2009.
- [43] L.G. Margolin. Introduction to “An arbitrary Lagrangian-Eulerian computing method for all flow speeds”. *J. Comput. Phys.*, 135 (2):198–202, 1997.
- [44] L.G. Margolin and M. Shashkov. Second-order sign-preserving remapping on general grids. *J. Comput. Phys.*, 184:266–298, 2003.
- [45] K. Matsushima, M. Murayama, and K. Nakahashi. Unstructured dynamic mesh for large movement and deformation. *40th AIAA Aerospace Sciences Meeting*, 2002.

- [46] Nathaniel R. Morgan, Jacob I. Waltz, Donald E. Burton, Marc R. Charest, Thomas R. Canfield, and John G. Wohlbier. A point-centered arbitrary Lagrangian Eulerian hydrodynamic approach for tetrahedral meshes. *J. Comput. Phys.*, 290:239–273, 2015.
- [47] N.R. Morgan, K.N. Lipnikov, D.E. Burton, and M.A. Kenamond. A Lagrangian staggered grid Godunov-like approach for hydrodynamics. *J. Comput. Phys.*, 259:568–597, 2014.
- [48] J.M. Morrell, P.K. Sweby, and A. Barlow. A cell by cell anisotropic adaptive mesh ALE scheme for the numerical solution of the Euler equations. *J. Comput. Phys.*, 226(1):1152–1180, 2007.
- [49] B. Nkonga. On the conservative and accurate CFD approximations for moving meshes and moving boundaries. *Comput. Methods Appl. Mech. Eng.*, 190:1801–1825, 2000.
- [50] W.F. Noh. Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux. *J. Comput. Phys.*, 72:78–120, 1987.
- [51] J. Michael Owen and Mikhail Shashkov. Arbitrary Lagrangian Eulerian remap treatments consistent with staggered compatible total energy conserving Lagrangian methods. *J. Comput. Phys.*, 273:520–547, 2014.
- [52] R. Liska and M. Shashkov and P. Váchal and B. Wendroff. Optimization-based synchronized flux-corrected conservative interpolation (remapping) of mass and momentum for arbitrary lagrangian-eulerian methods. *J. Comput. Phys.*, 229(5):1467–1497, 2010.
- [53] R. Liska and M. Shashkov and P. Váchal and B. Wendroff. Synchronized flux corrected remapping for ALE methods. *Comput. and Fluids*, 46(1):312–317, 2011.
- [54] Allen Robinson, Thomas Brunner, Susan Carroll, Richard Drake, Christopher Garasi, Thomas Gardiner, Thomas Hail, Heath Hanshaw, David Hensinger, Duane Labreche, Raymond Lemke, Edward Love, Christopher Luchini, Stewart Mosso, John Niederhaus, Curtis Ober, Sharon Petney, William Rider, Guglielmo Scovazzi, O Strack, Randall Summers, Timothy Trucano, V Weirs, Michael Wong, and Thomas

- Voth. ALEGRA: An Arbitrary Lagrangian-Eulerian Multimaterial, Multiphysics Code. 2012.
- [55] G. Scovazzi. Lagrangian shock hydrodynamics on tetrahedral meshes: A stable and accurate variational multiscale approach. *J. Comput. Phys.*, 231:8029–8069, 2012.
- [56] G. Scovazzi, J.N. Shadid, E. Love, and W.J. Rider. A conservative nodal variational multiscale method for Lagrangian shock hydrodynamics. *Comput. Methods Appl. Mech. Eng.*, 199:3059–3100, 2010.
- [57] G.A. Sod. A survey of finite difference methods for system of nonlinear conservation laws. *J. Comput. Phys.*, 27:1–31, 1978.
- [58] P. Váchal and B. Wendroff. A symmetry preserving dissipative artificial viscosity in an r-z staggered Lagrangian discretization. *J. Comput. Phys.*, 258:118–136, 2014.
- [59] J. Velechovsky, J. Breil, and R. Liska. Flux corrected remapping using piecewise parabolic reconstruction for 2D cell-centered ALE methods. *Int. J. Numer. Meth. Fluids*, 76(9):575–586, 2014.
- [60] J. Velechovsky, E. Kikinzon, N. Ray, H. Rakotoarivelo, A. Herring, M. Kenamond, K. Lipnikov, M. Shashkov, R. Garimella, and D. Shevitz. Multi-material swept face remapping on polyhedral meshes. *J. Comput. Phys.*, 469:111553, 2022.
- [61] J. von Neumann and R.D. Richtmyer. A method for the numerical calculations of hydrodynamical shocks. *J. Appl. Phys.*, 21:232–238, 1950.
- [62] J. Waltz, T. Canfield, N. Morgan, L. Risinger, and J. Wohlbier. Verification of a three-dimensional unstructured finite element method using analytic and manufactured solutions. *Comput. Fluids*, 81:57–67, 2013.
- [63] J. Waltz, N. Morgan, T. Canfield, M. Charest, and J. Wohlbier. A three-dimensional finite element Arbitrary Lagrangian-Eulerian method for shock hydrodynamics on unstructured grids. *Comput. Fluids*, 92:172–187, 2013.
- [64] B. Wendroff. A compact artificial viscosity equivalent to a tensor viscosity. *J. Comput. Phys.*, 229:6673–6675, 2010.

