CSMA 2026
17ème Colloque National en Calcul des Structures
18-22 Mai 2026, Presqu'île de Giens (Var)

Mesh Conformity in the X-Mesh Approach in 3D

Kerem Okyay[1], Antoine Quiriny[1], Jonathan Lambrechts[1], Nicolas Moës[1], Jean-François Remacle[1]

[1] *iMMC, Université catholique de Louvain (UCLouvain),*
{kerem.okyay, antoine.quiriny, jonathan.lambrechts, nicolas.moes, jean-francois.remacle}@uclouvain.be

Abstract — We present the first extension of the X-Mesh method to three-dimensional tetrahedral meshes. A bipartite minimum-vertex-cover formulation identifies the smallest set of mesh vertices that must be snapped to the interface. A foldover-free optimization step then relaxes the mesh and removes inverted elements while constraining snapped vertices to remain conformal to the interface. The resulting mesh conforms to the prescribed geometry but contains highly flattened tetrahedra. To compute on such meshes, we rely on the recently developed Tempered Finite Element Method (TFEM).
Keywords — sharp interface tracking; mesh deformation; minimum vertex cover; mesh untangling; tempered finite elements.

1 Introduction

Numerical simulations that involve a moving or geometrically complex interface such as multiphase flow simulations require two key ingredients: (i) a suitable representation of the interface and (ii) a computational mesh on which the governing equations are solved. Many interface-handling techniques fall into two broad categories. Capturing methods, such as level-set and Volume-of-Fluid (VOF) approaches, represent the interface implicitly through a scalar field, which could introduce a diffuse transition zone and therefore requires a careful treatment around the interface region. Tracking methods, including Arbitrary Lagrangian-Eulerian (ALE) and marker-based formulations, instead advect explicit geometric entities and therefore can in an easier way maintain a sharp interface.

When a geometrically sharp interface is required, tracking methods are an attractive choice, but they imply an additional challenge: the computational mesh must conform to the interface at all times. Maintaining this conformity becomes particularly difficult in 3D, where moving the mesh vertices to follow the interface can easily produce severely distorted or even inverted tetrahedra. Robust procedures are therefore needed to modify the mesh locally so that it remains valid while aligning exactly with the prescribed surface.

The extreme mesh deformation method, X-Mesh [1, 2], belongs to this second family: it performs interface tracking by locally deforming a fixed-connectivity background mesh so that its elements conform exactly to the evolving interface. In 2D, Quiriny et al. [1] have demonstrated that X-Mesh can have an accurate representation of moving interfaces throughout the deformation process in two-phase flows, as shown in Figure 1. Extending this strategy to three-dimensional tetrahedral meshes, however, raises significant challenges: pushing vertices toward a moving surface may create severely distorted or even inverted elements, which standard finite element formulations cannot accommodate.

This work presents the first steps toward a full 3D extension of the X-Mesh approach. We concentrate on the geometric aspect of the method—namely, producing a tetrahedral mesh that conforms sharply to a given 3D interface while preserving validity. Our contribution combines: a minimum vertex cover formulation that selects the minimal set of vertices to snap to the interface, ensuring conformity with minimal distortion, and a foldover-free optimization procedure heavily inspired by Garanzha et al. [3], enhanced with an interface-anchoring term to relax elements and eliminate negative volumes while keeping snapped vertices conformal to the interface. The resulting mesh is sharp and valid, but may still contain highly flattened tetrahedra near the interface. To compute on such meshes, we rely on the Tempered Finite Element Method (TFEM) [4], recently introduced to stabilize FEM on degenerate elements. We illustrate the pipeline by simulating a flow past a sphere on a mesh conformed using our method.

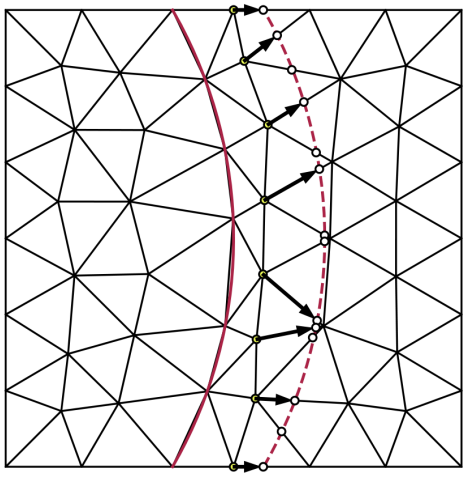


Figure 1: Deformation algorithm in the X-Mesh method in 2D. Continuous red representing the old interface and it is conformal, the dashed red represents the new interface at the following time step. The arrows show the nodal movements needed to be conformal to the new interface.

