Received: 27 September 2022 | Revised: 5 December 2022 | Accepted: 28 December 2022

DOI: 10.1002/nme.7200

RESEARCH ARTICLE

WILEY

The surface-reconstruction virtual-region mesh update method for problems with topology changes

Felipe A. González[1] | Stefanie Elgeti[1,2] | Marek Behr[1]

[1]Chair for Computational Analysis of Technical Systems (CATS), RWTH Aachen University, Aachen, Germany

[2]Institute of Lightweight Design and Structural Biomechanics, Vienna University of Technology, Vienna, Austria

Correspondence
Felipe A. González, Chair for Computational Analysis of Technical Systems (CATS), RWTH Aachen University, Schinkelstraße 2, 52062 Aachen, Germany.
Email: gonzalez@cats.rwth-aachen.de

Funding information
Bundesministerium für Bildung und Forschung; Comisión Nacional de Investigación Científica y Tecnológica, Grant/Award Number: 62180006; Deutscher Akademischer Austauschdienst; Forschungszentrum Jülich; National Agency for Research and Development (ANID)

Abstract

In this work, we introduce a novel boundary-conforming mesh-update method that is particularly designed for problems with large boundary displacements and topology changes. This method, which we call the surface-reconstruction virtual-region mesh update method, integrates the virtual region approach and a surface reconstruction process to handle complex boundary movements. The virtual region approach allows having an activated and deactivated part of the mesh, where elements can freely enter or leave the activated domain. Furthermore, the surface reconstruction ensures boundary conformity of the activated domain. The robustness of the proposed method is shown in two numerical examples: a variation of the benchmark Poiseuille flow, and the flow simulation during a closing diaphragm valve. In particular, the diaphragm valve simulation includes large boundary movement, complex geometry, and closing motion. For this case, both steady and transient simulation results at different closing conditions are presented.

KEYWORDS

constrained Delaunay triangulation, diaphragm valve, mesh update method, stabilized finite element method, topology change

1 | INTRODUCTION

Many problems in engineering involve moving boundaries or interfaces and topology changes; including for instance free-surface, fluid-structure interaction (FSI), and problems with imposed motion.

Two major classes of methods have been proposed in the literature: non-body-fitted approaches and body-fitted or boundary-conforming approaches.[1] The first class typically applies an immersed or embedded interface description on a non-body-fitted mesh; it is also known as interface-capturing approach. In this class, one finds the level-set approach,[2] the volume-of-fluid (VOF) method,[3] the marker and cell (MAC) method,[4] the phase-field method,[5] and the immersed boundary method.[6] The second class of methods uses an explicit boundary or interface description, for example, a tracking approach. In most cases, a boundary-conforming approach is combined with a mesh update method to address the mesh movement. In particular, this article focuses on this kind of technique, so we will pay special attention to the related literature.

In order to treat such problems in a boundary-conforming fashion, a variety of approaches are available such as the arbitrary Lagrangian–Eulerian (ALE) formulation[7-9] and the space-time (ST) formulation.[10,11] The key idea of ALE is to

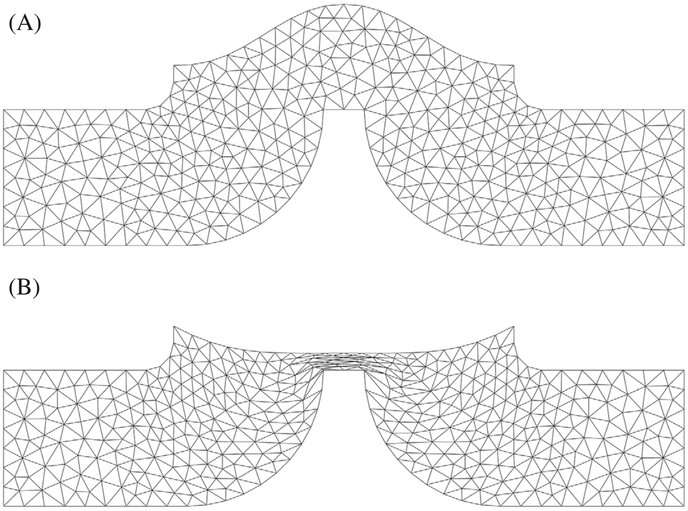

FIGURE 1 Mesh deformation of a diaphragm valve using a linear-elasticity-based method.

consider mesh velocity and material velocity as independent quantities, such allowing moving boundaries without the need for the mesh movement to follow the material. In a similar notion, ST employs finite element interpolation for both space and time, in contrast to only in space, which would be the more common approach. This brings the advantage that the variational form is written directly over the deformed mesh, thus automatically including the mesh deformation into the space-time domain.

It is important to note the difference between the domain topology and the mesh topology. The domain topology is directly related to the shape of the geometry. In contrast, the mesh topology refers to the underlying graph of the mesh. A change in mesh topology can for example be a result of a modification in the mesh connectivity. In general, any method that changes the mesh topology requires projection of the solution field between meshes, also called mesh-to-mesh interpolation, inevitably introducing projection errors.

Such projection errors can be avoided, if a mesh-update method based solely on modifying the mesh vertex position is used. One type of approach is based on interpolation functions, for instance, the radial basis functions (RBF)[12] or a variation of it.[13] The new position of the mesh nodes is obtained by interpolating the displacements of the boundary nodes to the whole mesh. It should be noted that a small system of equations, only involving the boundary nodes, has to be solved and no mesh-connectivity information is needed.

