

# 3D Mesh Regularization Based on a Weighted Line Sweeping Method

Guillaume Damour <sup>†</sup>

Sébastien Guisset <sup>‡</sup>

Jérôme Breil <sup>§</sup>

## Abstract

Mesh regularization methods are employed in computational simulations, particularly when dealing with problems modeled using the Lagrangian formalism. Indeed solving complex engineering problems using the Euler equations within a Lagrangian framework presents a critical challenge. In complex engineering applications that involve simulating strong shock waves, the computational mesh can deform significantly, closely tracking the movements of materials or fluids. In such cases, deteriorated or tangled meshes may compromise the integrity of the simulation. Combining regularization methods with Lagrangian simulation becomes indispensable, leading to the development of the Arbitrary Lagrangian-Eulerian (ALE) framework. In this work, we propose a regularization method for block-structured meshes which could be used within a 3D ALE code. Our proposed method aims to regularize Lagrangian mesh while preserving the interesting physical characteristics of the Lagrangian mesh using a 3D regularization method. This method is based on a weighted line sweeping method which can maintain non-uniform characteristics of the Lagrangian mesh.

## 1 Introduction.

Mesh regularization methods are often employed in computational simulations, particularly when dealing with problems modeled using the Lagrangian formalism. Solving the gas dynamics equations under the Lagrangian formalism is particularly appropriate for simulating complex engineering problems with strong shock waves. In simulations where the mesh tracks the movement of materials or fluids, such as in hydrodynamics or fluid dynamics simulations, the mesh can become distorted and tangled over time. On the other hand, the Eulerian description, with its fixed mesh, can handle significant distortions more easily but leads to strong numerical diffusion. The Arbitrary Lagrangian-Eulerian (ALE) formulation combines the benefits of both Lagrangian and Eulerian methods. It allows the mesh to move with the material during the Lagrangian phase, and then a rezoning step is initiated to construct a new regularized mesh using the nodes moved during Lagrangian phase. Regularization method [1] is the cornerstone of the ALE method as it helps restore and maintain the quality of the mesh. Regularization ensures the accuracy and reliability of the simulation results. Indeed, as materials or fluids move and deform within

the computational domain, mesh elements can become twisted or overlapped, leading to tangled or distorted cells. Regularization methods aim to identify and resolve these tangled elements, allowing the simulation to continue smoothly. Distorted meshes may also affect the numerical stability of simulations. Regularization helps stabilize the solution, preventing numerical issues that can arise from poorly conditioned meshes. In simulations that involve tracking physical features such as shock waves, regularization methods aim to maintain the integrity of these features. This is crucial for accurately capturing the physics of the problem. Overall, the use of regularization methods is essential for ensuring the accuracy, stability, and efficiency of simulations, particularly in scenarios where strong mesh deformation or tangled mesh may occur.

The numerical resolution is usually based on an indirect ALE strategy. More precisely, using such strategy, the first ingredient consists in solving the set of Euler equations in the Lagrangian framework. During the Lagrangian step, where the mesh follows the fluid velocity, the mesh quality may be considerably deteriorated. A regularization step intends to improve it while keeping some interesting features of the Lagrangian phase. Here starts a rezoning step in which, from the nodes moved during the Lagrangian phase, a new regularized grid is built up. The overall indirect ALE strategy is displayed in fig. 1.

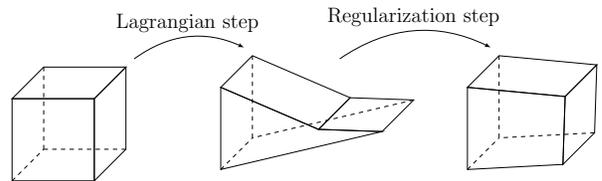


Figure 1: Indirect ALE strategy.

This work aims to present a regularization method for 3D block-structured meshes. The mesh regularization method used in this study is inspired by the Line-Sweeping regularization method introduced by Jin YAO [11, 10]. This iterative method regularizes each node with a given geometrical rule in a structured stencil consisting of the node itself and its neighboring nodes. As pointed out in [11, 10], the Line-Sweeping

<sup>†</sup>CEA-CESTA, Le Barp, France. guillaume.damour2@cea.fr

<sup>‡</sup>CEA-CESTA, Le Barp, France. sebastien.guisset@cea.fr

<sup>§</sup>CEA-CESTA, Le Barp, France. jerome.breil@cea.fr

regularization method gives similar results compared to other regularization methods (Laplacian smoothing, equi-potential relaxation or angle-based method). In addition, it prevents grid attraction effects [3], and produces good mesh quality on a concave boundary or at irregular connectivity point. This method has also been used to rezone 2D meshes in the context of aerothermodynamic reentry simulation which involves simulating the aerodynamic conditions and ablation experienced by a vehicle during its reentry into the Earth's atmosphere [7].

In this paper, we introduce a modified version of the equal space method introduced in [11] for 3D meshes. The main advance lies in the introduction of a weighted line sweeping method, which intends to preserve the interesting physical features of the Lagrangian mesh. This new version allows the regularization of meshes where points are not evenly spaced. In such cases, the result will not converge to the classical solution given by Laplacian smoothing or by the original equal space method. This aspect is of great interest in the context of ALE simulations.

The paper is structured into two main sections. In a first part, the regularization method is detailed. This method is implemented in a 3D code from the ideas proposed in [11, 10]. In particular, it demonstrates how a purely geometrical regularization methodology can effectively handle the regularization of highly deformed meshes. In this part, some regularization examples are displayed to show the efficiency of the method even in the case of 3D tangled mesh. This point out the flexibility of the method which handles the regularization of very strongly deformed meshes. In a second part, we introduce a weighted algorithm. This algorithm aims to further enhance the adaptability of the regularization method to Lagrangian hydrodynamics. To achieve this, we introduce a weighting mechanism to control the distribution of nodes in the final regularized mesh. This modified regularization approach is specifically designed to preserve the good properties of the Lagrangian simulation.

## 2 Regularization method

The mesh regularization method used in this study is inspired by the Line-Sweeping regularization method introduced in [10, 11]. The Line-Sweeping method is a local iterative geometric method for mesh regularization. The aim is to move each node according to the position of its neighbours until the regularization converges. The proposed method transforms the regularization of a multi-dimensional stencil into a sequence of one-dimensional stencil regularization steps. Since the methodology is devoted to 3D structured mesh, the

index (i,j,k) are introduced to represent the three dimensions of space. A 2D stencil is a set of nodes with one dimension i, j or k fixed. A 1D stencil is a set of nodes with two dimensions (i,j), (i,k) or (j,k) fixed.

### 2.1 Line Sweeping Method

For each 1D stencil, to calculate the regularized position at iteration  $m + 1$  of the node  $\mathbf{x}_i^m$  drawn in blue in fig. 2, it is sufficient to move the new node  $\mathbf{x}_i^{m+1}$  in red at equal distance from the neighbouring nodes  $\mathbf{x}_{i-1}^m$  and  $\mathbf{x}_{i+1}^m$  while following the geometry of the initial stencil.

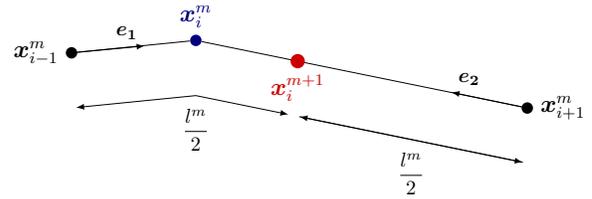


Figure 2: Regularization of a 1D stencil using equal-space-point method.