2 Mesh conformity to an interface in 3D

In 2D, the X-Mesh approach tracks an interface sharply by locally deforming a background triangulation so that the interface aligns precisely with element edges while preserving a valid mesh throughout the deformation. Extending these local deformations from 2D triangles to 3D tetrahedra is far from straightforward. A major difficulty is the creation of nearly zero-volume or even inverted tetrahedra when vertices are pushed toward a moving interface. For example, if a tetrahedron has four intersected edges and each endpoint is moved to its respective intersection point, the element collapses to zero volume. In standard FEM this is unacceptable, as such elements lead to ill-conditioned or undefined stiffness matrices. In our framework, however, these degeneracies can later be handled robustly by TFEM.

Computational domain and the interface representation

The interface is represented explicitly by a closed surface triangulation in 3D, denoted by Γ. The computational domain is discretized independently as a tetrahedral mesh $\mathcal{T}$. These two meshes are *a priori* unrelated: the vertices of $\mathcal{T}$ do not necessarily lie on Γ, and the interface may cut arbitrarily through the tetrahedra.

The objective of the deformation stage is to modify the vertex positions of $\mathcal{T}$ so that the domain mesh conforms exactly to Γ while maintaining an acceptable mesh quality and preventing inverted elements when possible. The deformation algorithm consists of two components, detailed in the subsections below.

2.1 Graph theoretic approach to conformity: Minimum vertex cover

The interface Γ partitions the vertices of the tetrahedral mesh $\mathcal{T}$ into inside and outside regions, a 2D example is shown in Figure 2 to clarify the concept. Any mesh edge with one endpoint inside and the other outside necessarily crosses Γ, and such edges are precisely the obstacles to conformity. Collecting all outside vertices in U, all inside vertices in V, and connecting $u \in U$ to $v \in V$ whenever edge (u,v) crosses Γ yields a bipartite graph $G = (U,V,E)$ encoding all interface violations.

To obtain an interface-conforming mesh, at least one endpoint of every crossing edge must be snapped. This is exactly a vertex-cover problem. While selecting all endpoints would trivially resolve all crossings, it would introduce excessive motion; instead, we seek the minimum vertex cover. Although it is NP-hard in general, Kőnig's theorem [5] ensures that on bipartite graphs it can be computed exactly and efficiently from a maximum matching. Snapping only the minimum cover vertices guarantees (i) removal of all crossings and (ii) minimal node motion.

To cast the problem in this form, having an interface that is regular with respect to the mesh size, we further rely on two simplifying assumptions. **(1) Each intersecting edge has one inside and one out-**

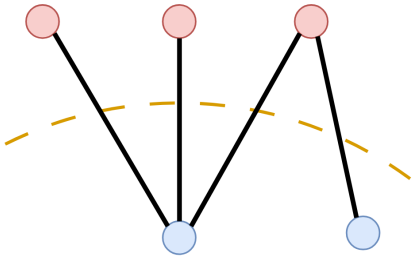


Figure 2: The interface in dashed yellow and the crossing edges in black. The outside vertices are colored red where the inside vertices are colored blue. The interface partitions the domain vertices in two groups "inside" and "outside". This forms the starting bipartite graph for the algorithm.

side endpoint. Same-side intersections (both endpoints inside or both outside) may occur on very coarse meshes, but are rare on reasonably resolved domains and are corrected automatically during the subsequent relaxation step. **(2) Snapping minimum cover vertices does not create new intersections**. All vertex snapping motions are restricted to existing mesh edges and are always smaller than an edge length. In the event that it generates new intersections, these would again be resolved during the relaxation stage.

Under these assumptions, the snapping stage reduces to computing the minimum vertex cover of the bipartite inside–outside graph. In practice, we first compute a maximum matching using the Hopcroft–Karp algorithm [6], which runs in $O(n^{2.5})$ for a graph with n vertices. By Kőnig's theorem, the size of a maximum matching equals the size of the minimum cover, and the cover itself is extracted through a standard alternating-path construction. This yields an exact minimum cover, guaranteeing that we snap the smallest possible set of vertices required to eliminate all interface crossings.