Another way for obtaining the updated mesh is by solving a partial differential equation (PDE). Different PDE-based approaches can be found in this category, as a review, see References 14-20. A commonly used PDE-based method relies on solving the elasticity equation, first introduced for the linear case,[21] and later extended to the nonlinear case.[22] Vertex-displacement-based methods generally support a wide range of boundary or interface motions, and they are computationally efficient since the mesh topology is preserved. However, one observes significant differences in their performance, mainly regarding computational cost and accumulated distortion. An interesting comparison of some of the aforementioned methods can be found in Reference 20.

Although vertex-displacement-based methods are widely used, they can still fail even for simple motions like large rigid body translation and rotation. Furthermore, large boundary displacements can easily lead to high mesh distortion and invalid meshes. Some elements can be undesirably stretched or compressed, which adversely affects the accuracy and stability of the numerical formulation. An example of the mesh deformation using a linear-elasticity-based method is shown in Figure 1.

Therefore, when it comes to large deformations in a boundary-conforming method, one needs more than simply vertex-displacement-based approaches. The most general solution is to perform a global or local remeshing. Examples can be found in References 23-25. The mesh is deformed as much as possible, and when the mesh quality deteriorates, the domain is globally or locally remeshed. Every time a new mesh is generated, a mesh-to-mesh interpolation is required. In general, projection between meshes tends to be computationally expensive for three-dimensional meshes. However, the main downside is that projections introduce inaccuracies to the numerical solution.

A way to overcome this issue is by applying a specific mesh update method suitable to a particular application or mesh movement. For instance, the shear-slip method (SSMUM)[26] has been developed for large mesh translation and rotation. The SSMUM method has been applied to problems with large, but regular mesh displacements, mainly rotation, for example, blood pumps.[27,28] In particular, this approach splits the mesh into static and moving subdomains. The subdomains are connected by a thin layer of structured elements, a so-called shear-slip layer. A modification of the mesh connectivity is applied only within the shear-slip layer and only if the element distortion exceeds a certain threshold. Later, this approach was extended to problems with large mesh translation by the virtual-ring shear-slip method (VRSSMUM).[29] The main difference to SSMUM is that the moving subdomain is equipped with two new concepts, the virtual ring and the virtual region. The virtual ring approach is responsible for connecting the "inlet" and "outlet" of the subdomain in a more abstract space, accounting for the periodicity of an object. The virtual region approach introduces the concept of having an activated and a deactivated part of the mesh. The virtual region approach is explained in detail in the following sections. An extension of VRSSMUM is the phantom-domain method,[30] which reuses only the concepts of the virtual ring and virtual region. Since the shear-slip feature of the original SSMUM method is not considered, the mesh topology is preserved during the simulation. Furthermore, this method is coupled with the linear elasticity method to address non-regular nodal displacements of the mesh. The phantom-domain method was successfully applied to FSI problems involving large mesh translation.[30]

Another option for addressing large rotating domains is by sliding mesh techniques.[31,32] Sliding-mesh methods are based on the coupling of surfaces, usually enhanced by using high-order interpolation functions, for example, non-uniform rational B-splines. In general, the mesh topology of each subdomain is preserved, the interface between subdomains is coupled, and the motion can be either imposed or computed using FSI.

Another mesh update technique called Snapping Reference Mesh Update Method (SRMUM) was presented in Reference 33. The SRMUM method has been applied to the flow simulation of the plastic melt inside single-screw[33,34] and twin-screw extruders.[35] In general, SRMUM is a method designed to handle intertwined rotating objects, which makes this method suitable for screw extruder simulations. The mesh topology is preserved, and the nodal positions are updated by using an algorithm based on only algebraic operations, making this approach computationally efficient.

In Reference 36, the space-time interface-tracking with topology change (ST-TC) method is introduced. This approach incorporates a special master-slave system that allows element degeneration and maintains the original mesh connectivity. In particular, elements can collapse and then can be set apart; thus, it makes it possible to represent the contact of surfaces and opening and closing motions. This mesh update technique requires the functionality of excluding collapsed elements from the integration of the equations. The ST-TC method has been applied to, for example, the FSI analysis of a heart valve.[37]

More arbitrary large mesh displacements can be handled by applying local mesh modifications, see References 38-43. This kind of approach applies local mesh optimizations in the vicinity of poor-quality elements, but overall tries to preserve the mesh topology of the initial mesh. Common local mesh optimization includes vertex insertion or removal, edge and face swapping or splitting, connectivity change, and vertex smoothing. Since the mesh topology changes with these operations, solution field projection is required for the modified elements.

The particle finite element method (PFEM)[44,45] offers an option to handle large mesh displacements by an efficient remeshing technique. This method exploits the Lagrange framework for the description of the motion, and the mesh nodes are treated as physical particles. To avoid generating a completely new mesh, PFEM keeps the vertex position, and only a new connectivity is generated. Another important feature is that mesh-to-mesh interpolation can be avoided when nodal variables are used. However, changing the mesh topology can still affect the convergence or the accuracy of the numerical solution.

The time-continuous simplex space-time (C-SST) method[46-48] has been introduced to study flow problems with moving domains, including opening or closing motions. In case of prescribed motion, the deformation of the domain is already included within the space-time mesh, so no mesh update method is required. The C-SST shows advantages regarding parallelization since the decomposition of the mesh is performed in both space and time.