At first, the length  $l^m$  of the branch is computed as

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

The index  $i$  is omitted for simplicity. Then the regularized point  $\mathbf{x}_i^{m+1}$  named equal-space-point is defined as

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

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

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

This choice offers the advantage of aligning the rezoned node with the initial geometry of the stencil, in contrast to the arithmetic average of nodes  $\mathbf{x}_{i-1}^m$  and  $\mathbf{x}_{i+1}^m$ . The regularization of a 1D stencil, as depicted in fig. 2, will be applied in the case of edges present in boundary conditions. For the sake of simplicity, the iteration index  $m$  is omitted in the following discussion as it is clear that the new position  $m + 1$  is computed

from the previous position at iteration  $m$ . In a 2D stencil, every triplet of points corresponds to its own equal-space-point. Consequently, in both directions of the stencil, a first relaxed set of points is calculated. As we are in a 2D stencil, six points are calculated:  $\mathbf{x}_{j-1}$ ,  $\mathbf{x}_j$ , and  $\mathbf{x}_{j+1}$  in one direction, and  $\mathbf{x}_{k-1}$ ,  $\mathbf{x}_k$ , and  $\mathbf{x}_{k+1}$  in the other direction. These points form two additional 1D stencils, denoted as  $\mathbf{x}_{j-1}, \mathbf{x}_j, \mathbf{x}_{j+1}$  and  $\mathbf{x}_{k-1}, \mathbf{x}_k, \mathbf{x}_{k+1}$ . They are illustrated in fig. 3. Again, another relaxed set of two new points  $\tilde{\mathbf{x}}_j$  and  $\tilde{\mathbf{x}}_k$  are computed as the equal-space-point 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 fig. 4.

$$\mathbf{x} = \frac{1}{2} (\tilde{\mathbf{x}}_j + \tilde{\mathbf{x}}_k).$$

The method depicted in fig. 3 and fig. 4 is also applied to planar faces encountered in boundary conditions. It's important to note that only boundary conditions featuring planar faces are regularized in the cases presented in this paper.

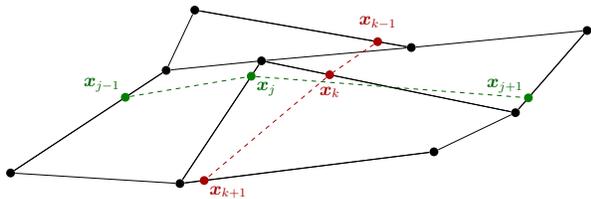


Figure 3: Regularization of a 2D stencil : green and red points correspond to 1D equal-space-points in both directions.

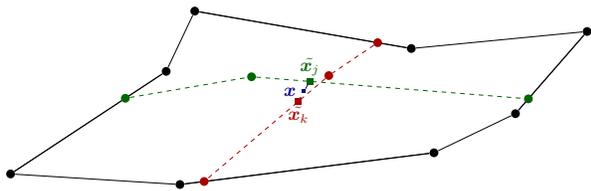


Figure 4: Regularization of a 2D stencil : green point  $\tilde{\mathbf{x}}_j$  and red point  $\tilde{\mathbf{x}}_k$  are respectively the 1D equal-space-point of the green branch and of the red branch.

This method naturally extends to 3D since each direction has three 2D stencils, as displayed in fig. 5. For each direction, three 2D equal-space-points methods are applied, resulting in a 1D stencil in each direction. They will respectively be denoted as  $\{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \mathbf{x}_{i,3}\}$ ,  $\{\mathbf{x}_{j,1}, \mathbf{x}_{j,2}, \mathbf{x}_{j,3}\}$  and  $\{\mathbf{x}_{k,1}, \mathbf{x}_{k,2}, \mathbf{x}_{k,3}\}$  for the  $i, j$ , and  $k$  directions. These 1D stencils are displayed in fig. 6.

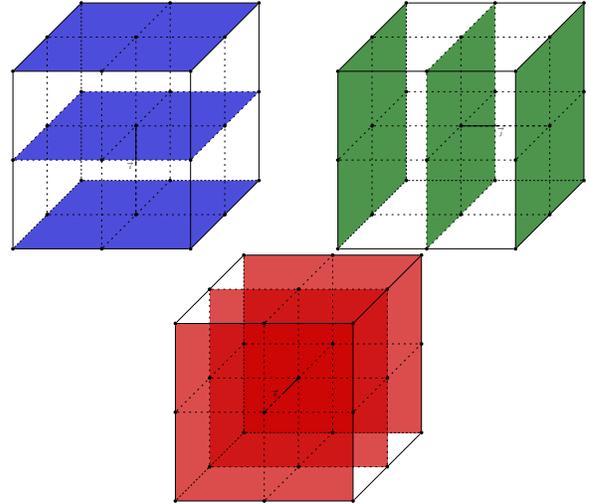


Figure 5: 2D stencils in each direction used for 3D regularization.

Again, the equal-space-points  $\tilde{\mathbf{x}}_i$ ,  $\tilde{\mathbf{x}}_j$  and  $\tilde{\mathbf{x}}_k$  of these 1D stencils are computed. The central node  $\mathbf{x}$  displayed in fig. 7 is finally positioned as the arithmetic mean of  $\tilde{\mathbf{x}}_i$ ,  $\tilde{\mathbf{x}}_j$  and  $\tilde{\mathbf{x}}_k$ .

$$\mathbf{x} = \frac{1}{3} (\tilde{\mathbf{x}}_i + \tilde{\mathbf{x}}_j + \tilde{\mathbf{x}}_k).$$

## 2.2 Regularization of cubic and cylindrical domain

To evaluate the efficacy of the regularization method, we conduct experiments involving the manipulation of nodes within an initially uniformly meshed structure. We generate perturbed meshes through the introduction of random deformations into these initially uniform meshes. The results presented here are for a small perturbation such that the reader can see the 3D effect of the regularization. Results with stronger perturbation are presented in appendix A. It's important to emphasize that only the inner nodes and the nodes associated with 1D stencils and 2D planar faces undergo perturbations during this deformation process. Subsequently, after introducing perturbations to the mesh, we apply the line sweeping method.

**2.2.1 Cubic domain** The initial result we present begins with a perturbed mesh of a cubic domain, as depicted in fig. 8.a. After undergoing several iterative regularization steps, the nodes within this mesh achieve the desired uniform distribution, as illustrated in fig. 8.b. The resulting mesh converges back to the initial uniform state. The method successfully removes the perturbation and enhances the mesh quality.

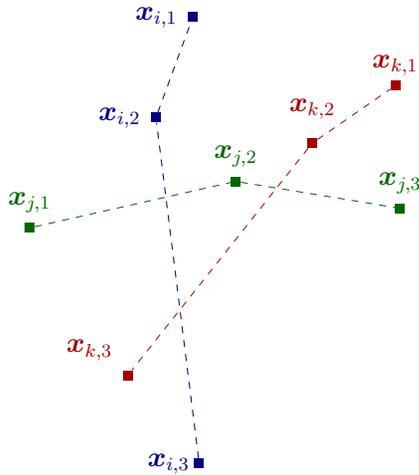


Figure 6: Regularization of a 3D stencil : blue, green and red points are the 2D equal-space-points of in each direction.