2.2 Foldover-free maps and mesh untangling

The snapping step resolves crossings but does not guarantee a visually or numerically high-quality mesh. As mentioned earlier, tetrahedra with four intersected edges may collapse to nearly zero volume once their vertices are moved to the intersection points, and some elements may even flip. To correct these issues, we use the foldover-free optimization method of Garanzha et al. [3], which moves vertices to eliminate inverted elements and relax deformed ones and it minimizes the following functional:

$$\lim_{\varepsilon \to 0^+} \arg\min_{U} F_{\mathcal{T}}(U, \varepsilon)$$

Where U denotes the vertex positions of $\mathcal{T}$ and

$$F_{\mathcal{T}}(U, \varepsilon) := \sum_{t \in \mathcal{T}} \left((1-\theta) f_\varepsilon(J_t) + \theta g_\varepsilon(J_t) \right) \operatorname{vol}(T_t)$$

The regularization function is

$$\chi(D, \varepsilon) := \frac{D + \sqrt{\varepsilon^2 + D^2}}{2},$$

and the two energy contributions are

$$f_\varepsilon(J) := \frac{\operatorname{tr} J^T J}{\left(\chi(\det J, \varepsilon)\right)^{2/d}}, \qquad g_\varepsilon(J) := \frac{\det^2 J + 1}{\chi(\det J, \varepsilon)}.$$

Here f_ε penalizes the element shapes and g_ε penalizes the areas in the optimization problem with respect to a reference tetrahedron. The parameter θ controls the relative weight of these effects.

Since some vertices have been snapped to the interface, we must ensure that they remain on Γ during relaxation. To enforce this, we introduce an additional interface-anchoring term,

$$F_\Gamma(U) = \sum_{\vec{u} \in \Gamma} \operatorname{dist}(\vec{u}, \Gamma)^2$$

and minimize the augmented functional

$$F(U,\varepsilon) = F_{\mathcal{T}}(U,\varepsilon) + \lambda F_{\Gamma}(U)$$

where $\lambda > 0$ controls the strength of the interface constraint. This ensures that snapped vertices remain on the interface while the rest of the mesh is relaxed. The resulting optimization problem is solved using the HLBFGS method, following the refinement strategy of Garanzha et al. [3] without modification.

Combination of components

The sequential application of these steps yields the results shown in Figure 5. First, the minimum vertex cover algorithm is employed to resolve the edge crossings identified in Figure 5b. While this effectively removes topological conflicts, the resulting vertex displacements introduce inverted elements (negative volumes). To correct this, we apply the mesh untangling procedure to restore the elements to positive volume and improve their quality. The quality metric we look at is the γ metric and this will be discussed more in the following section with Figure 6. Finally, by labeling the tetrahedra as strictly inside or outside the domain (Figure 5d), we can extract the specific facets that conform to the target interface, as shown in Figure 5c. This yields a watertight and 2-manifold representation of the interface, guaranteeing a distinct separation of physical domains without topological ambiguities.

3 Tempered finite elements

The mesh obtained after snapping and untangling conforms exactly to the interface, and all inverted tetrahedra have been removed. However, many elements located close to Γ may still have extremely small volume, therefore be in a degenerated configuration. Such elements are unavoidable when we snap nodes to the interface.

Even though these elements are geometrically valid, they cause severe numerical issues in standard finite element formulations: the determinant of the Jacobian of the reference-to-physical mapping becomes very small, the stiffness matrix becomes ill-conditioned, and the solution may "lock" in degenerate zones.

TFEM [4] was developed specifically to address this difficulty . Instead of modifying the mesh connectivity or enriching the finite element space, TFEM modifies the local contribution of degenerate elements so that their Jacobians cannot collapse below a prescribed threshold. This makes it possible to compute reliably on meshes that are sharp but contain very small or highly flattened tetrahedra.

The idea of TFEM can be understood by considering the simpler 2D case mapping from physical domain to reference domain tetrahedra. Take a physical domain triangle in 2D that becomes flattened as the interface approaches one of its edges.

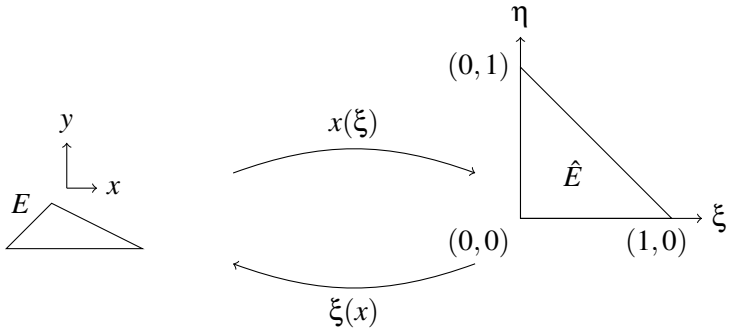


Figure 3: The mapping of physical to reference domain

In the standard FEM mapping

$$\nabla_x \phi = \nabla_\xi \phi \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}, \quad J = \det\left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right)$$

where J is proportional to the element area (volume in 3D).