This work introduces the surface-reconstruction virtual-region (SR-VR) mesh update method. This method is designed to handle large boundary displacements, including opening or closing motions. The concept of the virtual region[29] is reused here. Furthermore, the approach is enhanced by a surface reconstruction, which is based on the Steiner constrained Delaunay triangulation (Steiner CDT). The mesh topology changes only in a single layer of elements connected to the virtual region boundary, so that complex surface movement can be achieved. We use the framework of stabilized space-time finite element, but the SR-VR method is not limited to only this kind of discretization. The robustness of the proposed method is shown in two numerical examples: a Poiseuille flow in a moving channel, and a 3D simulation of the closing of a diaphragm valve. The steady and transient simulations show the potential of using a proper mesh update method to study this complex industrial problem.

The article is structured as follows: In Section 2, the governing equations and discretization are described. In Section 3, the proposed mesh update method is introduced. The numerical results are presented in Section 4, where two aforementioned cases are studied. Finally, the concluding remarks are given in Section 5.

2 | GOVERNING EQUATIONS

An incompressible viscous fluid is governed by the Navier–Stokes equations:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u}\right) - \nabla\cdot\boldsymbol{\sigma} = \rho\mathbf{b} \quad \text{on} \quad \Omega_t,\ \forall t \in (0,\mathrm{T}), \tag{1}$$

$$\boldsymbol{\nabla}\cdot\mathbf{u} = 0 \quad \text{on} \quad \Omega_t,\ \forall t \in (0,\mathrm{T}), \tag{2}$$

where **u** and p are the velocity and pressure, respectively; **b** collects the body forces, and ρ is the density.

The system of equations is subjected to initial conditions as well as boundary conditions of the Dirichlet and Neumann type, respectively written as:

$$\mathbf{u} = \mathbf{u_0} \ \text{on}\ \Omega_0, \quad \mathbf{u} = \mathbf{g}\ \text{on}\ \Gamma_g, \quad \mathbf{n}\cdot\boldsymbol{\sigma} = \mathbf{h}\ \text{on}\ \Gamma_h, \tag{3}$$

where Γ_g and Γ_h are complementary parts of Γ.

For Newtonian fluids, the stress tensor is defined as $\boldsymbol{\sigma} = -p\mathbf{I} + 2\eta\boldsymbol{\varepsilon}(\mathbf{u})$ where the rate of strain tensor is $\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^T)$, and η the dynamic viscosity.

The discretization of the governing equation in space and time is based on the deforming-spatial-domain/stabilized space-time finite element formulation.[10,11] This formulation constructs the weak form on a space-time domain; therefore, the deformation of the spatial domain over time is naturally incorporated into the formulation. More details can be found in References 49 and 28.

3 | MESH UPDATE METHOD

This section presents the implementation of the SR-VR mesh update method, which is suitable for problems with complex topology changes and large movement of boundaries.

This method reuses the virtual region concept of having an activated and deactivated portion of the mesh, where elements can freely enter or leave the activated domain. This idea was first introduced in Reference 29 and then extended to FSI problems.[30] In the works mentioned above, the mesh domain was subject to a high translational motion. In particular, the virtual region boundary Γ_{vr} is defined as a flat surface that stays fixed during the simulation, and the large displacement was applied to the background mesh.

The proposed method is focused on a more complex movement of the virtual region boundary Γ_{vr} on unstructured tetrahedral meshes, including large translation and opening or closing motions.

It is important to mention that although a space-time formulation is used, the SR-VR method is not limited to this kind of discretization. For instance, the method can be applied to a semi-discrete approach, that is, ALE formulation, with appropriate modifications.

3.1 | Surface-reconstruction virtual-region method (SR-VR)

The method starts by defining an activated domain Ω_A with respect to the mesh domain Ω_M as $\Omega_A \subset \Omega_M$. The complement is the deactivated domain $\Omega_D = \Omega_M \setminus \Omega_A$. Furthermore, the virtual region surface $\Gamma_{vr} = \Omega_A \cap \Omega_D$ divides both domains Ω_A and Ω_D, see Figure 2A. Note that Γ_{vr} is embedded in an arbitrary position of Ω_M, so there is no requirement for Γ_{vr} to match the background mesh. In this case, a surface reconstruction is applied to ensure boundary conformity on the activated domain Ω_A. It is important to also mention the subdomain of Ω_M formed by only activated elements Ω_A^{in}, and the triangulated domain $\Omega_{\mathcal{X}}$. Both domains are key pieces of the method, and a detailed description will be given in the following paragraphs.