- [65] M.L. Wilkins. Calculation of elastic plastic flow. *Methods Comput. Phys.*, 3:211–263, 1964.
- [66] A. M. Winslow. Equipotential zoning of two-dimensional meshes. *Technical Report UCRL-7312*, 1963.
- [67] A. M. Winslow. Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh. *J. Comput. Phys.*, 1(2):149–172, 1966.
- [68] J. Yao. A mesh relaxation study and other topics. *Lawrence Livermore National Laboratory Technical Report, LLNL-TR-637101*, 2013.
- [69] D. Zeng and C. Ross Ethier. A semi-torsional spring analogy model for updating unstructured meshes in 3D moving domains. *Finite Elem. Anal. Des.*, 41(11):1118–1139, 2005.

A Regularization of meshes with strong cells tangling

In this appendix, the numerical results obtained with the regularization method starting from strongly distorted meshes are presented. The regularization method is applied on a strongly distorted cubic domain as illustrated in Figure 28 (left figure). It is observed, that despite the very poor mesh regularity, the iterative method manages to reposition the nodes as desired (right figure). The second test consists in a strongly perturbed cylindrical domain mesh as illustrated in Figure 29 (left). Here again, after a few iterations, the nodes are evenly distributed as desired (right figure) despite the cylindrical geometry. These regularization results are impressive and demonstrate that the regularization procedure handle very large deformation.

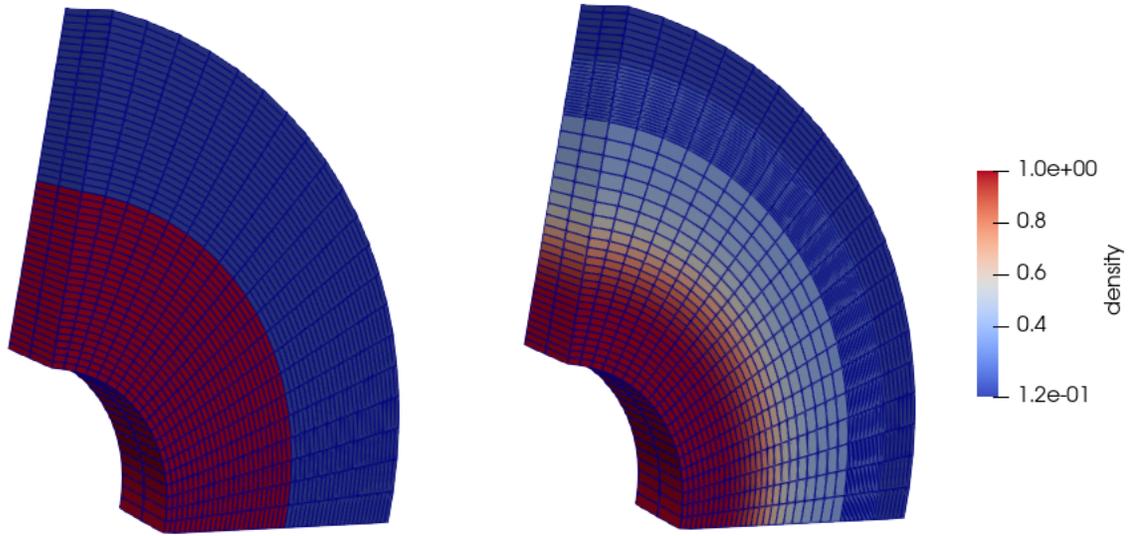


Figure 13: Density profiles at initial time and at time $t = 0.2$ obtained with the Lagrangian method working with a $50 \times 20 \times 2$ mesh (right).