$$\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}} = \left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right)^{-1} = \frac{1}{J}\begin{pmatrix} \dfrac{\partial y}{\partial \eta} & -\dfrac{\partial x}{\partial \eta} \\ -\dfrac{\partial y}{\partial \xi} & \dfrac{\partial x}{\partial \xi} \end{pmatrix}$$

$$J = \det\left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right) = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} = 2|A|$$

As the physical element collapses $|A| \to 0$, we have $J \to 0$ and therefore the J^{-1} blows up. The gradients of the basis functions become arbitrarily large and the element stiffness matrix becomes singular. The core idea of TFEM is to replace J that goes to 0 with $J_{\min}$.

$$J = \max\left(\det\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}},\, J_{\min}\right)$$

For a regular element ($J > J_{\min}$), the modified matrix coincides with the standard FEM matrix. For a degenerate element ($J < J_{\min}$), the local matrix is multiplied by the factor $\frac{J}{J_{\min}} < 1$, which relaxes the locking conditions. A suitable choice of $J_{\min}$ effectively eliminates the locking caused by clusters of degenerate elements while preserving good solution accuracy.

However, choosing $J_{\min}$ too large induces an excessive modification of the equations and thus a consistency error. As demonstrated in [4], the second-order convergence of the FEM solution is preserved with the TFEM formulation when one chooses:

$$J_{\min} = \frac{|u|_{W^{2,\infty}(E)}}{|u|_{W^{1,\infty}(E)}}\, h^3,$$

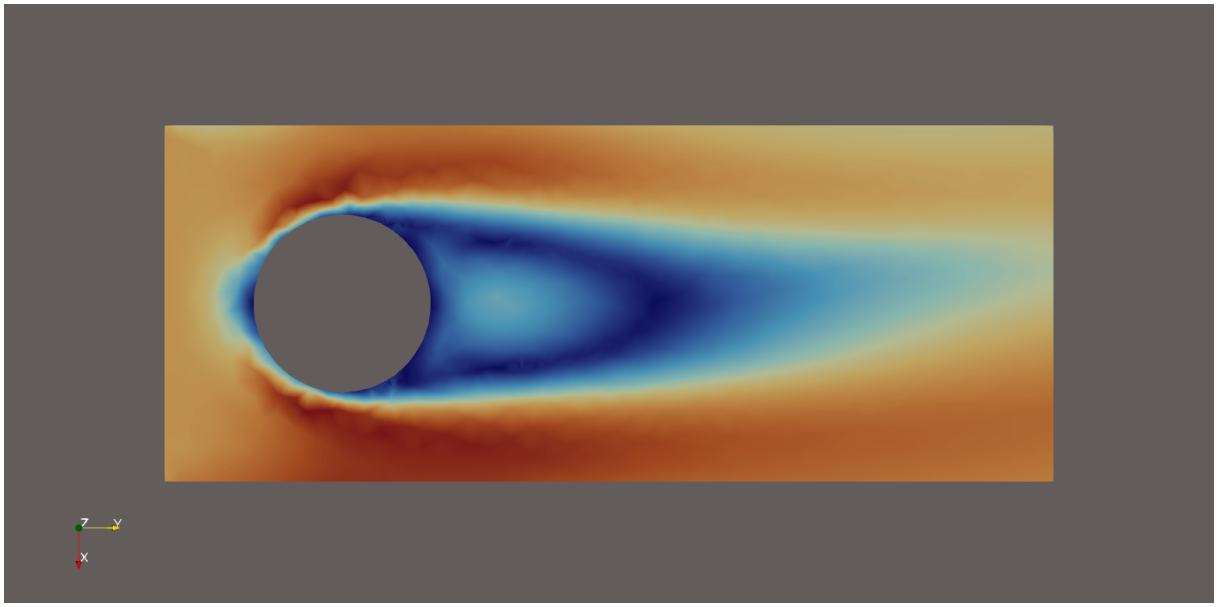
where h is the local size of the element, and $|u|_{W^{k,\infty}(E)}$ denotes the maximum norm of the k-th derivative of the exact solution u over the element E. In practice, since the exact solution is not known, these derivatives must be estimated. The convergence is not very sensitive to the exact value of $J_{\min}$ [4], which makes it easy to approximate these values.

