A cell-centered AMR-ALE framework for 3D multi-material hydrodynamics. Part II: linesweep ALE rezoning for nonconformal block-structured AMR meshes

A. Colaïtis[1,2], S. Guisset[3], J. Breil[3]

Abstract. The simulation of flows presenting contact discontinuities, vorticity, and large variations in spatial scales can be performed in a framework coupling Arbitrary Lagrangian Eulerian (ALE) algorithms and Adaptive Mesh Refinement (AMR). This coupling requires adaptation of ALE rezoning techniques to meshes containing nonconformal nodes arising from both the AMR topology and the junction of mesh blocks. In this paper, we present an ALE rezoning strategy that is compatible with such meshes, and that can also act as a disentangling algorithm. Emphasis is put on an algorithm that respects intrinsic Lagrangian mesh properties in order to preserve accuracy around discontinuities. To that end, we adapt the weighted linesweep algorithm to nonconformal block-structured AMR meshes. Then, we present control parameters introduced in the method for it to be applicable in practical situations. Notably, the method is coupled to a specific metric optimization in order to palliate some shortcomings of the linesweep method. Finally, numerical test cases are presented that feature the capabilities of the ALE-AMR algorithm for flows that present discontinuities, vorticity, and a variety of scales. Notably, we show that our ALE-AMR algorithm gives results at least similar to Euler-AMR, but provides better accuracy in cases where discontinuities are involved, thanks to a method that respects the Lagrangian features of the mesh. Additionally, it enables Euler-AMR-like computations on domains with temporally varying domain boundaries.

1. Introduction

The modeling of flows is prevalent in the study of natural phenomena and in engineering applications. In many cases, these flows involve processes that occur at a variety of spatial and temporal scales. It is often the case that the processes occurring at the smaller scales affect and are affected by the dynamics occurring at the larger scales. Such problems are well suited to *Adaptive Mesh Refinement* (AMR) methods, in which the mesh resolution dynamically evolves during the calculation. AMR methods are, for the most part, implemented within direct Euler algorithms, in which the Euler equations are resolved in the laboratory frame onto a structured mesh of cells. The direct Euler-AMR approach has been widely explored and presents significant advantages, notably related to ease of implementation and scalability. However, it suffers from drawbacks for modeling discontinuities, such as material interfaces and shocks. This is a strong limitation that is usually addressed by using higher-order schemes and/or higher mesh resolutions, eventually enabled by the use of AMR itself. Despite these advanced methods, approaches based on direct-Euler can exhibit reduced accuracy compared to Lagrangian approaches. A relevant example here is the accurate modeling of shocks and rarefaction waves, see e.g. results presented in Refs [1, 2, 3] for typical relevant test cases.

In general, the modeling of flows that present discontinuities is best addressed using the Lagrangian approach. In that case, the mesh cells naturally follow the material flow, leading to better resolution and

[1]Laboratory for Laser Energetics, 250 East River Rd, Rochester, NY 14623-1212
[2]Corresponding author: acol@lle.rochester.edu
[3]CEA/CESTA, 15 Avenue des Sablières CS 60001, 33116 Le Barp cedex, France

accuracy around discontinuities. The most prominent drawback of the Lagrangian method is its inability to model flows presenting significant vorticity, in which case the mesh quality will degrade significantly and lead to invalid mesh cells, thus stopping the calculation. As such, Lagrangian schemes are usually augmented by algorithms aimed at controlling and improving mesh quality, the so-called *indirect Arbitrary Lagrangian Eulerian* (ALE) approach. In indirect ALE, the hydrodynamic phase is resolved using a Lagrangian algorithm. It is followed by a regularization phase in which a new mesh is computed and a re-mapping algorithm projects the quantities from the Lagrangian mesh to the rezoned mesh. When the rezoned mesh is fixed, the algorithm is then operating in an indirect-Euler regime. In the general case, however, the aim is to keep the ALE rezoned mesh as close as possible with the Lagrangian one, in order to preserve accuracy around discontinuities. Since ALE schemes are based on an underlying Lagrangian scheme, they are usually formulated for unstructured meshes. They may however not be readily compatible with non-conformal nodes introduced by AMR meshes.

The various strengths and weaknesses of each of these approaches — direct-Euler, Lagrangian and ALE schemes — are the motivation behind the formulation of a 3D ALE-AMR method. The goal here is to leverage the Lagrangian formalism for the modeling of discontinuities, utilize the ALE framework to control the mesh quality in regions of strong vorticity, and use AMR to dynamically adapt the mesh resolution to spatially resolve the relevant processes. This approach is interesting because it is also generic: an implementation of indirect ALE-AMR allows one to carry out indirect-Euler-AMR calculations and Lagrangian-AMR calculations using the exact same algorithms. In addition, computing the hydrodynamic phase using a Lagrangian approach eases the implementation of physics models because the equations are resolved in the frame of the flow. In part I of this work[4], we have presented the algorithms for Lagrangian-AMR and indirect Euler-AMR in the 3D cell-centered framework. These algorithms included: management of a dynamic mesh topology created from a forest of octrees, conservative refinement and coarsening algorithms compatible with the cell-centered Lagrangian framework, refinement criteria suited for the Lagrangian-AMR approach, and a multimaterial remapping method for both the ALE and AMR remapping phases. In this Part II, we focus on ALE techniques adapted to the AMR framework, namely, *rezoning* and *mesh disentangling*. Notably, we recall that the disentangling method is part of the Lagrangian-AMR algorithm itself, as presented in Part I. This is because both Lagrangian coarsening and ALE preprocessing for Lagrangian refinement can lead to invalid meshes. Although the focus of this paper is on ALE-AMR rezoning, we will show that our adapted algorithm is also an ALE-AMR mesh disentangling algorithm.

Central to our ALE-AMR approach is also the ability to track multi-material interfaces accurately. There are two main strategies to deal with interfaces between materials. The first one, called diffuse interface method, consists in considering the interface between materials as diffused zones in which the materials are allowed to mix together (see e.g. [5, 6] for more details). This method has the advantage of being relatively simple and to facilitate the incorporation of additional physics models such as surface tension, drag forces, etc. The downside of diffuse interface approaches is that it introduces significant numerical smearing by construction, and must rely on anti-diffusive strategies. The second main approach is the sharp interface method, in which materials are not allowed to mix and the interface between the materials is tracked and reconstructed. Although this second method adds significant difficulties relating to the 3D geometrical interface reconstruction steps, it is also significantly more accurate and less diffusive. In this paper, we make use of a 3D sharp-interface reconstruction method, in which the separation between materials is done with planar interfaces. The method was detailed in Part I [4].

Control of the mesh quality in ALE methods can be achieved through various approaches. We briefly recall here some of the most used ones. One of the first approach was proposed by Winslow [7], and consists in smoothing the mesh by treating node positions as solutions of a carefully chosen partial differential equation system with a monitor function. While this method is usually stable and robust, it tends to yield very smooth mesh distributions. The method has limited flexibility as it does not directly incorporate desired element shape metrics. Because it is essentially a diffusion-like approach, it may also fail for complex distortion scenarios. A second method consists in explicitly defining mesh quality metrics that represent desired mesh properties and then optimizing the value of that metric by moving the nodes of the mesh. These metrics can be node-based [8, 9] and cell-based [10], and their optimization is usually performed through iterative systems. This metric-based optimization method can achieve targeted, high-quality meshes but suffers from

some drawbacks. It is dependant on the choice of the metric, which is not always straightforward. Especially, there can be multiple competing adaptation goals; size, smoothness, shape constraints, aspect ratios, etc. In that framework it is difficult to respect the Lagrangian mesh while guaranteeing element quality in regions of high vorticity. Another category of popular rezoning method is the Target-Matrix Optimization Paradigm (TMOP) [11]. TMOP defines an objective based on comparing the current element transformation to a target transformation. Several considerations can be encoded in the target matrices, such as desired element size, aspect ratios, skewness, anisotropy, etc. The main limitation of the method is that, though flexible, one must define a target matrix that correctly encodes the desired mesh characteristics, similarly to the metrics-based optimization method. This shared limitation of defining metrics or ideal elements can sometime be challenging for the preservation of Lagrangian mesh properties. It also makes these methods less generic. The AMR framework introduces additional difficulties in rezoning algorithms that rely on mesh metrics or encoding ideal elements. These are related to sudden variations in cell volume caused by variations in refinement depth, which are not a signature of a bad quality mesh and should be accounted for in the definition of the metrics. The nonconformal mesh also introduces a large variety of cell types that depends on the local refinement configuration. Finally, topologically hanging nodes, i.e. nodes which topologically split edges and faces, may also change the behavior of usual metrics and increase the complexity of the node movement algorithms.

Given these drawbacks and challenges, metric-based methods seem at first less attractive for the ALE-AMR framework. A different approach to mesh smoothing was recently proposed in Refs. [12, 13], called equal-space linesweep algorithm, with the goal of formulating a simpler and more generic method for improving mesh quality. This purely geometric approach uses the topological structure of the mesh itself to define organization patterns along logical lines. As such, it does not rely on the choice of a metric and on an optimization method. It is therefore attractive for its simplicity and generic aspect. In addition, it guarantees mesh validity by construction, which make it possible to adapt it for disentangling purposes. However, the method as proposed in Ref. [12] is poorly adapted to Lagrangian meshes, because it quickly regularizes meshes toward a uniformly-spaced ones. To that end, recent work in Ref. [14] proposed an adapted version that uses smoothed aspect ratios computed over the neighborhood to retain the local Lagrangian qualities of the mesh. This approach, called weighted linesweep algorithm, has not yet been adapted to nonconformal meshes and still suffers from some fundamental limitations occurring for strongly rotating flows.

In this paper, we adapt the weighted linesweep algorithm to a larger subset of mesh types. Namely, the method we present is applicable to any mesh with nonconformal nodes arising from the AMR construction of the mesh, and with nonconformal nodes arising from the junction of different mesh-blocks (i.e. macro-domains). The only requirement in the algorithm presented here is that the mesh remains 2:1 corner-balanced, i.e. that the AMR depth does not change by more than one level between cells sharing a node. This type of mesh arises from the forest-of-octree structure used in the 3D *p4est* AMR library [15], which we use here as described in Part I. The remaining limitations of the method are addressed by combining the linesweep with conventional metric optimization using a metric that is specifically formulated to address the limitations of the linesweep. We then present a variety of test cases that feature the advantages of ALE-AMR over the indirect Euler-AMR strategy. Notably, we show cases that feature discontinuities and vorticity at the same time, in which case we are looking to retain accuracy around discontinuities while ensuring mesh quality around vortices.

This paper proceeds as follows. We start by summarizing the key points of the 3D cell-centered Lagrangian AMR scheme in Sec. 2, including the AMR methodology and the representation of the mesh as a parallel forest of octants. Then, we introduce a linesweep algorithm for rezoning of AMR meshes in Sec. 3. We start by describing the linesweep method itself, followed by its weighted formulation, and the adaptations made for nonconformal block-structured AMR meshes. Applications of the linesweep in practical situations requires control methods and to address some of its limitations. This is described in Sec. 4, alongside with a linesweep-based disentanglement method. Finally, test cases for ALE-AMR are presented in Sec. 5.

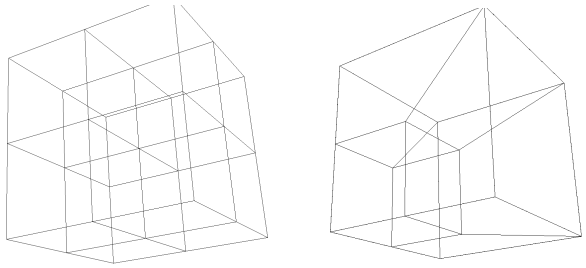


Figure 1: (left) example of a regular base connectivity, paving the 3 logical directions with 2 trees each, resulting in 8 trees. (right) example of a non-regular base connectivity, paving the 3 logical directions with 4 trees in total.

2. Summary of the cell-centered 3D AMR-Lagrangian scheme

We start by giving a brief summary of the framework of this work. The reader is invited to refer to Part I of this paper [4] for more details. The notations used in this paper are the same as in Part I, and will be recalled throughout.

2.1. Multi-material Lagrangian hydrodynamics

This work uses the finite volume approach of hydrodynamics, where all quantities (mass, velocity, total energy) are defined at the center of computational cells. The mesh is an unstructured ensemble of polyhedra (cells) arranged to produce a tessellation of the computational domain. In the 3D framework, cell faces are triangulated by the introduction of a virtual node $p_f^\star$ at the barycenter of each face. This choice enables to establish the relation between conservation of mass and conservation of volume for the cells (Global Conservation Law), while preserving symmetry for the flow. This is the EUCCLHYD scheme, described for the 2D case in Ref. [16] and extended for the 3D case in Refs. [17, 18]. The 3D EUCCLHYD scheme is extended at second order in space using a MUSCL procedure (Monotonic Upstream-centered Scheme for Conservation Laws) [19, 20] and second order in time using the Generalized Riemann Procedure [21] (see Part I and [18]). For the multi-material extension, we consider the equal strain-rate assumption in which all materials are assumed to follow the same deformation, which allows to write a closure relation for the pressure, density, energy and sound speed. A more detailed summary is given in Sec. 1 of Part I [4].

2.2. Lagrangian mesh as a forest of octrees

Following our coupling of AMR to an unstructured ALE framework presented in Ref. [4], we consider a mesh whose topology is described by a forest of octrees. An octree is a collection of octants described in 3D in logically Cartesian space. Each octant is constituted of 6 faces and 8 corners, and may be adjacent to 1 or 4 neighbors per face. Octants can be arbitrarily refined as long as there is no more than one refinement depth difference between octants that are corner-neighbors (2:1 corner-balance condition). Only the deepest octants in an octree are stored in memory. The forest of octrees is obtained by joining together different octrees through their faces. Note that this junction does not need to be logically Cartesian, as is illustrated in Fig. 1. Such configurations add flexibility for meshing computation domains, but also add complexity for the linesweep algorithm presented here. In this work, the topology is handled by the *p4est* library [15].

In the forest of octree approach, the octants are always composed of 8 corners and 6 faces, while the hydrodynamics considers an unstructured mesh of polyhedra defined from their nodes. We described in Part I a procedure used to convert between the two, which notably entails defining physical nodes from the octant's corners. Using this mesh topology, the polyhedra in the mesh have between 6 to 24 faces depending on the refinement configuration, with each face having from 4 to 8 nodes. Some of these nodes are called *hanging* when they topologically touch the face or edge of an octant without being a corner in that octant. Hanging nodes are not treated differently with respect to the hydrodynamic scheme, since the latter only considers the polygonal structure of the cell. However, topologically hanging nodes pose additional difficulties in ALE rezoning schemes, as will be discussed in Sec. 3.

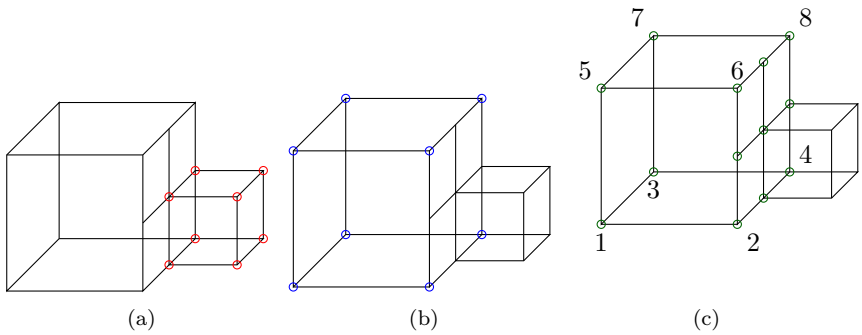


Figure 2: Two cells sharing a face and differing by one AMR level. Red and blue nodes in (a,b) indicate the corner nodes in the right and left cells, respectively. Green nodes in (c) show all the cell nodes in the left cell. The numbering in (c) indicates the local ordering for the corner nodes, following p4est's lexicographic ordering.

Some of our algorithms refer to *corner nodes* while others refer to *mesh nodes*. Corner nodes refer to the 8 nodes created from the topological octant which represents the cell, and are denoted $p \in \mathcal{P}(c)_o$. A cell has always a minimum of 8 nodes, which are these corner nodes. In addition, the corner nodes always follow the same local ordering scheme, which is derived from *p4est*'s lexicographic ordering (see Fig. 2 (c)). Through refinement of itself or neighbors, a cell may acquire more nodes. While these nodes are by definition corner nodes in some cells, they may not be corners in the current cell. More generally, we refer to mesh nodes as all the vertices of the polyhedral cell, which are denoted $p \in \mathcal{P}(c)$. This distinction is illustrated in Fig. 2.

2.3. Multi-material ALE Remapping

The remapping employs a second-order face-sweeping strategy in which fluxes between cells sharing a face are computed by integrating slope-limited fields in the face-swept polyhedra defined from the relation between the two meshes. In the multi-material version, these polyhedra are partial polytopes constructed by representing material boundaries in the cell as planar interfaces (Piece-Wise Linear Construction approach - PLIC [22]) and intersecting these material planes with the face-swept polyhedra. We use a surrogate volume approach [23], in which these partial polytopes are approximated by polyhedra of the same signed volumes and simplified geometry, in order to reduce the complexity of intersection calculations and ensure partial volume conservation.

By construction of the surrogate volume approach and of the face-sweeping strategy, the first-order projection is convex (a demonstration is given in Part I [4]). In order to ensure convexity of the projection at second order, we describe the face fluxes as a convex combination between the first- and second-order projections by defining a limiter on certain quantities (Flux Corrected Remapping - FCR [24]). This FCR is applied to the partial mass, partial momentum and partial internal energy. Conservation of the total energy is also ensured by adding a post-hoc correction based on a projection of the kinetic energy, as was described in Ref. [**?**]. The associated FCR coefficient for the corrected internal energy is described in Part I, Sec 2.8.2, alongside a longer summary on the present remapping method.

2.4. Lagrangian AMR

The AMR procedure consists in a collection of algorithms for re-gridding (refinement or coarsening) and re-partitioning (dynamic change of the parallel domain decomposition across the computing units) and are described in Part I. These algorithms are formulated for 2:1 corner-balanced meshes topologically represented as forests of octants. Notably, the refinement and coarsening algorithms are formulated for volume conservation using interlaced ALE remapping steps to pre-condition the mesh prior to changing the properties of the polyhedral cells. During re-gridding, the projection of quantities from the outgoing to the ingoing mesh is carried out using a second-order scheme inspired from the multi-material ALE remapping framework. Notably, the position of material interfaces is reconstructed during the re-gridding in order to project the partial quantities. The AMR scheme also uses a disentangling procedure in order to ensure

mesh validity during the ALE pre-processing. This procedure is described in the present paper, in Sec. 4.3. Handling of the octant topology, 2:1 balance, and re-partioning is carried out through the *p4est* library [15].

The mesh is refined and coarsened following two criteria. The first criterion, $\mathcal{A}_{o2}$, aims to trigger mesh refinement in regions where the second-order (spatial) scheme is losing accuracy. It is computed by comparing the slope-limited extrapolated values in neighbor cells with the actual values. The second criterion, noted $\mathcal{A}_{\text{diff}}$, aims at triggering refinement in regions where large absolute changes in values are observed. In both cases, the criteria are applied to partial pressure, partial volume fractions, material density, and velocity norm, and only activated when the local flow is above a fraction of the sound speed. These criteria are normalized, and the refinement threshold is denoted T, where a typical value is $T \simeq 0.05$. Coarsening of cells occurs when the criteria falls below $T\epsilon_T$ with $\epsilon_T = 0.2$. The re-gridding criteria and their relative behavior are described in Part I in Secs. 4.3 and 5.2, respectively. For simplicity, all tests presented in this paper use $\mathcal{A}_{o2}$ with $T = 0.02$.

3. An ALE-AMR linesweep algorithm for nonconformal block-structured meshes

The ALE framework is a powerful tool to improve the Lagrangian mesh quality during the calculation. The ALE algorithms can be divided in three components: (i) rezoning, in which a new mesh with desired properties is defined, (ii) disentangling, in which the validity of cells is verified and ensured, and (iii) remapping, in which a conservative projection from the Lagrangian mesh to the new mesh is performed. In this section we discuss rezoning and disentangling algorithms, focusing on robustness and application to AMR meshes. Remapping is conducted using the 3D multi-material framework described in Part I [4].

