3D Mesh Regularization within an ALE Code Using a Weighted Line Sweeping Method

Jérôme Breil[a,*], Guillaume Damour[a], Sébatien Guisset[a], Arnaud Colaïtis[b]

[a] *CEA CESTA, 15 Avenue des Sablières, CS 60001 33116 Le Barp cedex, France*
[b] *Laboratory for Laser Energetics, 250 East River Rd, Rochester, NY 14623-1212*

Abstract

The Lagrangian formalism is widely used to simulate hydrodynamic responses in complex engineering applications, particularly those involving strong shock waves. However, as the mesh moves with the fluid, it can become highly distorted, requiring a regularization step. This involves constructing a new grid and remapping conservative quantities onto it to restore mesh quality. This work introduces a regularization method for block-structured meshes within a 3D ALE (Arbitrary Lagrangian-Eulerian) code. The proposed approach prevents mesh tangling while preserving the anisotropic features of the initial Lagrangian mesh. This regularization technique incorporates aspect ratio-based weights to control mesh smoothing. Unlike uniform rezoning techniques, this weighted approach maintains proximity to the Lagrangian mesh while improving mesh quality. The method effectively handles concave geometries by mitigating the grid attraction phenomenon, which typically leads to mesh concentration along concave edges. Numerical experiments demonstrate its efficiency in regularizing severely deformed meshes,

*Corresponding author
Email address: jerome.breil@cea.fr (Jérôme Breil)

and its integration within the ALE framework is validated on challenging hydrodynamic test cases, including the triple point problem.
Keywords: Lagrangian methods, Arbitrary Lagrangian-Eulerian hydrodynamics, Mesh regularization, Multi-dimensional.

1. Introduction

Solving the gas dynamics equations using the Lagrangian formalism enables the simulation of complex flows with strong shock waves. This formulation is well suited for the numerical simulation of the dynamic behavior of materials because the Lagrangian mesh moves with the material, allowing for accurate representation of material motion and reducing numerical diffusion errors. However, the use of this formalism can lead to significant mesh deformations due to its inability to handle large distortions of the computational domain. Conversely, the Eulerian description, with its fixed mesh, can handle significant distortions more easily but leads to strong numerical diffusion.

The Arbitrary Lagrangian-Eulerian (ALE) formulation combines the benefits of both Lagrangian and Eulerian methods. It allows the mesh to move with the material during the Lagrangian phase, followed by a rezoning step to construct a new regularized mesh using the nodes displaced during the Lagrangian phase. The aim is to retain the interface definition while improving mesh quality. Finally, a remapping procedure projects the conservative quantities onto the new regularized mesh. The ALE method can handle larger distortions than the Lagrangian method, while providing higher resolution than the Eulerian approach.

This work introduces a regularization method for block-structured meshes within a 3D ALE parallel code. The mesh regularization method used in this study is inspired by the Line-Sweeping regularization method introduced by Jin Yao [1, 2]. This iterative method regularizes each node based on a given geometrical rule within a structured stencil consisting of the node itself and its neighboring nodes. As highlighted in [1, 2], the Line-Sweeping regularization method yields similar results compared to other regularization methods (equi-potential relaxation, angle-based method). Additionally, it prevents grid attraction effects and produces good mesh quality on concave boundaries.

The originality of this work lies in two key aspects. Firstly, the proposed regularization method effectively addresses severe mesh tangling and can be easily integrated into a 3D ALE code. Its simplicity not only makes it straightforward to implement but also allows for potential extensions. Secondly, unlike the equal-spacing method examined in [3], this approach does not converge toward a uniform mesh. Instead, it stays close to the Lagrangian mesh, preserving its physical characteristics. The use of mesh aspect ratios as weights in the Line Sweeping method ensures anisotropic regularization, which is particularly effective in preserving the anisotropic features and aspect ratios of the Lagrangian mesh including geometry with concave boundary.

The paper begins by introducing the general ALE framework, which follows an indirect strategy. A 3D Lagrangian phase, as described in [4], is succeeded by a regularization step and a remapping procedure. The regularization method is then presented in detail, extending the concepts proposed in

[1, 2] within an ALE context. This approach demonstrates how a geometrical regularization methodology can handle highly deformed meshes effectively. The method is modified by introducing weights to control node distribution in the final regularized mesh. Examples illustrate its efficiency in regularizing severely deformed meshes without hydrodynamic motion, showing how well it works. Finally, the modified regularization method is integrated into the ALE framework and tested on various hydrodynamic scenarios, proving its reliability and flexibility.

2. Governing equations and different formalisms for solving Euler equations

To accurately describe the conservation principles of mass, momentum, and energy within an Arbitrary Lagrangian-Eulerian (ALE) framework, it is necessary to establish a connection between the material (or total) time derivative, inherent in the conservation laws, and the referential time derivative. This relationship serves as a fundamental concept in ALE finite element and finite volume models.

In this section, we present the differential and integral formulations of the ALE conservation equations for mass, momentum, and energy. The numerical resolution of the proposed set of equations is based on an indirect ALE strategy. Specifically, the first step involves solving the set of equations in the Lagrangian framework. During the Lagrangian step, where the mesh follows the fluid velocity, the mesh quality may deteriorate considerably. A regularization step is then undertaken to improve mesh quality while preserving some of the beneficial features of the Lagrangian phase.

This process initiates a rezoning step, during which a new regularized grid is constructed from the nodes displaced during the Lagrangian phase. Finally, a remapping procedure is applied to project all conservative quantities onto the newly regularized grid.

2.1. Lagrangian formalism

The Euler equations describe the motion of inviscid fluids, neglecting viscosity effects. They consist of conservation equations for mass, momentum, and energy. Solving these equations is challenging due to their nonlinearity and the presence of discontinuities in the flow. The Euler equations in Lagrangian form are written as follows:

$$\begin{cases} \rho \dfrac{d}{dt}\dfrac{1}{\rho} = -\nabla \mathbf{V}, \\ \rho \dfrac{d\mathbf{V}}{dt} = -\nabla P, \\ \rho \dfrac{dE}{dt} = -\nabla \cdot (P\mathbf{V}), \end{cases} \tag{1}$$

where t is time (in s), ρ is the density of the fluid (in $kg\ m^{-3}$), $\mathbf{V}$ is the velocity vector of a fluid particle (in $m\ s^{-1}$), P is the pressure (in Pa) and E is the total energy (in J). In the Lagrangian formalism, the Euler equations (1) are solved by following the motion of individual fluid particles over time.

2.2. Eulerian formalism

To obtain the Eulerian form of the Euler equations, the domains D_X and D_x are defined as follows: The material domain D_X is the set of material particles (fluid or solid) X that belong to the initial configuration Ω_0. The

spatial domain D_x is a set of fixed points x in space defining the spatial domain Ω_t, which corresponds to the deformed configuration of Ω_0 at time t.

These two domains are connected by a bijective mapping x that associates each point X with a trajectory $x(X,t)$. In other words, each point x corresponds to a particle that was initially in X and has moved to x at time t. In order to obtain the Eulerian form, the Lagrangian derivative of $f(x,t)$ is used:

$$\frac{df}{dt} = \left.\frac{\partial f(x(X,t),t)}{\partial t}\right|_X = \left.\frac{\partial f}{\partial t}\right|_x (x(X,t),t) + \left.\frac{\partial x(X,t)}{\partial t}\right|_X \cdot \frac{\partial f}{\partial x}(x(X,t),t),$$

equivalent to,

$$\frac{df}{dt} = \left.\frac{\partial f}{\partial t}\right|_x + \mathbf{V}\cdot\nabla f, \tag{2}$$

with,

$$\boldsymbol{V} = \left.\frac{\partial x(X,t)}{\partial t}\right|_X.$$

By introducing the Lagrangian derivative (2) in (1), the Euler equations are written in Eulerian form as:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla\cdot(\rho\mathbf{V}) = 0, \\ \rho\left(\dfrac{\partial \mathbf{V}}{\partial t} + (\mathbf{V}\cdot\nabla)\mathbf{V}\right) = -\nabla P, \\ \rho\left(\dfrac{\partial E}{\partial t} + \mathbf{V}\cdot\nabla E\right) = -\nabla\cdot(P\mathbf{V}). \end{cases} \tag{3}$$

In the Eulerian formalism (3), the fluid is observed at fixed points in space, and fluid properties are defined in terms of these fixed points.

2.3. ALE formalism

To obtain the ALE form of the Euler equations, the domain D_χ is the set of mesh points χ that can move during the motion (Figure 1). At each

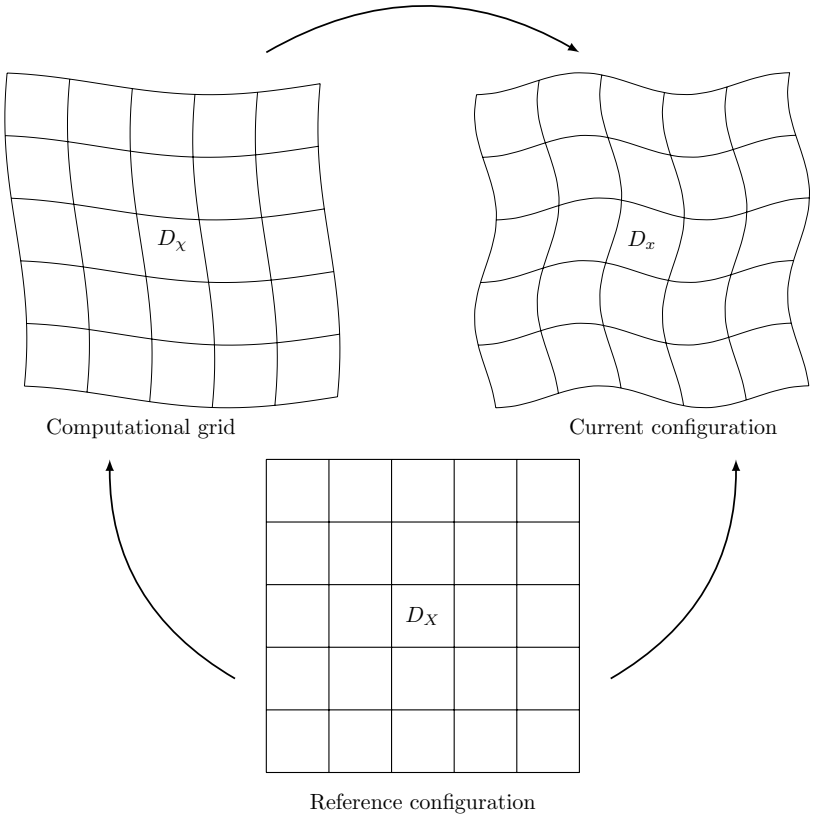


Figure 1: ALE computational mesh is independent of the material motion.