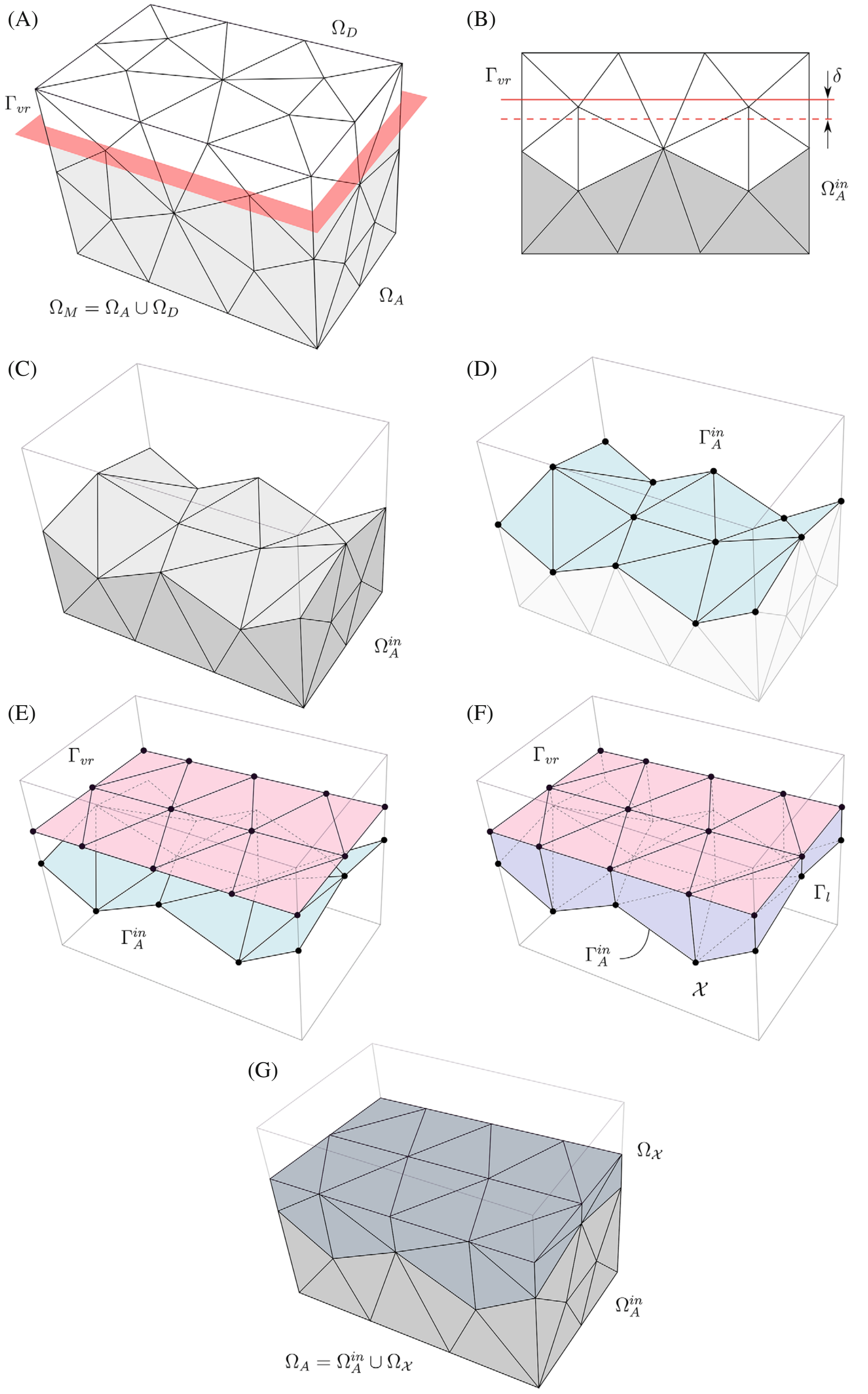

FIGURE 2 Surface reconstruction process: (A) Mesh domain Ω_M, activated domain Ω_A, deactivated domain Ω_D, and virtual region boundary Γ_{vr}. (B) Side view of Ω_M, Γ_{vr}, tolerance δ, and activated elements domain Ω_A^{in}. (C) Full view of Ω_A^{in}. (D) Activated elements surface mesh Γ_A^{in}. (E) Surface mesh Γ_A^{in} projected onto Γ_{vr}. (F) Surface mesh $\mathcal{X} = \Gamma_A^{\text{in}} \cup \Gamma_{vr} \cup \Gamma_l$. (G) Final mesh in the activated domain $\Omega_A = \Omega_A^{\text{in}} \cup \Omega_{\mathcal{X}}$.

The concept of activation refers to the fact that during the assembly of the equation system, only the contribution of activated elements is added to the system, whereas deactivated elements are skipped and do not provide any contribution.[29]

Generally, the surface reconstruction process can be broken down into the following steps:

1. Set up the node and element activity distribution.
2. Define the activated surface mesh Γ_A^{in}.
3. Project Γ_A^{in} onto Γ_{vr}.
4. Smooth Γ_{vr}.
5. Define the surface mesh $\mathcal{X}$ from Γ_A^{in}, Γ_{vr}, and Γ_l.
6. Triangulate $\mathcal{X}$ by the Steiner CDT.
7. Connect the activated domains $\Omega_A = \Omega_A^{\text{in}} \cup \Omega_{\mathcal{X}}$.

Figure 2 shows all the steps of the surface reconstruction. The first step is to set up the node and element activity distribution. In particular, a node is activated if it is located inside the activated domain Ω_A. Furthermore, solely for the node activation, the position of Γ_{vr} is relaxed by a tolerance δ to avoid activating nodes very close to Γ_{vr} that can end in badly shaped elements. Next, the element activity is set up. Particularly, elements inside the activated domain are activated, that is, all elements with completely activated nodes. As a result, the domain formed by only activated elements Ω_A^{in} is defined.