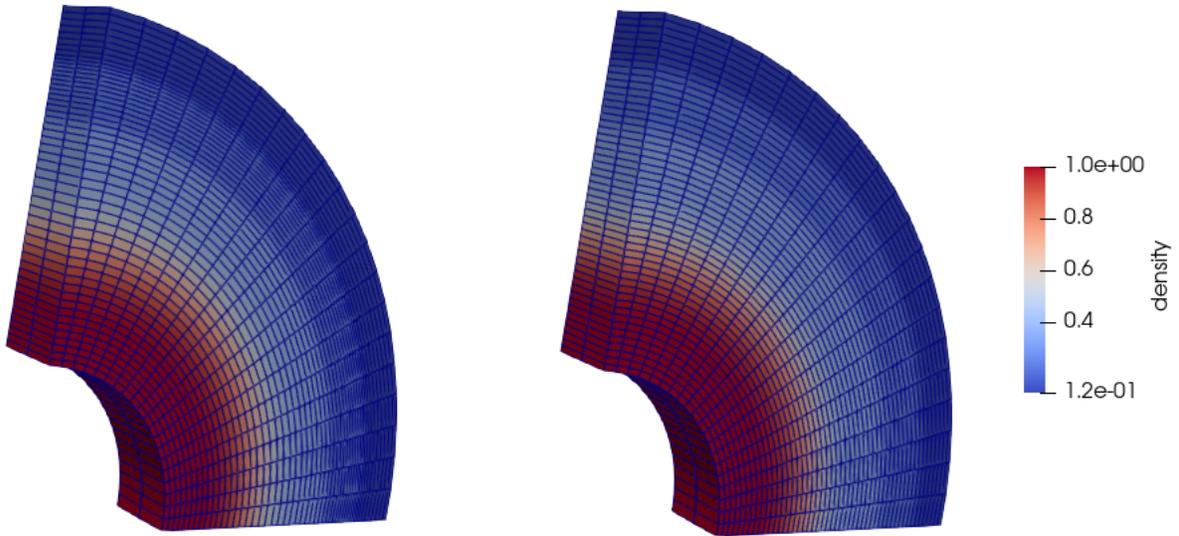


Figure 14: Density profiles obtained with the ALE method working with a $50 \times 20 \times 2$ mesh at time $t = 0.2$. For both pictures the regularisation process is applied after 10 Lagrangian steps but with 10 regularisation iterations (left) and 100 regularisation iterations (right).

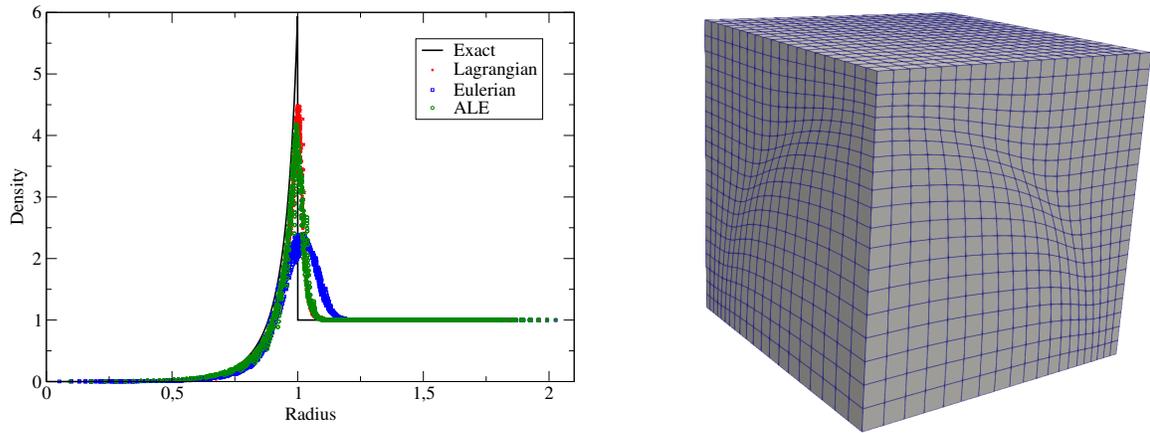


Figure 15: Density profiles obtained for the Lagrangian (red), Eulerian (blue) and ALE (green) working with $20 \times 20 \times 20$ cells. Representation of the ALE mesh at time $t = 1$.