moment, this reference domain is connected to the deformed spatial domain Ω_t through a bijective mapping $x^*(\chi, t)$. In other words, each point x^* corresponds to a mesh point that was initially in χ and has moved to x^* at time t following the mesh with velocity $\mathbf{V}^{\text{ALE}}$.

$$\left.\frac{\partial f(x^*(\chi,t),t)}{\partial t}\right|_\chi = \left.\frac{\partial f}{\partial t}\right|_x (x^*(\chi,t),t) + \left.\frac{\partial x^*(\chi,t)}{\partial t}\right|_\chi \cdot \frac{\partial f}{\partial x}(x^*(\chi,t),t), \qquad (4)$$

equivalent to,

$$\left.\frac{\partial f}{\partial t}\right|_{\chi} = \left.\frac{\partial f}{\partial t}\right|_{x} + \mathbf{V}^{\mathrm{ALE}} \cdot \nabla f, \tag{5}$$

By subtracting equation (5) from equation (2),

$$\frac{df}{dt} = \left.\frac{\partial f}{\partial t}\right|_{\chi} + (\boldsymbol{V} - \mathbf{V}^{\mathrm{ALE}}) \cdot \nabla f. \tag{6}$$

The ALE form is obtained by replacing the particle derivative in the material reference with the particle derivative in the ALE reference (6), in Euler equations (1):

$$\begin{cases} \left.\frac{\partial \rho}{\partial t}\right|_{\chi} + (\mathbf{V} - \mathbf{V}^{\mathrm{ALE}}) \cdot \nabla \rho = -\rho \nabla \cdot \mathbf{V}, \\ \rho \left(\left.\frac{\partial \mathbf{V}}{\partial t}\right|_{\chi} + ((\mathbf{V} - \mathbf{V}^{\mathrm{ALE}}) \cdot \nabla)\mathbf{V} \right) = -\nabla P, \\ \rho \left(\left.\frac{\partial E}{\partial t}\right|_{\chi} + (\mathbf{V} - \mathbf{V}^{\mathrm{ALE}}) \cdot \nabla E \right) = -\nabla \cdot (P\mathbf{V}), \end{cases} \tag{7}$$

where $\mathbf{V}^{\mathrm{ALE}}$ is the velocity of the mesh (in $m.s^{-1}$).

Deriving the ALE integral form of the conservation equations begins with the Reynolds transport theorem applied to an arbitrary volume $\omega(t)$ whose boundary $\partial\omega(t)$ moves with the mesh velocity $\mathbf{V}^{\mathrm{ALE}}$.

$$\left.\frac{\partial}{\partial t}\right|_{\chi} \int_{\omega(t)} f \, dv = \int_{\omega(t)} \left.\frac{\partial f}{\partial t}\right|_{x} dv + \oint_{\partial\omega(t)} f \mathbf{V}^{\mathrm{ALE}} \cdot \mathbf{n} \, dS.$$

By substituting the spatial time derivative $\left.\frac{\partial f}{\partial t}\right|_{x}$ with the expressions given in (3) for the mass, momentum, and energy equations, the ALE integral form

can be derived:

$$\begin{cases} \left.\dfrac{\partial}{\partial t}\right|_{\chi} \displaystyle\int_{\omega(t)} \rho \,\mathrm{d}v + \int_{\partial\omega(t)} \rho \left(\boldsymbol{V} - \mathbf{V}^{\mathrm{ALE}}\right) \cdot \boldsymbol{n} \, ds = 0, \\ \left.\dfrac{\partial}{\partial t}\right|_{\chi} \displaystyle\int_{\omega(t)} \rho \boldsymbol{V} \,\mathrm{d}v + \int_{\partial\omega(t)} \left(\rho \left(\boldsymbol{V} - \mathbf{V}^{\mathrm{ALE}}\right) \cdot \boldsymbol{n} \, \boldsymbol{V} + P\boldsymbol{n}\right) \mathrm{d}s = \mathbf{0}, \\ \left.\dfrac{\partial}{\partial t}\right|_{\chi} \displaystyle\int_{\omega(t)} \rho E \,\mathrm{d}v + \int_{\partial\omega(t)} \left(\rho E \left(\boldsymbol{V} - \mathbf{V}^{\mathrm{ALE}}\right) + P\boldsymbol{V}\right) \cdot \boldsymbol{n} \,\mathrm{d}s = 0, \\ \left.\dfrac{\partial}{\partial t}\right|_{\chi} \displaystyle\int_{\omega(t)} \mathrm{d}v - \int_{\partial\omega(t)} \mathbf{V}^{\mathrm{ALE}} \cdot \boldsymbol{n} ds = 0, \end{cases} \tag{8}$$

where $\boldsymbol{n}$ denotes the unit outward normal of the volume $\partial\omega(t)$. This set of equations is commonly referred to as the ALE integral form of the Euler equations (3), corresponding to mass conservation, momentum conservation, and total energy conservation, respectively. The final equation represents volume conservation and is equivalent to the trajectory relation:

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

where $\boldsymbol{x}^{\mathrm{ALE}}$ defines the position of a node on the control volume surface, while $\boldsymbol{x}_0^{\mathrm{ALE}}$ represents its initial position. This governing set of equations must be closed using an equation of state. In the following, a closure relation of the form $P = P(\rho, \varepsilon)$ will be considered, where the internal energy ε is defined as $\varepsilon = E - \frac{1}{2}\boldsymbol{V}^2$. Note that in the special case where $\mathbf{V}^{\mathrm{ALE}} = \boldsymbol{V}$, the standard Lagrangian formalism is recovered (the control volume moves with the fluid velocity), while for $\mathbf{V}^{\mathrm{ALE}} = \mathbf{0}$, the usual Eulerian description is obtained. In this work, equations (8) are not solved directly. Instead, equations (1) are first solved, followed by a regularization step to provide the rezoned mesh χ, and then a remapping step, which corresponds to (6).

3. Lagrangian step

3.1. Mesh notations and geometrical splitting for GCL condition

To explain the overall numerical procedure in detail, we now introduce some mesh notations. We follow the notation conventions established in [4]. The spatial domain of interest, denoted as $\omega(t)$, is covered by non-overlapping polyhedra ω_c such that $\omega(t) = \bigcup_c \omega_c$. Each polyhedron is a volume bounded by polygonal faces. In this 3D context, since the faces may not be planar, the definitions of outward normal and area are not straightforward. In [4], the cells are subdivided by introducing an additional point p_f^* for each face, as illustrated in Figure 2. This approach is significant for the derivation of the ALE strategy. Firstly, as noted in [4], it helps establish the Geometric Conservation Law (GCL), which is a discrete compatibility relation between mass conservation and mesh geometry. Additionally, it ensures proper discrete symmetry preservation of the flow. Furthermore, as will be explained in the subsequent sections, this face splitting method is particularly useful for deriving the remapping procedure in 3D configurations. Regarding the notations introduced in [4], the index c represents a cell ω_c, f denotes a face, and p refers to a node. The triangles formed by the face splitting are denoted t_r. For completeness, we summarize all the notations used below:

- $\mathcal{P}(c)$ is the set of nodes p of cell c without the nodes p_f^*,
- $\mathcal{P}(f)$ is the set of nodes p of face f without the node p_f^*,
- $\mathcal{F}(c,p)$ is the set of faces f of cell c and sharing point p,
- $\mathcal{C}(p)$ is the set of cells c' holding node p,

- $\mathcal{T}(c)$ is the set of all the triangles t_r resulting from the splitting of the faces of cell c,

- $\mathcal{T}(c, f)$ is the set of triangles t_r resulting from the splitting of the face f of cell c,

- $\mathcal{T}(c, f, p)$ is the set of triangles t_r resulting from the splitting of the face f of cell c and sharing point p.

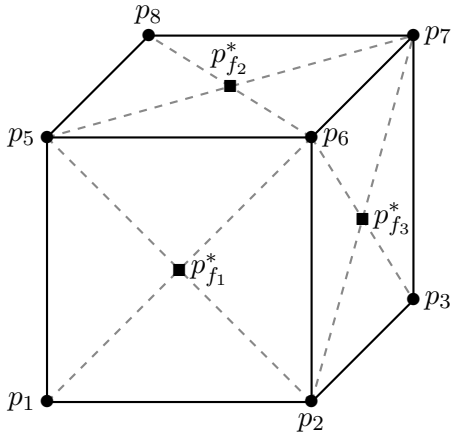


Figure 2: Taken from [4]. Result of the splitting of the cell faces into triangles using the face barycenter p^*_f. Simple case of a hexahedral cell with square faces.

An important result for the GCL condition enforcement from [4] is the definition of the face area vector $S_{pf}\boldsymbol{n}_{pf}$:

$$S_{pf}\boldsymbol{n}_{pf} = \frac{1}{3}\left(\sum_{t_r \in \mathcal{T}(c,f,p)} S_{t_r}\boldsymbol{n}_{t_r} + \sum_{t_r \in \mathcal{T}(c,f)} \frac{1}{N_{p,f}} S_{t_r}\boldsymbol{n}_{t_r}\right), \tag{9}$$

where $N_{p,f}$ is the number of nodes on the face f (without p^*_f). The $S_{pf}\boldsymbol{n}_{pf}$ term called face area vector may be seen as the contribution of face f

to the corner area vector [5] where the corner area vector writes $\boldsymbol{n}_p = \sum_{f\in\mathcal{F}(c,p)} S_{pf}\boldsymbol{n}_{pf}$.