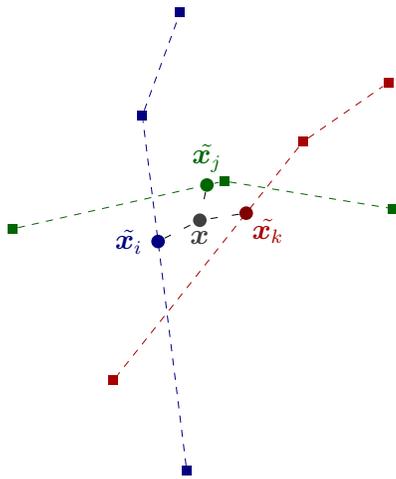
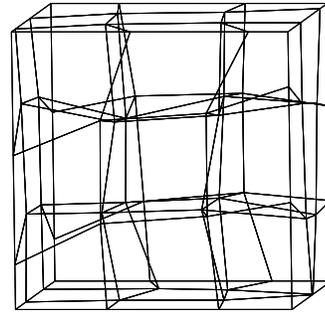
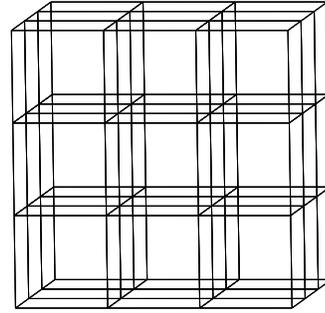


Figure 7: Regularization of a 3D stencil : blue point  $\tilde{x}_i$ , green point  $\tilde{x}_j$  and red point  $\tilde{x}_k$  are the 1D equal-space-points in the three directions of the stencil.

**2.2.2 Cylindrical domain** The second set of results concerns a cylindrical domain meshed and randomly perturbed, as depicted in fig. 9.a. Following a series of iterative regularization steps, the nodes within this mesh attain the desired even distribution, as exemplified in fig. 9.b. The proposed method demonstrates its capability to produce elements of uniform size, even in the presence of concave boundaries. This stands in contrast to the conventional equi-potential relaxation-based method [9, 8] or the regularization based on functional minimization like in [5, 6], which may exhibit grid attraction phenomena when applied to curvilinear meshes, resulting in reduced mesh quality near concave



(a)



(b)

Figure 8: Regularization of a cubic domain mesh.

boundaries.

### 2.3 Regularization of singular points in block structured mesh

Meshing complex geometries in a structured manner often leads to undesirable mesh distortion. The strategic placement of nodes, referred to as singularities, in regions where the grid structure is disrupted can substantially mitigate overall mesh distortion and enhance its practicality. A block-structured mesh is subdivided into discrete sections known as blocks, with each block containing a regular grid. In certain cases, such as when dealing with circular or curved geometries, a special grid pattern called an 'O-grid' may be employed. Singularities often emerge at the intersections of these blocks. It's important to note that the regularization of a singular stencil (fig. 10) differs from that of a regular stencil due to the incomplete nature of the stencil compared to structured one.

The objective is to apply the method previously discussed in section 2.1. A key distinction between a standard 2D stencil and a singular 2D stencil is the number of neighboring cells connected to the central node. In the case of the singular stencil (as shown in fig. 11.b), the central node is linked to only three neighboring cells, whereas the regular stencil (as depicted in fig. 11.a) features four neighboring cells.

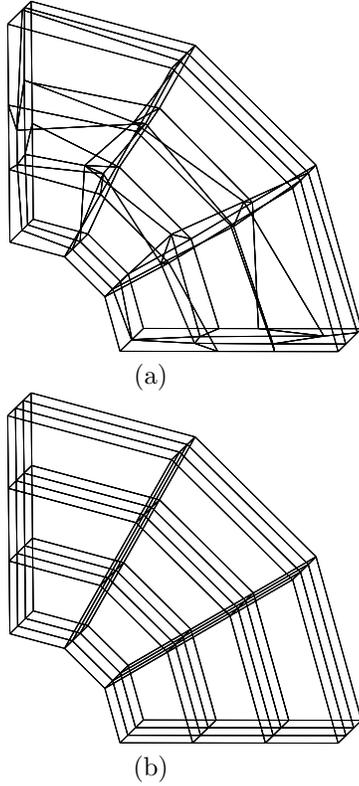


Figure 9: Regularization of a cylindrical domain mesh.

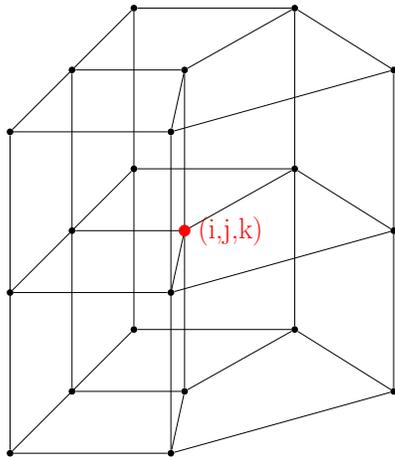


Figure 10: Stencil for a singular node (singular stencil).

The main concept involves considering a singular stencil as an extreme case of a regular stencil. Specifically, a node situated at a corner of the singularity formed by points A, B, or C (as shown in fig. 11.b) can be conceptualized as the merging of three individual nodes. Consequently, for each corner A, B, or C,

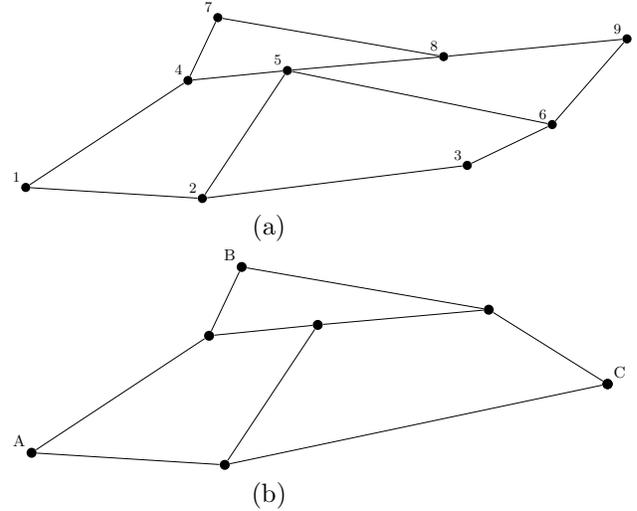


Figure 11: Comparison between regular 2D stencil (a) and singular 2D stencil (b).

it becomes possible to reconstruct a regular stencil, as depicted in fig. 12. The case in fig. 12.a corresponds to the situation where points 1, 2 and 3 of fig. 11.a merge to the same position and we have  $A = 1 = 2 = 3$ . Same for fig. 12.b where points 7, 8 and 9 of fig. 12.a merge to the point  $B$ . Finally, in fig. 12.c points 3, 6 and 9 of fig. 12.a merge to the point  $C$ .

The method detailed in section 2.1 can be applied in conjunction with these three regular stencils. The equal-space-point  $\mathbf{x}$  is positioned as the arithmetic mean of the three regularized points  $\tilde{\mathbf{x}}_A$ ,  $\tilde{\mathbf{x}}_B$ , and  $\tilde{\mathbf{x}}_C$ :

$$\mathbf{x} = \frac{1}{3} (\tilde{\mathbf{x}}_A + \tilde{\mathbf{x}}_B + \tilde{\mathbf{x}}_C).$$

The methodology for regularizing 3D singular stencils remains consistent with that described in section 2.1.