A common approach for rezoning consists in defining a mesh quality metric and using an optimization algorithm to improve mesh quality by displacing the mesh nodes. This approach can be delicate for two main reasons. First, mesh quality metrics behave in a wide variety of ways which may be counter-productive to the problem at hand, e.g. decreasing mesh resolution in regions of interest or not detecting invalid elements. To remedy this, damping functions are often introduced which restrict node movement during rezoning. This damping can be difficult to define in a generic manner as it is often physics-dependent. Furthermore, depending on the choice of damping functions, the rezoned mesh may not be valid. The second main issue stems from the method employed to optimize the metric function. When a node is displaced during rezoning, it affects the metric function value of the neighbors. In order to avoid solving large non-linear systems, the nodes are usually moved in the direction of decrease of the metric function computed at the previous step. While iterative systems can be employed to converge, inline rezoning usually relies on a single descent step, which does not guarantee decrease of the quality metric due to the displacement of neighbor nodes. Similar issues arise with disentangling, which is essentially a rezoning step with a focus on cell validity.

An alternative rezoning technique, not based on optimization of a metric function, was introduced in Ref. [12]. The method consists in using lines of logically aligned mesh nodes to define rezoned nodes at the middle of segments, and is called *equal space* algorithm. More generally, this class of method can be referred to as a *linesweep* method. In order to respect the shape of the mesh-lines, this equal-space point is taken along the mesh-line and not at the barycenter between adjacent nodes. This process is repeated for each logical direction (e.g. 3 directions in 3D), and an average is taken to define the final node position. The algorithm is iterated until convergence.

This purely geometrical approach is both simple and robust, as it guarantees mesh disentanglement and produces equal-spaced elements at convergence. However, it is not well suited to Lagrangian meshes in its original formulation because it strongly coarsens high resolution regions. To remedy these shortcomings, Refs. [25, 14] reformulated the method with a nodal weighting scheme designed to preserve local aspect ratios. They also introduced a damping algorithm aiming to allow for relaxation of the aspect ratios toward the equal-space point. The simplicity and robustness of this method makes it attractive for the Lagrangian framework. It also provides a simple algorithm to ensure disentanglement of mesh cells. However, the formulations of Refs. [12, 25, 14] are not adapted to nonconformal meshes.

In this section, we present several linesweep-based ALE methods for the framework of nonconformal block-structured meshes built from parallel forests of octrees. This section is organized as follows. First,

we present the original equal-space and weighted linesweep algorithms in Sec. 3.1. Then, we discuss modifications for nonconformal meshes in Sec. 3.2. This includes two types of nonconformal nodes: those introduced by the AMR, and those that can arise from the junction of mesh blocks. We also define how mesh-lines are identified in that framework. At that point, we show that the linesweep method itself has two main short-comings: strong non-local range and null-mode for cell shearing. These limitations are addressed in Sec. 4.

3.1. Weighted linesweep rezoning algorithm

The linesweep method consists in rezoning mesh nodes by assuming that they are organized along *mesh-lines*. These lines usually arise from the topology of the mesh and from the initialization of the nodes when meshing the domain. However, their definition for unstructured meshes and at the junction of domain blocks is not always trivial. An algorithm to identify these lines is given in Sec. 3.2.4. For now we assume that nodes are organized along 3 mesh-lines (in 3D) denoted $l = [1, 2, 3]$. The linesweep method starts by defining an initial weight $\Gamma_p^{l,0}$ associated with point p along line l, computed from the distances from the neighbor points:

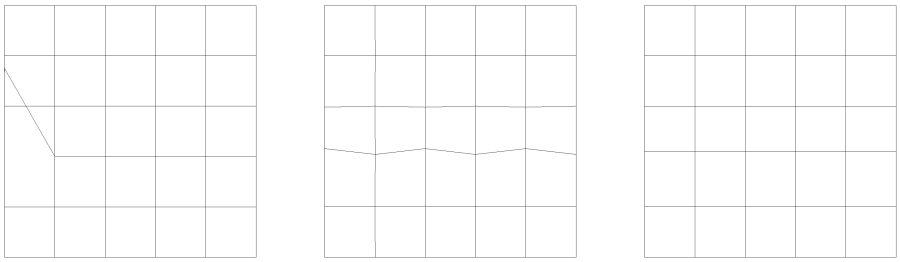
$$\Gamma_p^{l,0} = \frac{|\mathbf{x}_p - \mathbf{x}_{p_l^-}|}{|\mathbf{x}_p - \mathbf{x}_{p_l^-}| + |\mathbf{x}_p - \mathbf{x}_{p_l^+}|}, \tag{1}$$

where p_l^- and p_l^+ designate the previous and next node from p along line l, respectively, and the notation $\mathbf{x}$ denotes a position vector. $\Gamma_p^{l,0}$ represents the initial local mesh aspect ratio computed on the oriented line $[p_l^-, p, p_l^+]$. Refs. [14, 25] then introduce the weights Γ_p^l, which represent the desired per-direction mesh aspect ratio after convergence of the rezoning algorithm. These are combined with the initial weights to define the rezoned point $\tilde{\mathbf{x}_p}$ as:

$$\begin{aligned}
\tilde{\mathbf{x}_p} &= \frac{1}{3} \sum_{l \in [1,3]} \tilde{\mathbf{x}_p^l}, \\
\tilde{\mathbf{x}_p^l} &= \begin{cases} \mathbf{x}_{p_l^-} + \frac{\Gamma_p^l}{\Gamma_p^{l,0}} (\mathbf{x}_p - \mathbf{x}_{p_l^-}) \text{ if } \quad \frac{\Gamma_p^l}{\Gamma_p^{l,0}} \leq 1, \\ \mathbf{x}_{p_l^+} + \frac{1-\Gamma_p^l}{1-\Gamma_p^{l,0}} (\mathbf{x}_p - \mathbf{x}_{p_l^+}) \text{ otherwise.} \end{cases}
\end{aligned} \tag{2}$$

Here it is useful to mention how Eq. (2) behaves in two cases. In the case where $\Gamma_p^l = \Gamma_p^{l,0}$, i.e. if we require an aspect ratio equal to the initial aspect ratio, then the node does not move. Similarly, if we set $\Gamma_p^l = 0.5$, then, $\tilde{\mathbf{x}_p}$ is the equal space point of Ref. [12]. This can be seen by setting $\Gamma_p^l = 0.5$, assuming $\Gamma_p^l \leq \Gamma_p^l$ and using Eq. (1) we get $\tilde{\mathbf{x}_p^l} = \mathbf{x}_{p_l^-} + 0.5(|\mathbf{x}_p - \mathbf{x}_{p_l^-}| + |\mathbf{x}_p - \mathbf{x}_{p_l^+}|)(\mathbf{x}_p - \mathbf{x}_{p_l^-})/|\mathbf{x}_p - \mathbf{x}_{p_l^-}|$ which is by definition the equal space point along the mesh-line $[p_l^-, p_l^+]$. The same result is obtained if $\Gamma_p^l > \Gamma_p^l$. In order to specify the weight Γ_p^l in Eq. (2), Refs. [14, 25] propose to compute an average over neighbor nodes, where the nodes are selected along the cross-logical lines to l. For example, the weight along direction $l = 2$ would be averaged on the direct node neighbors chosen along directions $m = 1$ and $m = 3$, using their local aspect ratio value along l. The averaging process must be iterated to obtain a converged value, which represents smoothed aspect ratios from neighbor to neighbor. While this approach works well in most cases, it can lead to oscillating behavior when the weight values converge to a 2:1 alternated pattern, as illustrated in Fig. 3. A straightforward solution to this problem is to also average the weight Γ_p^l along line l with the value of the current point, in addition with the previous and next nodes. With this modification, the weight $\Gamma_p^{l,q+1}$ associated with point p, along logical direction $l = [1, 2, 3]$, and calculated at iteration $q + 1$ then reads:

$$\Gamma_p^{l,q+1} = \frac{1}{2} \sum_{m \in [1,3], m \neq l} \frac{\Gamma_{p_m^-}^{l,q} + \Gamma_p^{l,q} + \Gamma_{p_m^+}^{l,q}}{3} \tag{3}$$



(a) (b) (c)

Figure 3: Illustration of oscillations that can appear due to boundary problems in the original weighted line sweeping method. (a) initial mesh perturbation, (b) converged solution of the weighted linesweeping method of Ref. [25], (c) converged solution with the formulation proposed in Eq. 7.

where $\Gamma_p^{l,q=0} \equiv \Gamma_p^{l,0}$ from Eq. (1), and we will now denote $\Gamma_p^{l,q}$ as the converged value of the iterative process. In general, the iterative process yields $\Gamma_p^{l,q} \neq \Gamma_p^{l,0}$, such that applying Eq. (2) yields $\tilde{\mathbf{x}_p} \neq \mathbf{x}_p$. However, when the mesh cells tighten, or near convergence (in terms of global aspect ratio), one gets $\Gamma_p^{l,q} \simeq \Gamma_p^{l,0}$ such that $\tilde{\mathbf{x}_p} \simeq \mathbf{x}_p$. In that case, the algorithm no longer rezones nodes, which may be an issue if the mesh quality is not adequate. To remedy that, Ref. [25] introduces an additional relaxation parameter $\nu \in [0, 0.5]$ and defines the relaxed weight $\Gamma_p^{l,\nu}$ applied after iterative convergence of $\Gamma_p^{l,q}$ as:

$$\Gamma_p^{l,\nu} = (1-\nu)\Gamma_p^{l,q} + \nu(1-\Gamma_p^{l,q}). \tag{4}$$

Setting $\nu = 0$ in Eq. (4), i.e. disabling relaxation, we recover $\Gamma_p^{l,\nu} = \Gamma_p^{l,q}$, that is the weighted linesweep which fully preserves average aspect ratio. On the contrary, setting $\nu = 0.5$ we get $\Gamma_p^{l,\nu} = 0.5$ everywhere, which corresponds to the original equal space point of Ref. [12]. As such, the parameter ν allows to combine some aspect-preserving features while still allowing for mesh relaxation.

3.2. Adaptation to AMR meshes

The context of AMR adds two difficulties to the rezoning algorithm presented in Sec. (3.1). First, hanging nodes introduce nonconformal faces and edges in the mesh such that logical mesh-lines are locally truncated, as is the case at mesh boundaries. Second, adjacent cells with different refinement levels introduce variations in local aspect ratio. Without modifications, these variations break the regularity of the algorithm and lead to deformed cells. In this section we present modifications required to adapt the weighted linesweep method to AMR meshes. We also detail the handling of boundary conditions. The modified weighted algorithm introduces two new elements: (i) virtual nodes added to guide the regularization towards a well-behaved equilibrium, and (ii) virtual weights added to guide regularization at boundaries and on some nonconformal interfaces.

3.2.1. Initial weights in AMR

We start by modifying the initial weights $\Gamma_p^{l,0}$. The AMR case introduces new configurations that must be dealt with appropriately so that the algorithm correctly conserves aspect ratios. An illustration of these

specific configurations is shown in Fig. 4. These are treated by defining $\Gamma_p^{l,0}$ as follows:

$$\Gamma_p^{l,0} = \frac{|\mathbf{x}_p - \hat{\mathbf{x}}_{p_l^-}|}{|\mathbf{x}_p - \hat{\mathbf{x}}_{p_l^-}| + |\mathbf{x}_p - \hat{\mathbf{x}}_{p_l^+}|} \text{ with the modified mesh-line nodes:}$$

$$\begin{cases} \text{(i)} & \hat{\mathbf{x}}_{p_l^-} = \mathbf{x}_{p_l^+} + 2|(\mathbf{x}_p - \mathbf{x}_{p_l^+}).\mathbf{n}_p^l|\mathbf{n}_p^l \text{ if } p \text{ is incident on a boundary along } l^- \\ \text{(ii)} & \hat{\mathbf{x}}_{p_l^-} = \frac{\mathbf{x}_p + \mathbf{x}_{p_l^-}}{2} \text{ if } [\mathbf{x}_p, \mathbf{x}_{p_l^-}] \text{ is a double size segment} \\ \text{(iii)} & \hat{\mathbf{x}}_{p_l^-} = \frac{1}{8}\sum_{p' \in \mathcal{P}(c)_o} \mathbf{x}_{p'} \text{ if } p \text{ is face-hanging in cell } c \text{ along } l^- \\ \text{(iv)} & \hat{\mathbf{x}}_{p_l^-} = \frac{1}{4}\sum_{p' \in \mathcal{P}(f)_o} \mathbf{x}_{p'} \text{ if } p \text{ is edge-hanging on face } f \text{ along } l^- \\ \text{(v)} & \hat{\mathbf{x}}_{p_l^-} = \mathbf{x}_{p_l^-} \;\&\; \hat{\mathbf{x}}_{p_l^+} = \mathbf{x}_{p_l^+} \text{otherwise} \end{cases} \tag{5}$$

where these conditions are written for p_{l^-}, which implies that $\hat{\mathbf{x}}_{p_l^+} = \mathbf{x}_{p_l^+}$. These conditions are also duplicated by swapping $(+)$ for $(-)$ and setting $\hat{\mathbf{x}}_{p_l^-} = \mathbf{x}_{p_l^-}$. The normalized vector $\mathbf{n}_p^l$ is the local outward normal incident on node p and along logical direction l. This normal is either fixed for a symmetry boundary condition, or can be a local average normal for a free boundary condition. We recall that $\mathcal{P}(c)_o$ designates the corner nodes in cell c. Similarly, $\mathcal{P}(f)_o$ designates the nodes of face f which are corner nodes in any cell incident on that face. These conditions are illustrated in Fig. 4.

Condition (i) deals with boundaries by creating a virtual node outside of the mesh and at equal distance from the other half of the mesh-line. This is equivalent to setting $\Gamma_p^{l,0} = 0.5$ for node p if the logical direction l is incident on the boundary, i.e. not parallel to the face, as shown in Fig. 4 (a).

Condition (ii) involves *double size segments*, i.e. mesh-line segments on which the refinement depth changes by 1. The part of the segment which doubles in logical size (i.e. is coarser) is called double-size. This is illustrated in Fig. 4.

Condition (iii) is used for nodes that are face hanging along the logical direction l. It defines a virtual node at the barycenter of the logically larger cell in which it is hanging, calculated only using the corner nodes.

Condition (iv) is the equivalent to (iii) for edge-hanging nodes. In that case, a virtual node should be created only if there is a change in refinement depth along l: it defines the node at the barycenter of the logically larger face, and is illustrated in Fig. 4 (c).

Condition (v) is the general case, where $\hat{\mathbf{x}}_{p_l^-}$ is simply the previous node from p along logical line l (see an example in Fig. 4 (d)).

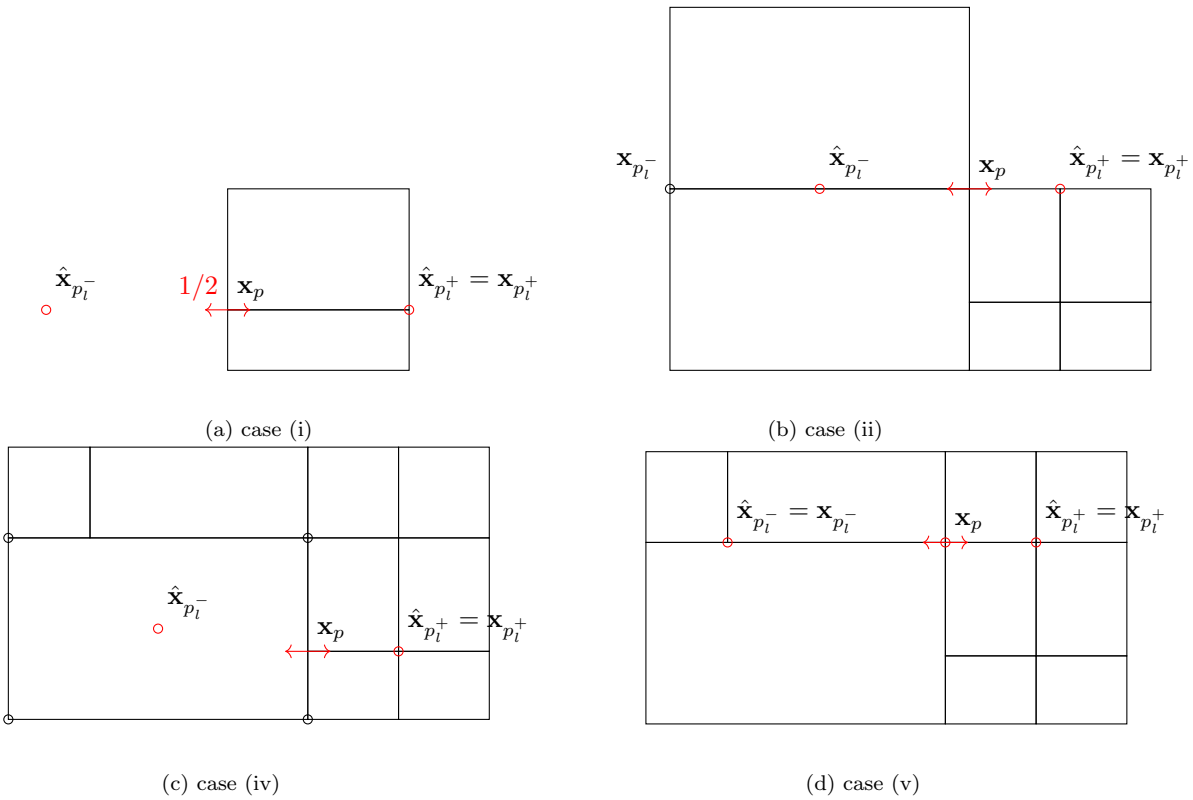
3.2.2. Iterative weights in AMR

Now that the initial weights $\Gamma_p^{l,0}$ are defined, we must modify the iterative weights $\Gamma_p^{l,q}$. Some of the problematic configurations and how they are dealt with are illustrated in Fig. 5. The list of modifications to $\Gamma_p^{l,q}$ for boundaries and face/edge-hanging nodes is as follows:

$$\Gamma_p^{l,q+1} = \frac{1}{2}\sum_{m \in [1,3], m \neq l} \frac{\hat{\Gamma}_{p_m^-}^{l,q} + \Gamma_p^{l,q} + \hat{\Gamma}_{p_m^+}^{l,q}}{3} \text{ with the modified iterative weights:} \tag{6}$$

$$\begin{cases} \text{(i)} & \hat{\Gamma}_{p_m^-}^{l,q} = \Gamma_{p_m^+}^{l,q} \text{ if } p \text{ is incident on a boundary along } m^- \\ \text{(ii)} & \hat{\Gamma}_{p_m^-}^{l,q} = 1/2 \text{ if } p \text{ is face-hanging in cell } c \text{ along } m^- \\ \text{(iii)} & \hat{\Gamma}_{p_m^-}^{l,q} = 1/2 \text{ if } p \text{ is edge-hanging on face } f \text{ along } m^- \\ \text{(iv)} & \hat{\Gamma}_{p_m^-}^{l,q} = \Gamma_{p_m^+}^{l,q} \text{ if } p \text{ is edge-hanging on face } f \text{ but not along direction } m \\ \text{(v)} & \hat{\Gamma}_{p_m^-}^{l,q} = \Gamma_{p_m^-}^{l,q} \;\&\; \hat{\Gamma}_{p_m^+}^{l,q} = \Gamma_{p_m^+}^{l,q} \text{ otherwise.} \end{cases} \tag{7}$$

It should be noted here that along logical line m, only one of the directions at a time (m^- or m^+) can require a specific treatment as given above. Said differently, the mesh-line cannot be truncated on both



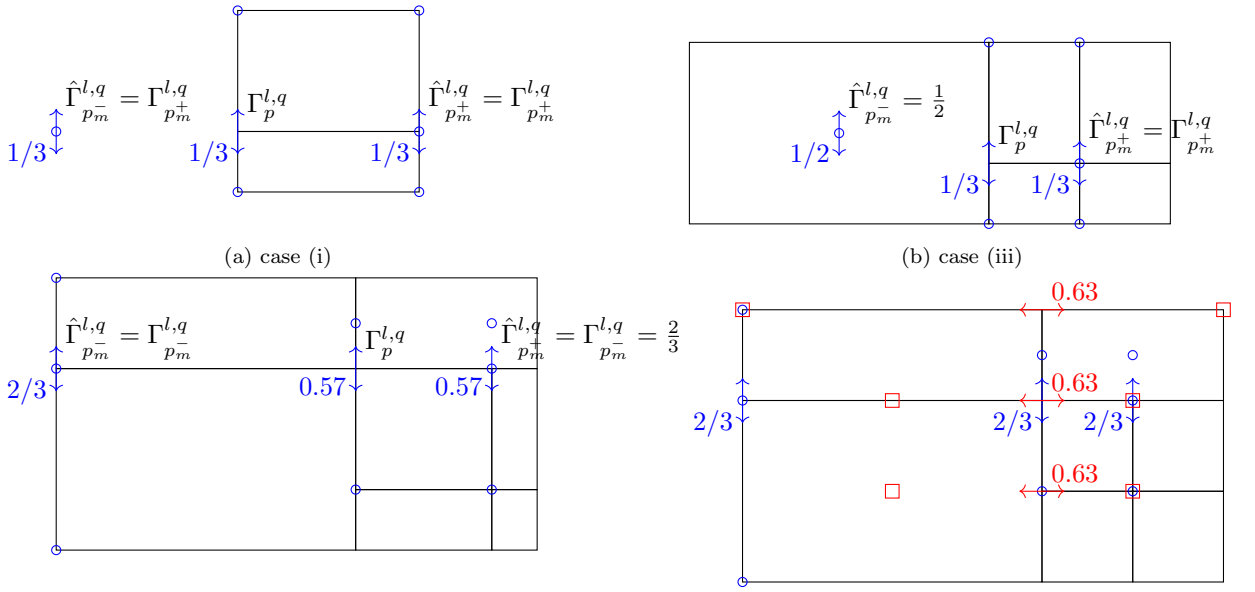
(a) case (i)

(b) case (ii)

(c) case (iv)

(d) case (v)

Figure 4: Illustration of real/virtual nodes used for the calculation of $\Gamma_p^{l,0}$ in Eq. (5), for a mesh-line incident on (a) a mesh boundary, (b) a double-size segment due to a jump in AMR depth, (c) a edge-hanging node along l and (d) a standard case. The black circles in (c) show the 4 *p4est* face nodes from which the barycenter $\hat{\mathbf{x}}_{p_l^-}$ is computed. In all panels, the red circles show the positions used to compute $\Gamma_p^{l,0}$, and the red arrow symbolizes the sweep direction l. Note that the case of a face-hanging node (case (iii) in Eq. (5)) is conceptually similar to (c).