Figure 2B shows the side view of the Ω_M, where the virtual region surface Γ_{vr} and the surface including the tolerance δ are represented by the solid and dashed red lines, respectively. The domain Ω_A^{in} that contains all activated elements is also shown. A full view of Ω_A^{in} is shown in Figure 2C.

The second step is to identify the activated surface mesh Γ_A^{in}, which is obtained from Ω_A^{in}. Note that Γ_A^{in} is defined as a simplicial complex, that is, vertices, segments, and facets. To ensure a consistent Γ_A^{in}, we check for invalid element faces. In general, an invalid element face is a face that, after projecting its nodes into Γ_{vr} will form a collapsed or negative facet. In particular, the normal vector of an element face $\mathbf{n}_i$ and the normal of the virtual boundary $\mathbf{n}_{vr}$ are used to deactivate invalid elements. It is important to mention that $\mathbf{n}_{vr}$ is defined pointing outward from the activated domain. Thus, elements with faces normal $\mathbf{n}_i$ oriented in the opposite direction to the virtual boundary normal $\mathbf{n}_{vr}$ are deactivated, that is, $\mathbf{n}_i \cdot \mathbf{n}_{vr} \leq 0$. If an element on Ω_A^{in} is deactivated by this procedure, Ω_A^{in} needs to be recomputed. Once no new elements are deactivated, a consistent Γ_A^{in} has been created, since all facets are oriented in the same direction of Γ_{vr}. Figure 2D shows the activated surface mesh Γ_A^{in} resulting from the Ω_A^{in}.

The next step is to project the surface mesh Γ_A^{in} onto Γ_{vr}. In particular, vertices on Γ_A^{in} are unidirectionally projected on Γ_{vr}, as shown in Figure 2E. Although unprojectable faces were already deactivated in the previous step, the vertex projection can still lead to badly shaped facets on Γ_{vr}, and consequently, poor quality triangulated cells. Thus, a surface-mesh smoothing scheme is performed on Γ_{vr} vertices to improve the quality of facets. Particularly, a weighted angle-based smoothing scheme is applied in this work. The vertex position on the boundary of Γ_{vr} is constrained, so the smoothing only modifies the position of the inner vertices of Γ_{vr}. It is important to mention that smoothing schemes that modify the connectivity of Γ_{vr} can also be applied, for example, area-based smoothing. The only requirement is that the boundary of Γ_{vr} remains unchanged, so it maintains the exact shape of Γ_A^{in}. For more details about the smoothing approach and the library used, see References 50 and 51.

Once the surface meshes Γ_A^{in} and Γ_{vr} are set up, we proceed to define the lateral surface Γ_l, as shown in Figure 2F. Γ_l is formed by quadrilateral facets generated by using only Γ_A^{in} and Γ_{vr} boundary vertices.

Then, the union of the three aforementioned surfaces Γ_A^{in}, Γ_{vr}, and Γ_l, form the surface $\mathcal{X} = \Gamma_A^{\text{in}} \cup \Gamma_{vr} \cup \Gamma_l$, which is a piecewise linear complex. The domain enclosed by $\mathcal{X}$ is denoted as $\Omega_{\mathcal{X}}$, which is the domain to be triangulated. A triangulation $\mathcal{T}$ of $\mathcal{X}$ is a simplicial complex that decomposes $\Omega_{\mathcal{X}}$ into a simplicial mesh.

In particular, $\mathcal{X}$ is a non-convex hull, so the standard Delaunay triangulation cannot be applied. Instead, the Steiner CDT is used to triangulate $\mathcal{X}$.[52] A Steiner CDT is a triangulation of $\mathcal{X} \cup S$, where S is an additional set of vertices added to it, known as Steiner points. Adding Steiner points to the CDT has three main purposes: creating a valid $\mathcal{T}$, recovering the boundaries of a CDT, and improving the mesh quality of $\mathcal{T}$.

The Steiner CDT algorithm introduced in Reference 53 allows for constrained boundaries of $\mathcal{X}$ by recovering facets and reallocating Steiner points. This algorithm is essential for the SR-VR method since the mesh topology on Γ_A^{in} cannot change during the triangulation. In other words, the mesh faces on Γ_A^{in} and the surface facets of Γ_A^{in} must be topologically equivalent. It should be noted that the Steiner CDT algorithm adds Steiner points only inside $\Omega_{\mathcal{X}}$, and not on the facets of Γ_A^{in}, so Γ_A^{in} remains unaffected. The implementation of the Steiner CDT algorithm used in this work is the one provided by the TetGen library, version 1.6.0. For more details, see Reference 54.

It is important to mention that an additional mesh on Ω_{add} is defined, such that $\Omega_{\mathcal{X}} \subset \Omega_{\text{add}}$. This additional mesh works as a mesh reservoir for the triangulated mesh $\mathcal{T}$ on $\Omega_{\mathcal{X}}$. Thus, the main mesh domain Ω_M remains unchanged. In most cases, the triangulation $\mathcal{T}$ is only a single layer of elements that conforms to the boundary Γ_{vr}, so Ω_{add} is notably smaller than Ω_M. Note also that elements on $\Omega_{\mathcal{X}}$ are activated, whereas the remaining unused elements on $\Omega_{\text{add}} \cap \Omega_{\mathcal{X}}$ are deactivated.