**2.3.1 Shell domain** The method developed for the regularization of singular stencils is now tested with a shell domain mesh. This mesh consists of block-structured meshes with singular nodes. After several regularization iterations, the singular nodes of the perturbed mesh (fig. 13.a and fig. 14.a) migrate towards the center of the domain, resulting in each block being uniformly meshed in the final regularized mesh (fig. 13.b and fig. 14.b).

Mesh tangling issues are also effectively addressed by our method, particularly in cases involving singular points within block-structured meshes. This is demonstrated in the mesh examples presented in appendix A, illustrated in fig. 25.

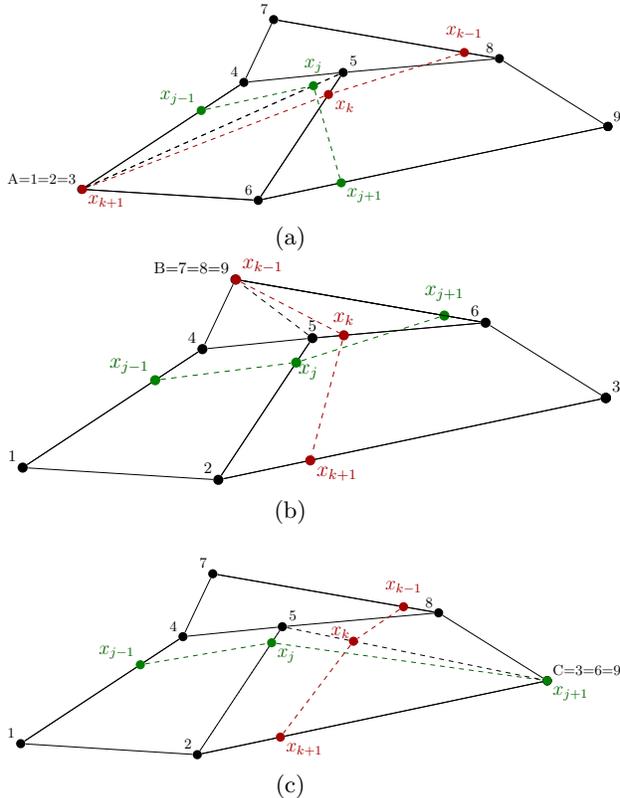


Figure 12: The three regular 2D stencils used to regularize the singular 2D stencil.

## 2.4 Regularization of a distorted Lagrangian mesh

The primary objective of the method presented in the previous sections is to efficiently regularize strongly distorted Lagrangian meshes within an ALE simulation code. To illustrate its effectiveness, we present the results obtained with the method for the triple point case after numerous Lagrangian iterations. This test case [4] aims to simulate the complex behavior observed in shock physics where three different states intersect at a single point, creating a triple point scenario. The mesh used in this test case corresponds to the real Lagrangian solution of the triple point problem with only one material and three different states. This challenging simulation involves the interaction of multiple shocks, resulting in the generation of vortices and the formation of tangled cells. Notably, at the triple point where the three existing domains coincide, the mesh quality is compromised due to these complex interactions.

As observed in fig. 15.b and fig. 15.d, the maximum aspect Frobenius mesh quality metric in Paraview is noticeably improved, and mesh crossings become untangled when applying the regularization method. However, the mesh is smoothed as a consequence. The regu-

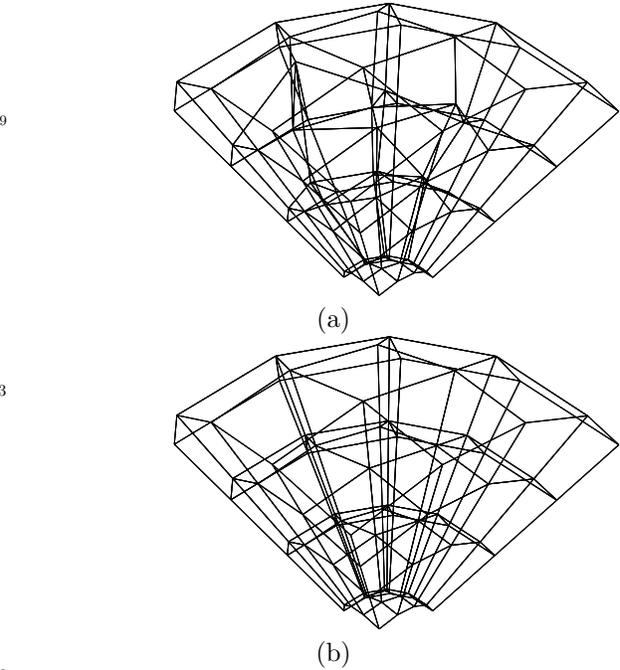


Figure 13: Regularization of a block structured mesh with singular stencils at the junction between three blocks (view from side).

larization process causes nodes to move significantly far from their Lagrangian positions, as depicted in fig. 15.a and fig. 15.c. Specifically, the method regularizes the mesh towards a uniform distribution, resulting in a loss of accuracy in the Lagrangian step.

The challenge lies in determining when to halt the iterative process to strike a balance between remaining close to the Lagrange mesh while achieving a satisfactory mesh quality. Ideally, the method should converge towards a non-uniform mesh that is closely aligned with the initial Lagrangian mesh while maintaining good mesh quality. This example underscores the need for the development of a new regularization method, which is the focus of the following section.

## 3 Weighted Regularization Method

As shown in the test case of section 2.4, the regularisation method introduced by Jin Yao corrects tangled meshes and produces a good quality mesh. However, the new mesh may be very different from the Lagrangian mesh, resulting in a loss of accuracy. Indeed, the method tends to make the mesh uniform while the irregular Lagrangian mesh contains interesting physical features. To address this issue and to ensure that the regularized mesh remains close to the Lagrangian mesh,

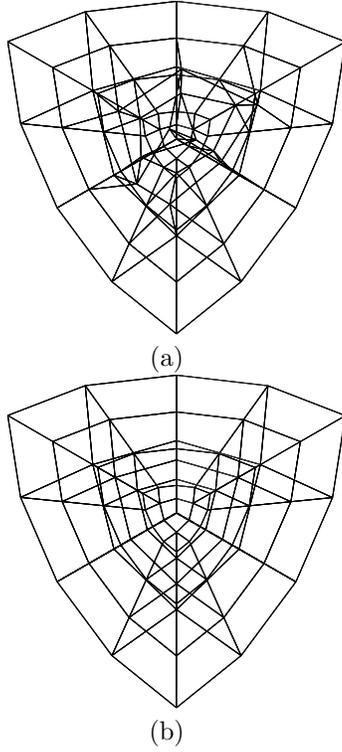


Figure 14: Regularization of a block structured mesh with singular stencils at the junction between three blocks (view from above).

the Weighted Line Sweeping Method is now introduced

### 3.1 Weighted Line Sweeping Method

To prevent the method from rezoning the mesh uniformly, the idea is to introduce a weighting factor to the Line Sweeping Method. Considering  $\Gamma_i \in [0, 1]$  as the corresponding weight, the regularized position  $\mathbf{x}_i^{m+1}$  of  $\mathbf{x}_i^m$  is no longer positioned at the centre of the stencil but is now weighted as displayed in fig. 16.