(a) case (i)

(b) case (iii)

(c) Example configuration prior to iteration convergence

(d) Example configuration after to iteration convergence

Figure 5: Illustration of some special cases in the computation of the iterative weights $\Gamma^{l,q}_p$ using the neighbors along the cross-logical lines $m \in [1,3], m \neq l$ (Eq. (7)). Panel (a) illustrates case (i), i.e. a boundary condition. Panel (b) illustrates case (iii), i.e. p is edge-hanging along m and the weight is averaged toward a barycentric equilibrium. This is required due to the choice of the virtual node position in that case, that is at the barycenter of the face in which the node is hanging. Panel (c) combines several special cases. In the vertical direction, node p is along a double-size segment, is horizontally adjacent to a regular node and to a node edge-hanging in a face. The configuration is mirrored in the horizontal direction. In cases (a-c), the colored numbers indicate the current weight values for those examples. Panel (d) shows the converged weights after iteration of (Eq. (7), both for the horizontal (red) and vertical (blue) directions. The blue circles indicate the corresponding real/virtual nodes used for the vertical direction weights, and the red squares indicate the real/virtual nodes used for the horizontal direction weights.

sides. As such, the above condition written for p_{m^-} implicitly means that here $\hat{\Gamma}^{l,q}_{p^+_m} = \Gamma^{l,q}_{p^+_m}$. When the mesh line is truncated along m^+, then the corresponding conditions are obtained by swapping $(+)$ for $(-)$ in Eq. (7) and in that case we have $\hat{\Gamma}^{l,q}_{p^-_m} = \Gamma^{l,q}_{p^-_m}$.

Condition (i) states that if p is on a boundary, and the cross-sweep direction m is incident into that boundary (not parallel to the boundary face), then a virtual weight is defined from the other node in the mesh-line. This is illustrated in Fig. 5 (a).

Condition (ii) and (iii) forces a barycenter equilibrium for face-hanging and edge-hanging nodes. In the case of edge-hanging node, this is only done when the cross-sweep direction m is aligned with the edge which contains the hanging node (see Fig. 5 (b) for the edge-hanging case (iii)). When that is not the case, condition (iv) states that the weight is copied from the other node on the cross-swept mesh-line. Note that this case can only occur in 3D configurations, when a edge-hanging node may be not be hanging in faces along all directions.

3.2.3. Rezoned point

The position of the final rezoned point is computed in the same way as Eq. (2), with the exception that the mesh-line nodes $\mathbf{x}_{p_l^-}$ and $\mathbf{x}_{p_l^+}$ are substituted for the modified nodes $\hat{\mathbf{x}}_{p_l^-}$ and $\hat{\mathbf{x}}_{p_l^+}$ defined for the modified initial weights in Eqs. (5).

3.2.4. Identification of logical lines and degenerate nodes

Up until now we have assumed the logical lines $l = [1,2,3]$ are well defined and ordered in the whole domain. There is however some difficulty in defining logical lines in meshes which combine nonconformal

faces (with edge- and face-hanging nodes) and which are composed of mesh blocks with arbitrary orientations (see Sec. 2.2). When several blocks are connected with different orientations, or on more than one face, degenerate lines can appear, as was already noted in Refs. [12, 14]. Here, we take the case of an 8^{th}-sphere mesh to illustrate how to define logical lines when combining AMR with degenerate lines from block junctions.

Handling of degenerate mesh-lines. For a 3D 2:1 corner-balanced AMR mesh structured by blocks, there is a finite number of configurations that can arise for the connectivity around a given node p. These are illustrated in Fig. 6 for nodes on a boundary and Fig. 7 for non-boundary nodes. We can classify these configurations depending on the number of boundaries, nodes and cells incident on p, and its hanging status. This classification is given in Tab. 1. Using these four pieces of data as input allows to determine if a node is *degenerate* in terms of local connectivity.

In this paper, degenerate nodes are not used for iterative weight computation. While it is possible in principle to define some weights in the degenerate directions (see Ref. [25] for the conformal case), this is not done here for simplicity. Instead, we perform two modifications: (i) degenerate nodes are not rezoned using the linesweep algorithm, but using a specific procedure detailed below, and (ii) nodes adjacent to degenerate nodes do not use the latter to compute their iterative weights, which requires modifying the average in Eq. (6) to ignore these nodes.

The degenerate nodes, identified using Tab. 1, can be classified in 4 categories: (i) the degenerate node is not hanging, is on a boundary, and joins 3 mesh blocks, it is degenerate on two directions defined from 3 nodes, (ii) the degenerate node is edge- or not hanging, is not on a boundary, joins 3 or 6 mesh blocks, it is degenerate on two directions defined from 3 nodes, (iii) the degenerate node is edge-hanging, is not on a boundary, joins 3 mesh blocks, it is degenerate on two directions defined from 2 nodes and the face into which the node is hanging, and (iv) the degenerate node is not hanging, is not on a boundary, joins 4 mesh blocks, it is degenerate in all directions and is incident on 4 nodes.

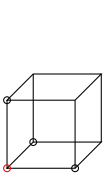
Once a degenerate node is found, we identify the nodes it depends on (sharing the degenerate logical direction), and compute the barycentric coordinates λ_i of p in the polygon formed by these nodes it depends on:

- (i) and (ii): barycentric coordinates of the node in the triangle defined by the 3 nodes in the 2 degenerate directions
- (iii): same as (i-ii) but the third coordinate for the triangle is set to the barycenter of $\mathcal{P}(f)_o$, where f is the face in which the node is edge-hanging in
- (iv): barycentric coordinates of the node in the tetrahedron defined by the 4 nodes incident on p

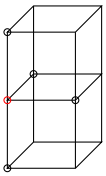
where we note that these barycentric coordinates are computed assuming an equal-space mesh. The rezoned coordinates for the degenerate nodes are then simply computed from the barycentric coefficients λ_i and the real-space coordinates of the nodes in the degenerate directions around p. This rezoning of the degenerate nodes is performed before each call to the linesweep.

Identification and orientation of mesh-lines. Given the variety of cases presented in Figs. 6 and 7, the definition of what constitutes a *mesh-line* is warranted. From a topological point of view, we define a mesh-line around node p and direction l as the two nodes p_l^- and p_l^+ such that $[p_l^-, p, p_l^+]$ respects the following conditions:

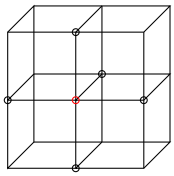
- each node in $\mathcal{P}(p)$ can only belong to one mesh-line at p
- if node p is either face-hanging or not hanging, then two nodes p_1 and p_2 in $\mathcal{P}(p)$ cannot be on the same mesh-line if they share a face (see Figs. 6 (b,c,e) and 7 (a,b))
- if node p is edge-hanging, then the above determination can only be made in faces belonging to cells into which the node is not hanging (an example where the above would otherwise fail is Fig. 6 (d)). In addition, we must add that p_1 and p_2 cannot be in the same cell if both p_1 and p_2 are face-hanging nodes (see Fig. 7 (c)),



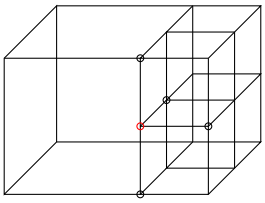
(a) regular node on 3 boundaries, 3 neighbors in 1 cell



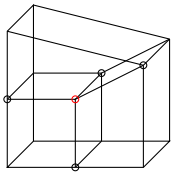
(b) regular node on 2 boundaries, 4 neighbors in 2 cells



(c) regular node on 1 boundary, 5 neighbors in 4 cells



(d) (edge-)hanging node on 1 boundary, 4 neighbors in 3 cells



(e) regular node on 1 boundary, 4 neighbors in 3 cells (degenerate)

Figure 6: All possible topological configurations for a node on at least one boundary for a 2:1 corner-balanced block-structured mesh. Configurations (a-c) and (e) are non-hanging nodes. Configuration (e) occurs at a boundary junction between 3 logical blocks. Black circles indicate the node neighbors to the current node highlighted in red.

where we recall that $\mathcal{P}(p)$ is the set of nodes incident on node p through an edge.

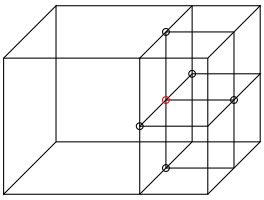
The ordering of nodes p_l^- and p_l^+ around p in direction l must be consistent in the whole mesh for the weighted linesweep method, since the latter is equivalent to an aspect ratio smoothing for each l. For a mesh where all the logical blocks have the same orientation, this determination is trivial and can be made from the logical coordinates. In the case of logical block rotations at block junctions (as for the 8^{th}-sphere mesh), the ordering is not trivial. One solution, implemented here, is to allow for a node-local numbering of the mesh-line $l \in [1, 3]$ that is different for each node. We then build a table that relates the local orientation of l for p to those of p_l^- and p_l^+. This orientation matching is performed by computing logical vector directions for each node in logical space, and comparing to its neighbors.

3.2.5. Weighted linesweep in meshes with non-uniform initial aspect ratio

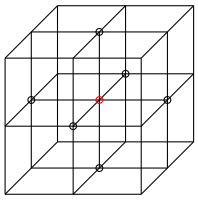
Conceptually, the weighted linesweep method will seek to equilibrate aspect ratios along each logical direction toward an average value computed from the neighbors. This procedure has the side effect that a mesh with an initially non-uniform aspect ratio along a given logical direction is not at *equilibrium* with respect to the weighted linesweep. A typical example is a spherical mesh constituted of polygonal elements with flat faces. Since the sphere cannot be paved regularly with such elements, initial aspect ratio variations will exist and the rezoning procedure may significantly alter the quality of the numerical solution by moving the mesh nodes along the angular direction. We implement here a simple solution, whereby meshes that have non-constant initial aspect ratios along direction l when evaluated along the cross-directions $m \neq l$, are pre-relaxed prior to the calculation using 500 iterations of the linesweep algorithm. In practice in this paper, this is applied to all test cases that make use of the spherical mesh.

3.2.6. Illustration of the modified linesweep method

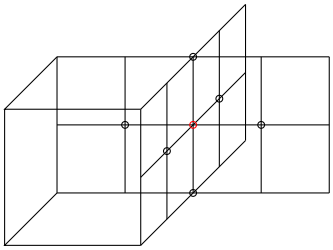
We provide here some illustrations of the rezoning capabilities of our algorithm. We first consider a cubic domain with non-linear mesh spacing along the x and z directions. This spacing must be preserved by the weighted algorithm when $\nu = 0$. In addition, we refine a part of the domain to a depth of 2. Finally, we add perturbations to the nodes inside the domain, while keeping the boundary conditions un-perturbed. If the algorithm performs correctly, the nodes inside the domain should be rezoned to a structure close to that present on the boundary. The initial mesh and final mesh configurations are shown in Fig. 8 for the cases of $\nu = 0$ and $\nu = 0.5$. In both cases, the weights $\Gamma_p^{l,q}$ are computed using 50 iterations.



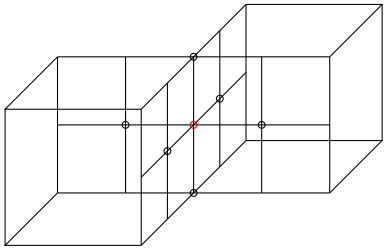
(a) face-hanging interior node, 5 neighbors in 5 cells



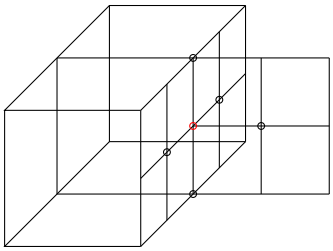
(b) regular interior node, 6 neighbors in 8 cells



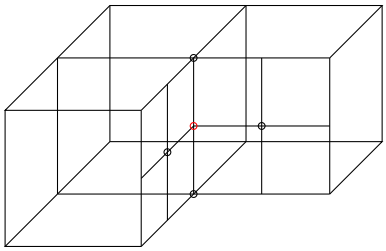
(c) edge-hanging interior node, 6 neighbors in 7 cells



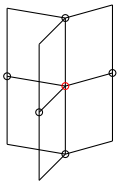
(d) edge-hanging interior node, 6 neighbors in 6 cells



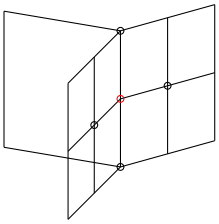
(e) edge-hanging interior node, 5 neighbors in 6 cells



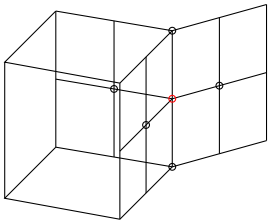
(f) edge-hanging interior node, 4 neighbors in 5 cells



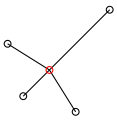
(g) regular interior node, 5 neighbors in 6 cells (degenerate)



(h) edge-hanging interior node, 4 neighbors in 4 cells (degenerate)

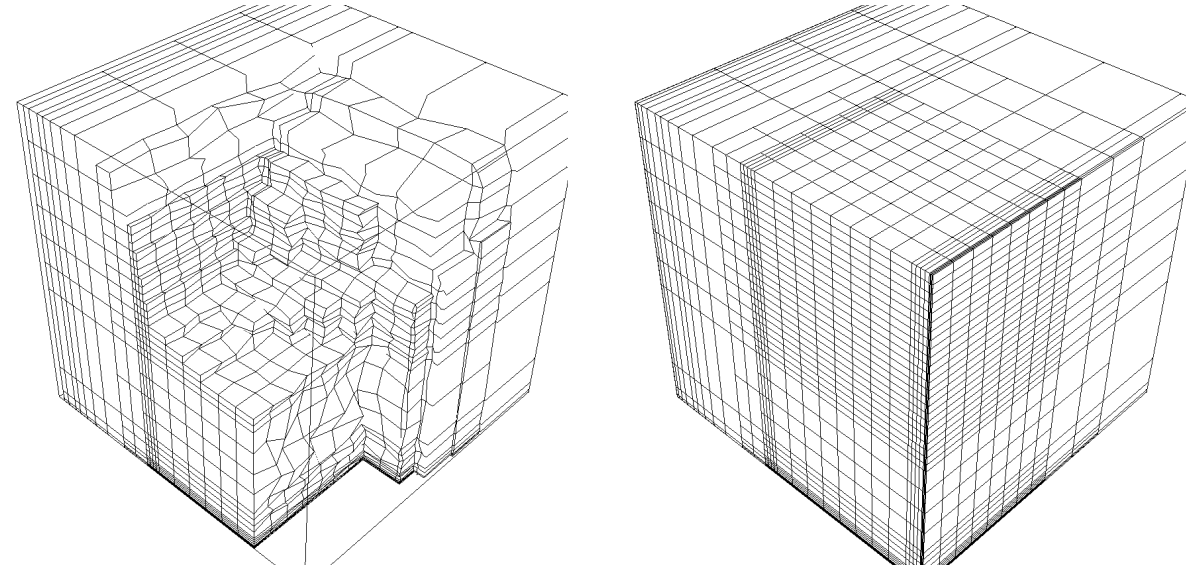


(i) edge-hanging interior node, 5 neighbors in 5 cells (degenerate)



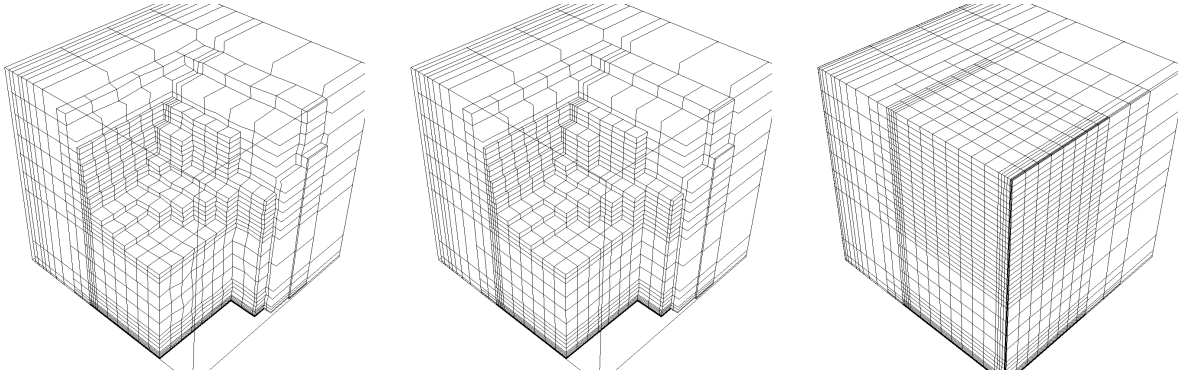
(j) regular interior node, 4 neighbors in 4 cells (degenerate)

Figure 7: All possible topological configurations for an interior node of a 2:1 corner-balanced block-structured mesh. For simplicity, not all cells are drawn. Configurations (a-f) are non-degenerate, (g) occurs at the junction between 3 or 6 logical blocks, (h,i) occur for an edge-hanging node at the junction of 3 logical blocks, and (j) for a regular node at the junction of 3 logical blocks. Black circles indicate the node neighbors to the current node highlighted in red.