Once the mesh of $\Omega_{\mathcal{X}}$ is generated, both meshes Ω_A^{in} and $\Omega_{\mathcal{X}}$ are connected. The connection of both meshes is performed by updating the connectivity of $\Omega_{\mathcal{X}}$ elements located on the boundary Γ_A^{in}. Note that the connectivity of Ω_M remains unchanged and only the connectivity of elements of $\Omega_{\mathcal{X}}$ changes. Finally, the mesh on the activated domain is $\Omega_A = \Omega_A^{\text{in}} \cup \Omega_{\mathcal{X}}$, as shown in Figure 2G.

Remark 1. In general, $\mathcal{T}$ only changes when the SR-VR method is performed, and this happens as frequently as required. Once SR-VR is performed, no information of the solution vector of the previous time step is available for nodes on $\mathcal{T}$, so interpolation of the solution vector is required. Particularly, not all nodes on $\mathcal{T}$ need interpolation, only Steiner points and those located on Γ_{vr}.

Remark 2. The SR-VR method is performed on space domains only. In particular, for space-time formulation, the mesh update method is performed on the upper-level space-time slab. Then, the mesh information, that is, coordinates and connectivity, is transferred to the lower-level space-time slab. In the case of moving domains, the coordinates on Γ_{vr} are adapted according to the movement of Γ_{vr}.

Remark 3. Regarding parallelization, all triangulation operations are computed in a single partition, and afterwards, the new connectivity information is shared with the partitions that contain mesh information of Ω_{add}. Generally, such sequential operations reduce parallel efficiency. However, the effect is not significant since the domain where they are applied is just a small portion of the entire mesh. We notice that the time required by the mesh update method is considerably less than that of the linear system solver

Remark 4. The total number of nodes and elements are those that belong to both meshes Ω_M and Ω_{add}, and do not change during the simulation. This is particularly important in a multiprocessor environment since the mesh partitioning is performed only once before the simulation starts. However, updating the mesh connectivity still requires updating the communication information among processors during runtime. Note also that the implementation of the method is independent of how both meshes are partitioned. Yet, the communication is more efficient when the number of partitions with information of Ω_{add} is minimal.

Remark 5. In general, the tolerance δ depends on the mesh size where Γ_{vr} is located. We notice that the triangulation can fail only in some particular cases where δ is not defined. One can still find poorly shaped elements for small and large values of δ, but as long as δ is defined, a valid triangulation is obtained. In particular, a value of half of the element size was chosen in our simulations since it gave the best mesh quality.

4 | NUMERICAL RESULTS

4.1 | Poiseuille flow

A variation of the Poiseuille flow, including a moving domain, is investigated to validate the proposed approach. The moving domain is obtained by adding a constant motion on one of the lateral walls. Consequently, this wall is pushing fluid in one direction as a piston would do. A fully developed Poiseuille flow is expected at the outlet of the channel. A sketch of the problem is shown in Figure 3A. The activated Ω_A and deactivated Ω_D domains are shown, where the mesh domain is $\Omega_M = \Omega_A \cup \Omega_D$. Γ_{vr} is positioned at a distance Δx from the left wall, and it moves at a constant speed U. The bottom and top walls are split into a slip and no-slip boundary condition, as shown in the figure. On Γ_{outlet}, a traction-free boundary condition is imposed.

The analytical solution of the flow in the channel is a parabolic velocity profile:

$$u_a(y) = \frac{6Q}{H^3} y \left(H - y\right), \tag{4}$$

where $Q = UH$ is the flow rate, and H is the height of the channel.

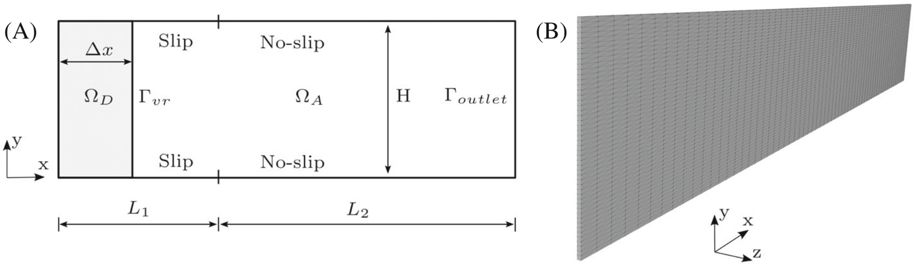

FIGURE 3 Poiseuille flow: (A) Dimensions, boundary conditions, and activated domain. (B) Structured tetrahedral mesh.

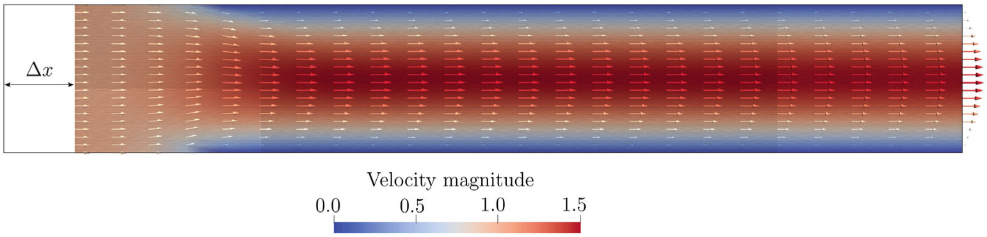

FIGURE 4 Velocity flow field when Γ_{vr} is positioned at $\Delta x = 0.5$.