The weighted regularized point  $\mathbf{x}_i^{m+1}$  is defined as

$$\mathbf{x}_i^{m+1} = (\mathbf{x}_{i-1}^m + \Gamma_i l^m \mathbf{e}_1) \delta_{\{\Gamma_i l^m \leq l_1^m\}} + (\mathbf{x}_{i+1}^m + (1 - \Gamma_i) l^m \mathbf{e}_2) \delta_{\{(1 - \Gamma_i) l^m \leq l_2^m\}}$$

with the same notations as in section 2.1.

### 3.2 Definition of aspect ratios and weights

The aim of the method introduced in the previous section is to regularize a mesh with a given set of aspect ratios into a mesh with another set of known aspect ratios. Indeed, each mesh has its own node distribution and its own aspect ratios.

The aspect ratio  $\gamma_i$  within a 1D stencil (fig. 17) is defined as:

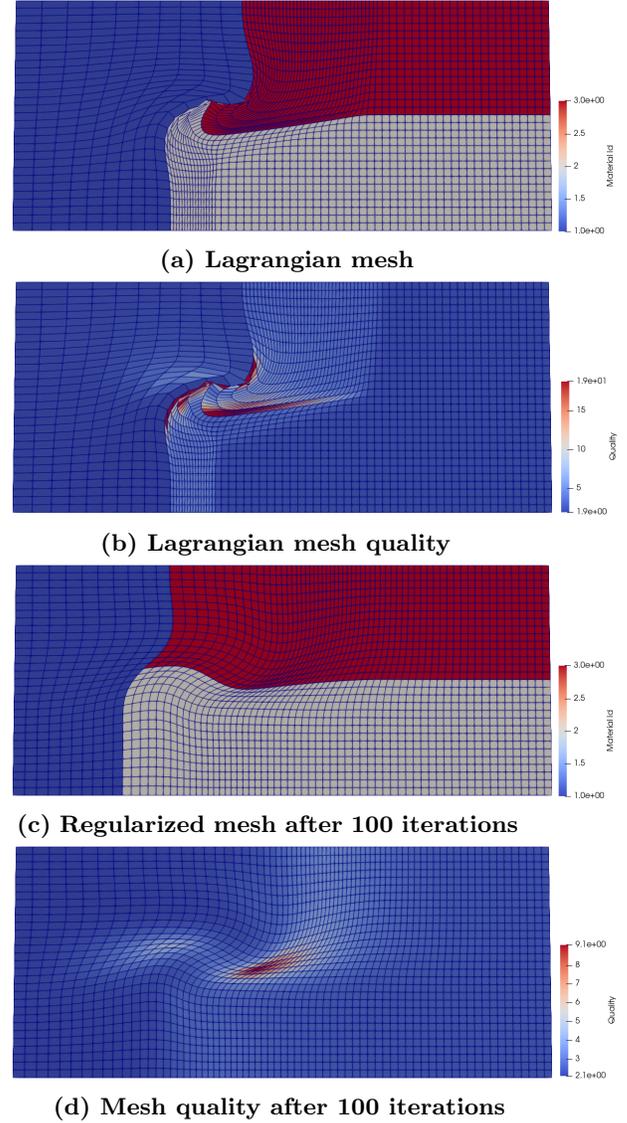


Figure 15: Regularization of a Lagrangian mesh with  $70 \times 30 \times 2$  cells.

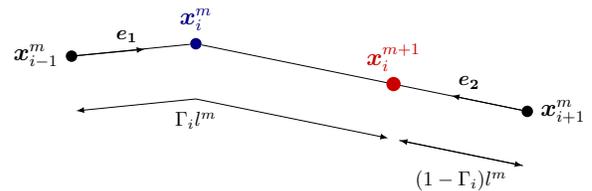


Figure 16: Weighted regularization of 1D stencil.

$$\gamma_i = \frac{l_1}{l_1 + l_2}, \quad l_1 = |\mathbf{x}_i - \mathbf{x}_{i-1}|, \\ \text{and} \quad l_2 = |\mathbf{x}_i - \mathbf{x}_{i+1}|.$$

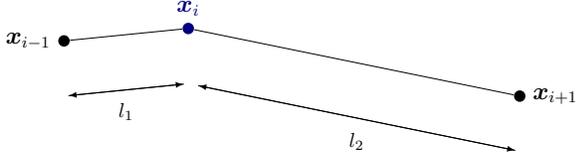


Figure 17: The initial aspect ratio of a 1D stencil is defined as  $\gamma_i = \frac{l_1}{l_1 + l_2}$ .

One should note that the weight  $\gamma_i$  is limited to the range  $[0, 1]$ . To regularize a 1D mesh initially defined by a set of aspect ratios  $\gamma_i$  into a new 1D mesh with predefined aspect ratios  $\Gamma_i$ , we initialize the aspect ratios  $\Gamma_i$  as regularization weights. During regularization iteration, the aspect ratio  $\gamma_i^m$  undergoes changes and gradually converges towards the target weight  $\Gamma_i$ . This process is iterative because the nodes  $x_{i-1}^m$  and  $x_{i+1}^m$  move independently in relation to  $x_i^m$ , leading to the general case where  $\gamma_i^{m+1} \neq \Gamma_i$ . The approach for regularizing 2D and 3D stencils remains consistent, incorporating the introduction of weights. In a 2D stencil, where the central node is part of two 1D stencils, two aspect ratios can be defined, and two weights  $\Gamma_i$  and  $\Gamma_j$  can be associated with the node. Similarly, in a 3D stencil, where the central node is part of three 1D stencils, three aspect ratios can be defined, and three weights  $\Gamma_i$ ,  $\Gamma_j$ , and  $\Gamma_k$  can be linked to the node.

The weight computed directly from the mesh can also be limited. It's important to note that when  $\Gamma_i = \Gamma_j = \Gamma_k = 0.5$  for all  $i$ , the Weighted Line Sweeping Method coincides with the Line Sweeping Method introduced in section 2.1. On the other hand, if the weighted line sweeping method is applied to a non-uniform mesh with initial weights, the mesh will not move, as the initial grid is a solution of the regularization algorithm. This behavior is completely different from the equal space method. To achieve an intermediate state between these different behaviors, the initial weights can be limited using a  $\Gamma_{lim}$  coefficient. If we define  $\Gamma_{min} = \Gamma_{lim}$  and  $\Gamma_{max} = 1 - \Gamma_{lim}$  then we can compute new weight  $\tilde{\Gamma}$  using equation

$$\tilde{\Gamma}_i = \Gamma_{min}\Gamma_i + (1 - \Gamma_i)\Gamma_{max}$$

The  $\Gamma_{lim}$  coefficient should be chosen within the range of  $[0, 0.5]$ . The limitation process yields new weights, denoted as  $\tilde{\Gamma}_i$ , which are in the interval  $[\Gamma_{min}, \Gamma_{max}]$ . Specifically, if  $\Gamma_{lim}$  is set to 0, the initial weights remain unchanged. Conversely, when  $\Gamma_{lim}$  is set to 0.5, the method converges towards the equal space approach.