(a) Initial mesh interior

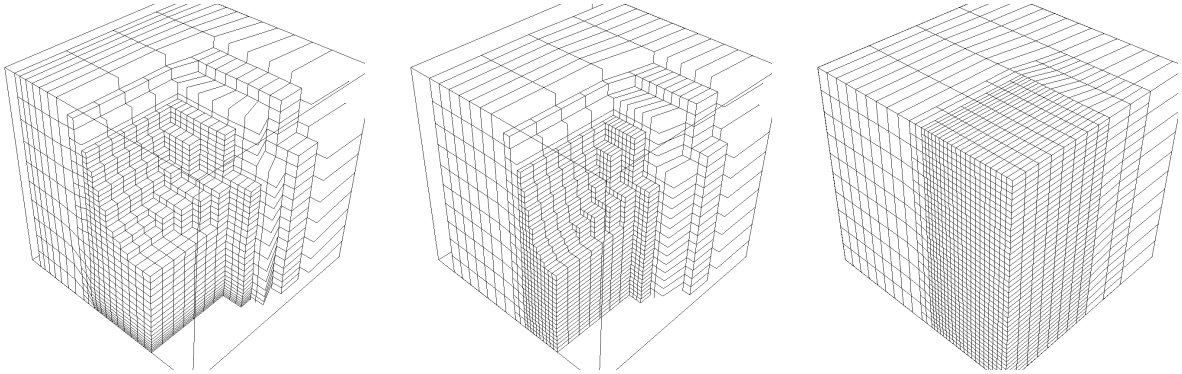
(b) Initial mesh exterior



(c) $\nu = 0$, 1 iteration

(d) $\nu = 0$, 50 iterations

(e) $\nu = 0$, 50 iterations



(f) $\nu = 0.5$, 50 iterations

(g) $\nu = 0.5$, 500 iterations

(h) $\nu = 0.5$, 500 iterations

Figure 8: Comparison of the mesh interior and exterior for an initial cubic mesh with various refinement depths and initial aspect ratios. The initial mesh interior is randomly perturbed, with some tangled elements. The iterations refer to iteration of the whole algorithm, not the iterative calculations of the weights.

card($\mathcal{B}_p$)	card($\mathcal{P}_p$)	card($\mathcal{C}_p$)	Hang.	Degen.	Fig.	Nb. full lines	fixup type
3	3	1	no	no	6(a)	0	
2	4	2	no	no	6(b)	1	
1	5	4	no	no	6(c)	2	
1	4	3	edge	no	6(d)	1	
1	4	3	no	yes	6(e)	0	(i)
0	6	8	no	no	7(b)	3	
0	6	7	edge	no	7(c)	3	
0	6	6	edge	no	7(d)	3	
0	5	6	edge	no	7(e)	2	
0	5	6	no	yes	7(g)	1	(ii)
0	5	5	face	no	7(a)	2	
0	5	5	edge	yes	7(i)	1	(ii)
0	4	5	edge	no	7(f)	1	
0	4	4	edge	yes	7(h)	1	(iii)
0	4	4	no	yes	7(j)	0	(iv)

Table 1: Classification of the connectivity types for the definition of the logical lines for the linesweep algorithm at node p. The configurations are organized first by number of boundaries (card($\mathcal{B}_p$)), then by number of neighbor nodes (card($\mathcal{P}_p$)), then by number of cells the node belong to (card($\mathcal{C}_p$)), and finally by weather the node is edge-, face- or not hanging. This table shows that these 4 inputs are sufficient to build a decision tree that can distinguish if the node is on a degenerate block junction. The corresponding Fig. number is given for reference. The second-to-last column indicates the number of full logical lines that the node belong too, where *full* means that there is a node before and after p on that line. The last column indicates the rezoning type for the degenerate cases: (i) triangular barycentric, (ii) triangular barycentric perpendicular to the full line, (iii) same as (ii) but completing the triangle using the barycenter of the face into which the node is edge-hanging, (iv) tetrahedral barycentric. These conditions are further explicited in Sec. 3.2.4.

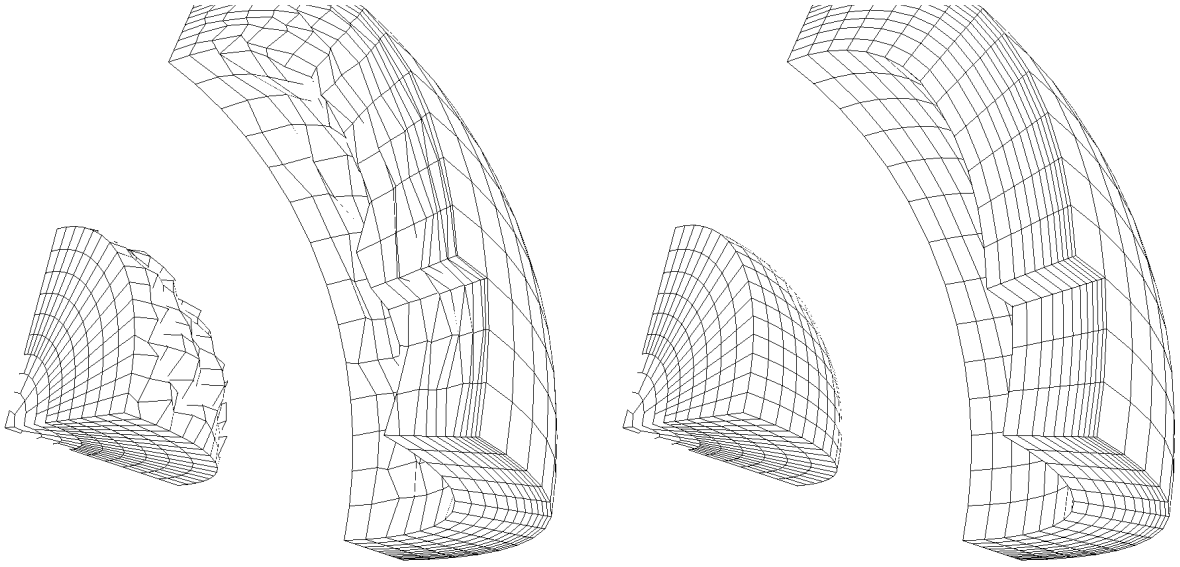
This test shows that the algorithm formulated without damping ($\nu = 0$) performs well to preserve the local mesh aspect ratio on both the interior and the exterior of the domain. A single iteration of the algorithm provides most of the mesh improvement, with little gain with more iterations. This is due to the interplay between the rezoned position and aspect-ratio-preserving weights highlighted in Sec. 3.1. The adaptations presented in Sec. 3.2 preserve resolution jumps due to AMR, while relaxing AMR cells in an equal-space manner. While this means that AMR cells loose the aspect-ratio-preserving property of the scheme, this approach is consistant with our AMR formulation that refines cells using barycenters (see part I [4]). Using $\nu = 0.5$, one can see that the mesh relaxes progressively towards a more homogeneous mesh, but that process takes many iterations. Since the rezoning rate is different for different nodes, the intermediate states contain deformed cells. In addition, the algorithm will not converge to a fully cartesian-like mesh because of nodes that share AMR levels in a triple-point-like manner. This can notably be seen on panel (h). A correct adaptation of the scheme to $\nu > 0$ would require to apply the damping only on the nodes of the original un-refined grid, thus guiding the rest of the rezoning on the finer mesh. This is however not currently compatible with our framework.

A second example for an 8^{th}-sphere domain is given in Fig. 9. The mesh has an initial radial aspect ratio variation around the outer boundary of the mesh, and an angular aspect ratio variation created by the non-uniform spherical mesh itself. As for the previous case, the initial positions of the nodes is perturbed except at mesh boundaries. The rezoned mesh recovers a small radial aspect ratio close to the outer boundary and preserves the angular aspect ratio of the spherical mesh structure itself.

4. Adaptation of the linesweep algorithm for practical applications

There are three main limits to the algorithm as it is presented up to this point:

- *Range problem*: the weighted line spacing method is a global method, i.e. it acts on the entire mesh since the weight function of Eq. 7 acts from neighbor to neighbor. As such, a small aspect ratio



(a) Initial mesh interior (b) $\nu = 0$, 100 iterations

Figure 9: Inner and outer cuts of an 8^{th}-sphere mesh with (a) an initial interior node position perturbation and (b) after 100 iterations of the undamped weighted linesweep algorithm.

perturbation arising in a given location causes nodes to be displaced on a large scale (see Fig. 10 (a,b)). This is detrimental (i) in the AMR framework since the AMR refinement criteria may then be triggered in those regions of propagating mesh perturbations, and (ii) for physics applications, where cold regions or interfaces should be preserved.

- *Shearing problem*: the un-damped linesweep method does not prevent cell shearing, since by definition it respects the mesh-line when defining the new node position. An example is given in Fig. 10 (c).
- *Bunching problem*: the damped linesweep may not prevent cells from bunching together because nodes may be locally already at an equal-space point. An example is given in Fig. 10 (d).

Here, we propose an adaptation of the algorithms presented until now that address these issues. For that purpose, we introduce (i) a relative line-angle control metric ω_p to activate the linesweep and set its damping value ν_p, and (ii) couple the linesweep method with conventional metric optimization, where we will define an optimization metric Ω_p^{opt}, activated by a similar metric denoted Ω_p. Finally, we also describe a robust and minimum-impact disentangling procedure.

4.1. linesweep activation and bunching control through damping

The linesweep activation region is controlled within the iterative procedure that computes $\Gamma_p^{l,q+1}$ (Eq. (7)), by defining the modified weights $\tilde{\Gamma}_p^{l,q+1}$ as:

$$\tilde{\Gamma}_p^{l,q+1} = \begin{cases} \Gamma_p^{l,q+1} \text{ if: } \quad \omega_p > \omega_{\min} \\ \Gamma_p^{l,0} \text{ otherwise} \end{cases} \tag{8}$$

where $\omega_{\min}$ is a threshold relating to the relative line-angle control metric ω_p. This defines an operative region where the linesweep is active. It differs from using Eq. (4), even with $\nu = 0$, since the latter does not prevent node movement but controls how much local aspect ratios are preserved. As mentioned before, ω_p is used to set both the linesweep activation and its damping ν. The assumption here is that a node that requires a lot of linesweep rezoning (high value of ω_p) should also be relaxed with respect to aspect ratios

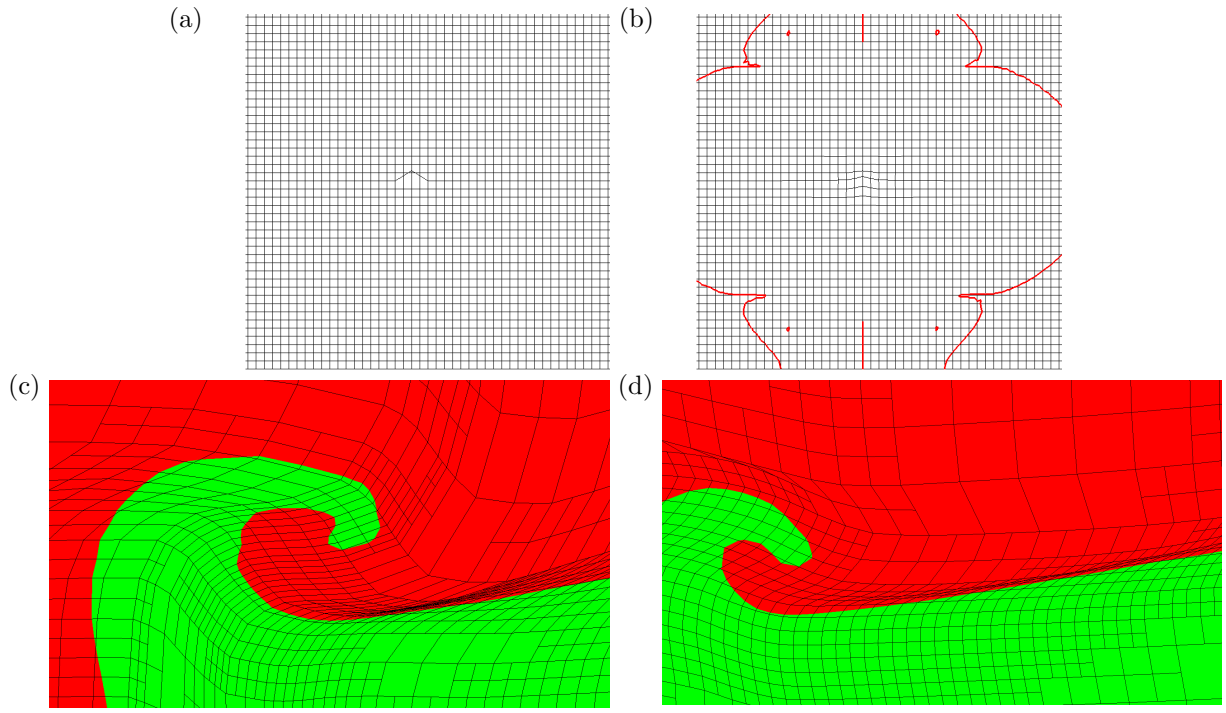


Figure 10: Illustrations of two limitations of the weighted linesweeping method. (a) local mesh perturbation and (b) corresponding range of nodes that were moved by one iteration of the weighted linesweeping method ($\nu = 0$) shown inside the red contour. Example of cell bunching occuring around a vortex for the linesweeping method with (c) $\nu = 0$ and (d) $\nu = 0.5$.

by authorizing $\nu > 0$. As such, we introduce a simple dependency of ν on ω. We define an angle threshold ω_{ν_0} under which $\nu = 0$ (no damping), and an angle threshold $\omega_{\nu_{1/2}}$ over which $\nu = 1/2$ (maximum damping, i.e. equal space algorithm). Then, we set ν to vary linearly with ω between these two thresholds, such that ν_p reads:

$$\nu_p = \max\left(0, \frac{1}{2}\min\left(1, \frac{\omega_{\nu_0} - \omega_p}{\omega_{\nu_0} - \omega_{\nu_{1/2}}}\right)\right). \tag{9}$$

We must now define the metric ω_p. This metric should be normalized, since we have introduced a variety of threshold values. To do so, we propose to consider a generic statement about the mesh structure: nodes along a mesh line will require more rezoning as a mesh line rotates with respect to its original orientation. This is because the root topology of the mesh does not change with AMR (contrary to ReALE methods [26]), such that mesh lines cannot deviate too strongly from their original orientation before leading to potential tangling, bunching or shearing. In this method, we can see that only shear in the mesh line will trigger linesweep rezoning and damping. Since in Lagrangian simulations mesh lines are usually aligned with the expected main flow direction at $t = 0$, this method ensures that no ALE is used when the flow can be described in pure Lagrangian mode.

To define mesh-line rotation, we start by defining 3 cell-local directions from the corner nodes:

$$\mathbf{l}_c = \frac{1}{4}\sum_{p_1,p_2\in\{e_l\}} \frac{\mathbf{x}_{p_2} - \mathbf{x}_{p_1}}{|\mathbf{x}_{p_2} - \mathbf{x}_{p_1}|}, \tag{10}$$

where $\{e_l\}$ denotes the set of 4 corner-edges aligned with logical direction l, and p_1, p_2 are the two corner nodes along that edge. As an example, direction $l = 2$ has 4 edges formed by $\{\{1,5\},\{2,6\},\{3,7\},\{4,8\}\}$ in our local indexing (see Fig. 2). We also denote the initial cell directions, computed at t=0, as $\mathbf{l}_c^0$.

We then define the Root Mean Square (RMS) average cell rotation θ_c as:

$$\theta_c = \sqrt{\frac{1}{3}\sum_l (\arccos(\mathbf{l}_c \cdot \mathbf{l}_c^0))^2}, \tag{11}$$

from which wet set the node-based control parameter ω_p as the RMS average over the cells sharing the node:

$$\omega_p = \sqrt{\frac{1}{\text{card}(\mathcal{C}_p)}\sum_{\mathcal{C}_p} \theta_c^2}. \tag{12}$$

We have introduced three control parameters that describe how the linesweep algorithm is adjusted from undamped (for line angles ω_p below $\omega_{\min}$), to minimally damped (from ω_{ν_0}) to equal space (at and above $\omega_{1/2}$). How much these values affect the solution is problem-dependent, but there are typical operating regimes that are rather safe to use. First, it is recommended to use a small non-zero $\omega_{\min}$ (e.g. 5°). This means the weighted linesweep algorithm can safely be activated on large regions of the mesh without sacrificing the physical information it contains. A non-zero value avoids rezoning in regions of the mesh that have not been activated or deviated from their initial orientation. The choice of $\omega_{1/2}$ depends on how much vorticity is expected to be present. Lower values will trigger the equal space algorithm faster, which will lead to a more diffusive but more stable solution. A typical value is in the 30-50° range. ω_{ν_0} can be set in-between (e.g. 20°) by choosing the threshold at which one wants to depart from the unweighted solution. An illustration of how the solution depends on these parameters is given in Sec. 5.2 for the triple-point case.

4.2. Shearing control

Since the linesweep method cannot prevent the mesh from shearing locally (unless it is iterated many times over a large stencil), we combine the linesweep algorithm with a conventional metric function optimization. We recall that, as discussed in the introduction, we do not aim to solely rely on metric optimization to regularize the mesh, as an effort to retain a generic approach to mesh improvement. As such, here, we choose a metric that specifically addresses the issues of mesh shearing in order to palliate the shortcomings of the linesweep.

4.2.1. Definition of tetrahedra used for metric function optimization

We start by a few definitions required to proceed with the definition of the metrics. The functions that will be optimized are mesh quality metrics defined at the node of the mesh (so-called nodal objective function optimization [9]). The usual approach is to compute the function value in each tetrahedra incident on node p. These tetrahedra are formed by considering the three faces in cell c incident on p, which defines four nodes. However in our case, such tetrahedra are not well-defined for topologically hanging nodes. In that case, one would need to modify the definition of the tetrahedra incident on p for cells where p is a hanging node. While we think this is possible in general, e.g. by introducing a different decomposition in those cases, extending metric function optimization in a fully consistent way to nonconformal meshes is not the subject of the present work. In our case, we will simply ignore the tetrahedra contribution in cells into which p is hanging. As such in what follows, $\mathcal{T}_p$ denotes the set of tetrahedras incident on p with the exclusion of the contributions from cells into which p is hanging.

For a given tetrahedra $t \in \mathcal{T}_p$, we denote $\mathbf{x}_{t,i}$ as the i-th node incident on p in t, where $i \in [1,3]$. We define the edge vectors for t with respect to p as $\boldsymbol{\delta}_{t,i} = (\mathbf{x}_{t,i} - \mathbf{x}_p)$. The signed volume α_t of the tetrahedra is $\alpha_t = \boldsymbol{\delta}_{t,1}.(\boldsymbol{\delta}_{t,2} \times \boldsymbol{\delta}_{t,3})$. The tetrahedra nodes i are numbered such that α_t is positive for a valid (non-tangled) tetrahedra. We now proceed with the definition of the optimization metrics.

4.2.2. Definition of the control and optimization metrics

The metric optimization method is intended to address cell shearing. There are a variety of cell-shearing metrics proposed in the literature, see e.g. in Ref. [10]. In this work, we propose to adapt the Condition Number (CN) metric (see Ref. [9] and *aspect Frobenius* metric in Ref. [10]) to a normalized version. Similar

results can be obtained with other shearing metrics and it is not the goal of this paper to discuss the merit and drawback of each. We introduce the normalization so that the metric is not sensitive to the volume difference of the tetrahedra incident on p, which the original CN is. This is important here because the AMR mesh produces discontinuities in cell volumes associated with refinement depth, which would lead the CN metric to incorrectly relax the mesh. The CN metric is otherwise chosen because it has a high value in sheared and skewed cells, while also diverging for invalid elements. We will refer to this metric as normalized-CN, denote its value in a tetrahedra as Ω_t and the nodal value of the metric as Ω_p. Those read:

$$\begin{aligned}
\Omega_p &= \max_{t \in \mathcal{T}_p} \left| \frac{\Omega_t - 1}{2} \right|, \\
\Omega_t &= \frac{1}{\alpha_t \sqrt{3}} \sqrt{\frac{||\boldsymbol{\delta}_{t,1} \times \boldsymbol{\delta}_{t,2}||^2}{||\boldsymbol{\delta}_{t,1}||^2 ||\boldsymbol{\delta}_{t,2}||^2} + \frac{||\boldsymbol{\delta}_{t,1} \times \boldsymbol{\delta}_{t,3}||^2}{||\boldsymbol{\delta}_{t,1}||^2 ||\boldsymbol{\delta}_{t,3}||^2} + \frac{||\boldsymbol{\delta}_{t,2} \times \boldsymbol{\delta}_{t,3}||^2}{||\boldsymbol{\delta}_{t,2}||^2 ||\boldsymbol{\delta}_{t,3}||^2}} \text{ if } \quad \alpha_t > 0 \ , \\
\Omega_t &= 10^{100} \text{ otherwise} \quad ,
\end{aligned} \tag{13}$$

and note that some factors were introduced such that $\Omega_p = 0$ corresponds to the ideal element. From this formula, Ω_p is a normalized measure of the worst element quality incident on node p, from the normalized-CN metric sense.