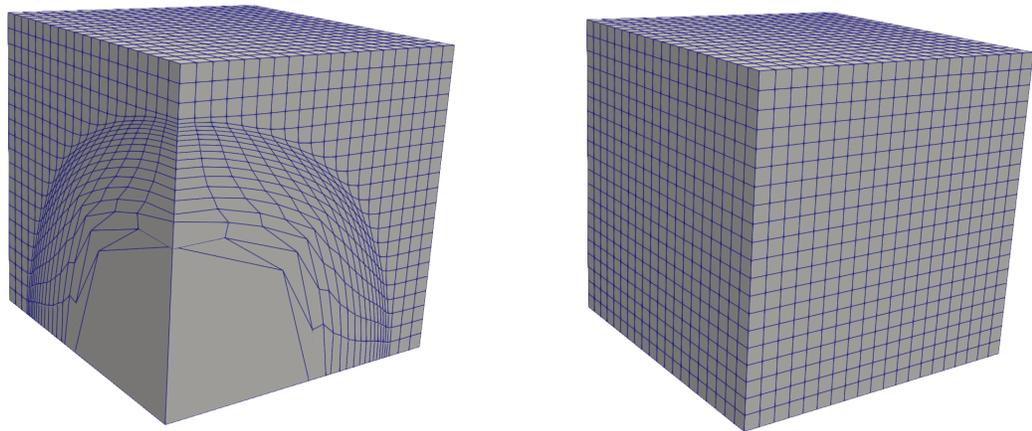


Figure 16: Representation of the Lagrangian mesh (left) and Eulerian mesh (right) at time $t = 1$.

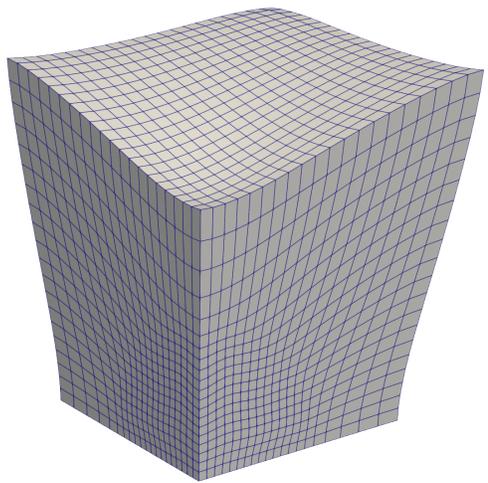
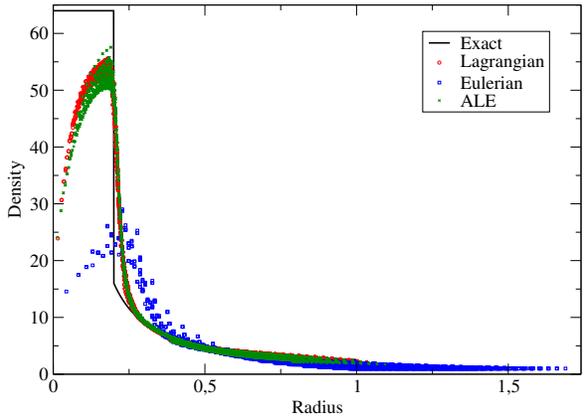


Figure 17: Density profiles obtained for the Lagrangian (red), Eulerian (blue) and ALE (green) working with $20 \times 20 \times 20$ cells. Representation of the ALE mesh at time $t = 0.6$ (right figure).

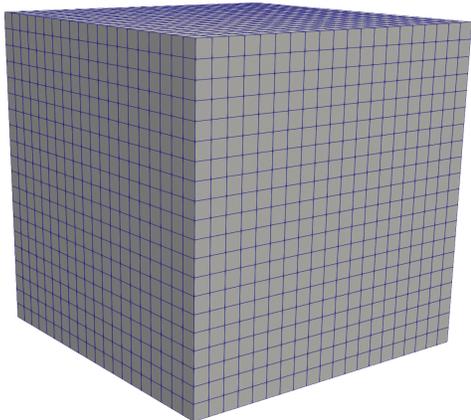
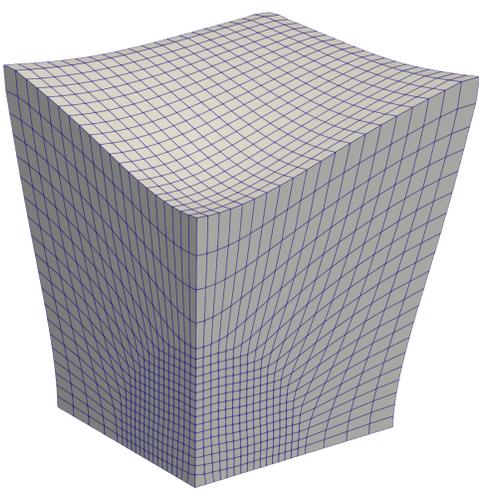


Figure 18: Representation of the Lagrangian mesh (left) and Eulerian mesh (right) at time $t = 0.6$.