TFEM in our pipeline

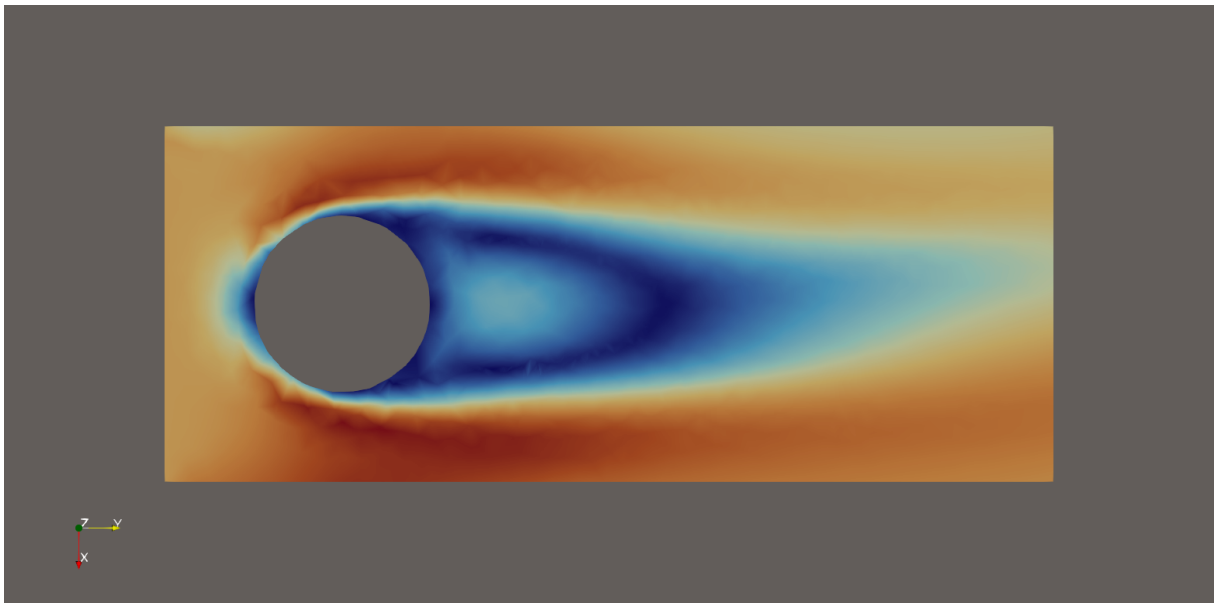
We utilize the TFEM method implemented within the Migflow framework [7]. The conforming algorithm generates a mesh that captures the interface sharpness; however, this process inevitably produces extremely small, sliver-shaped tetrahedra near the interface Γ. Standard finite element formulations often fail on such elements, but TFEM stabilizes the discretization, allowing us to solve the incompressible Navier–Stokes equations directly without requiring remeshing or local refinement.

To demonstrate this capability, we compare a simulation run on our conformed mesh against a reference solution. The simulation setup extends the domain parameters used in Figure 5 along the y axis. The expected flow field is characterized by a wake downstream of the sphere, exhibiting a minor asymmetry due to a slight tilt in the inlet velocity. Regarding boundary conditions, we apply a no-slip condition on the spherical interface and symmetry conditions on the domain walls.

Figure 4 presents a comparison between the two meshing approaches. To quantify mesh quality, we employ the metric γ (ratio of inscribed to circumscribed radius), where values approaching zero indicate degenerate tetrahedra. Figure 4a displays the reference solution computed on a standard mesh generated via `gmsh` boolean operations [8]. This benchmark mesh is well-controlled, maintaining a minimum quality of $\gamma \approx 0.34$. In contrast, Figure 4b shows the solution computed on our conformed mesh. As detailed in the quality analysis of Figure 6, this mesh contains severely degenerate elements ($\gamma \approx 0$) near the interface. Despite this geometric degeneracy, the TFEM formulation remains stable, and the resulting velocity field shows excellent qualitative agreement with the reference solution.



(a) Reference solution: Flow section computed on a high-quality mesh generated via `gmsh` boolean operations ($\gamma_{min} = 0.34$).



(b) Proposed method: Flow section computed on the conformed mesh. Despite containing degenerate elements near the sphere ($\gamma_{min} \approx 0$), the field remains smooth.

Figure 4: Comparison of velocity magnitude fields at $t = 2.15\,\mathrm{s}$. (a) The reference simulation using a standard high-quality mesh. (b) The simulation using the proposed conforming method. The close agreement between the two results demonstrates that TFEM effectively handles the local mesh degeneracy introduced by the conforming process.