3.2. 3D Lagrangian scheme

The Lagrangian step used is directly taken from [4]. In a Lagrangian scheme, the mass conservation equation requires the cell mass m_c to remain constant. Within this framework, the semi-discrete forms of the momentum and total energy conservation equations are expressed as follows:

$$\begin{aligned} m_c\frac{d\boldsymbol{V}_c}{dt} + \sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{F}(c,p)} S_{pf}P_{cfp}\boldsymbol{n}_{pf} &= \boldsymbol{0}, \\ m_c\frac{dE_c}{dt} + \sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{F}(c,p)} S_{pf}P_{cfp}\boldsymbol{V}_p\cdot\boldsymbol{n}_{pf} &= 0, \end{aligned} \tag{10}$$

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

$$\frac{d\boldsymbol{x}_p}{dt} = \boldsymbol{V}_p. \tag{11}$$

The nodal fluxes P_{cfp} and $\boldsymbol{V}_p$ are the remaining unknowns to be determined. Following [6] in order to ensure a positive entropy production, the pressure jumps may be written in terms of the velocity jump as follow:

$$P_{cfp} - P_c = Z_c\left(\boldsymbol{V}_c - \boldsymbol{V}_p\right)\cdot\boldsymbol{n}_{pf}, \tag{12}$$

where $Z_c = (\rho a)_c$ defines the acoustic impedance inside cell c and $a = \sqrt{(dP/d\rho)_\eta}$ the speed of sound. Considering the total momentum and energy conservation [4], the nodal velocity $\boldsymbol{V}_p$ may be computed by imposing a momentum balance around the node p. In this case, the node velocity $\boldsymbol{V}_p$ is

defined by:

$$\mathbb{M}_p \boldsymbol{V}_p = \boldsymbol{B}, \tag{13}$$

where

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

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

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

3.3. Linear reconstruction of the velocity and pressure fields

The second order extension of this Godunov-type scheme is made following the MUSCL method [4]. The pressure and velocity fields are linearly reconstructed in each cell Ω_c. It enables to determine more accurate fluxes which are then used in the Riemann solver for calculating the node velocity. Those linear reconstructions are written

$$\begin{cases} \tilde{P}_c(\boldsymbol{X}) = \bar{P}_c + \nabla P_c.(\boldsymbol{X} - \boldsymbol{X_c}), & (14) \\ \tilde{\boldsymbol{v_c}}(\boldsymbol{X}) = \bar{\boldsymbol{v_c}} + \overline{\overline{\nabla}} \boldsymbol{v_c}.(\boldsymbol{X} - \boldsymbol{X_c}), & (15) \end{cases}$$

where $\tilde{P}_c(\boldsymbol{X})$ and $\tilde{\boldsymbol{v_c}}(\boldsymbol{X})$ are the pressure and velocity extrapolated values at point $\boldsymbol{X} \in \Omega_c$, $\bar{P}_c$ and $\bar{\boldsymbol{v_c}}$ the mean values in cell c and $\boldsymbol{X_c}$ the cell centroid. The Barth-Jespersen limiter is used here as slope limiters [7].

3.4. Time discretization

The scheme presented here can be written under the general semi-discrete form

$$\frac{d\mathcal{U}}{dt} = \mathcal{F}(\mathcal{U}), \tag{16}$$

where $\mathcal{U}$ is the vector unknown and $\mathcal{F}(\mathcal{U})$ the numerical flux. To reach the second order in time we apply a Predictor-Corrector scheme defined as

$$\begin{cases} \mathcal{U}_c^{(1)} = \mathcal{U}_c^n + \Delta t \mathcal{F}\left(\mathcal{U}^n\right), & (17) \\ \mathcal{U}_c^{n+1} = \mathcal{U}_c^n + \dfrac{\Delta t}{2}\Big[\mathcal{F}\left(t\mathcal{U}^n\right) + \mathcal{F}\left(\mathcal{U}^{(1)}\right)\Big], & (18) \\ t^{n+1} = t^n + \Delta t. & (19) \end{cases}$$

The CFL condition used here writes

$$\Delta t = CFL \min_c \frac{\lambda_c}{a_c}, \tag{20}$$

where $CFL \in [0, 1]$ (Courant-Friedrichs-Lewy condition) is a corrective scalar, λ_c the characteristic space length defined as the cell volume divided by the maximal face area of the same cell and a_c the speed of sound. In all this study, the CFL condition has been set to $CFL = 0.45$.

4. Remapping step

The remapping step employed here is consistent with the method described in [3]. However, they are reiterated in this paper to enhance readability and ensure internal consistency. During the Lagrangian phase, various quantities such as density, velocity, pressure, and other conservative variables are computed and updated on the Lagrangian mesh. Depending on the mesh deformations, a regularization step may be necessary. After the

rezoning step, where the new regularized mesh is constructed, these quantities must be remapped onto the new mesh. This section provides a detailed description of the remapping process. Specifically, it is demonstrated how the face splitting technique introduced earlier is extensively used for effective projections. This numerical strategy involves a 3D sweeping procedure. To ensure robustness while maintaining accuracy, a Flux-Corrected Remap (FCR) strategy is employed, although it is not described here. Interested readers can find detailed information in [3, 8].

4.1. 3D sweeping strategy

The methodology presented in [9, 10, 11] is now extended to the 3D context as in [3]. This is achieved by splitting the cell faces into triangles using the face barycenter as described in the previous section. Consider ψ a conservative variable (obtained at the end of the Lagrangian step) defined on a collection of non-overlapping polygons c. The goal is to compute this variable on a new collection of cells $\tilde{c}$. Practically, given the conservative variable ψ on the collection c, the objective is to compute

$$\psi_{\tilde{c}} = \frac{1}{V_{\tilde{c}}} \left(\int_c \psi, dv + \sum_{p,\, p^+,\, p^{++} \in \mathcal{T}(c)} \int_{\mathcal{P}p,\, p^+,\, p^{++}} \psi, dv \right), \tag{21}$$

where $V\tilde{c}$ is the volume of the new cell $\tilde{c}$, computed as the sum of the volume of the original cell V_c and the signed volumes of the regions swept by the triangular faces during the mesh displacement.

The integral over the swept region $\mathcal{P}_{p,\, p^+,\, p^{++}}$, represented as prisms, is evaluated using piecewise linear reconstructions of ψ with a standard upwind strategy.

The reconstruction of ψ is performed as a piecewise linear function:

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

where ψ_c is the mean value in cell c, and $(\boldsymbol{\nabla}\psi)_c$ is a gradient computed using a least-squares approach [4]. To ensure monotonicity and prevent the generation of spurious oscillations, a Barth-Jesperson slope limiter is applied [7].

The volume $V_{\tilde{c}}$ of the new cell is determined by adding the volume of the old cell V_c to the sum of the volumes of the regions swept by the triangles associated with the faces. These swept regions, represented as prisms $\mathcal{P}p,,p^+,,p^{++}$, contribute a volume $V(\mathcal{P}p,,p^+,,p^{++})$, as shown in Figure 3. Using this swept region computation in the remapping procedure ensures the conservation of mass, total energy, and momentum throughout the remapping phase.

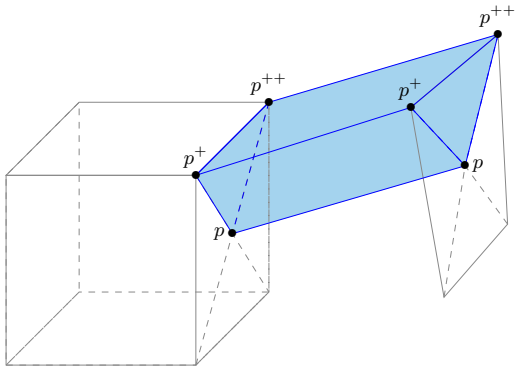


Figure 3: Representation of the swept region in blue, starting from an old triangle (formed by the points $p,,p^+,,p^{++}$) to the new one.

4.2. Weighted Line Sweeping Method

The regularization method presented here is the same as that described in the proceedings [12] from the IMR conference. It is reintroduced in this paper

to demonstrate its application within the ALE framework. This method differs from the one used in [3], highlighting its distinct approach and its coupling with the ALE code. The regularization method introduced in [3] corrects mesh tanglings and produces a good quality mesh. However, the new mesh may be very different from the Lagrangian mesh, resulting in a loss of accuracy. Indeed, the method tends to make the mesh uniform while the irregular Lagrangian mesh contains some physic features. To address this issue and to ensure that the regularized mesh remains close to the Lagrangian mesh, the Weighted Line Sweeping Method is now introduced.

To prevent the method from rezoning the mesh uniformly, the idea is to introduce a weighting factor to the Line Sweeping Method. Considering $\gamma_i \in [0,1]$ as the corresponding weight, the regularized position x_i^{m+1} of x_i^m is no longer positioned at the centre of the stencil but is now weighted as displayed in Figure 4.

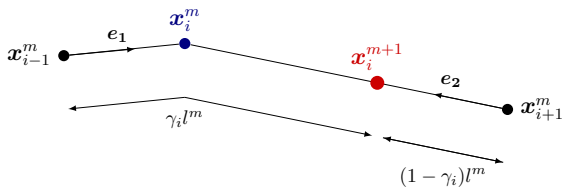


Figure 4: Weighted regularization of 1D stencil.