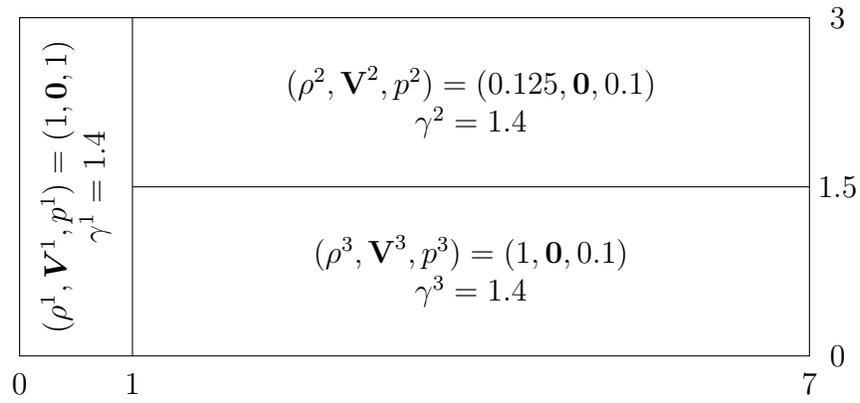


Figure 19: 2D triple point layout and initial conditions.

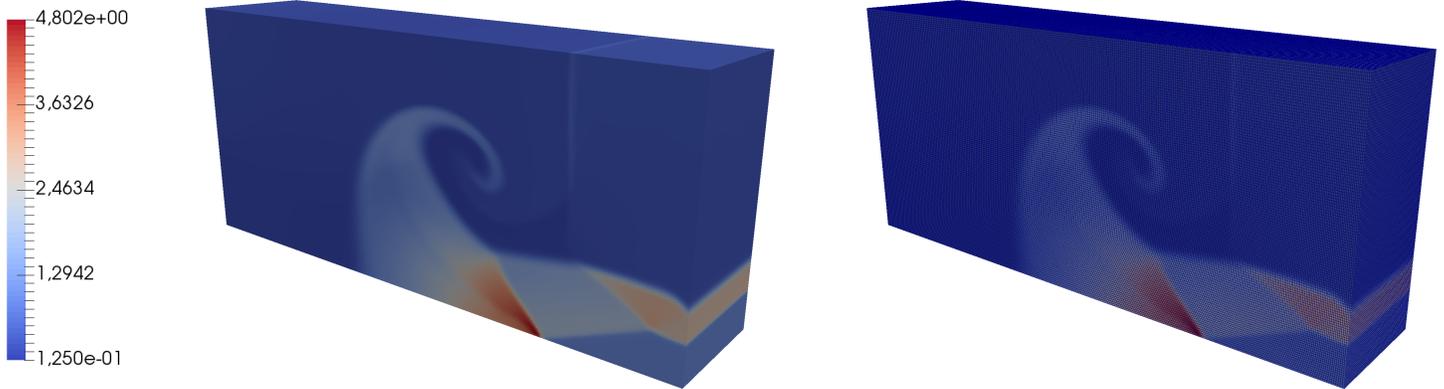


Figure 20: Density profiles with $280 \times 60 \times 120$ cells. The mesh is displayed on the right Figure.