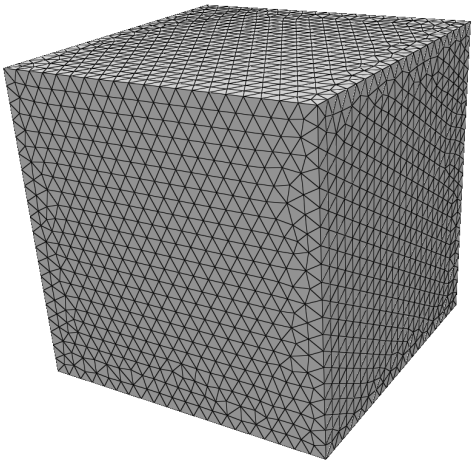
This comparison primarily serves as a proof of concept, confirming that TFEM maintains numerical stability even in the presence of severe mesh degeneracy ($\gamma \approx 0$). While a comprehensive physical validation—such as extending the domain size to analyze vortex shedding frequencies—is beyond the current scope, these results definitively establish the computational feasibility of the proposed pipeline on conformed meshes.

4 Conclusion

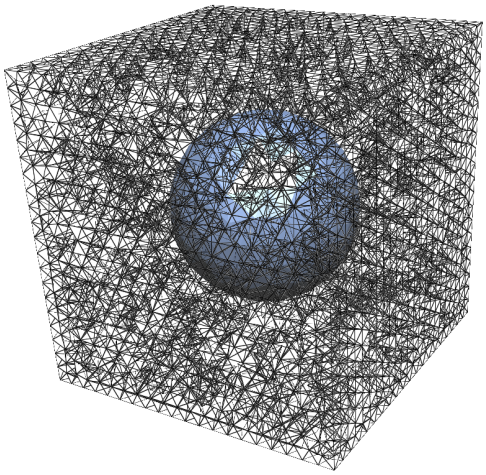
This work presents the first steps toward extending the X-Mesh methodology to three-dimensional tetrahedral meshes. Our focus has been on achieving a sharp, interface-conforming domain mesh starting from an independent background discretization. The proposed pipeline combines two complementary ingredients: a graph-theoretic snapping strategy based on a minimum vertex cover, which guarantees the minimal set of vertex movements required to eliminate all interface crossings, and a foldover-free relaxation procedure that removes inverted elements and improves mesh quality while preserving the interface geometry.

The resulting mesh conforms exactly to the prescribed surface Γ, yet inevitably contains highly flattened tetrahedra in the vicinity of the interface. To compute robustly on such meshes, we rely on the TFEM, which stabilizes the discretization by tempering the Jacobian contribution of degenerate elements. Using the TFEM implementation in Migflow, we demonstrate that physically relevant flows—such as flow past a conforming sphere—can be computed stably on meshes produced by our conformity algorithm.

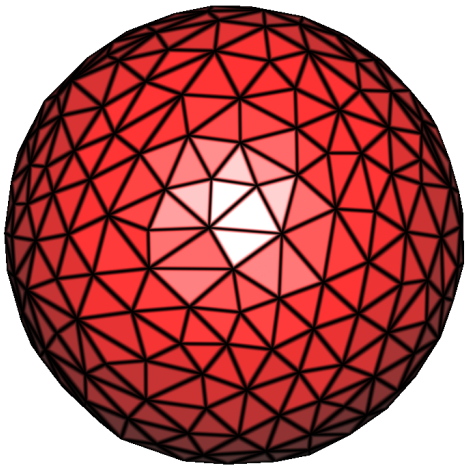
These results indicate that sharp, interface-aligned simulations in 3D can be carried out without remeshing or topology changes, even in the presence of strong geometric constraints. Future work will extend this framework to fully two-phase flows, including rising-bubble and droplet-dynamics problems, where explicit interfaces and conforming meshes naturally facilitate the imposition of jump conditions and surface tension forces.



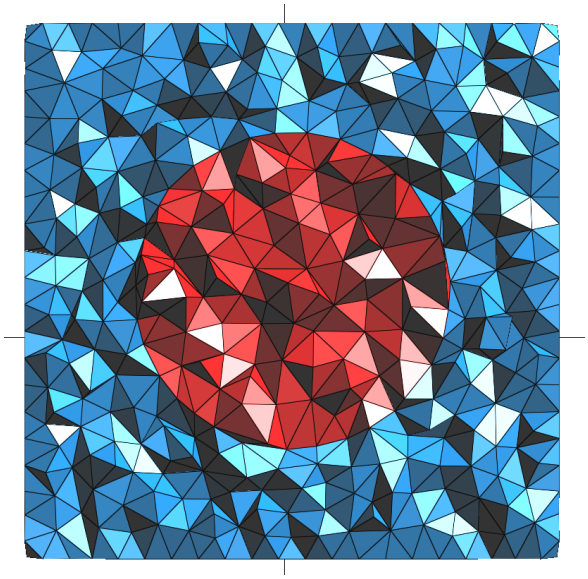
(a) Initial tetrahedral domain



(b) Domain edges intersected by the target interface (blue sphere)