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

$$\boldsymbol{x}_i^{m+1} = \left(\boldsymbol{x}_{i-1}^m + \gamma_i l^m \boldsymbol{e_1}\right) \delta_{\{\gamma_i l^m \leq l_1^m\}} + \left(\boldsymbol{x}_{i+1}^m + (1-\gamma_i) l^m \boldsymbol{e_2}\right) \delta_{\{(1-\gamma_i) l^m < l_2^m\}}$$

where

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

and

$$\boldsymbol{e}_1 = \frac{\boldsymbol{x}_i^m - \boldsymbol{x}_{i-1}^m}{l_1^m},\ \boldsymbol{e}_2 = \frac{\boldsymbol{x}_i^m - \boldsymbol{x}_{i+1}^m}{l_2^m},$$

δ functions are define as

$$\delta_{\{\gamma_i l^m \leq l_1^m\}} = \begin{cases} 1, & \text{if } \gamma_i l^m \leq l_1^m \\ 0, & \text{otherwise} \end{cases},\ \delta_{\{(1-\gamma_i) l^m < l_2^m\}} = \begin{cases} 1, & \text{if } (1-\gamma_i) l^m < l_2^m \\ 0, & \text{otherwise} \end{cases}$$

In a 2D stencil, the regularized point $\boldsymbol{x}$ is computed through the following steps. First, two 1D stencils are extracted from the 2D stencil, using their respective triplets: $\{\boldsymbol{x}_{j-1}, \boldsymbol{x}_j, \boldsymbol{x}_{j+1}\}$ in direction j and $\{\boldsymbol{x}_{k-1}, \boldsymbol{x}_k, \boldsymbol{x}_{k+1}\}$ in direction k. The regularized points $\tilde{\boldsymbol{x}}_j$ and $\tilde{\boldsymbol{x}}_k$ are then calculated for each 1D stencil, as shown in Figure 5. Finally, the 2D regularized point $\boldsymbol{x}$ is obtained by averaging these intermediate points:

$$\boldsymbol{x} = \frac{\tilde{\boldsymbol{x}}_j + \tilde{\boldsymbol{x}}_k}{2}.$$

This approach provides the regularized point $\tilde{\boldsymbol{x}}$ of the 2D stencil, as illustrated in Figure 5.

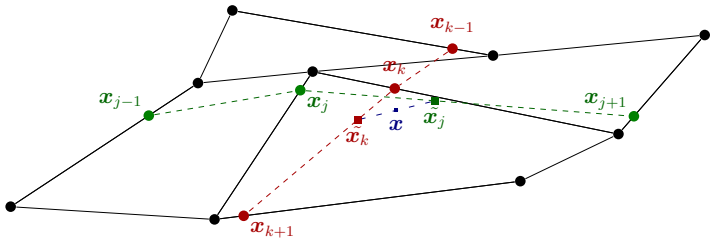


Figure 5: Regularization of a 2D stencil: the green point $\tilde{\boldsymbol{x}}_j$ and red point $\tilde{\boldsymbol{x}}_k$ correspond respectively to the 1D regularized points of the green and red branches. The final regularized point $\boldsymbol{x}$ is the arithmetic mean of these two points.

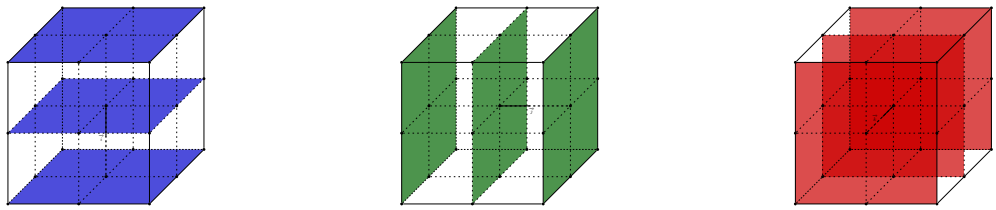


Figure 6: 2D stencils in each direction used for 3D regularization.

This method naturally extends to 3D, as each direction has three 2D stencils, as shown in Figure 6. For each direction, three regularized points are computed, resulting in a 1D stencil in each direction. These will respectively be denoted by $\{\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \boldsymbol{x}_{i,3}\}$, $\{\boldsymbol{x}_{j,1}, \boldsymbol{x}_{j,2}, \boldsymbol{x}_{j,3}\}$, and $\{\boldsymbol{x}_{k,1}, \boldsymbol{x}_{k,2}, \boldsymbol{x}_{k,3}\}$ for the i, j, and k directions. These 1D stencils are displayed in Figure 7.

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

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

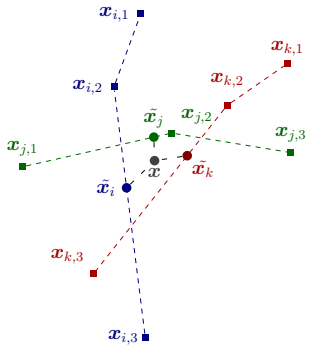


Figure 7: Regularization of a 3D stencil: blue point $\tilde{\boldsymbol{x}}_i$, green point $\tilde{\boldsymbol{x}}_j$, and red point $\tilde{\boldsymbol{x}}_k$ are the 1D regularized points in the three directions of the stencil.

4.3. Weighted regularization and Boundary conditions

The effectiveness of the weighted method is tested by demonstrating its ability to generate non-uniform meshes. A cubic domain with a non-uniform mesh is considered in Figure 8.a. The aspect ratios of this mesh are calculated and used to initialize the weights γ_i, γ_j, and γ_k at each node. The mesh is then randomly perturbed (Figure 8.b) to a degree that causes cells to overlap, resulting in a tangled configuration. The weighted line sweeping method is applied, and after 35 iterations, the tangled mesh is rezoned into the regularized mesh shown in Figure 8.f.

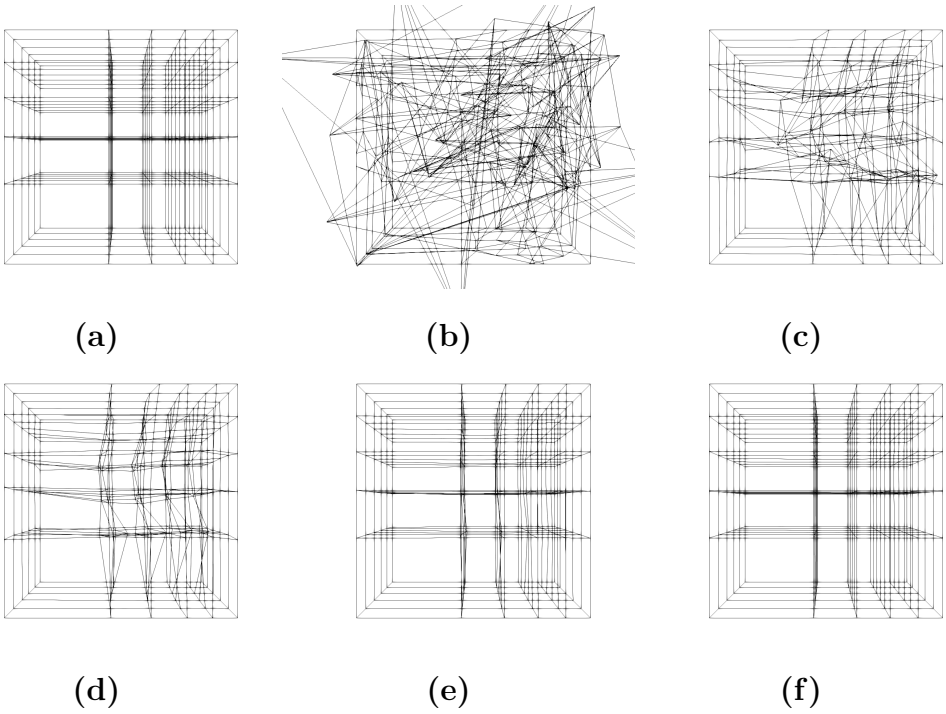


Figure 8: Regularization of a non uniform mesh with tangled cells. (a) initial mesh, (b) randomly perturbed mesh. (c), (d), (e) and (f) meshes respectively after 5, 10, 20 and 35 iterations of the weighted line sweeping regularization.

For boundary conditions, the points located on the faces of the cube are also perturbed while remaining constrained to the boundary planes. The regularization of these points follows the approach illustrated for the 2D plane in Figure 5. Similarly, for the edges of the cube, the nodes are displaced along the edges in accordance with the methodology shown in Figure 4. This approach ensures that the weighted regularization respects both planar and edge boundary conditions, preserving the coherence and consistency of the mesh structure. This phenomenon is illustrated during the iteration process shown in Figure 8.

4.4. Weights adapted to the Lagrangian mesh

In the previous section, the weights γ_i, γ_j, and γ_k at each node were determined based on the initial mesh, which corresponds to the desired regularized configuration. In this section, we define the weights required to regularize a Lagrangian mesh. This mesh possesses its own node distribution and corresponding aspect ratios, denoted as $\gamma_{L,i}$, $\gamma_{L,j}$, and $\gamma_{L,k}$ at each node. These aspect ratios carry physical information and are now used to initialize the weights Γ_i, Γ_j, and Γ_k for the regularization process. The weights Γ_i, Γ_j, and Γ_k assigned to the central node within a stencil are computed by averaging the Lagrangian aspect ratios of the neighboring nodes. This process is then iterated to refine the weights for optimal results.