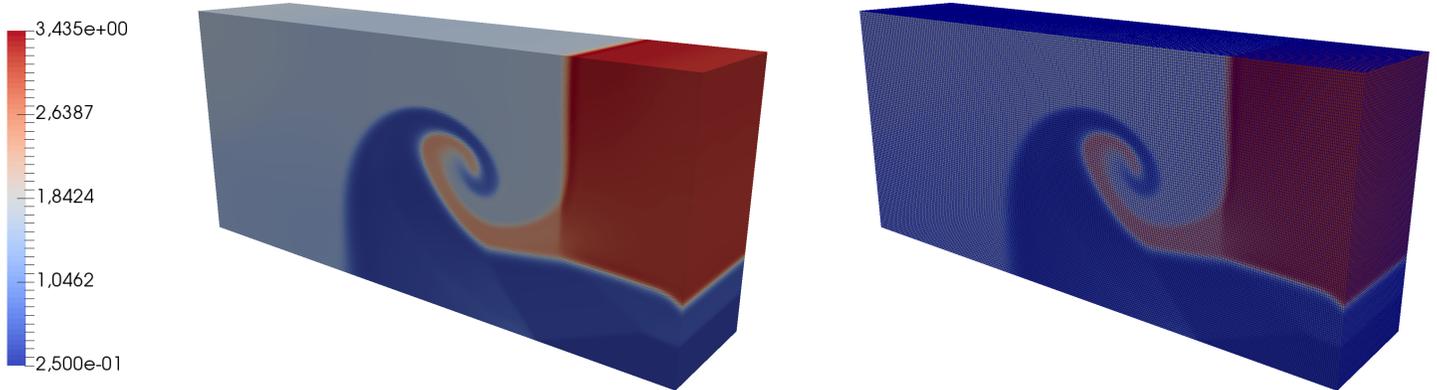


Figure 21: Internal energy profiles with $280 \times 60 \times 120$ cells. The mesh is displayed on the right Figure.

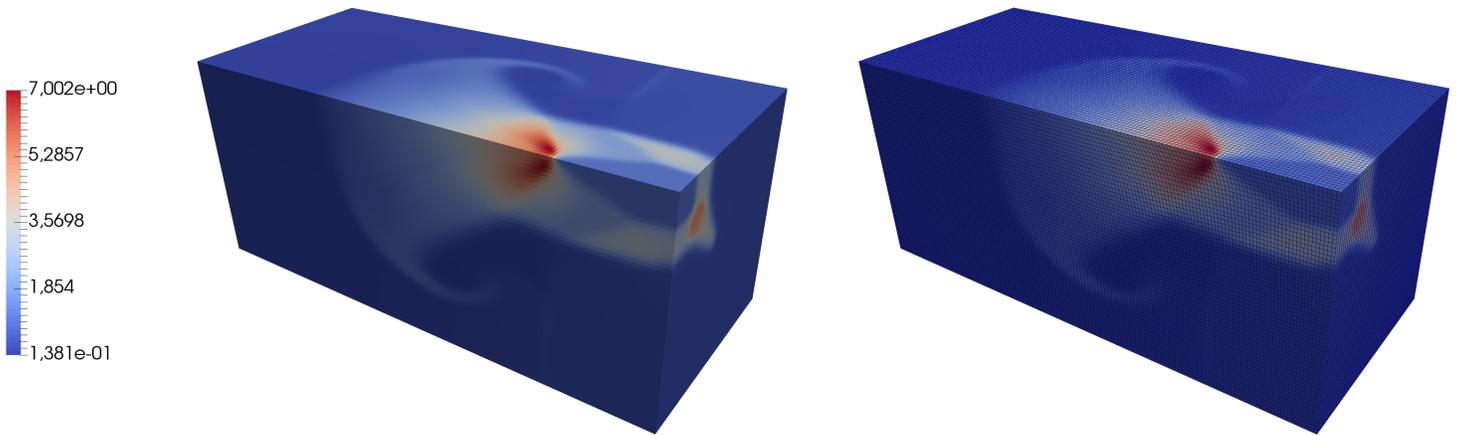


Figure 22: Density profiles with $140 \times 60 \times 60$ cells. The ALE mesh at final time is displayed on the right picture.