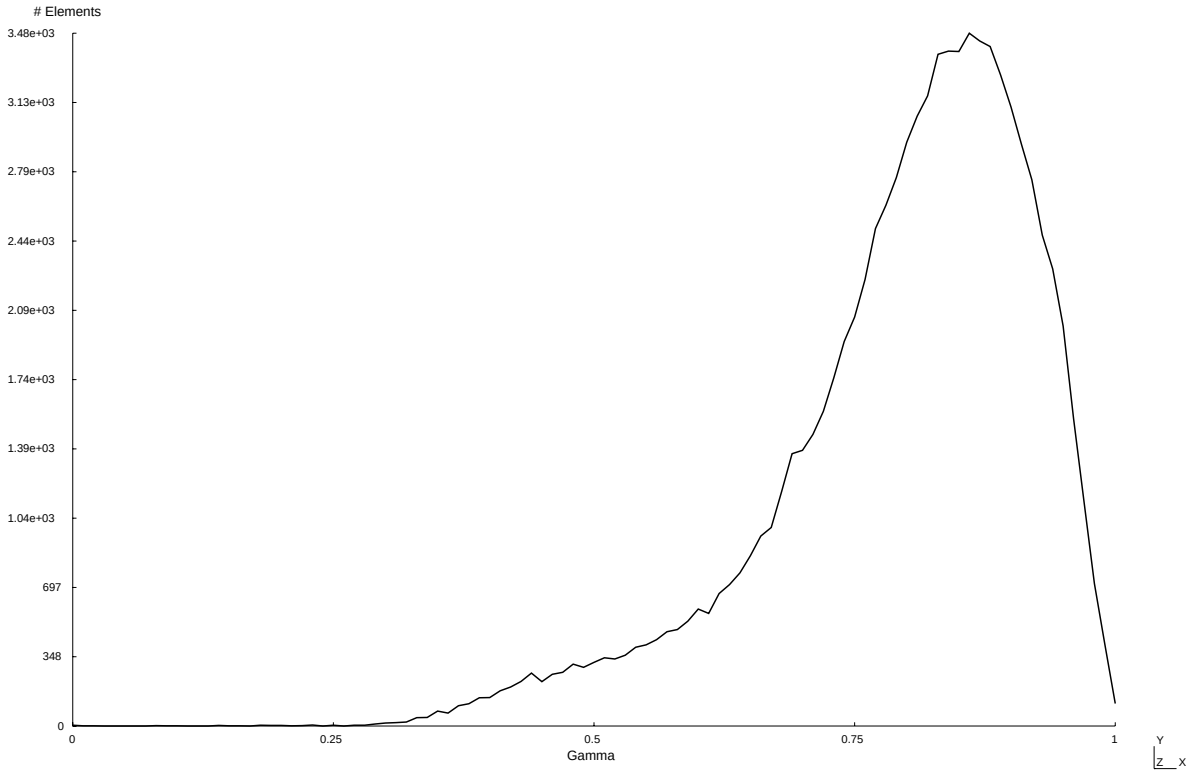


(c) Facets of the conformed mesh covering the interface

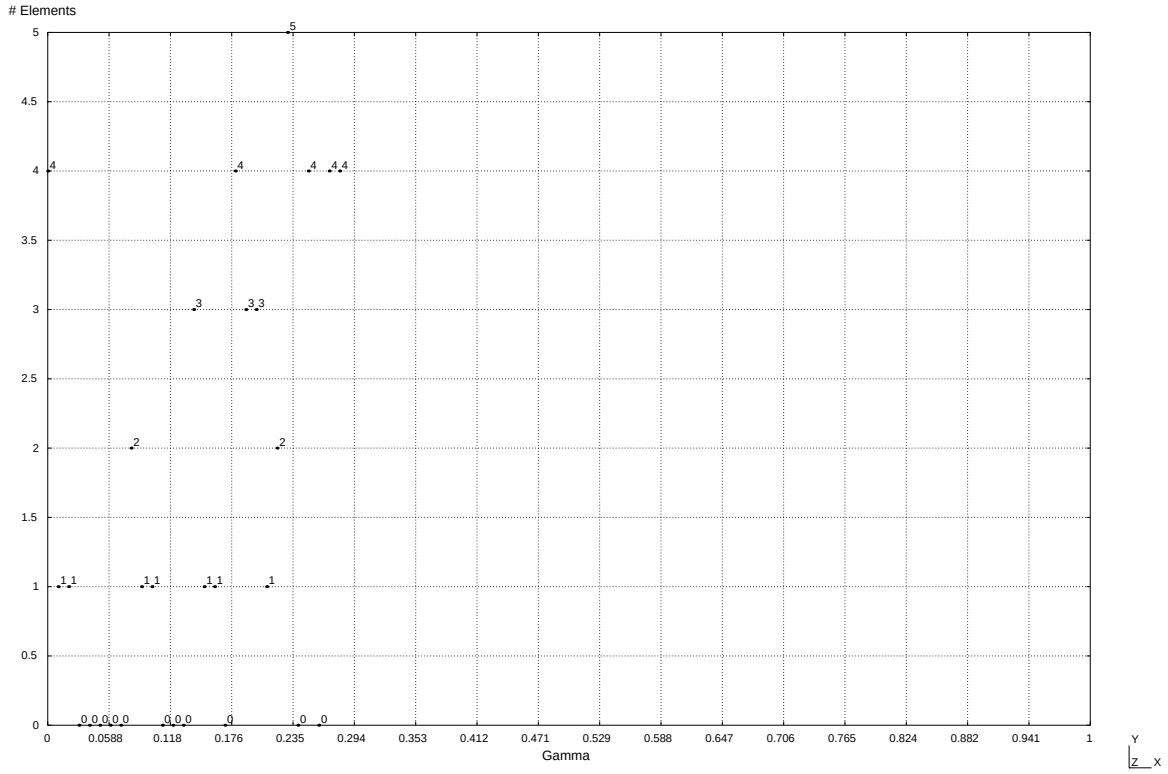


(d) Cross-section showing interior (red) and exterior (blue) elements

Figure 5: Process of conforming a tetrahedral mesh to a spherical interface. (a) The initial 3D domain. (b) The domain edges overlaid with the target interface (blue). Note that the interface is initially independent of the domain, causing arbitrary edge intersections. (c) The watertight, 2-manifold selection of facets representing the interface after conforming. (d) A clipped view demonstrating the sharp boundary between tetrahedra labeled as inside (red) and outside (blue).



(a) Global distribution of the mesh quality metric γ across the domain.



(b) Magnified view of the low-quality tail. Annotations indicate the exact count of elements at specific quality levels.

Figure 6: Assessment of mesh quality using the metric γ (ratio of inscribed to circumscribed radius). Values approaching zero indicate degenerate (sliver) elements. While the global distribution (a) shows a healthy mesh, the zoomed view (b) reveals the existence of four elements with extreme degeneracy ($\gamma \approx 0$) resulting from the interface conforming process.

References