For a 2D stencil (Figure 9), the weights Γ_i^p and Γ_j^p are initialized with the Lagrangian aspect ratios γ_L and are computed as follow:

$$\Gamma_i^{p+1} = \frac{\Gamma_{i-1}^p + \Gamma_{i+1}^p}{2} \qquad \Gamma_j^{p+1} = \frac{\Gamma_{j-1}^p + \Gamma_{j+1}^p}{2} \quad \text{with} \quad \Gamma_i^0 = \gamma_{L,i} \quad \text{and} \quad \Gamma_j^0 = \gamma_{L,j}.$$

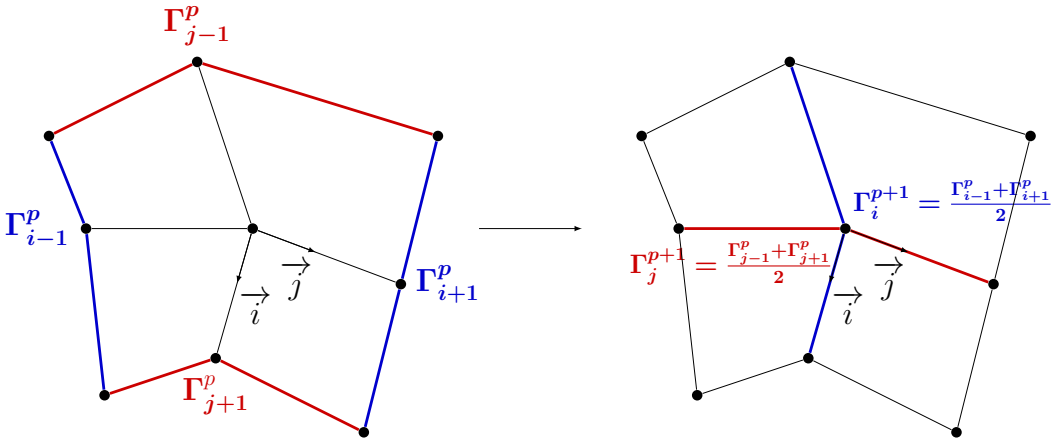
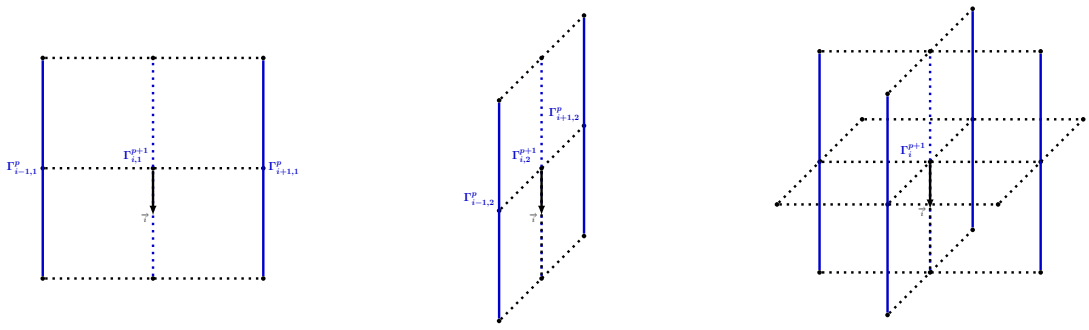


Figure 9: Computation of Γ_i^{p+1} and Γ_j^{p+1} for a 2D stencil.

For a 3D stencil, in each direction, there are four 1D stencils which are in the same direction as the main 1D stencil where the weight is to be fixed. The 2D stencils containing these 1D stencils with fully colored lines are considered, and the method for 2D stencils is reused. For example, the weight Γ_i^{p+1} is computed by averaging the weights $\Gamma_{i,1}^{p+1}$ and $\Gamma_{i,2}^{p+1}$ defined in the 2D stencils containing the 1D stencils with fully colored lines (Figure 10).

After a chosen number of iterations, the weights are fixed and used for mesh regularization. By considering these weights during the mesh regularization, the method can preserve the characteristics of the Lagrangian mesh while still achieving a better mesh regularity.

The weights computed directly from the mesh can also be adjusted. Notably, when $\Gamma_i = \Gamma_j = \Gamma_k = 0.5$, the Weighted Line Sweeping Method becomes identical to the Line Sweeping Method described in [3]. Conversely, if the Weighted Line Sweeping Method is applied to a non-uniform mesh with its initial weights, the mesh remains stationary, as the initial grid satisfies



Computation of $\Gamma_{i,1}^{p+1}$

Computation of $\Gamma_{i,2}^{p+1}$

Computation of $\Gamma_i^{p+1} = \dfrac{\Gamma_{i,1}^{p+1} + \Gamma_{i,2}^{p+1}}{2}$

Figure 10: The weight calculation method for 2D stencil is used twice to calculate the weight Γ_i^{p+1} in the i direction of a 3D stencil.

the regularization algorithm.

This behavior contrasts significantly with the equal-spacing method. To achieve a balance between these distinct behaviors, the initial weights can be adjusted using a coefficient θ. The modified weights, $\tilde{\Gamma}$, are calculated using the following equation:

$$\tilde{\Gamma}_i = \theta\Gamma_i + (1-\theta)0.5.$$

The θ coefficient should be chosen within the range $[0, 1]$. This adjustment process generates new weights, $\tilde{\Gamma}_i$, which enable tuning between the two approaches. Specifically, setting $\theta = 1$ preserves the initial weights, while $\theta = 0$ results in convergence towards the equal space method.

5. 3D ALE test cases

In this section, various classical test cases are computed using the meth-

ods previously introduced: the ALE with the Line Sweeping Method (also referred to as the equal space method or ALE with 0.5 weights, corresponding to $\theta = 0$, as explained in Section 4.4), the ALE with the Weighted Line Sweeping Method (also called ALE with adapted weights), the Lagrangian formulation, and the indirect Eulerian strategy, where the solution of the Lagrangian computation is directly remapped onto the initial grid. In the test cases presented here, the fluid is assumed to be a perfect gas, such that the fluid pressure is given by

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

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

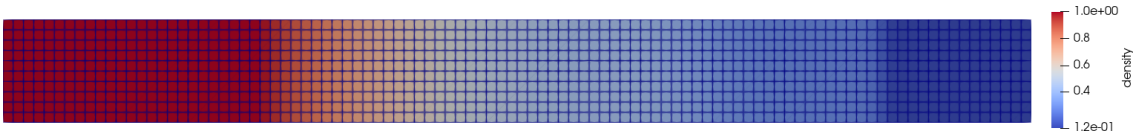
$$a = \sqrt{\frac{\gamma p}{\rho}}.$$

5.1. Sod test case

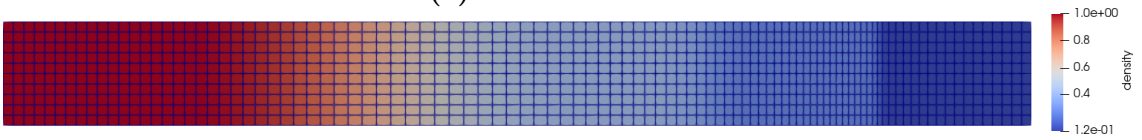
This numerical section starts with the well-known Sod test case [13]. Consider a spatial domain $x \in [0, 1]$ filled with a diatomic gas under the following initial conditions:

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

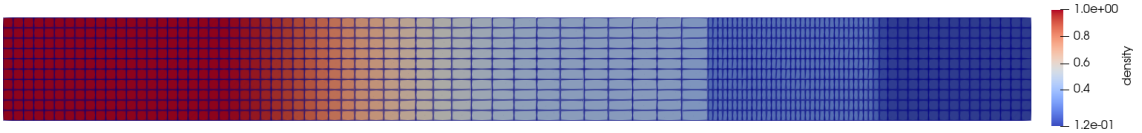
Even though this case is a simple 1D problem, it is performed using a 3D domain to assess the correct behavior of the presented numerical strategy. The regularization step is performed every 20 Lagrangian iterations, and before the mesh is regularized, the weights are initialized with 2 iterations.



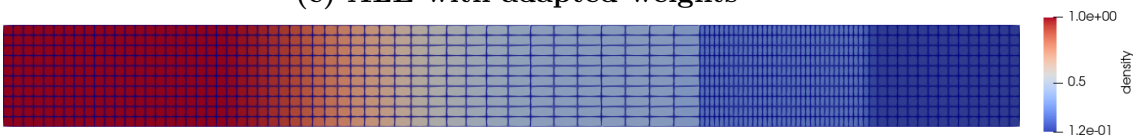
(a) Eulerian



(b) ALE with 0.5 weights



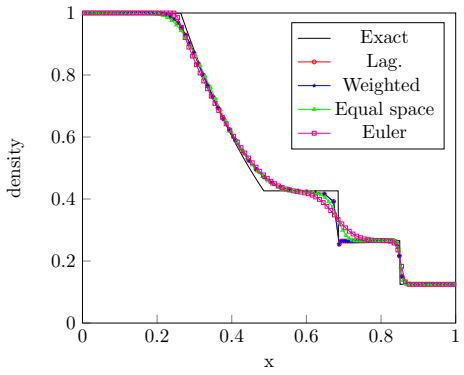
(c) ALE with adapted weights



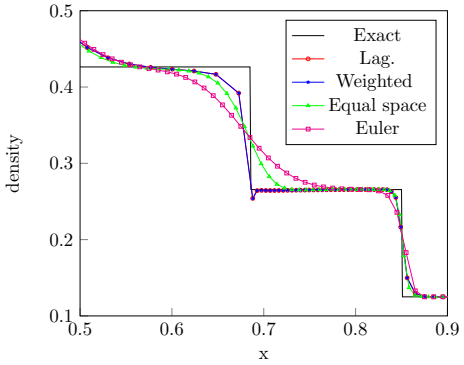
(d) Lagrangian

Figure 11: Comparison between the different methods for Sod simulation at time $t = 0.2s$ with $100 \times 10 \times 10$ cells.

The results are fairly similar between the two methods (Figures 11.c and 11.d). However, the density and the mesh computed with ALE using adapted weights coincide exactly with the purely Lagrangian simulation preserving the resolution of the shocks and the contact discontinuity (Figures 12.a and 12.b). The solution diffuses numerically for the ALE simulation with 0.5 weights and the indirect Eulerian simulation. This test case shows that the method with adapted weights retains the type of refinement shown in Figure 11.d.



(a) Density plots



(b) Zoom of density plots

Figure 12: Density plots comparing different methods for the Sod shock tube simulation at $t = 0.2s$ with $100 \times 10 \times 10$ cells.