### 3.3 Weighted regularization of cubic domain

The effectiveness of the weighted method is evaluated by demonstrating its capability to generate non-uniform meshes. In the context of this evaluation, a cubic domain featuring a non-uniform mesh is illustrated in fig. 18.b. To initialize the weights  $\Gamma_i$ ,  $\Gamma_j$ , and  $\Gamma_k$  for each node within this mesh, the aspect ratios of this mesh are computed and utilized. If the weighted line sweeping method is directly applied to this non-uniform mesh with initial weights, the mesh will not move. Subsequently, this mesh is subjected to random perturbation, as depicted in fig. 18.a, and the weighted line sweeping method is applied. Through this process, the perturbed mesh converges to the mesh shown in fig. 18.b which was the initial mesh without perturbation. Again, the method is able to correct tangled mesh, as demonstrated in appendix A in fig. 26.

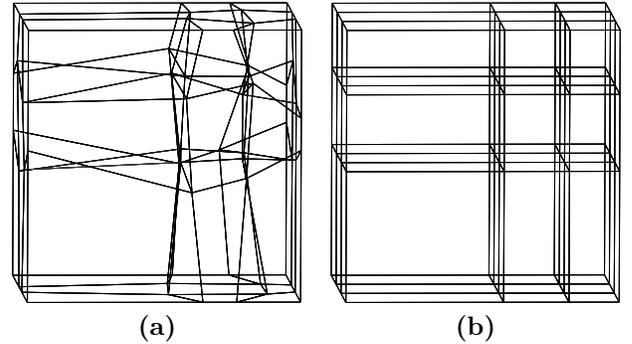


Figure 18: Regularization of a non-uniform (in two directions) mesh.

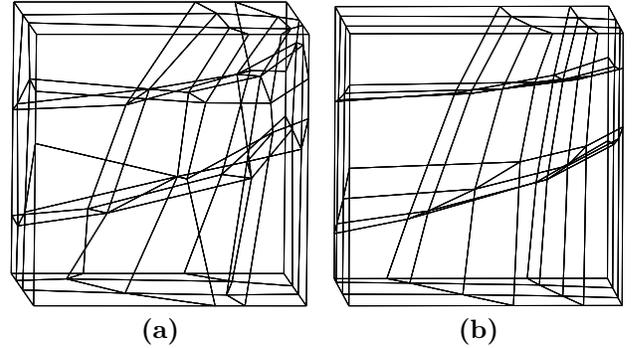


Figure 19: Regularization of a 3D non-uniform mesh.

Considering a fully 3D non-uniform mesh in fig. 19.b, we apply the same methodology. The perturbed mesh shown in fig. 19.a is then rezoned into the mesh displayed in fig. 19.b. Consequently, through this process, the perturbed mesh converges back to the original mesh shown in fig. 19.b, which was the same as the initial unperturbed mesh.

### 3.4 Weighted limitation of a vortex

In this section, we introduce a new test case inspired by the Taylor Green vortex [2]. This test case is particularly challenging for Lagrangian methods since the mesh is severely distorted but not tangled. For this test case, we use the limiting coefficient introduced in section 3.2.

In fig. 20.a, the initial Lagrangian mesh is presented. If we set the limiting coefficient to  $\Gamma_{lim} = 0$ , then the mesh does not move, regardless of the number of iterations performed. Subsequently, the limiting coefficient is set to  $\Gamma_{lim} = 0.25$ , and after 50 iterations of regularization, the resulting mesh is shown in fig. 20.b. Finally, the limiting coefficient is set to  $\Gamma_{lim} = 0.5$ . In this case, the weighted method is equivalent to the classical line sweeping method. After 50 iterations, the regularized mesh is presented in fig. 20.c.

### 3.5 Weights adapted to the Lagrangian mesh

In the previous section, we had the advantage of knowing the weights  $\Gamma_i$ ,  $\Gamma_j$ , and  $\Gamma_k$  at each node because the desired regularized mesh was predetermined. However, in this section, we introduce a method for defining the weights used to regularize a Lagrangian mesh. The Lagrangian mesh possesses its unique node distribution and is characterized by its specific set of aspect ratios, denoted as  $\gamma_{L,i}$ ,  $\gamma_{L,j}$ , and  $\gamma_{L,k}$  at each node. This collection of aspect ratios contains valuable physical information and is now employed to initialize the weights  $\Gamma_i$ ,  $\Gamma_j$ , and  $\Gamma_k$  for the mesh regularization procedure. To determine these weights  $\Gamma_i$ ,  $\Gamma_j$ , and  $\Gamma_k$  associated with the central node of a stencil, a method is used whereby the Lagrangian aspect ratios of neighboring nodes are averaged, and an iterative process is applied to refine these weights for optimal results.

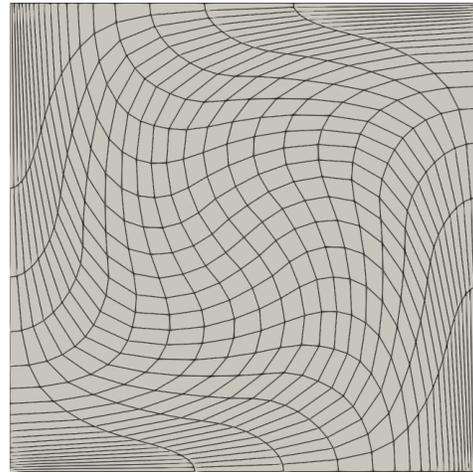
For a 2D stencil (fig. 21), the weights  $\Gamma_i^p$  and  $\Gamma_j^p$  are initialized with the Lagrangian aspect ratios  $\gamma_L$  and are computed as follow:

$$\Gamma_i^{p+1} = \frac{\Gamma_{i-1}^p + \Gamma_{i+1}^p}{2} \quad \Gamma_j^{p+1} = \frac{\Gamma_{j-1}^p + \Gamma_{j+1}^p}{2}$$

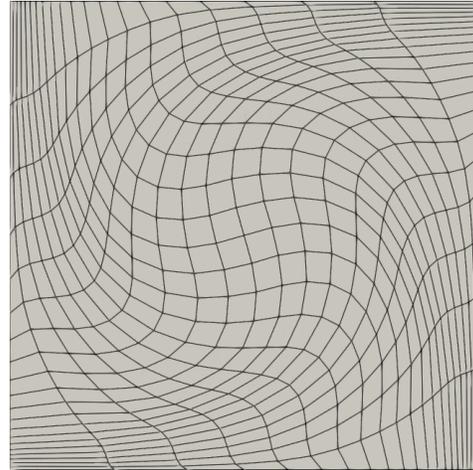
with  $\Gamma_i^0 = \gamma_{L,i}$  and  $\Gamma_j^0 = \gamma_{L,j}$ .

In the case of a 3D stencil, there exist four 1D stencils in each direction (depicted as fully colored lines in fig. 22) aligning with the primary 1D stencil (shown as a colored dotted line in fig. 22) where the weight is intended to be established.

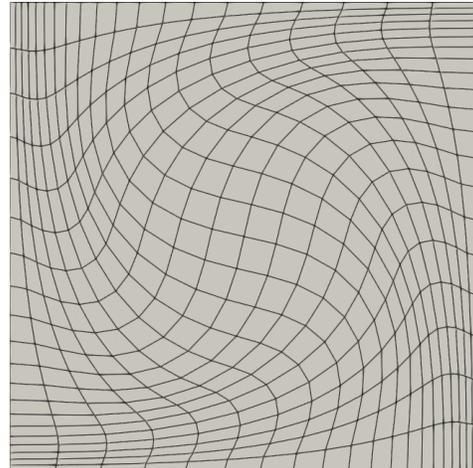
Next, we examine the 2D stencils encompassing these 1D stencils represented by the fully colored lines. We apply the same methodology as for 2D stencils. As an instance, the weights  $\Gamma_i^{p+1}$  are determined by averaging the weights  $\Gamma_{i-1}^p$  and  $\Gamma_{i+1}^p$  which are defined in the 2D stencils containing the 1D stencils as in fig. 21.