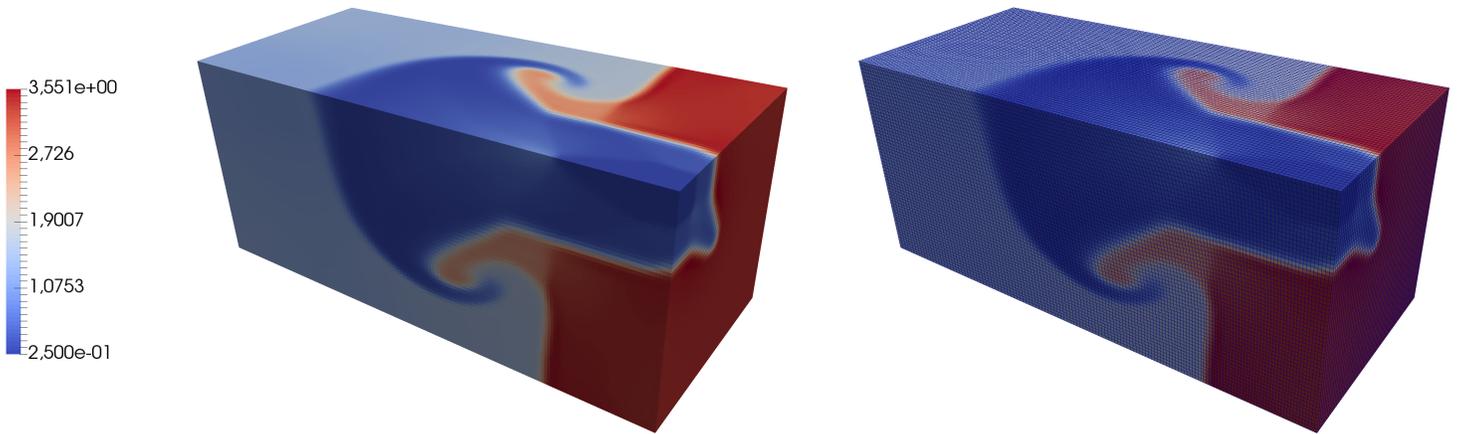


Figure 23: Internal energy profiles with $140 \times 60 \times 60$ cells. The ALE mesh at final time is displayed on the right picture.

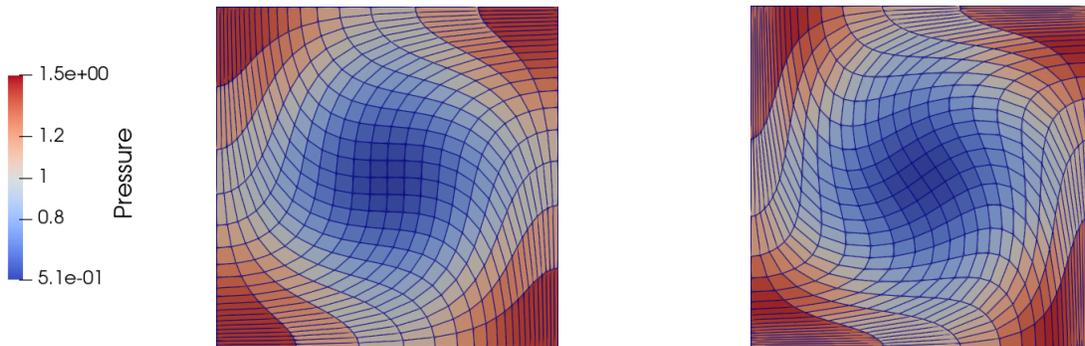


Figure 24: Pressure profiles obtained with the Lagrangian method working with a $20 \times 20 \times 1$ mesh at time $t = 0.5$ (left) and $t = 0.7$ (right).

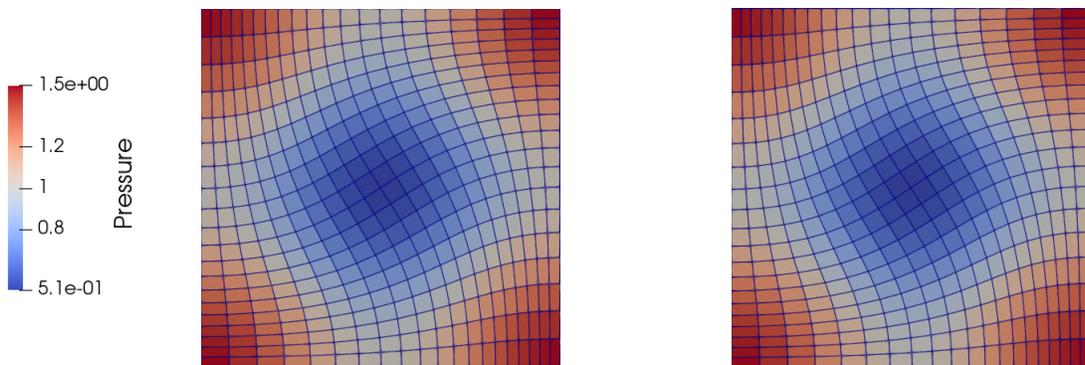


Figure 25: Pressure profiles obtained with the ALE method working with a $20 \times 20 \times 1$ mesh at time $t = 0.5$ (left) and $t = 0.7$ (right).