The following parameters are used in the simulation: $U = 1$, $H = 1$, $L_1 = 1.5$, $L_2 = 5$, viscosity $\eta = 0.1$, and density $\rho = 1$. It is important to mention that the geometry is chosen long enough to obtain a developed Poiseuille flow at the outlet.

The time step is chosen according to a Courant number $C \approx 0.2$, computed with respect to the inflow fluid velocity U and the longitudinal element size.

The initial condition is the steady solution of the whole mesh, that is, $\Delta x = 0$. Then, the unsteady simulation starts, and the Γ_{vr} moves from $\Delta x = 0$ until a steady-state of the moving mesh is reached, which happens for values close to $\Delta x = 0.7$.

A structured tetrahedral mesh Ω_M is used, as shown in Figure 3B. A single element layer is considered in the depth direction. The element shape is defined by an aspect ratio of 5 between the longitudinal and transverse element size. The additional mesh Ω_{add} described in Section 3.1 is not presented, but consists of 2% of elements of Ω_M.

Figure 4 shows the velocity field for Γ_{vr} positioned at $\Delta x = 0.5$. Three states of the fluid can be visualized: a constant velocity profile, a transition flow once the boundary condition changes to no-slip, and a developed parabolic velocity profile at the outlet.

A mesh-convergence study is performed, and the maximum absolute difference $\max(|u - u_a|)$ between the velocity on Γ_{outlet} and the analytical solution (4) is computed. Figure 5 shows the results of the maximum absolute error once the steady-state of the problem is reached at different dimensionless transverse element sizes $h^* = h/H$. The convergence analysis shows that the error decreases quadratically with the mesh size.

4.2 | 3D diaphragm valve simulation

This section shows the result of the steady and the transient simulation of a diaphragm valve. The same geometry, mesh, and fluid properties are used for both cases. It is important to mention that the SR-VR method allows using the same mesh in all simulations. Thus, creating meshes for different opening positions is avoided.

An incompressible viscous fluid with viscosity $\eta = 0.01$Pa s and density $\rho = 1000$ kg/m^3 is considered. The flow is defined by a Reynolds number $Re \sim 200$ based on the channel diameter and the maximum inlet velocity.

A sketch of the problem is shown in Figure 6. Dimensions are presented in Table 1. The geometry is based on Reference 55. The opening distance of the diaphragm l varies from 0 to 15 mm, being closed and fully open, respectively. The diaphragm, represented by the virtual region boundary Γ_{vr}, is shown in its half-open state. The figure also shows the activated Ω_A and deactivated Ω_D domains, where the mesh domain is $\Omega_M = \Omega_A \cup \Omega_D$. The additional mesh Ω_{add} described

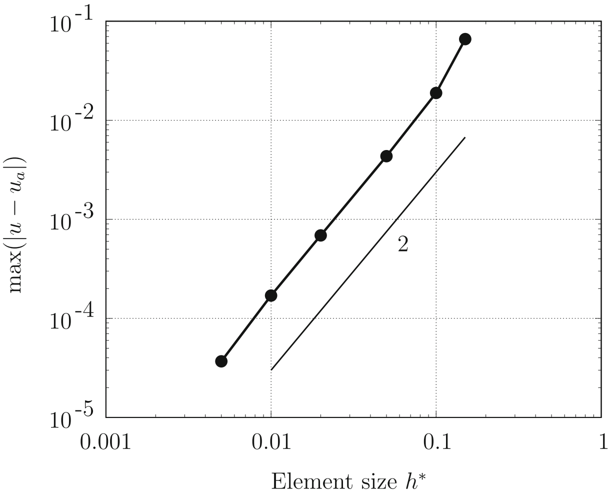

FIGURE 5 Mesh convergence study: Maximum velocity difference at different element sizes.

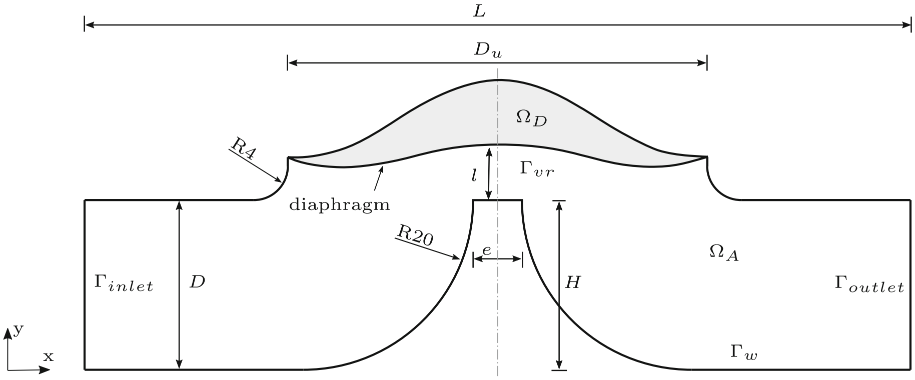

FIGURE 6 Diaphragm valve: Dimensions, boundary conditions, and activated domain.

TABLE 1 Diaphragm valve dimensions in mm.

L	D	D_u	H	l	e
100	20	50	20	0–15	6

in Section 3.1 is not shown, but consists of 8% of the number of elements on Ω_M. The activated domain Ω_A is enclosed by the virtual region boundary Γ_{vr}, the inlet Γ_{inlet}, the outlet Γ_{outlet}, and the remaining wall boundaries Γ_w.