Finally, we introduce a threshold value $\Omega_{\min}$ such that the objective function optimization is only enabled for $\Omega_p > \Omega_{\min}$. A typical value is $\Omega_{\min} \simeq 0.02$. Above the threshold, the optimizatio proceeds as follows. We start by computing the descent direction of the metric function. Then, we take a single step in that descent direction, using a line-search algorithm to determine an optimal step size. The maximum step size is weighted by $\Omega_p - \Omega_{\min}$ to accelerate regularization when the mesh quality is low. Boundary conditions are handled by orienting the line-search direction along the boundary itself.

4.3. Mesh disentanglement

The last component for the rezoning is to guarantee the validity of the mesh. The disentangling of the AMR mesh using the weighted linesweeping procedure is only guaranteed if $\nu = 0.5$ and if the rezoning is run on the entire mesh. The objective here is to find the smallest stencil around an invalid node required to disentangle the mesh, and to respect local aspect ratios if possible. In order to detect invalid elements, we first define a node-based invalidity metric $\mathcal{I}_p$ such that $\mathcal{I}_p \geq 0$ if node p is invalid and $\mathcal{I}_p < 0$ otherwise. The *validity* of a node is evaluated with respect to node-based mesh quality metrics that present a well-defined behavior for tangled elements. A simple metric to detect invalid elements is defined as:

$$\mathcal{I}_p = \max_{t \in \mathcal{T}_p} (-\alpha_t), \tag{14}$$

which yields $\mathcal{I}_p \geq 0$ if any of the tetrahedra incident on p has a negative (or zero) signed volume. The damping factor ω_p is set from $\mathcal{I}_p$ using an iterative procedure. Denoting ω_p^q as the factor computed at iterative step q, we define:

$$\begin{aligned}
\omega_p^{q+1} &= \frac{1}{\text{card}(\mathcal{C}_p)} \max_{c \in \mathcal{C}_p} \left[\frac{1}{\text{card}(\mathcal{P}_c)} \max_{p' \in \mathcal{P}_c} \omega_{p'}^q \right]. \\
\omega_p^0 &= 1 \text{ if } \mathcal{I}_p > 0 \ , \quad \omega_p^0 = 0 \text{ otherwise.}
\end{aligned} \tag{15}$$

The disentanglement procedure is initialized by calculating ω_p^q using Eq. (15) with $q_{\max} = 2$, i.e. on a 2 cells neighborhood. We then apply the modified weighted line sweeping algorithm of Eq. (8), and do not apply mesh relaxation (i.e. set $\nu = 0$). If this procedure does not successfully disentangle the mesh, then we restart from Eq. (15) by increasing $q_{\max}$. In case this procedure does not converge, we progressively increase ν on the reduced stencil, up to $\nu = 0.5$. This procedure is guaranteed to succeed, since at worst, ω_p^q will be set to 1 on the whole mesh and ν will be set to 0.5. In most practical applications, we find that the mesh is usually disentangled with $q_{\max} < 6$ and $\nu < 0.125$.

4.4. Summary of the mesh-improvement procedure

The rezoning procedure is summarized as follows:

- compute the average line rotation angles ω_p (Eq. 12) on the Lagrangian mesh to define the linesweep activation region above the minimum line rotation $\omega_p^{\min}$ (Eq. 8)
- apply the damped weighted linesweep with ν calculated from the line angle, using Eq. (9)
- update Ω_p (Eq. 13) to define the metric optimization region above $\Omega_p^{\min}$ and optimize the mesh to reduce cell shearing
- check and disentangle the mesh using procedure of Sec. 4.3 to make sure the mesh is valid after the rezone step.

This procedure defines new mesh coordinates, which are then used for a multi-material second-order remapping step using interface reconstruction.

5. Test cases

We now present multi-material test cases to assess the accuracy and performance of the 3D ALE-AMR algorithm presented in this paper. We focus on tests that are difficult to run in pure Lagrange, thus requiring ALE. Benchmark of Lagrange-AMR results were conducted in Part I of this paper. Here, ALE-AMR results are compared to indirect Euler-AMR calculations. We explore results for the rotor in Sec. 5.1, triple point in Sec. 5.2, Rayleigh-Taylor single mode growth in Sec. 5.3, and ideal and perturbed ICF-like implosion in Sec. 5.4. Timing and number of cells comparisons are summarized for 3D cases in Tab. 2.

5.1. 2D rotor

The rotor test case [27] consists in initializing a uniformly rotating disk of radius 0.1 radius and density of 10, in a background medium of density and pressure of 1. The total domain size is $0.1 \times 1.5 \times 1.5$. This test is usually employed for testing magneto-hydrodynamics codes, but it can be adapted to test multi-material algorithms in rotating flows. Notably, as the disk rotates, the central region should remain at constant density and with a linear velocity profile. The rotation also causes density lumps to form at the edge of the disk, such that the maximum density reached can be used to assess the accuracy of the scheme. We initialize two materials with the same properties and initially separated in the middle of the disk. We compare a pure Lagrangian calculation initialized on $1 \times 200 \times 200$ cells with an ALE-AMR and Euler-AMR calculation, both initialized on $1 \times 26 \times 26$ mesh and with a maximum refinement depth of $l_{\max} = 3$. The ALE parameters are set to $\omega_{\min} = 5°$, $\omega_{\nu_0} = 10°$, $\omega_{\nu_{1/2}} = 30°$ and $\Omega_{\min} = 0.02$. The calculation is run to t = 0.018 for the Lagrangian case (the Lagrangian mesh loses quality soon after) and to t = 0.1 for the AMR cases. The resulting 2D maps of density and partial volume fractions at t = 0.1 are shown in Fig. 11, and show how the ALE scheme is able to maintain a region of high mesh quality in the rotating part while adapting the mesh lines to the unperturbed mesh outside the rotating region. The map of density also appears to reach higher values, thanks to the mesh that retained higher resolution in the bulge region. Radial profiles of density and velocity at t = 0.018 are given in Fig. 12. These illustrate that the ALE-AMR calculation is much less diffusive, yielding results closer to the pure Lagrangian calculation. For both the Euler-AMR and ALE-AMR, we note that the multi-material interface remains flat in the central region. The rotor test cases shows that the ALE-AMR method can be more accurate than Euler-AMR thanks to a rezoning strategy that aims to respect features of the Lagrangian mesh. The result is found to be density and velocity profiles that are more symmetric, and reach higher peak values. However, the improvement here remains modest.

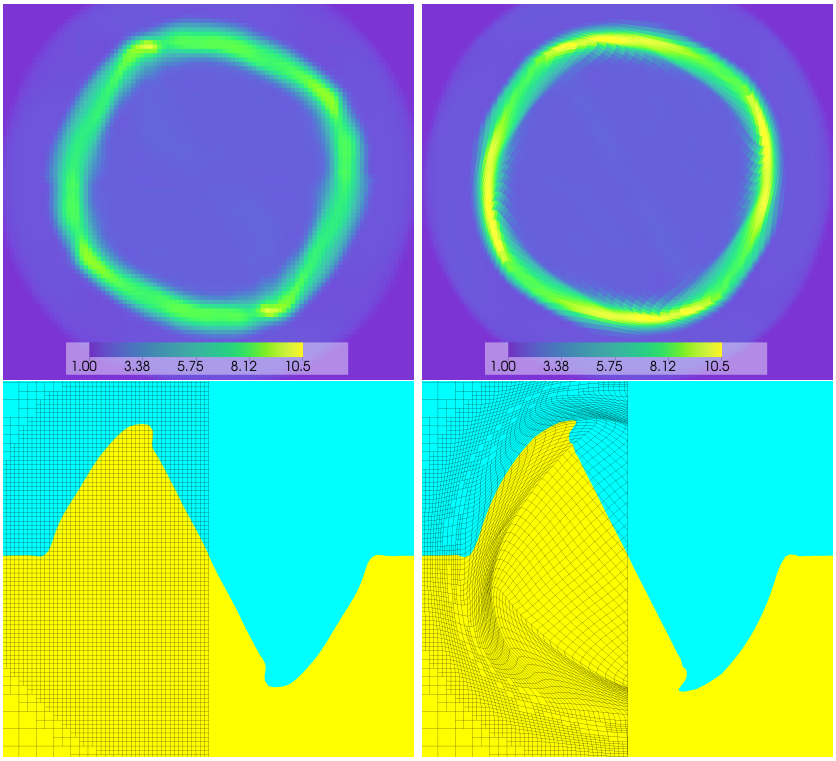


Figure 11: Results from the rotor test case at t=0.1 s for the (top row) density and (bottom row) multi-material interface position, for the (left) Euler-AMR case and (right) ALE-AMR case. Both cases are initialized on $1 \times 26 \times 26$ cells with $l_{\max} = 3$. The bottom plots show part of the mesh.

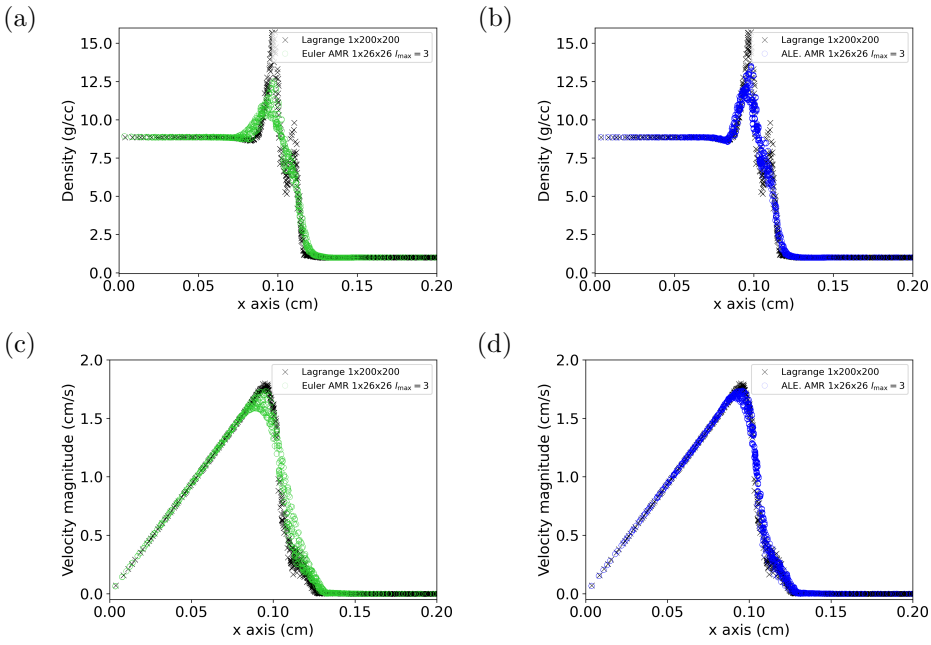


Figure 12: Radial profiles of (a,b) density and (c,d) velocity for the rotor test problem at t=0.018 s, for the (a,c) Euler-AMR case and (b,d) ALE-AMR case. Pure Lagrangian simulation results on the $1 \times 200 \times 200$ mesh are shown as black crosses. AMR simulations are initialized on a $1 \times 26 \times 26$ mesh with $l_{\max} = 3$.

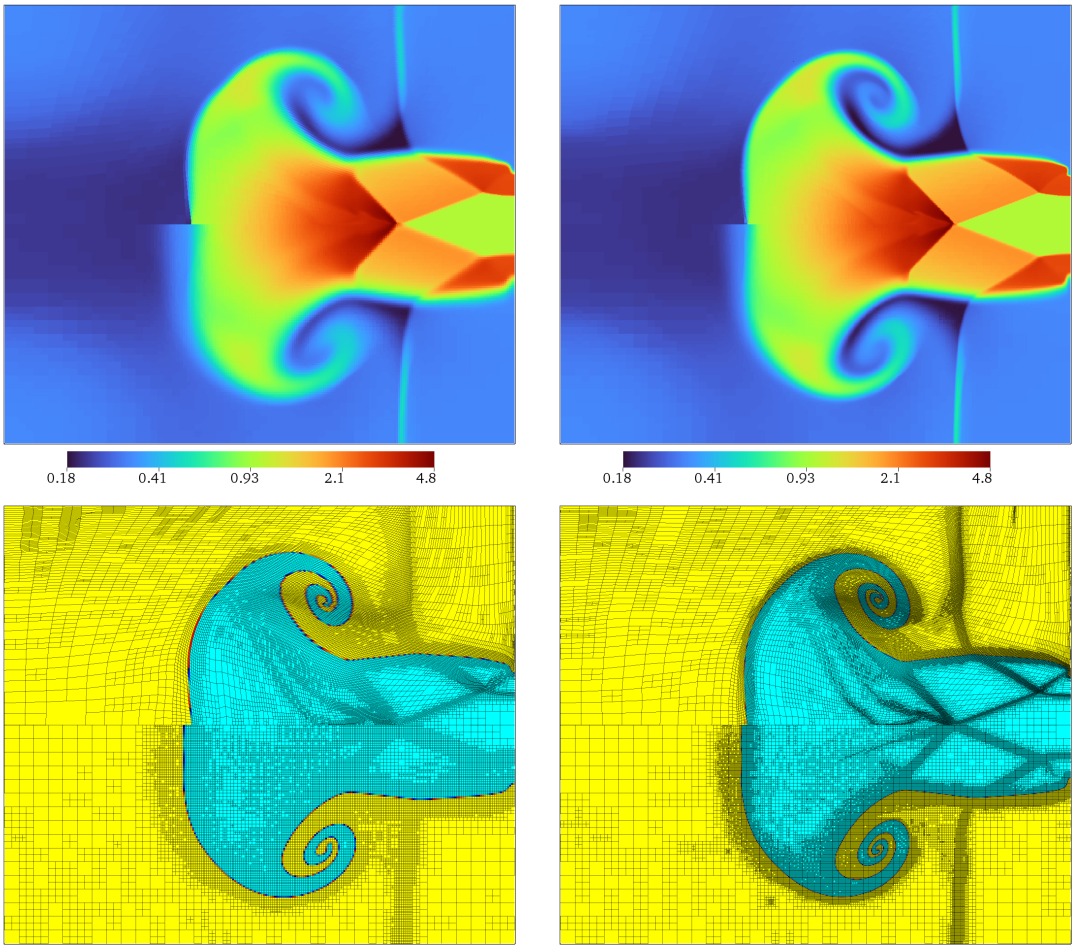


Figure 13: Numerical results for the 2D triple-point calculation at t=0.5, showing the material density in log scale (top) and the material VOF interface (bottom), for a maximum refinement level of 3 (left row) and of 4 (right row). In each panel, the upper part are results using ALE-AMR and the lower part are results using Euler-AMR.

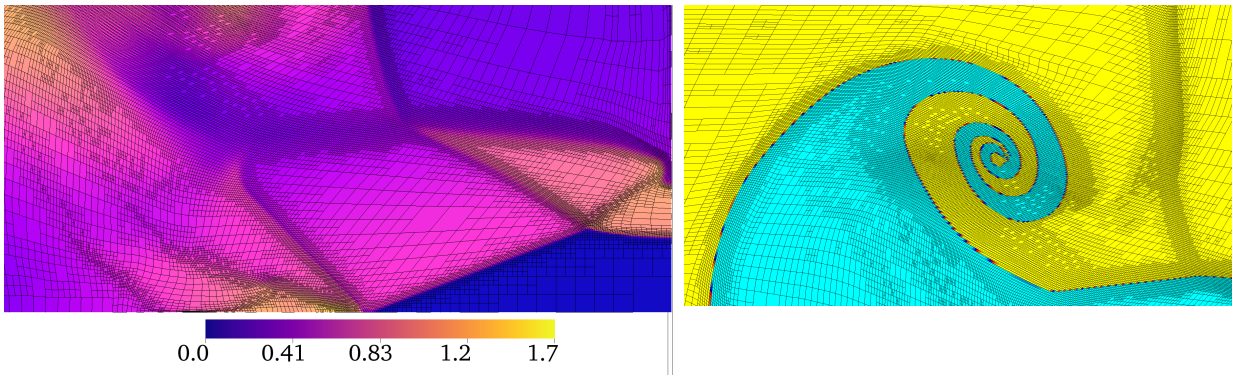


Figure 14: Close-up on the mesh of the ALE-AMR 2D triple-point calculation with $l_{\max} = 4$, showing (left) the Mach number around the various discontinuities traveling in the plume, and (right) the material interface on the vortex roll-up.

5.2. Triple point

We now consider a 3D extension of the 2D static triple point problem [28], which was also considered in Part I [4]. The mesh is divided into a region of pressure and density of 1 left of $x = 2$, and assigned a material id 1 with $\gamma = 1.4$. The right region has a lower pressure of 0.1 and is divided in the transverse direction between a corner at density of 1 and assigned with material 2 with $\gamma = 1.5$, and the rest is assigned material 1 with a density of 0.125. The box dimensions are $7 \times 3 \times 3$ and the boundary conditions are symmetries.

A first set of 2D calculations is carried out using ALE-AMR and compared to a Euler-AMR. The initial mesh resolution is $35 \times 1 \times 16$ and the maximum refinement depth is varied from 3 to 4. The ALE parameters are set to $\omega_{\min} = 5°$, $\omega_{\nu_0} = 20°$, $\omega_{\nu_{1/2}} = 35°$ and $\Omega_{\min} = 0.02$. Results for the material interface, density and Mach number at t=0.5 are shown in Fig. 13. While the ALE-AMR and Euler-AMR results are qualitatively similar, there are key differences that should be highlighted. Notably, we find that in the ALE-AMR case: (i) the density profiles are significantly sharper in the compression part (front) of the plume, (ii) the material interface position is shifted, (iii) the density peaks higher on axis, and (iv) discontinuities traveling inside the plume are better captured, which can be seen both in the physical profiles and in the arrangement of the mesh cells. We also see that there is significant distortion of the mesh in the ALE case, but that the actual quality of the mesh remains locally satisfying, owing to the combination of rezoning techniques we have employed. A close-up of the mesh on the roll-up and discontinuities in the ALE-AMR $l_{\max} = 4$ case is given in Fig. 14 and showcases the mesh quality.

These results suggest that there is a significant advantage to using the ALE-AMR framework when shocks are involved, when compared to Euler-AMR at the same level of maximum AMR refinement and for a second order indirect-Euler scheme. Our rezoning scheme is also shown to produce cells of good quality even in configurations combining strong vortices and contact discontinuities.

We now conduct 3D calculations and compare ALE-AMR and Euler-AMR results. We also carry out an Eulerian reference calculation for timing comparison. The non-AMR calculation is initialized on a $280 \times 128 \times 128$ grid, while the AMR calculations are initialized on $35 \times 16 \times 16$ cells and the maximum refinement depths is set to 3. Results are shown in Fig. 15. Comparing the position of the material interface, we find similar qualitative results between the two calculations. However, comparing the mach number profiles reveal that, as in the 2D case, discontinuities are better captured using ALE-AMR. A close-up of the back of the plume shows how the mesh in the ALE case is still moving with the flow to provide higher resolution (Fig. 15(bottom)). We also note that in both cases the flow has preserved its symmetry, which shows that neither the hydrodynamics nor the ALE/Euler algorithms broke the symmetry.