Mesh	\mathcal{E}_{L_1}	\mathcal{O}_{L_1}	\mathcal{E}_{L_2}	\mathcal{O}_{L_2}	\mathcal{E}_{L_∞}	\mathcal{O}_{L_∞}
10×10	$4.76 \cdot 10^{-2}$	-	$5.54 \cdot 10^{-2}$	-	$12.0 \cdot 10^{-2}$	-
20×20	$2.01 \cdot 10^{-2}$	1.24	$2.90 \cdot 10^{-2}$	0.93	$8.92 \cdot 10^{-2}$	0.43
40×40	$5.79 \cdot 10^{-3}$	1.80	$1.18 \cdot 10^{-2}$	1.29	$6.33 \cdot 10^{-2}$	0.49
80×80	$1.20 \cdot 10^{-3}$	2.27	$3.08 \cdot 10^{-3}$	1.94	$2.24 \cdot 10^{-2}$	1.50
160×160	$3.16 \cdot 10^{-4}$	1.92	$6.40 \cdot 10^{-4}$	2.26	$4.73 \cdot 10^{-3}$	2.24

Figure 26: Taylor-Green vortex - Table of convergence of the pressure field at final time $t = 0.7$ obtained with the Lagrangian method.

Mesh	\mathcal{E}_{L_1}	\mathcal{O}_{L_1}	\mathcal{E}_{L_2}	\mathcal{O}_{L_2}	\mathcal{E}_{L_∞}	\mathcal{O}_{L_∞}
10×10	$2.06 \cdot 10^{-2}$	-	$2.43 \cdot 10^{-2}$	-	$4.55 \cdot 10^{-2}$	-
20×20	$3.39 \cdot 10^{-3}$	2.60	$3.93 \cdot 10^{-3}$	2.63	$6.75 \cdot 10^{-3}$	2.75
40×40	$9.99 \cdot 10^{-4}$	1.76	$1.20 \cdot 10^{-3}$	1.72	$2.78 \cdot 10^{-3}$	1.28
80×80	$3.24 \cdot 10^{-4}$	1.62	$3.97 \cdot 10^{-4}$	1.59	$9.88 \cdot 10^{-4}$	1.49
160×160	$1.00 \cdot 10^{-4}$	1.70	$1.26 \cdot 10^{-4}$	1.66	$3.48 \cdot 10^{-4}$	1.50

Figure 27: Taylor-Green vortex - Table of convergence of the pressure field at final time $t = 0.7$ obtained with the ALE method.

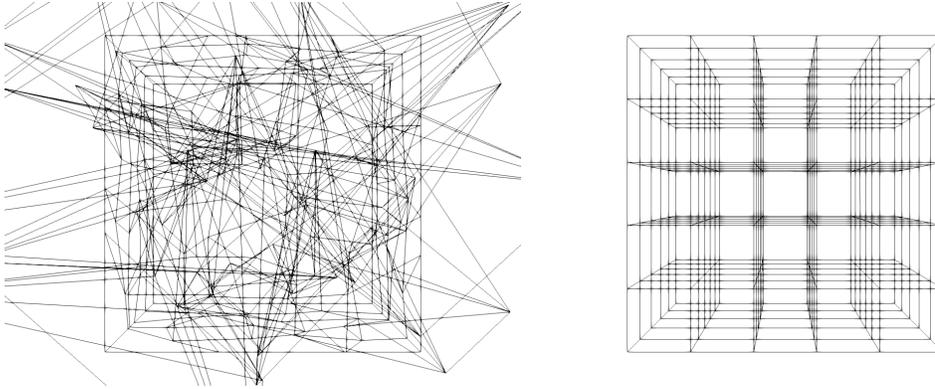


Figure 28: Regularization results obtained starting from a cubic domain mesh with a very strong cell tangling (left picture) obtained after 30 iterations (right picture).

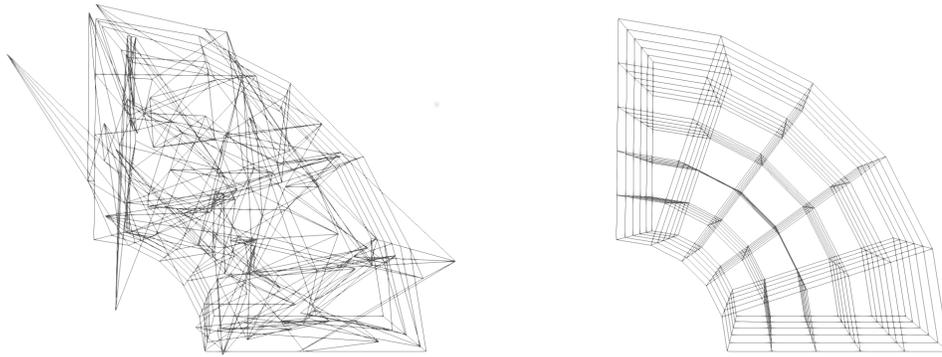


Figure 29: Regularization results obtained starting from a cylindrical domain mesh with a very strong cell tangling (left picture) obtained after 30 iterations (right picture).