(a) Initial Lagrangian mesh



(b) 50 iterations with  $\Gamma_{lim} = 0.25$



(c) 50 iterations with  $\Gamma_{lim} = 0.5$

Figure 20: Regularization of a Lagrangian mesh using limitation coefficient  $\Gamma_{lim}$ .

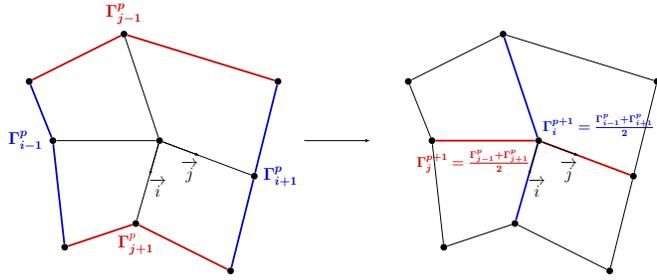


Figure 21: Computation of  $\Gamma_i^{p+1}$  and  $\Gamma_j^{p+1}$  for a 2D stencil.

Once a designated number of iterations  $p$  is completed, the weights become constant and are employed in the mesh regularization process. By incorporating these weights in the mesh regularization, the approach can preserve the distinctive features of the Lagrangian mesh while simultaneously achieving enhanced mesh regularity.

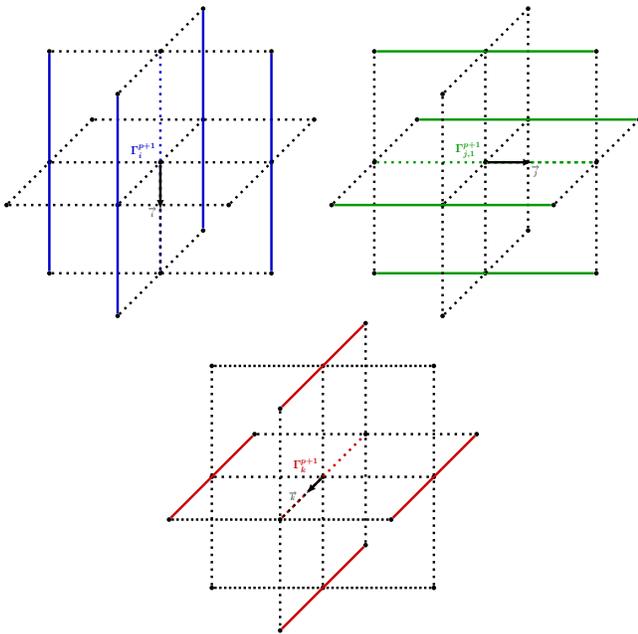
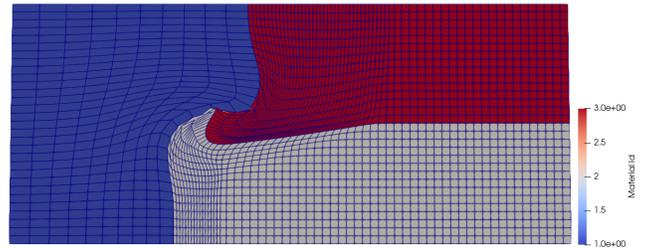


Figure 22: The weights defined at the fully colored lines are used to compute the weight at the colored dotted line.

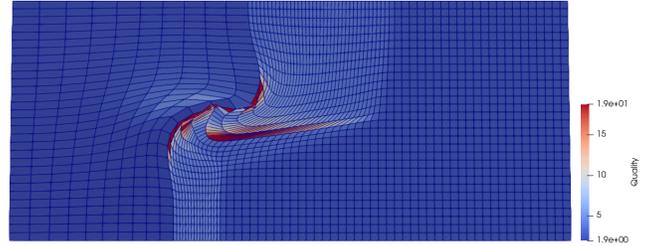
### 3.6 Regularization of a distorted Lagrangian mesh using adapted weights

The aim of introducing weights is to improve the result obtained in the section 2.4. The same test case is considered using the weighted regularization method. Prior to regularisation, the weights are fixed after 100

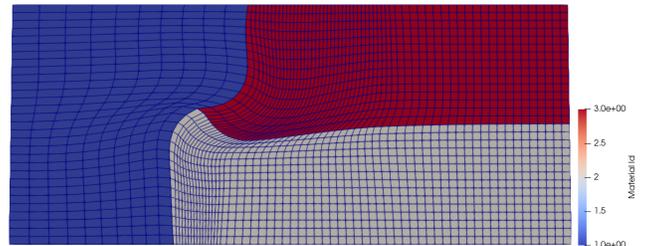
iterations of the method presented in section 3.5.



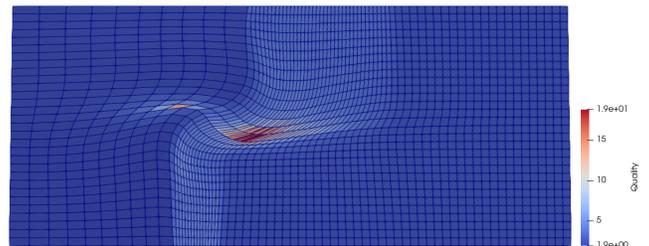
(a) Lagrangian mesh



(b) Lagrangian mesh quality



(c) Regularized mesh after 100 iterations



(d) Mesh quality after 100 iterations

Figure 23: Regularization of a Lagrangian mesh with  $70 \times 30 \times 2$  cells.

The weighted regularization method, as depicted in fig. 23.c and fig. 23.d, successfully maintains the aspect ratio of the Lagrangian mesh while enhancing the maximum aspect Frobenius quality metric. Notably, the regularization process does not necessarily displace nodes away from their original Lagrangian positions, especially along the edges where nodes remain stationary. It's worth mentioning that the number of iterations required for adjusting the weights may vary in ALE test cases. However, in scenarios such as the one described

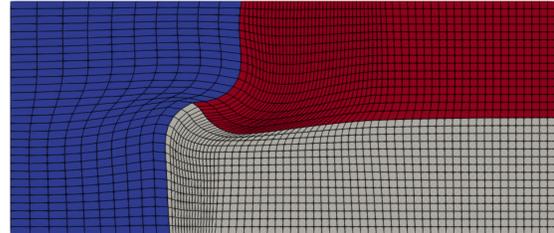
here, where cells become tangled at domain intersections, multiple iterations are necessary for effective regularization.

### 3.7 Regularization of a distorted Lagrangian mesh using adapted weights and limitation

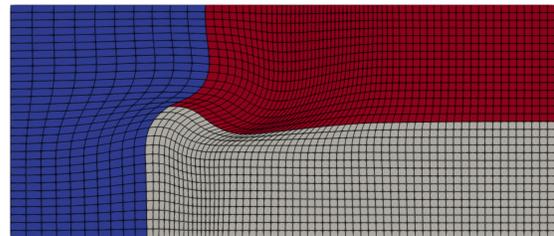
In this section, we introduce adapted weights coupled with a limitation. The purpose is to present regularization results intermediate to those obtained by limiting with initial weights computed on a Lagrangian geometry, as shown in fig. 24.a, and the results in fig. 24.c with a limitation coefficient  $\Gamma_{lim} = 0.5$ . In all the cases presented in fig. 24, adapted weights from section 3.5 are employed with 100 iterations, followed by the limitation step outlined in section 3.2. For fig. 24.a, the limitation coefficient is set to  $\Gamma_{lim} = 0$ . In this case, we recover the results presented in section 3.6. The limitation, corresponding to the classical line sweeping method with  $\Gamma_{lim} = 0.5$ , is presented in fig. 24.c. The case, representing a balance between fig. 24.a and fig. 24.c, is obtained using a limitation coefficient  $\Gamma_{lim} = 0.25$  and is presented in fig. 24.b.