[1] Antoine Quiriny, Jonathan Lambrechts, Nicolas Moës, and Jean-François Remacle. X-mesh: A new approach for the simulation of two-phase flow with sharp interface. *Journal of Computational Physics*, 501:112775, 2024.

[2] Nicolas Moes, Jean-François Remacle, Jonathan Lambrechts, Benoît Lé, and Nicolas Chevaugeon. The eXtreme mesh deformation approach (X-MESH) for the Stefan phase change model. *Journal of Computational Physics*, 477:111878, 2023.

[3] Vladimir Garanzha, Igor Kaporin, Liudmila Kudryavtseva, François Protais, Nicolas Ray, and Dmitry Sokolov. Foldover-free maps in 50 lines of code. *ACM Transactions on Graphics (TOG)*, 40(4):1–16, 2021.

[4] Antoine Quiriny, Václav Kučera, Jonathan Lambrechts, Nicolas Moës, and Jean-François Remacle. The tempered finite element method. *arXiv preprint arXiv:2411.17564*, 2024.

[5] Gábor Szárnyas. Graphs and matrices: A translation of "Graphok és Matrixok" by Dénes Kőnig (1931). *arXiv preprint arXiv:2009.03780*, 2020.

[6] John E Hopcroft and Richard M Karp. An nˆ5/2 algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231, 1973.

[7] Matthieu Constant, Frédéric Dubois, Jonathan Lambrechts, and Vincent Legat. Implementation of an unresolved stabilised fem–dem model to solve immersed granular flows. *Computational Particle Mechanics*, 6(2):213–226, 2019.

[8] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009.