Finally, we present in Fig. 16 the effect of varying the linesweep control parameters. This parameter scan is performed on the 2D ALE-AMR case with $l_{\max} = 3$ and at $t = 0.25$ where we can easily compare the steepness of the density profile and the roll-up. These results show several typical features: (i) the equal-space algorithm produces high smoothness meshes but is more diffusive (see the front of the plume in the density profile), which is reminiscent of the Euler-AMR case, (ii) small variations in linesweep parameters have virtually no effects on the solution, (iii) too-high values of $\omega_{1/2}$ lead to poor-quality mesh around vortical regions (here, discontinuity in the material interface, irregular cells right of the roll-up), and (iv) too-high values of $\omega_{\min}$ can lead to bad quality cells in some regions (here, sliver cells in the front of the plume, which lead to smaller timesteps).

5.3. Rayleigh-Taylor single mode growth

We now consider the growth of a single-mode Rayleigh-Taylor spike. The test is an extension from Ref. [29] Sec 4.6, and was also considered in 2D for EUCCLHYD in [30]. This problem considers the unstable evolution of two immiscible fluids at hydrostatic equilibrium and in a gravity field, with a perturbed initial interface. The direction of the gravity field is taken to be $-\mathbf{z}$ with an amplitude of $g = 0.1$. The heavy fluid is on top (positive z), with a density of $\rho_h = 2$, and the light fluid is on the bottom with $\rho_l = 1$ (negative z). The interface between the fluids is located around $z = 0$. The initial interface position $z_i(x, y)$ is:

$$z_i(x, y) = 10^{-2} \cos(2\pi x/\lambda) \cos(2\pi y/\lambda) \tag{16}$$

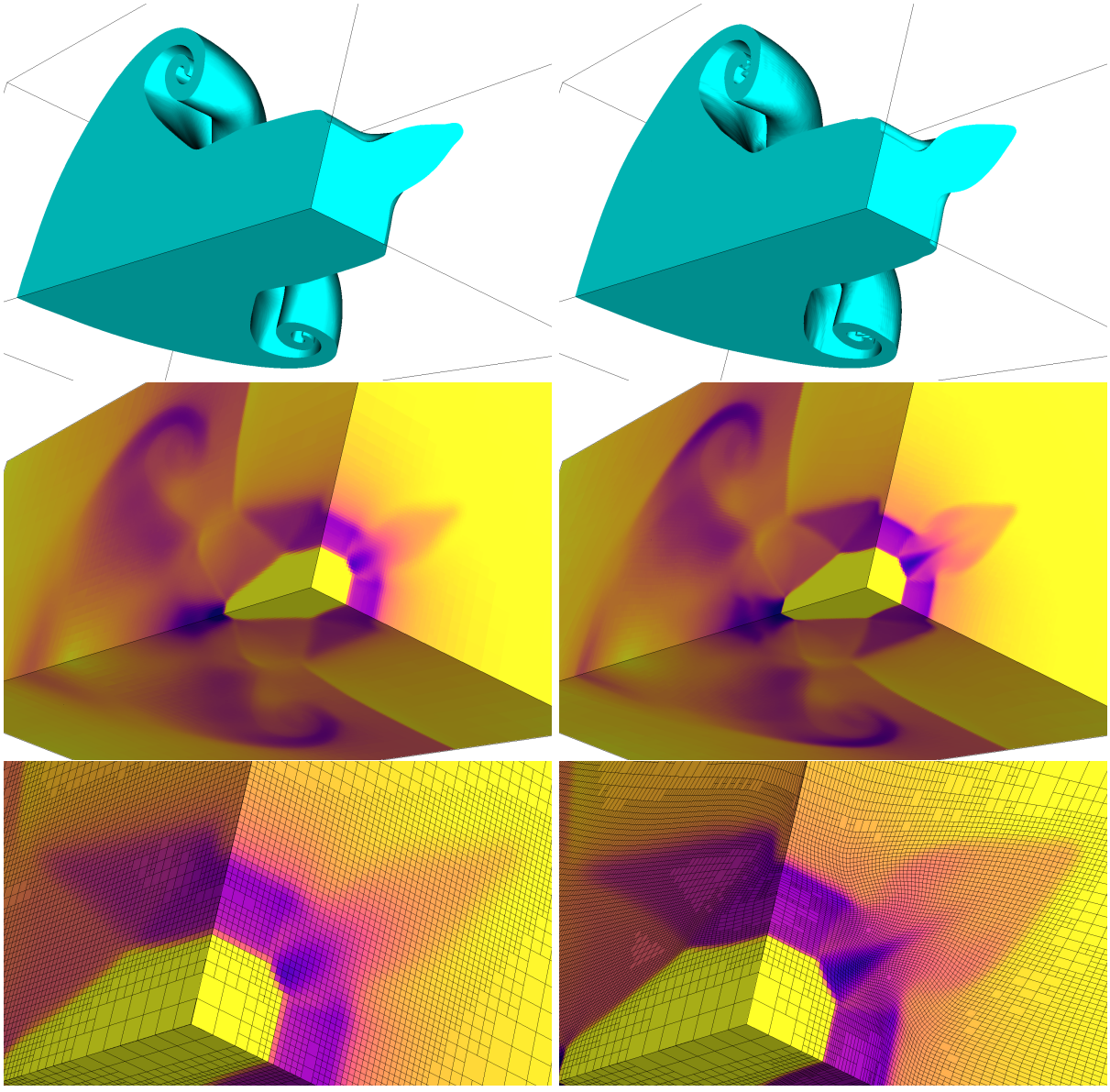
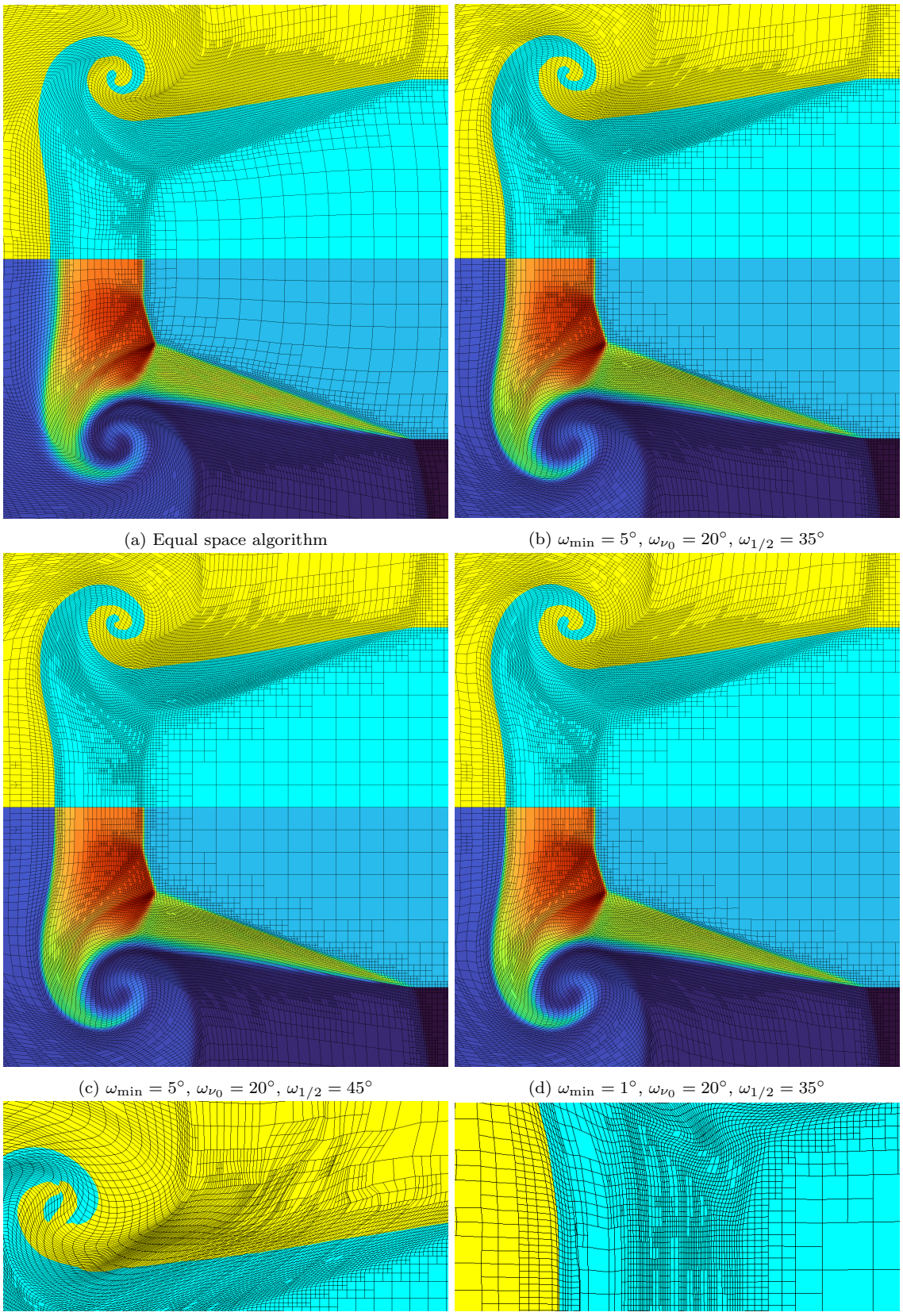


Figure 15: Numerical results for the 3D triple-point calculation at t=0.5, showing the material VOF interface (top) the Mach number (middle), and a close-up of the Mach number and mesh (bottom). Results are for (left) Euler-AMR and (right) ALE-AMR.



(a) Equal space algorithm

(b) $\omega_{\min} = 5°$, $\omega_{\nu_0} = 20°$, $\omega_{1/2} = 35°$

(c) $\omega_{\min} = 5°$, $\omega_{\nu_0} = 20°$, $\omega_{1/2} = 45°$

(d) $\omega_{\min} = 1°$, $\omega_{\nu_0} = 20°$, $\omega_{1/2} = 35°$

(e) $\omega_{\min} = 5°$, $\omega_{\nu_0} = 40°$, $\omega_{1/2} = 90°$
close-up right of the roll

(f) $\omega_{\min} = 10°$, $\omega_{\nu_0} = 20°$, $\omega_{1/2} = 35°$
close-up on the front of the plume

Figure 16: Variations in the triple point simulation results as a function of different linesweep control parameters. Results are taken at $t = 0.25$ in the 2D ALE-AMR case. The top parts of each panels shows the material volume fraction. Except for the bottom two figures, the bottom part of the panels show the material density. The bottom two figures are close-up on lower quality mesh features.

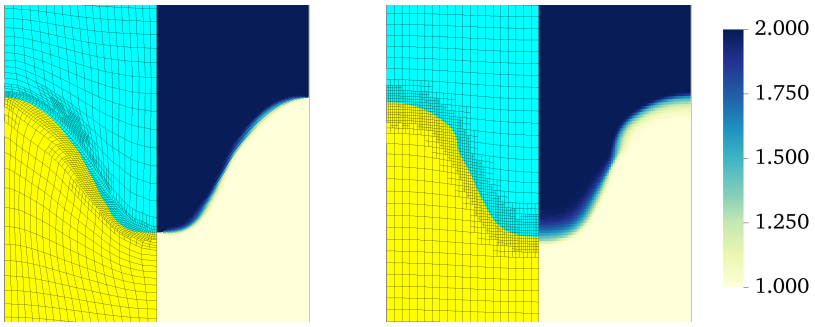


Figure 17: Results of the 2D RT test case at the end of the linear phase $t^\star = 0.45$ for (left) ALE-AMR and (right) Euler-AMR, both with $l_{\max} = 2$. In each panel, the left half shows the partial volume fraction of the materials, and the right half shows the density.

with $\lambda = 1/3$, such that one recovers the 2D case of [29] by setting $y = 0$. Note that in that configuration the wavenumber of the dominant mode is $k = 6\pi$ cm^{-1} and the Atwood number is $A_t = (\rho_h - \rho_l)/(\rho_h + \rho_l) = 1/3$, yielding a classical RT growth rate of $\sigma = \sqrt{A_t g k} = 0.793$. All simulations are ran to $t = 8.5$ (as in [29]) which corresponds to $t^\star = t\sqrt{A_t g/\lambda} = 0.9$, such that the instability is expected to be in its early non-linear phase.

The computation domain considers half a wavelength and we only model half of the spike, such that the box dimensions are $[\lambda/2, \lambda/2, 1]$ (and $[\lambda/2, 10^{-4}, 1]$ in 2D), with reflecting boundary conditions on all sides. Both fluids have the same polytropic index of $\gamma = 1.4$, and are initialized at pressure p corresponding to the hydrostatic equilibrium:

$$p(x,y,z) = \begin{cases} 1 + \rho_h g(z - z_{\max}) & \text{, if } z > z_i(x,y) \\ 1 + \rho_h g(z_i(x,y) - z_{\max}) + \rho_l g(z - z_i(x,y)) & \text{, otherwise.} \end{cases} \tag{17}$$

We conduct a first series of simulations in 2D. The simulations are initialized with 20x1x80 cells. The refinement depth is varied from 1 to 4, and the ALE parameters are set to $\omega_{\min} = 2°$, $\omega_{\nu_0} = 10°$, $\omega_{\nu_{1/2}} = 20°$ and $\Omega_{\min} = 0.02$. We compare Euler-AMR and ALE-AMR results at $t^\star = 0.45$ (end of the linear stage) in Fig. 17 for $l_{\max} = 2$. Both cases exhibit similar bubble height, with a slightly higher bubble in the Euler-AMR case and a smeared density profile. This comparison suggests that both cases have similar linear growth. We now compare the results at the final time $t^\star = 0.9$ for all cases in Fig. 18. Again, we note a slight difference in bubble height between the Euler-AMR and ALE-AMR cases, which we can link back to a slight difference in linear growth in the more diffusive Euler-AMR case. Most notably, we find that the RT bubble has a more arrow-head shape in the ALE case than in the Euler case, in which case the bubble head is flatter. There is here no reference to compare to, as even the results presented in Ref. [29] show a wide range of behavior depending on the algorithms chosen. Qualitatively, our Euler-AMR results are closer to the CFLFh and LL results in [29], while our ALE-AMR results for $l_{\max} = 1$ are very close to those reported for the BLAST code [31] (results can be found at [32]) for Q3-Q4 and Q7-Q8 finite elements (results at higher resolutions are not available for BLAST, and results for Q1-Q2 differ significantly). Finally, we provide a close-up on the vortex to show the mesh refinement and quality in Fig. 19. In the Euler-AMR case, the multi-material interface in the roll-up has started to break down. In the ALE-AMR case, the interface has made one less roll, and is still continuous. In that latter case one can see that the ALE has retained a good mesh quality.

We now run a second series of simulations in 3D using 20x20x80 cells and $l_{\max} = 2$. We also run a non-AMR Euler simulation with 80x80x320 cells in order to compare run-times. The ALE parameters are the same as in the 2D test. Results between ALE-AMR and Euler-AMR at t=8.5 for the position of the material interfaces are shown in Fig. 20. We observe similar qualitative differences as in the 2D case, with the ALE-AMR results yielding a slightly deeper bubble penetration. These results between the two

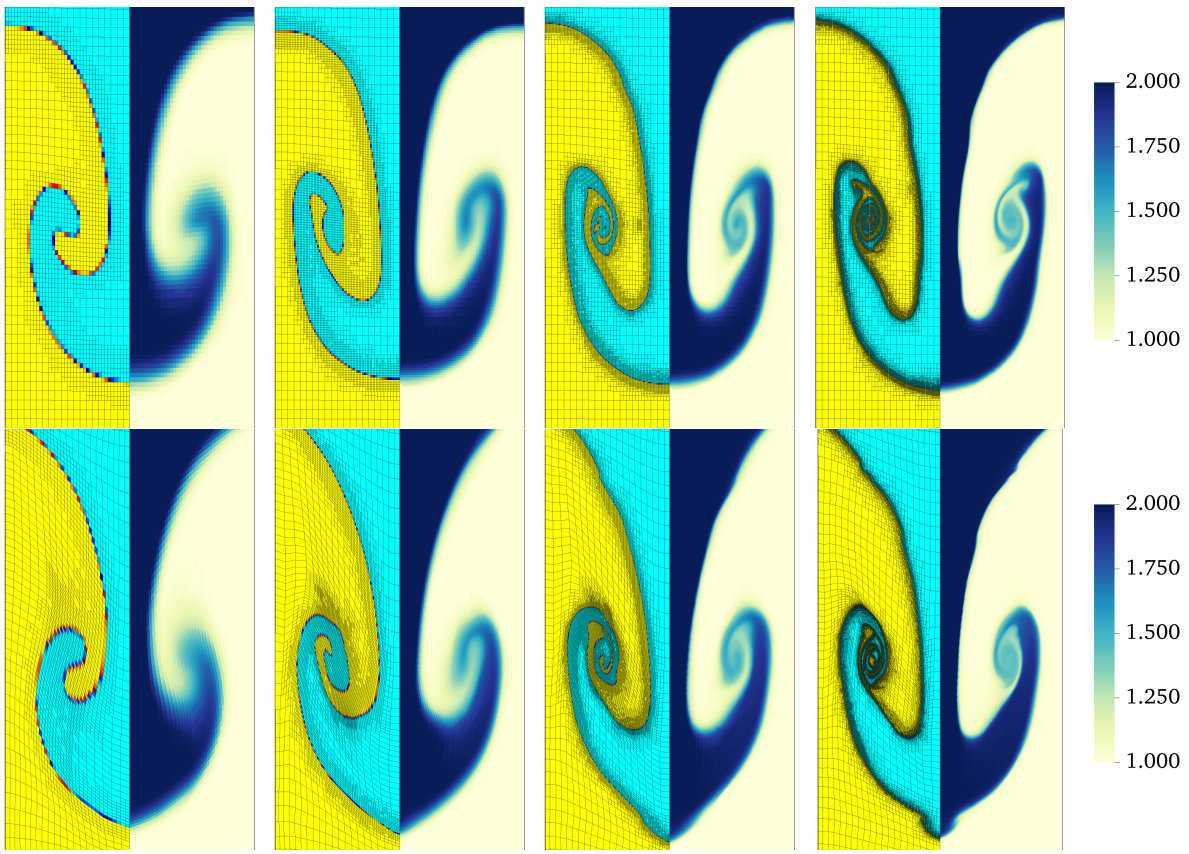


Figure 18: Results of the 2D RT test case with increasing degrees of refinement, from $l_{\max} = 1$ (left column) to $l_{\max} = 4$ (right column) at $t^\star = 0.9$. The top row shows Euler-AMR results and the bottom row are ALE-AMR results. In each panel, the left half shows the partial volume fraction of the materials, and the right half shows the density.

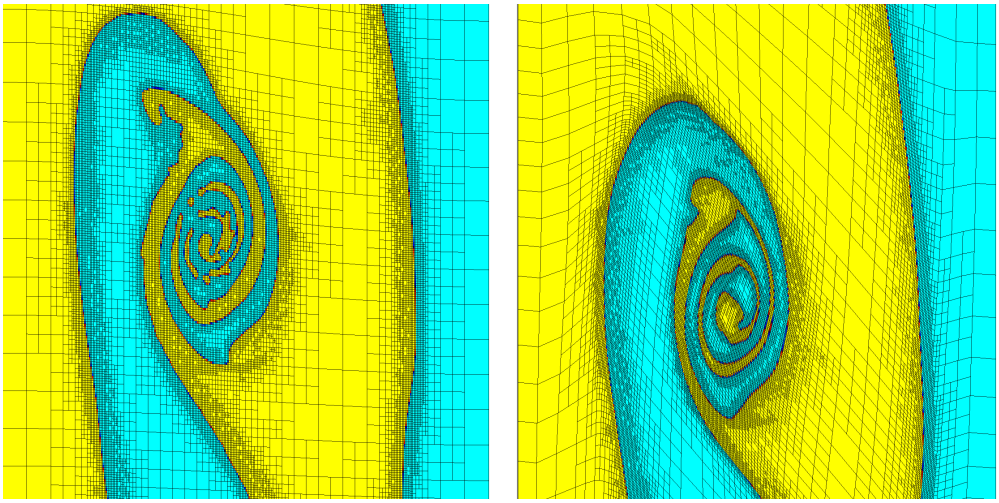


Figure 19: Close-up on the mesh for the 2D RT test case with $l_{\max} = 4$ for (left) Euler-AMR and (right) ALE-AMR. The color shows the partial volume fraction of the materials.