## 4 Conclusion

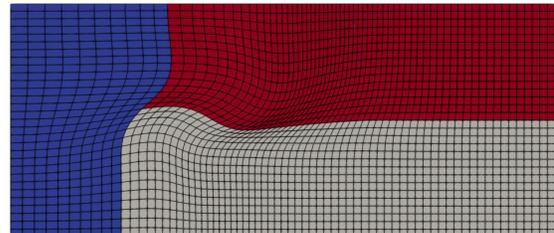
When solving the Euler equations, the use of the Lagrangian formalism can lead to significant mesh deformations. Thus, an efficient regularization method is necessary for the indirect ALE strategy in order to prevent tangled cell and mesh quality decrease. The Line-Sweeping method [10, 11] was first considered because of its simplicity based on geometric considerations. However, despite its efficiency, it has been observed that the final regularized mesh is uniform and may be far from the initial Lagrangian one (even if it is not necessary). To prevent the method from rezoning the mesh uniformly, weights were introduced in the Line-Sweeping method. More precisely, the weighted method allows to regularize an initial mesh with a given set of aspect ratios into a new mesh with another set of known aspect ratios. Here, a key idea consists in using the aspect ratios of the initial Lagrangian mesh to define the weights for the regularisation step. This new weighted method is promising as it yields results close to the Lagrangian ones but with a better mesh quality. In addition, it can successfully handle unfeasible cases with the Lagrangian formalism as the triple point test case. Various perspectives are to be considered. For example, the regularisation process can still be improved combining the Line-Sweeping method with the weighted one. Also, the regularization method should be coupled to a hydrodynamic code in order to dynamically regularize the mesh while solving gas dynamics equations under the Lagrangian formalism.



(a) Regularized mesh after 100 iterations and  $\Gamma_{lim} = 0$



(b) Regularized mesh after 100 iterations and  $\Gamma_{lim} = 0.25$



(c) Regularized mesh after 100 iterations and  $\Gamma_{lim} = 0.5$

Figure 24: Regularization of a Lagrangian mesh with adapted weights and limitation.

## References

- [1] V. DOBREV, P. KNUPP, T. KOLEV, K. MITTAL, R. RIEBEN, AND V. TOMOV, *Simulation-driven optimization of high-order meshes in ale hydrodynamics*, *Comput. Fluids*, 208 (2020), p. 104602.
- [2] V. DOBREV, T. KOLEV, AND R. RIEBEN, *High-order curvilinear finite element methods for Lagrangian hydrodynamics*, *SIAM J. Sci. Comput.*, 34 (2012), pp. 606–641.
- [3] M. B. FRIESS, J. BREIL, P.-H. MAIRE, AND M. SHASHKOV, *A multi-material ccafe-mof approach in cylindrical geometry*, *Commun. Comput. Phys.*, 15 (2014), p. 330–364.
- [4] 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 (2010), pp. 5755–5787.
- [5] 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. Meth. Engng.*, 48 (2000), pp. 401–420.
- [6] P. KNUPP, L. G. MARGOLIN, AND M. J. SHASHKOV, *Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods*, *J. Comput. Phys.*, 176 (2002), pp. 93–128.
- [7] C. ROCHE, J. BREIL, AND M. OLAZABAL, *Mesh regularization of ablating hypersonic vehicles*.
- [8] A. WINSLOW, *Equipotential zoning of two-dimensional meshes*, Tech. Rep. UCRL-7312, Lawrence Livermore National Laboratory, 1963.
- [9] A. M. WINSLOW, *Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh*, *J. comput. phys.*, 1 (1966), pp. 149–172.
- [10] J. YAO, *A mesh relaxation study and other topics*, LLNL-TR-637101, (2013).
- [11] J. YAO AND D. STILLMAN, *An equal-space algorithm for block-mesh improvement*, *Procedia Engineering*, 163 (2016), pp. 199–211. 25th International Meshing Roundtable.

### A Regularization of meshes with tangled cells

In this section, we aim to emphasize the capability of the method to untangle 3D meshes in scenarios involving shell mesh (as depicted in fig. 25) or non-uniform mesh (as illustrated in fig. 26). For all test cases, we deliberately introduce substantial random perturbations, resulting in the formation of tangled cells. We then apply the regularization process. After 40 iterations, the mesh becomes untangled, and a perfectly uniform mesh is restored.

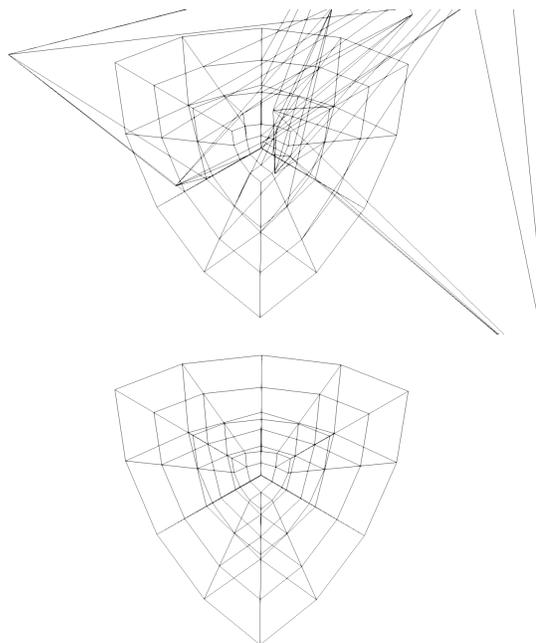


Figure 25: Regularization of a shell domain mesh (view from above) with tangled cells after 40 iterations (Line Sweeping Method)

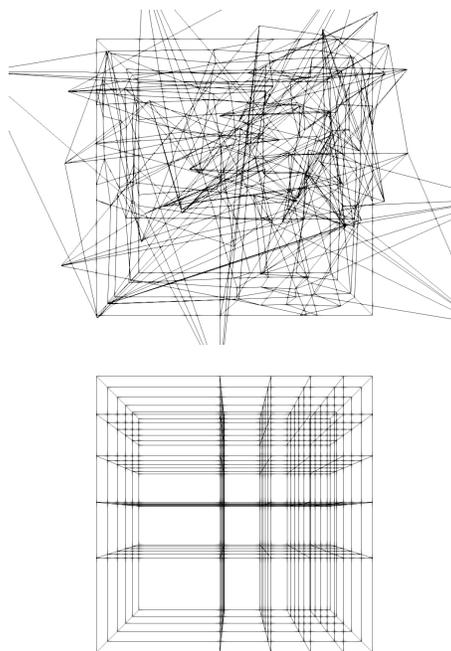


Figure 26: Regularization of a cubic domain mesh with tangled cells after 40 iterations (Weighted Line Sweeping Method)