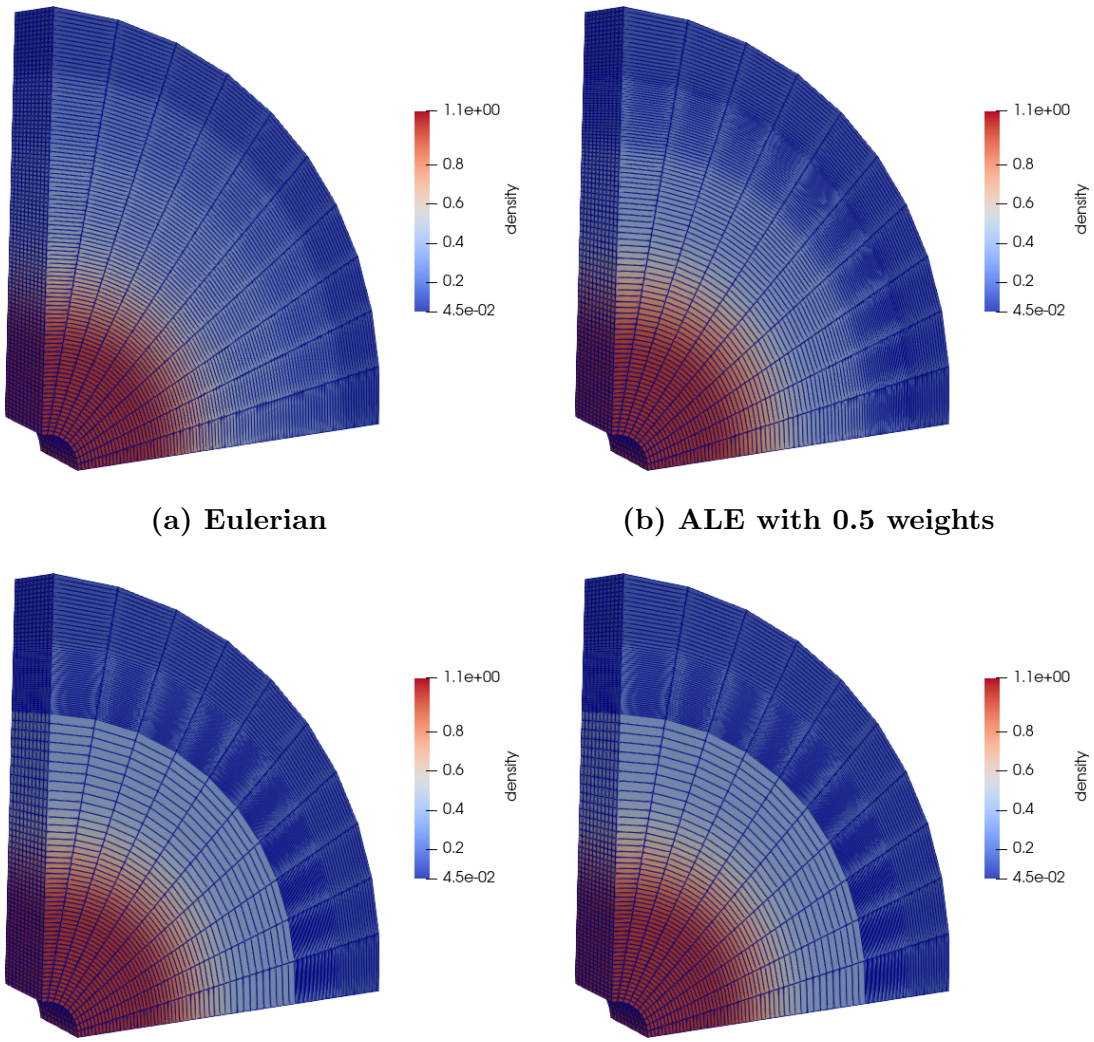
5.2. Cylindrical Sod test case

This test case is an adaptation of the Sod problem in a cylindrical configuration. The domain is a cylindrical shell with $r \in [0.1, 1.1]$, filled with a diatomic gas subject to the following initial conditions:

$$\begin{cases} (\rho_l, p_l, \boldsymbol{V}_l) = (1.0, 1.0, \mathbf{0}), & \text{for} \quad r \leq 0.6, \\ (\rho_r, p_r, \boldsymbol{V}_r) = (0.125, 0.1, \mathbf{0}), & \text{for} \quad r \geq 0.6. \end{cases}$$

For this test, a $100 \times 10 \times 10$ mesh is used, and the simulation is stopped at $t = 0.2s$. In cylindrical geometries with concave boundaries, the regularization step often results in an accumulation of cells near the origin (close to the internal radius in this case).

Figure 13.d shows the density profiles at the time $t = 0.2s$, using the Lagrangian method. More interestingly, Figure 13.b displays the density profiles obtained with the ALE method using the equal space method and,



(a) Eulerian **(b) ALE with 0.5 weights**

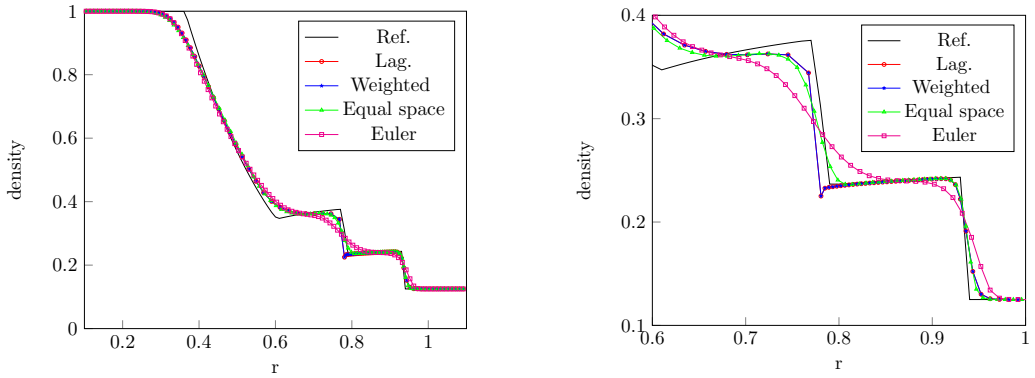
(c) ALE with adapted weights **(d) Lagrangian**

Figure 13: Comparison between the different methods for a cylindrical Sod problem at time $t = 0.2s$ with $100 \times 10 \times 10$ cells.

Figure 13.c shows the mesh using the adapted weights. In both cases, the regularization process is applied after 20 Lagrangian steps and weights are initialized with 2 iterations.

Firstly, the ALE regularization process presented here does not induce a collapse of the cell layers near the origin, even in the presence of concave boundaries. Secondly, unlike the behavior reported in [3], it is observed that for the ALE using adapted weights (Figure 13.c), the resulting mesh is identical to the Lagrangian mesh (Figure 13.d).

The density plots computed with ALE using adapted weights coincide exactly on the purely Lagrangian simulation (Figures 14.a and 14.b). Like for the previous Sod test case the solution diffuses numerically for the ALE simulation with 0.5 weights and the indirect Eulerian simulation. The solutions are compared to a reference simulation obtained from an Eulerian simulation on 10000 cells.



(a) Density plots **(b) Zoom of density plots**

Figure 14: Density plots comparing different methods for the cylindrical Sod shock tube simulation at $t = 0.2s$ with $100 \times 10 \times 10$ cells.

5.3. Sedov test case

The second test is the Sedov test case [14]. It consists in the propagation of a spherical shock wave. The space domain chosen is $[0; 1.2]^3$ filled with a diatomic gas with the following initial conditions

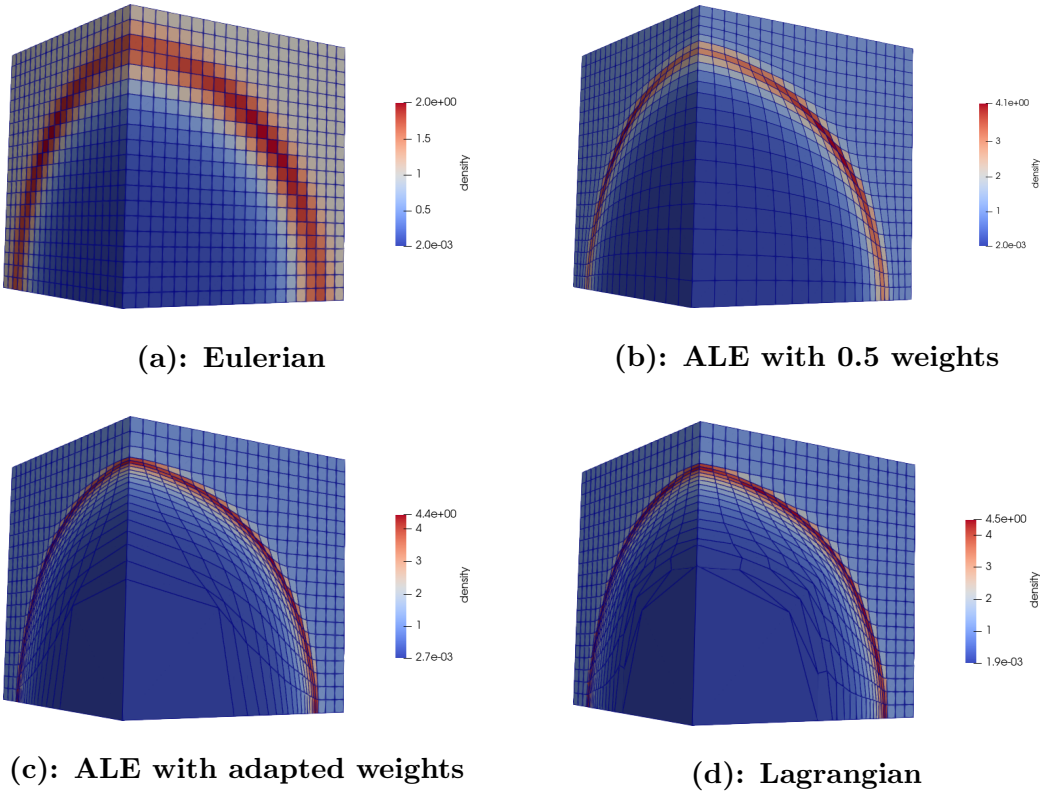
$$(\rho, p, \boldsymbol{V}) = (1, 10^{-6}, \mathbf{0}).$$

We follow the initial conditions given in [4] enforcing an initial pressure $p = (\gamma - 1)\varepsilon/V$ in the cells containing the space origin, with V the cell volume and $\varepsilon = 0.106384$ the initial specific internal energy. As recalled in [15], the maximum shock density reached at time $t = 1$ is 6 with a shock radius of 1.