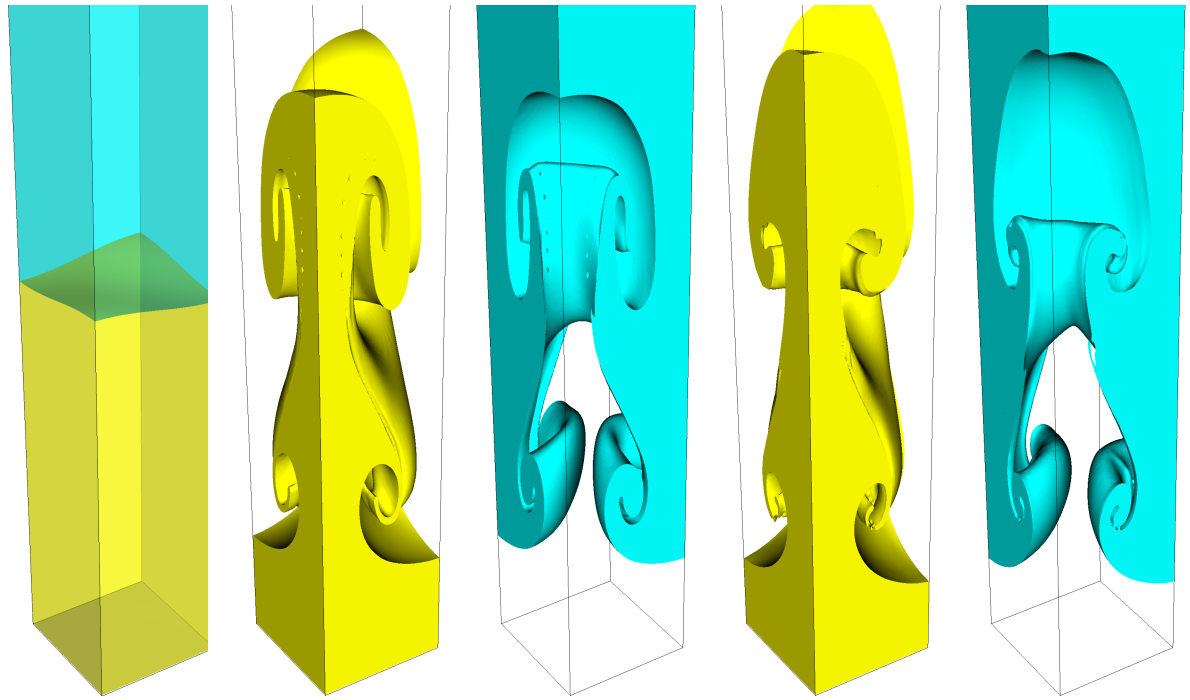
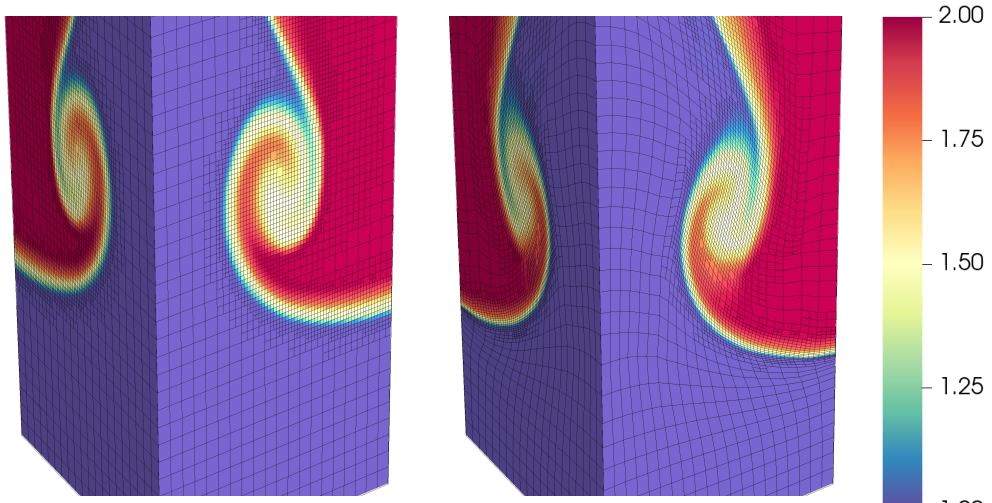


Figure 20: Results for the 3D RT test case showing the material interfaces: (a) initial state, (b,c) Euler-AMR at $t = 8.5$ s and (d,e) ALE-AMR at $t = 8.5$ s.

approaches are close nonetheless for the material interface position. A close-up of the mesh and material density around the bubble head for both cases is given in Fig. 21. The mesh has retained an good quality in the ALE-AMR case. As could be expected, cells are bunching up on the side of the bubble penetration, which leads to a density profile that is much steeper there than in the Euler-AMR case. While this may not be critical in this test case, it is a desirable and advantageous feature of ALE-AMR, that leads to less diffuse fronts.

5.4. ICF-like spherical implosion

We consider the implosion of a spherical target in a scenario relevant to ICF. A similar test was proposed for 3D cylindrical geometries in [33] and also explored in 2D in [30]. Similar tests in 2D and 3D are featured on the BLAST [31] webpage [32] (high-order finite element hydrodynamics research code). The test considers a spherical domain composed of an inner ball of radius 1, density 0.05, pressure of 0.1, and an outer layer of thickness 0.2, density 1 and pressure of 0.1. The two layers are composed of immiscible materials with the same polytropic index $\gamma = 5/3$. The outward boundary condition is set to an inward velocity of 5. The implosion of the shell drives a shock wave inside the target, which breaks out of the shell around t = 0.03, and then the target center at t = 0.085. The shock then rebounds and collides with the imploding shell layer around t = 0.33. A series of shock reflection occurs, and a shock reaches the boundary condition at t = 0.144. The simulation is stopped at t = 0.15. A diagram of the implosion dynamics is given in Fig. 22. In this setup, the interface between the two materials is both Richtmyer-Meshkov and Rayleigh-Taylor unstable, such that errors introduced by the numerical scheme will be amplified as a function of time. As such, we start by assessing the ability of the Lagrangian hydrodynamics to preserve 1D symmetry during the implosion. We note here that such errors can be introduced by the unstructured nature of the mesh. Here, an eigth of a sphere is meshed using 4 mesh blocks non-conformally joined to each other (see Fig. 1, such that there is an angular variation in the initial volume of the cells, as well as topological triple point features present in the mesh.



(a) (b)

Figure 21: Results for the 3D RT test case showing a close-up on the mesh of around the plume at $t = 8.5$ for (a) Euler-AMR and (b) ALE-AMR. The color shows the material density.

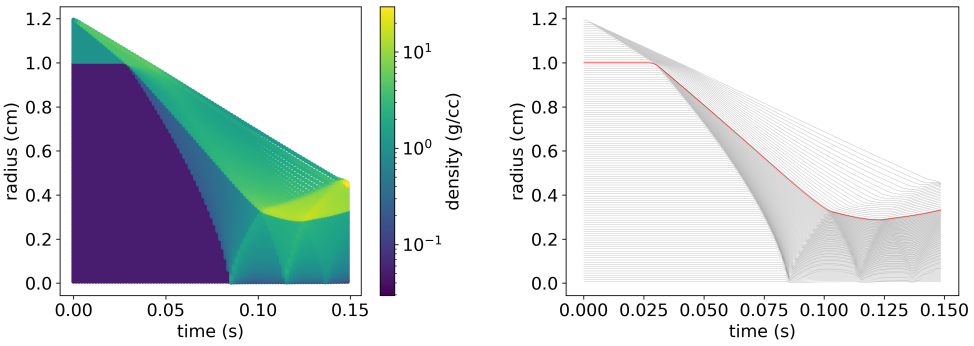


Figure 22: Results for the 1D ICF-like spherical implosion problem, showing (left) a flow diagram of the cell density as a function of position and time, and (right) the position of the Lagrangian cell centers during the implosion, whi the interface between the two materials shown in red.

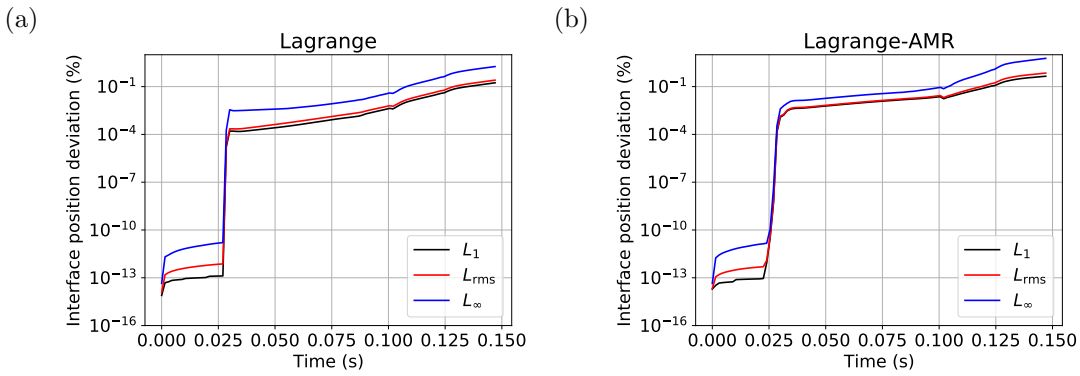
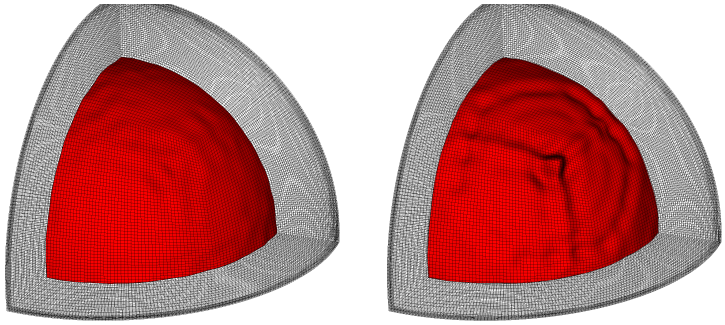


Figure 23: Comparison of the L_1, L_{rms} and L_∞ norm errors (in percents) of the position of the nodes at the interface between the two materials in the ICF-like spherical implosion problem. Results are shown for (a) Lagrange on 240 radial cells and 512 cells per great circle, and (b,c) Lagrange-AMR initialized on 60 radial cells and 128 cells per great circle and with a maximum refinement depth of 2.



(a) Lagrange (b) Lagrange-AMR

Figure 24: Multi-material interface at t=0.15 for the 3D ICF-like implosion test, for (a) the Lagrange and (b) Lagrange-AMR calculations. Perturbations are visible on the interface in both cases, and much stronger in the Lagrange-AMR case. This is thought to be related to the least-square procedure used for the second-order reconstructions during the ALE pre-processing and AMR steps.

5.4.1. Preservation of flow symmetry

We start by comparing a 1D Lagrange simulation ran on 240 cells with a 3D Lagrange simulation initialized on a 8-th sphere mesh of 240 radial cells and 512 cells per great circle (about 128 cells per angular direction on the 8-th sphere). The flow symmetry is assessed by tracking the position of the interface between the two materials, as is done in Refs. [30, 31]. We compare results for different norms: L_1, L_{rms} and L_∞ in Fig. 23 (a). At the final time of t = 0.15 s, the RMS symmetry error in that case is of 0.254%. For comparison, results given for Ref. [31] using high-order finite element Lagrangian simulation on unstructured meshes reach about 0.25% for Q1-Q0 (see [32]).

A second series of calculations are performed using Lagrange-AMR. The calculations are initialized on a mesh of 60 radial cells and 128 cells per great circle, and the maximum refinement depth is set to $l_{\text{max}} = 2$. Since this case is particularly sensitive to the growth of perturbations from noise, we set the refinement threshold to 0.01 instead of the value of 0.02 used in the other test cases. As usual, the regridding criterion used is the average second-order reconstruction error. Results for the symmetry of the interface position are shown in Fig. 23(b), and are found to be higher than the pure Lagrange result, reaching 0.75% RMS asymmetry. The final position of the material interface is shown in Fig. 24 alongside the pure Lagrange result. Significant perturbations are found to stem from the center of the mesh, at the non-conformal junction between the mesh blocks. We underline here that this test case is challenging: the mesh is not

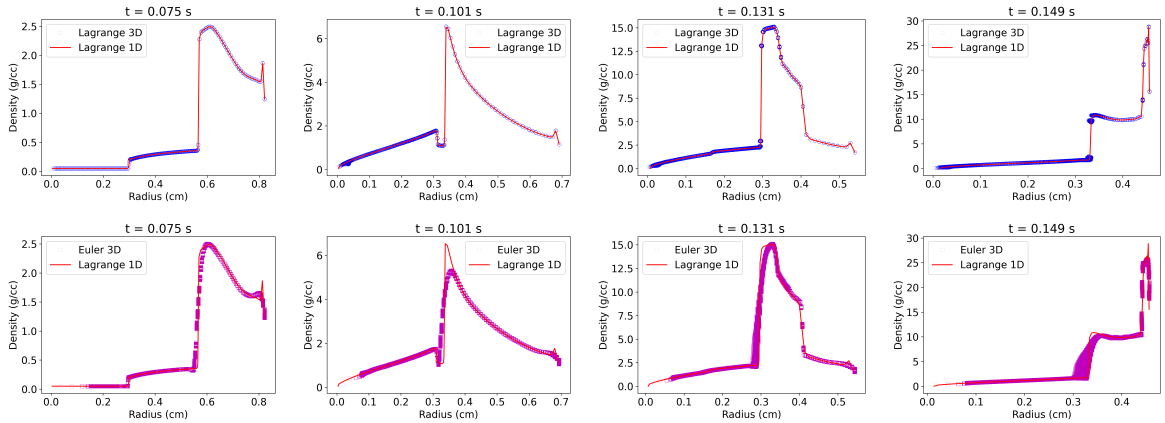


Figure 25: Comparison of density profiles between 1D Lagrangian in red and (top) 3D Lagrange in blue and (bottom) 3D ALE equal-space in purple, at various times for the symmetric ICF-like implosion test case. The different times were chosen: before the shock bounces at the center (t=0.075), after one bounce at the center but before collision with the incoming material interface (t=0.1), after two bounces at the center and bufer the second collision with the material interface (t=0.13) and at the final time, after 3 bounces at the center and 2 bounces on the interface (t=0.15).

angularly isotropic, the topology is degenerate around the mesh blocks junctions, the AMR is not spherical (degrees of freedom are not introduced at constant radii), and our second order scheme can only approximate the spherical geometry locally. Future work will be conducted to assess the exact origin of this perturbation in order to reduce it. As will be shown later, that lower accuracy in the Lagrange-AMR case does not significantly affect cases when perturbations are pre-imposed, which are more relevant of physical systems.

5.4.2. Accuracy of shock dynamics

We now compare density profiles with the 1D solution at different times for the 3D Lagrange-AMR case. For this test case, we are not able to perform Euler calculations. This is because we are unable to formulate a boundary condition that is equivalent to a velocity condition imposed on the moving outer mesh boundary in the Lagrangian case. However, we can emulate the accuracy of an Eulerian calculation by conducting an ALE calculation using the equal-space algorithm, such that the mesh is essentially Eulerian on a shrinking domain described by a Lagrangian moving boundary. The 1D Lagrange, 3D Lagrange-AMR and 3D ALE-AMR equal-space results are compared at various times of interest in Fig.; before the shock collapse at the target center (t = 0.75), after the shock collapse and before the first collision with the material interface (t = 0.1), after two shock rebounds on the center and two collisions with the interface (t = 0.13), and at the final time of t = 0.15. The results comparing 1D Lagrange and 3D Lagrange are shown in Fig. 25 (top). The 3D solution is found to track well the 1D density profile. Some loss of symmetry is observed at the final time around the density peak near the domain boundary. Comparison with the 3D ALE equal-space is shown in Fig. 25 (bottom). Not surprisingly, this quasi-Euler result is more diffusive and a less accurate representation of the various shocks. We also observe a higher degree of symmetry loss, which we believe is related to the aforementioned accuracy loss in the ALE least-square gradient computation procedure.

5.4.3. Perturbed case

We now modify the interface between the two materials using a spherical harmonics mode $l = 10$ and $m = 6$ perturbation:

$$r_i(\theta,\phi) = 1 + \delta r \frac{\Re(Y_{10}^{6}(\theta,\phi)))}{\frac{24}{4\pi}\frac{4!}{16!}} \tag{18}$$

where r_i denotes the radial position of the interface, δr is the initial perturbation amplitude, and the factor on the denominator is here to normalize the spherical harmonics to 1. We recall the definition of the spherical

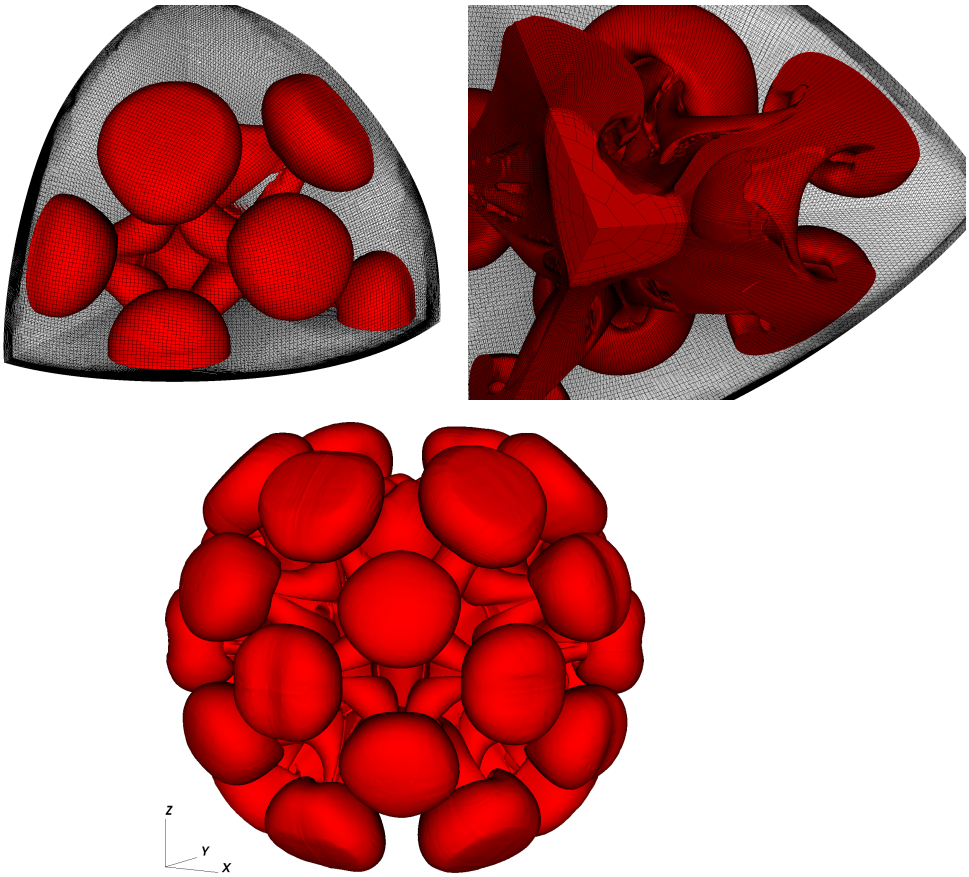


Figure 26: Visualization of the 3D material interface for the ICF-like implosion test case at the final time $t = 0.15$. The view is given from the front side (top-left) and from the back side through the equatorial plane (top-right). The full reconstructed interface using the mesh symmetries is shown in the bottom panel. The basis is shown to indicate the z-axis being vertical, for easier comparison with expected spherical harmonics modes.

harmonics Y_l^m:

$$Y_l^m(\theta,\phi) = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}} P_l^m(\cos(\theta))\exp(\imath m\varphi) \tag{19}$$

where P_l^m denotes the associated Legendre polynomials. The interface is resolved by modulating directly the initial position of the nodes in the mesh such that there are no mixed cells at $t = 0$.

The perturbed case is ran using a perturbation of $\delta r = 10^{-2}$ using ALE-AMR. The ALE parameters are set to $\omega_{\min} = 5°$, $\omega_{\nu_0} = 10°$, $\omega_{\nu_{1/2}} = 20°$ and $\Omega_{\min} = 0.025$. The final multimaterial interface is shown in Fig. 26 (top) from the front and back sides. On the front side, no peculiar perturbations that could be attributed to the triple-point boundary are visible, which suggests that the pre-imposed perturbation dominated the flow dynamics. The back side image better shows the roll-up on the plume head, similar to the Rayleigh-Taylor case. Using the mesh symmetries, a full 3D visualization of the flow is given in Fig. 26 (right), which allows us to recognize clearly the spherical harmonics $l = 10$, $m = 6$ pattern that was pre-imposed.

Close-ups of the mesh and density profiles on the equatorial-plane boundary (z=0) are shown in Fig. 27 from start to end time, every $\Delta t = 0.01875$. We can see how the mesh remains Lagrangian during part of the implosion, especially near the shock before it rebounds on the material interface. At later times and where perturbations are developing away from radial symmetry, the mesh is clearly being smoothed

through ALE. There is no visible mesh-attraction problem here, which stems from the use of the linesweep and normalized-CN, none of which are sensitive to volume variation around nodes.

5.5. Timings summary for 3D test cases

A summary table of the timings and number of cells for the various test cases is provided in Tab. 2 for 3D cases. While AMR is found to systematically reduce runtime, the speed-up factor for these pure hydrodynamics cases is variable, with factors ranging from ~ 1.5 to 6.5. The number of cells range from 0.2% of the full mesh to $\sim 10-60\%$ time-averaged sizes. While ALE-AMR is faster than ALE, the runtime of ALE vs. Euler is dictated by the rezoner's ability to prevent bunched and sheared cells, which is case-dependent. Here, this is controlled by the optimization metric part of the rezoner. Future efforts will enhance the control of bunched and sheared cells.

Case	Initial resolution	$l_{\max}$	AMR criterion	τ	min/max/ave nb. cells
3D Triple point					
Pure Euler	$280 \times 128 \times 128$	—	—	1	1/1/1
Euler–AMR	$35 \times 16 \times 16$	3	$\mathcal{A}_{o2}$ $(T = 0.02)$	0.42	0.002/0.33/0.23
ALE–AMR	$35 \times 16 \times 16$	3	$\mathcal{A}_{o2}$ $(T = 0.02)$	2.7	0.002/0.47/0.31
3D Rayleigh–Taylor					
Pure Euler	$80 \times 80 \times 320$	—	—	1	1/1/1
Euler–AMR	$20 \times 20 \times 80$	2	$\mathcal{A}_{o2}$ $(T = 0.02)$	0.15	0.03/0.39/0.12
ALE	$80 \times 80 \times 320$	—	—	8.6	1/1/1
ALE–AMR	$20 \times 20 \times 80$	2	$\mathcal{A}_{o2}$ $(T = 0.02)$	2.4	0.03/0.32/0.09
3D ICF-like implosion (non-perturbed)					
Lagrange	$240 \times 128 \times 128$	—	—	1	1/1/1
Lagrange-AMR	$64 \times 32 \times 32$	2	$\mathcal{A}_{o2}$ $(T = 0.01)$	0.19	0.041/0.99/0.58
Equal-space	$240 \times 128 \times 128$	—	—	3.45	1/1/1
3D ICF-like implosion (perturbed)					
ALE	$240 \times 128 \times 128$	—	—	1	1/1/1
ALE-AMR	$20 \times 20 \times 80$	2	$\mathcal{A}_{o2}$ $(T = 0.01)$	0.68	0.045/0.92/0.32

Table 2: Summary of normalized timings and number of cells for 3D cases in Part II, comparing AMR to non-AMR calculations for configurations with equal maximum mesh resolutions. Note that the Lagrange, ALE and Euler cases do not provide the same accuracy in general. For each case, the runtime τ is normalized to the reference non-AMR case. Number of cells are given for the minimum and maximum during the simulation, as well as an average calculated as $\int N_{\text{cells}} \mathrm{d}t / \int \mathrm{d}t$. These are also normalized to the number of cells of the non-AMR reference.

6. Conclusion

We have presented an algorithm for the regularization of nonconformal meshes in the ALE-AMR framework. This work follows on a first part that described second-order AMR algorithms for multi-material ALE hydrodynamics. While the first part focused on Lagrangian-AMR and Eulerian-AMR, this second part tackles the ALE-AMR aspects, in terms of mesh rezoning and disentangling. In this present paper, a heavy emphasis is put on several key points: (i) the Lagrangian aspect of the mesh should be respected, (ii) the method should not depend on the specific choice of an ad-hoc ideal element, and (iii) the method should be compatible with unstructured meshes and nonconformal nodes introduced by both AMR and the junction of mesh blocks. One algorithm that addresses the first point is a linesweep strategy in which nodes are rezoned along mesh lines. The second point is respected by using a weighted version of this linesweep algorithm, which aims at producing a rezoned mesh whose aspect ratios are computed from the smoothing of the Lagrangian aspect ratios along each mesh line direction. The weighted algorithm also feature a damping mechanism through which aspect ratios can be relaxed if required.

Starting from this existing technique, we presented significant modifications that should be made in order to adapt the algorithm to unstructured meshes presenting non-conformal nodes introduced by both

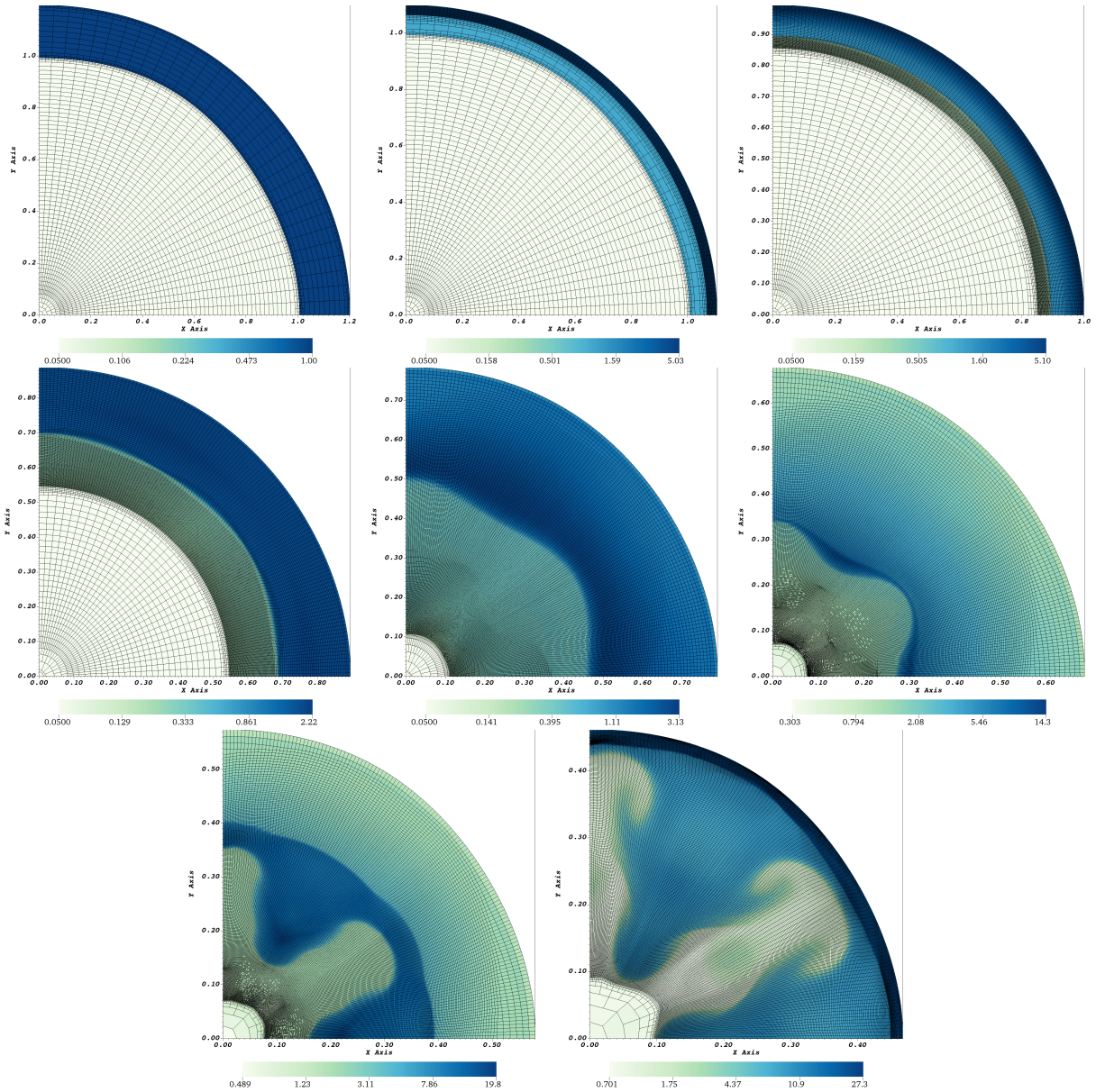


Figure 27: Slices of the 3D mesh and density along the equatorial plane ($z = 0$), shown every $\Delta t = 0.01875$ from start to end of the ICF-like implosion test case. The shock reaches the target center between the 5th and the 6th panel. Note that the axis are scaled so that the target fills the figure in each panel.

AMR and the junction of mesh blocks. These modifications entailed several points. First, we proposed an algorithm to identify mesh-lines in an unstructured framework. One solution proposed to handle non-conformal nodes and interfaces between mesh blocks is to use a local numbering system where there is no continuous line index across neighborhing nodes. We also showed that the junction of mesh blocks creates degenerate nodes which are not hanging, and proposed a method to deal with those. Second, we showed that the weighted linesweep algorithm could be recovered by introducing virtual nodes and virtual weights along mesh lines that are truncated by hanging nodes. Incidentally, this method was also applied to propose a generic handling of boundary conditions, where mesh-lines are also truncated. We showed that these adaptations allowed to recover a balanced weighted linesweep in the undamped case.

Having formulated the linesweep modifications, we showed how the algorithm still suffers from inherent flaws in practical situations. Most prominently, it is not able to prevent and resolve mesh shearing and bunching unless a high number of iterations are used with significant relaxation toward the equal-space method. A second flaw is that the weighted algorithm tends to move mesh nodes on large regions around aspect-ratio perturbations. To address the range and cell bunching issues, we proposed to control the activation and damping of the linesweep from an activation metric. For the sake of generality, we define that metric as being a measure of the local rotation of the mesh line with respect to its original rotation. This metric is generic because it is a fundamental limitation for the mesh to lose quality when mesh lines rotate significantly. To address the shear issue, we combined the linesweep with a standard node-based quality metric optimization, where we have chosen a metric that measures shear and smoothness. Throughout this process, we introduced four parameters to control the linesweep. Three of them are based on the measure of the line rotation and control, with increasing values; the activation of the method, the start of the damping of the linesweep weights, and the reaching of the maximum damping. The last parameter is a threshold value to activate the metric optimization. Using this framework, we also formulated a simple disentangling algorithm that aims to find the smallest stencil onto which nodes can be disentangled using the linesweep algorithm.

Finally, we presented a series of test cases to showcase the performance of the ALE-AMR algorithm. We performed comparisons between Lagrangian, Lagrange-AMR, ALE-AMR, Euler and Euler-AMR calculations in a variety of settings. We showed that, in general, ALE calculations tended to produce more accurate results than their Euler counterparts in terms of sharpness of discontinuities and shock tracking. We also found that the AMR configurations would in general run faster than their non-AMR counterparts, by factors typically ranging from 1.5 to 6.5 for modest refinement depths. In most cases, the AMR cases were found to be as accurate as their non-AMR counterpart, with the exception of the implosion test case, which showed a higher departure from symmetry in that case. However, in the perturbed implosion case, that departure from symmetry was no longer visible. The number of cells in the AMR cases was significantly smaller, ranging from time-averaged values of 10 to 60% of that of the non-AMR case. This is a significant advantage for physics models that scale non-linearly with number of cells, such as diffusion operators.

Overall, these results are promising for the ALE-AMR framework. They show the strength of ALE, that is to conserve the accuracy of the Lagrangian approach for discontinuity tracking, while allowing to model vortical flows as in the Euler approach. We have shown that the addition of AMR allows similar gains as in the Euler framework; reducing computation times while maintaining similar or acceptable accuracy. Another advantage of the ALE-AMR framework is that it allows to run calculations in Lagrange, Lagrange-AMR, ALE, ALE-AMR, Euler and Euler-AMR using the same set of algorithms, thus providing greater flexibility to model physical processes.

This work has perspectives that may be explored in the near future. First, we will explore if the discrete cell gradient approach used in the hydrodynamic solver can be generalized to other physical quantities. These generalized discrete gradients would then be applied to the ALE and AMR second-order algorithms in the hope of reducing the symmetry loss observed in the ICF-like implosion test case. Second, we envision that further improvements of the linesweep method can be made to reduce its reliance on control parameters. One such approach could be to guarantee that a maximum mesh-line rotation should not be exceeded, which would make the linesweep efficacy less dependent on the timestep. Third, the current runtime of ALE vs. Euler is mostly dictated by the ability of the rezoner to prevent cell shearing and bunching, which relies here on the metric optimization coupled to the linesweep. In that regard, the method we have presented

lacks control and would merit more work on the metric part. Finally, we could explore a more restrictive version of ALE-AMR in which nodes on the boundary of a given AMR level could be constrained in terms of position.

7. Acknowledgment

This work is supported by the U.S. Department of Energy (DOE), Office of Science, Fusion Energy Sciences, under Award No. DE-SC0024863: IFE-STAR and AWD-00007026: INFUSE.

References

[1] Clawpack Development Team. Clawpack software. http://www.clawpack.org, 2024. Version 5.10.0.
[2] B. Fryxell, K. Olson, P. Ricker, F. X. Timmes, M. Zingale, D. Q. Lamb, P. MacNeice, R. Rosner, J. W. Truran, and H. Tufo. Flash: An adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes. *Astrophysical Journal, Supplement*, 131:273–334, 2000.
[3] Ann Almgren, Maria Barrios Sazo, John Bell, Alice Harpole, Max Katz, Jean Sexton, Donald Willcox, Weiqun Zhang, and Michael Zingale. Castro: A massively parallel compressible astrophysics simulation code. *Journal of Open Source Software*, 5(54):2513, 2020.
[4] A. Colaïtis, S. Guisset, and J. Breil. An amr cell-centered scheme for 3d multi-material ale hydrodynamics. part i: Lagrangian and indirect euler amr algorithms. *Submitted for publication.*
[5] Richard Saurel and Carlos Pantano. Diffuse-interface capturing methods for compressible two-phase flows. *Annual Review of Fluid Mechanics*, 50(Volume 50, 2018):105–130, 2018.
[6] V. Maltsev, M. Skote, and P. Tsoutsanis. High-order methods for diffuse-interface models in compressible multi-medium flows: A review. *Physics of Fluids*, 34(2):021301, 02 2022.
[7] Alan M Winslow. Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh. *Journal of Computational Physics*, 1(2):149–172, 1966.
[8] Patrick M. 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.
[9] Patrick M. Knupp. Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantities. part ii—a framework for volume mesh optimization and the condition number of the jacobian matrix. *Int. J. Numer. Methods Eng.*, 48(8):1165–1185, 2000.
[10] C. J. Stimpson, C. D. Ernst, P. Knupp, P. P. Pébay, and DD. Thompson. "the verdict library reference manual". Technical report, https://itk.org/Wiki/images/6/6b/VerdictManual-revA.pdf, 2007.
[11] Patrick M. Knupp. Introducing the target-matrix paradigm for mesh optimization via node-movement. *Engineering with Computers*, 28:419–429, 2012.
[12] Jin Yao and Douglas Stillman. An equal-space algorithm for block-mesh improvement. *Procedia Engineering*, 163:199–211, 2016. 25th International Meshing Roundtable.
[13] S. Guisset, G. Damour, and J. Breil. Cell-centered indirect arbitrary lagrangian-eulerian numerical strategy for solving 3d gas dynamics equations. *Journal of Computational Physics*, 505:112903, 2024.
[14] Jérôme Breil, Guillaume Damour, Sébastien Guisset, and Arnaud Colaïtis. 3d mesh regularization within an ALE code using a weighted line sweeping method. *Computers & Fluids*, 292:106591, 2025.
[15] Carsten Burstedde, Lucas C. Wilcox, and Omar Ghattas. `p4est`: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing*, 33(3):1103–1133, 2011.
[16] Pierre-Henri Maire and Jérôme Breil. A second-order cell-centered lagrangian scheme for two-dimensional compressible flow problems. *International Journal for Numerical Methods in Fluids*, 56(8):1417–1423, 2008.
[17] G. Georges. *Développement d'un schéma aux volumes finis centré lagrangien pour la résolution 3D des équations de l'hydrodynamique et de l'hyperélasticité.* PhD thesis, Université de Bordeaux, 2016.
[18] Gabriel Georges, Jérôme Breil, and Pierre-Henri Maire. A 3D GCL Compatible Cell-Centered Lagrangian Scheme for Solving Gas Dynamics Equations. *J. Comput. Phys.*, 305(C):921–941, jan 2016.
[19] Bram van Leer. Towards the ultimate conservative difference scheme. ii. monotonicity and conservation combined in a second-order scheme. *J. Comput. Phys.*, 14(4):361–370, 1974.
[20] Bram van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *J. Comput. Phys.*, 32(1):101–136, 1979.
[21] Pierre-Henri Maire, Rémi Abgrall, Jérôme Breil, Raphaël Loubère, and Bernard Rebourcet. A nominally second-order cell-centered lagrangian scheme for simulating elastic–plastic flows on two-dimensional unstructured grids. *J. Comp. Phys.*, 235:626–665, 2013.
[22] D. L. Youngs. Time-dependent multi-material flow with large fluid distortion. In K. W. Morton and M. J. Baines, editors, *Numerical Methods for Fluid Dynamics*, 1982.
[23] Jan Velechovsky, Evgeny Kikinzon, Navamita Ray, Hoby Rakotoarivelo, Angela Herring, Mack Kenamond, Konstantin Lipnikov, Mikhail Shashkov, Rao Garimella, and Daniel Shevitz. Multi-material swept face remapping on polyhedral meshes. *J. Comput. Phys*, 469:111553, 2022.

[24] 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.
[25] Guillaume Damour, Sébastien Guisset, and Jérôme Breil. 3d mesh regularization based on a weighted line sweeping method. In *Proceedings of the 2024 International Meshing Roundtable (IMR)*, pages 132–143.
[26] Jérôme Breil, Hubert Alcin, and Pierre-Henri Maire. A swept intersection-based remapping method for axisymmetric reale computation. *International Journal for Numerical Methods in Fluids*, 77(11):694–706, 2015.
[27] Gábor Tóth. The divb=0 constraint in shock-capturing magnetohydrodynamics codes. *Journal of Computational Physics*, 161(2):605–652, 2000.
[28] Milan Kucharik, Rao V. Garimella, Samuel P. Schofield, and Mikhail J. Shashkov. A comparative study of interface reconstruction methods for multi-material ALE simulations. *J. Comp. Phys.*, 229(7):2432–2452, 2010.
[29] Richard Liska and Burton Wendroff. Comparison of several difference schemes on 1d and 2d test problems for the euler equations. *SIAM Journal on Scientific Computing*, 25(3):995–1017, 2003.
[30] J. 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, https://hal.science/tel-01467157, 2016.
[31] Robert W. Anderson, Veselin A. Dobrev, Tzanio V. Kolev, Robert N. Rieben, and Vladimir Z. Tomov. High-order multi-material ale hydrodynamics. *SIAM Journal on Scientific Computing*, 40(1):B32–B58, 2018.
[32] BLAST web page. `https://computing.llnl.gov/projects/blast`.
[33] D. L. Youngs. Multi-mode implosion in cylindrical 3d geometry. In *11th International Workshop on the Physics of Compressible Turbulent Mixing (IWPCTM11)*, 2008.