The regularization step is performed every 20 Lagrangian iterations and before the mesh is regularized, the weights are initialised with 2 iterations. The ALE simulation with adapted weights (Figure 15.c) produces a mesh with a quality level that is intermediate between the purely Lagrangian simulation, characterized by concave and tangled cells (Figure 15.d), and the ALE simulation using the equal space method (Figure 15.b). This mesh provides better quality than the Lagrangian mesh and allows less numerical diffusion at shock level than the equal space method. This explains why the following results can be seen in the Figure 16 : $\max(\rho_{Lag}) > \max(\rho_{ALE_{weighted}}) > \max(\rho_{ALE_{\frac{1}{2}}}) > \max(\rho_{Eul})$.

5.4. Noh test case

The fourth test case considered is the Noh problem [16]. It consists in a inwards moving shock wave at a constant speed $D = 1/3$. This test case is a good one to assess the robustness of the studied numerical method. The initial conditions consists of a monoatomic gas defined as follows $(\rho, p, \boldsymbol{V}) =$



(a): Eulerian **(b): ALE with 0.5 weights**

(c): ALE with adapted weights **(d): Lagrangian**

Figure 15: Comparison between the different methods for Sedov simulation at time $t = 1.0$ s with $20 \times 20 \times 20$ cells.

$(1, 10^{-6}, \boldsymbol{e}_r)$ where $\boldsymbol{e}_r$ is the radial vector. Symmetry conditions are enforced on the boundaries holding the origin. Boundary conditions with $p = 10^{-6}$ are fixed on the other boundaries.

The regularization step is performed every 20 Lagrangian iterations. Before mesh regularization, the weights are initialized with 2 iterations. As observed in the Sedov test case, the Eulerian approach shows significant numerical diffusion, while the ALE simulations remain closer to the Lagrangian solution but are still slightly more diffusive (Figure 18). Regarding the mesh, the ALE meshes shown in Figures 17.b and 17.c are very similar to the La-

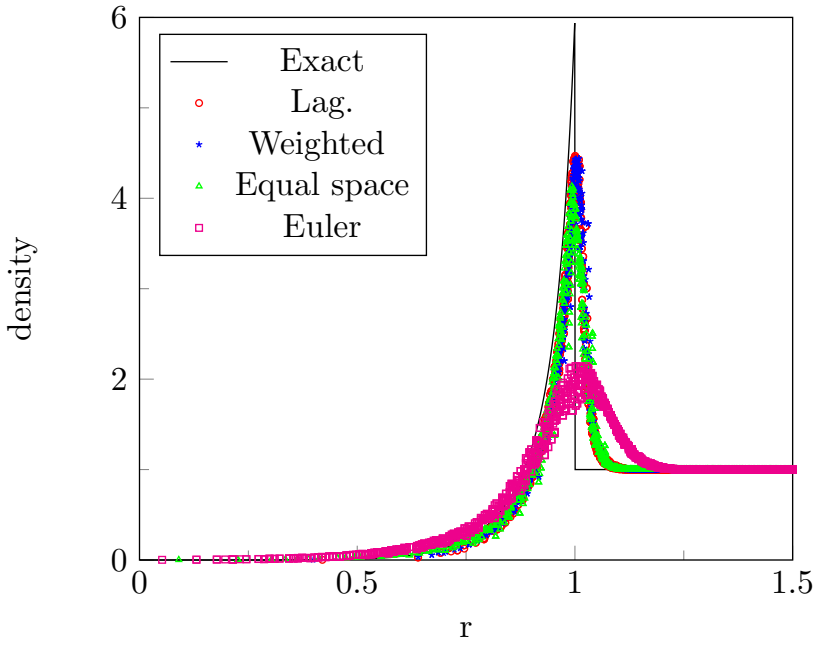


Figure 16: Density plots comparing different methods for Sedov simulation at time $t = 1.0s$ with $20 \times 20 \times 20$ cells.

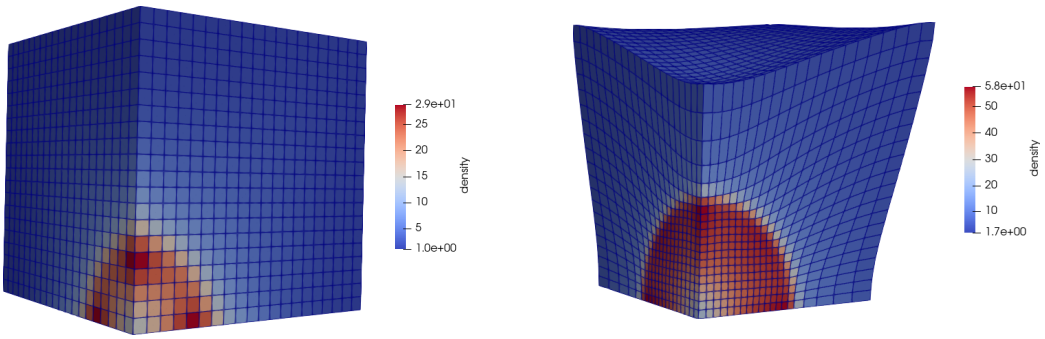
grangian mesh but exhibit some differences.

The method with adapted weights is more suitable, as the density plot is less scattered compared to the method with 0.5 weights in Figure 18. It should be noted that for this case, the FCR projection [3] not described in this paper is used to obtain ALE results.

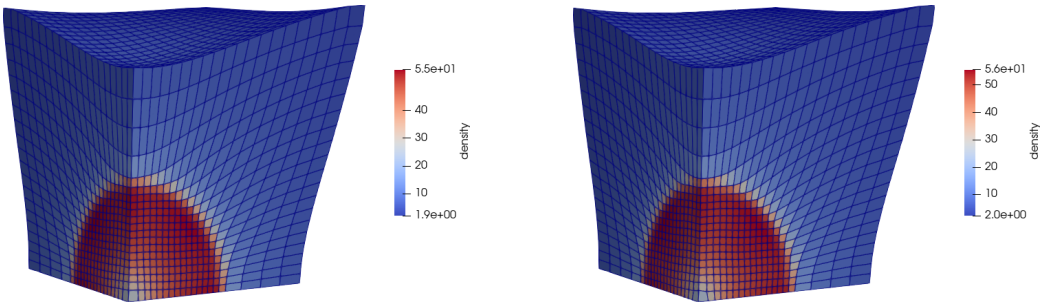
5.5. Triple point problem

The triple point test case is particularly interesting in order to assess the robustness of the ALE method. The triple point considers the evolution of three regions with three states detailed in Figure 19.

The regularization step is performed every Lagrangian iterations and be-



(a): Eulerian **(b): ALE with 0.5 weights**



(c): ALE with adapted weights **(d): Lagrangian**

Figure 17: Comparison between the different methods for Noh simulation at time $t = 0.6$ s with $20 \times 20 \times 20$ cells.

fore the mesh is regularized, the weights are initialised with 5 iterations and then 10 regularization iterations are done. The ALE simulation without weight is carried out with the same parameters, but the weights are fixed at 0.5.

Comparison with a full Lagrangian computation can not be performed since this test case suffers from severe mesh tangling as shown in Figure 20.b. The weighted method provides an almost Lagrangian simulation of the triple point while for the same number of iterations, as shown in section in

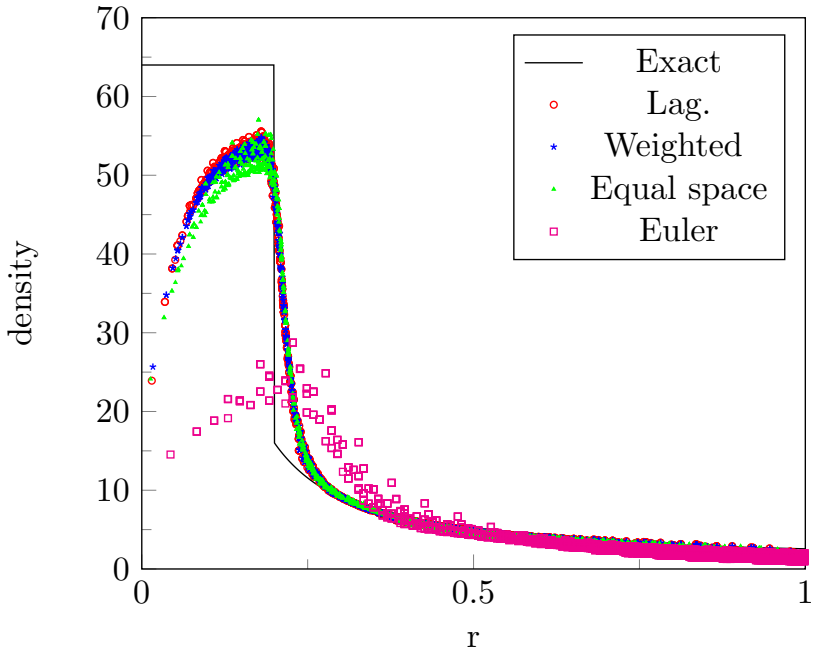


Figure 18: Density plots comparing different methods for Noh simulation at time $t = 0.6s$ with $20 \times 20 \times 20$ cells.

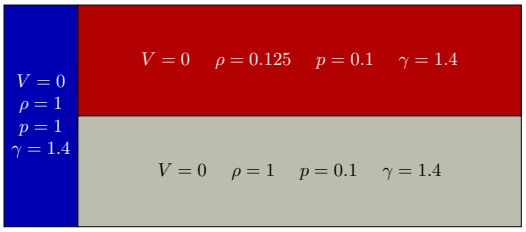
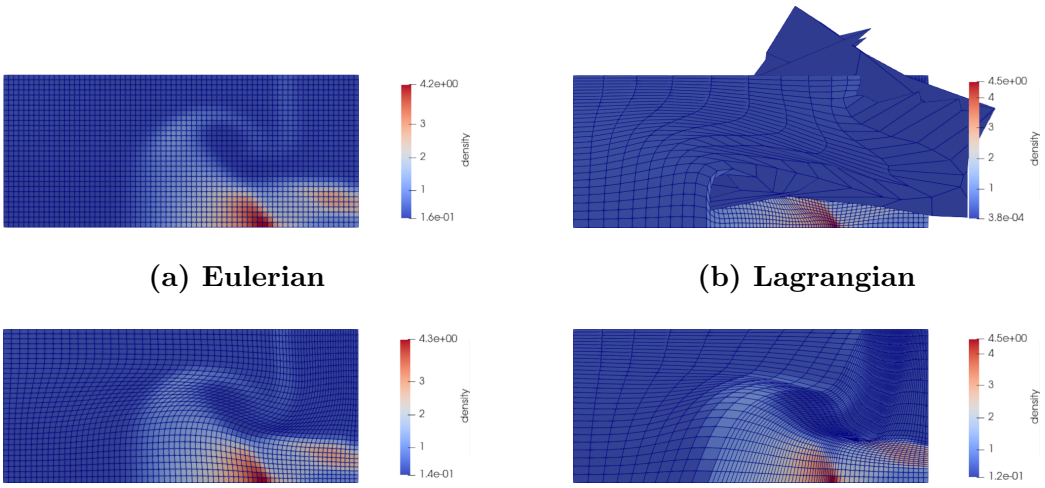


Figure 19: Triple point domain definition.

Figure 20.c, the ALE method with 0.5 weights makes the mesh too uniform and the simulation is comparable to the indirect Eulerian simulation in 20.a. However, it is observed that the weighted method over-preserves aspect ratios at the winding boundary. This aspect ratio, which is close to the Lagrangian



(a) Eulerian **(b) Lagrangian**

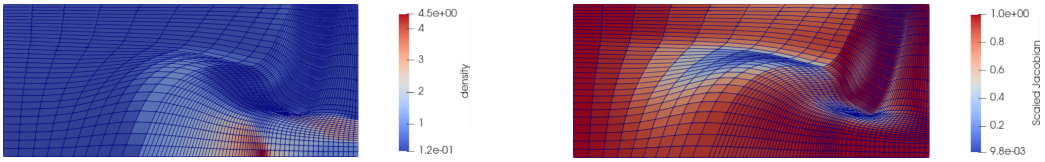
(c) ALE with 0.5 weights **(d) ALE with adapted weights**

Figure 20: Comparison between the different methods for triple point simulation at time $t = 5.0$ s with $70 \times 30 \times 2$ cells.

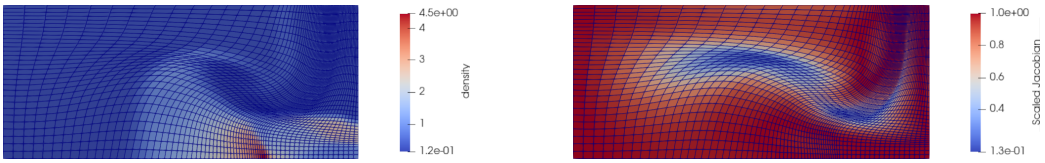
one, leads to an excessive numerical diffusion due to the projection step. To obtain a more progressive mesh, this regularization process can be improved combining the Line-Sweeping method with the weighted one.

This is demonstrated in Figure 21, where the parameter θ is set to $\theta = 10^{-3}$. The resulting mesh better captures the vortex at the triple point. Figures 21.b and 21.d show the scaled Jacobian for computations with $\theta = 0$ and $\theta = 10^{-3}$, respectively. The scaled Jacobian is a quality metric used to assess mesh integrity. Notably, it becomes negative in cases of tangled meshes. Here, we observe that the values remain positive, indicating that the mesh maintains a valid configuration.

The weighted line sweeping regularization method generates a mesh that is very similar to the results obtained with a high-order method [18, 17], as illustrated by the high-order ALE and Lagrangian computation results in

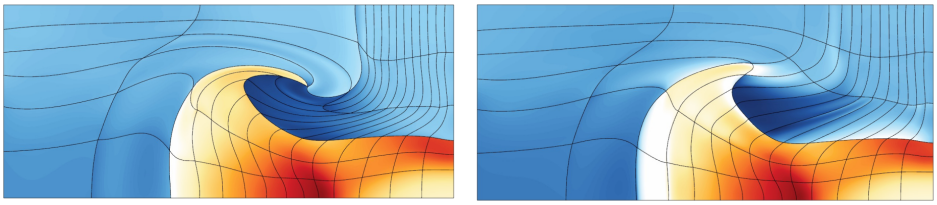


(a) Weights and $\theta = 0$ **(b) Mesh quality**



(c) Weights and $\theta = 10^{-3}$ **(d) Mesh quality**

Figure 21: Comparison between $\theta = 0$ in (a-b) and $\theta = 10^{-3}$ in (c-d) for triple point simulation at time $t = 5.0$ s with $70 \times 30 \times 2$ cells.



(a) High order Lagrangian **(b) High order ALE**

Figure 22: High order solution from [17].

Figure 22. This computation has been run with a very coarse mesh of 14×6 cells. The same curvature and significant differences in aspect ratios within the vortex are observed, highlighting the accuracy of the method. To further enhance mesh quality, it would be beneficial to use AMR (Adaptive Mesh Refinement) to refine areas where the aspect ratio of the cells becomes too large.

6. Conclusion

When solving the Euler equations, the use of the Lagrangian formalism can lead to significant mesh deformations. Therefore, an efficient regularization method is necessary for the indirect ALE strategy to prevent cell tangling and a decrease in mesh quality. The Line-Sweeping method [2, 3] was initially considered due to its simplicity and geometric basis. However, despite its efficiency, it has been observed that the final regularized mesh tends to be uniform and may deviate significantly from the initial Lagrangian mesh, even when this is unnecessary.

To avoid uniform rezoning of the mesh, weights were introduced into the Line-Sweeping method. Specifically, the weighted method allows the regularization of an initial mesh with a given set of aspect ratios into a new mesh with another set of known aspect ratios. The key idea here is to use the aspect ratios of the initial Lagrangian mesh to define the weights for the regularization step. This new weighted method is promising, as it produces results close to the Lagrangian ones but with improved mesh quality. Additionally, it can successfully handle challenging cases, such as the triple point test case, which is problematic for the Lagrangian formalism.

Looking forward, various perspectives can be considered. For example, the regularization process can be further improved by refining the aspect ratios used in the weight calculation. Exploring optimization-based methods for computing these weights could potentially provide even better control over mesh quality. Additionally, we plan to incorporate Adaptive Mesh Refinement (AMR) to further refine regions where the aspect ratios of the mesh cells become too large.

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.
The authors also gratefully acknowledge Dr. Vladimir Tomov (LLNL) for providing high-order results for the triple point obtained with the BLAST code, a high-order finite element hydrodynamics research code developed at LLNL.

References

[1] J. Yao, D. Stillman, An equal-space algorithm for block-mesh improvement, Procedia Engineering 163 (2016) 199–211, 25th International Meshing Roundtable.

[2] J. Yao, A mesh relaxation study and other topics, LLNL-TR-637101 (2013).

[3] S. Guisset, G. Damour, J. Breil, Cell-centered indirect arbitrary lagrangian-eulerian numerical strategy for solving 3d gas dynamics equations, Journal of Computational Physics 505 (2024) 112903.

[4] G. Gabriel, J. Breil, P.-H. Maire, A 3D GCL compatible cell-centered Lagrangian scheme for solving gas dynamics equations, J. Comput. Phys. 305 (2016) 921–941.

[5] B. Després, C. Mazeran, Lagrangian gas dynamics in dimension two and Lagrangian systems, Arch. Ration. Mech. Anal. 178 (2005) 327–372.

[6] P.-H. Maire, R. Abgrall, J. Breil, J. Ovadia, A cell-centered Lagrangian scheme for two-dimensional compressible flows problems, SIAM J. Sci. Comput. 29 (2007) 1781–1824.

[7] T. Barth, D. Jespersen, The design and application of upwind schemes on unstructured meshes, AIAA Paper, 27th Aerospace Sciences Meeting, Reno, Nevada 89-0366 (1989).

[8] J. Velechovsky, J. Breil, R. Liska, Flux corrected remapping using piecewise parabolic reconstruction for 2d cell-centered ale methods, International Journal for Numerical Methods in Fluids 76 (9) (2014) 575–586.

[9] R. Liska, M. Shashkov, P. Váchal, B. Wendroff., Optimization-based synchronized flux-corrected conservative interpolation (remapping) of mass and momentum for arbitrary lagrangian-eulerian methods., J. Comput. Phys. 229(5) (2010) 1467–1497.

[10] R. Liska and M. Shashkov and P. Váchal and B. Wendroff., Synchronized flux corrected remapping for ALE methods., Comput. and Fluids 46(1) (2011) 312–317.

[11] S. Galera, P.-H. Maire, J. Breil, A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction, J. Comput. Phys. 229 (2010) 5755–5787.

[12] G. Damour, S. Guisset, J. Breil, 3d mesh regularization based on a weighted line sweeping method, in: Proceedings of the 2024 International Meshing Roundtable (IMR), 2024, pp. 132–143.

[13] G. Sod, A survey of finite difference methods for system of nonlinear conservation laws, J. Comput. Phys. 27 (1978) 1–31.

[14] J. Kamm, F. Timmes, On efficient generation of numerically robust Sedov solutions, Technical report, Los Alamos National Laboratory LA-UR-07-2849 (2007).

[15] J. Breil, Numerical methods for Lagrangian and Arbitrary-Lagrangian-Eulerian Hydrodynamic Contribution to the simulation of High-Energy-Density-Physics Problems, Ph.D. thesis (2016).

[16] W. Noh, Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux, J. Comput. Phys. 72 (1987) 78–120.

[17] V. A. Dobrev, P. Knupp, T. Kolev, K. Mittal, R. N. Rieben, M. Stees, V. Z. Tomov, Asymptotic analysis of compound volume+shape metrics for mesh optimization, in: Proceedings of the 2024 International Meshing Roundtable (IMR).

[18] R. W. Anderson, V. A. Dobrev, T. V. Kolev, R. N. Rieben, V. Z. Tomov, High-order multi-material ale hydrodynamics, SIAM Journal on Scientific Computing 40 (1) (2018) B32–B58.