On Γ_{outlet}, a traction-free condition is imposed. For internal flows in channels, an inflow velocity profile is typically imposed. Here, this is not possible due to the unknown behavior of the flow rate during the closing process. Therefore, a mixed boundary condition is imposed on Γ_{inlet} instead. This condition prescribes a zero Dirichlet tangent and bitangent velocity, and an inhomogeneous Neumann traction in the normal direction, as follows:

$$\mathbf{u} \cdot \mathbf{t} = 0 \text{ on } \Gamma_{\text{inlet}}, \quad \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{h} \cdot \mathbf{n} \text{ on } \Gamma_{\text{inlet}}. \tag{5}$$

Note that the normal component of the strain rate tensor is small on the inlet boundary, so the magnitude of the traction vector is close to the pressure value on the inlet, that is, $\mathbf{h} \cdot \mathbf{n} \sim p_{\text{in}}$. It is important to mention that for all simulations, the inlet pressure shows variations no larger than 5%, and a parabolic profile is recovered for steady problems. Therefore, by using this mixed boundary condition, a pressure p_{in} can indirectly be imposed, and a consistent inflow velocity is obtained. Furthermore, a pressure value close to zero $p_{\text{out}} \sim 0$ can be found on the outlet due to the traction-free condition. Consequently, a constant pressure drop Δp is obtained. In particular, $p_{\text{in}} = 5.7$ Pa is used in all simulations.

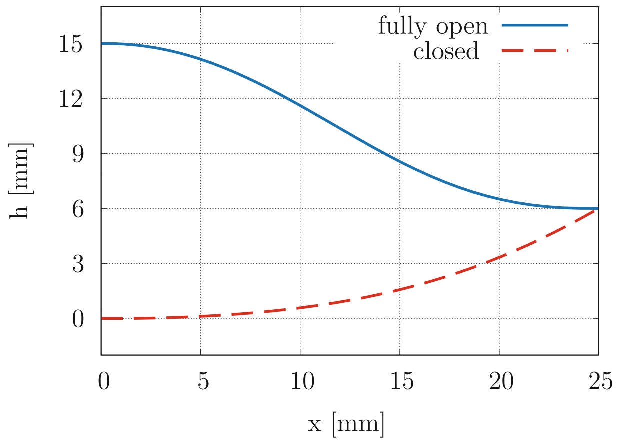

FIGURE 7 Diaphragm valve: Exact position of the diaphragm at two opening positions, fully open and closed.

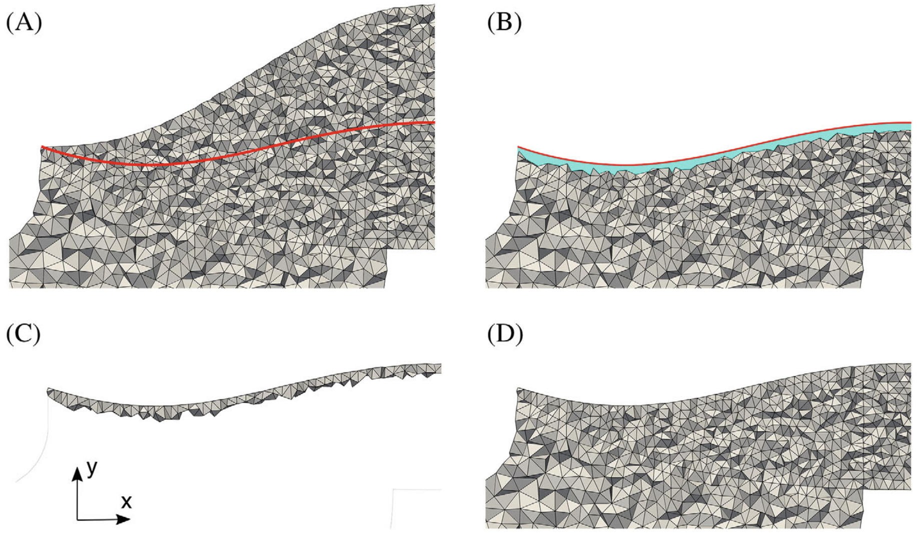

FIGURE 8 Diaphragm valve: Example of the surface reconstruction of the half-open valve position. (A) Main mesh Ω_M and diaphragm position Γ_{vr} (red); (B) elements inside the activated domain Ω_A^{in} and the domain that needs to be triangulated $\Omega_{\mathcal{X}}$ (cian); (C) triangulated mesh on $\Omega_{\mathcal{X}}$; (D) mesh of the activated domain $\Omega_A = \Omega_A^{\text{in}} \cup \Omega_{\mathcal{X}}$.

On the moving boundary Γ_{vr}, we imposed a no-slip condition for steady cases and the mesh velocity for unsteady flows. The mesh velocity is computed from the Γ_{vr} position. Finally, a no-slip boundary condition is imposed on the remaining walls Γ_w.

In Figure 7, the exact position of the fully open and closed diaphragm valve is shown. These two states are the initial and the final ones, respectively. Since the simulations required intermediate times, a Lagrange cubic interpolation in time is used. In general, cubic interpolations require four evaluation points. So, for simplicity, we just duplicate the initial and final curves at equal times. In addition, a cubic interpolation offers a smooth derivative interpolation; in other words, a smooth mesh velocity on Γ_{vr} is imposed.

As example, the surface reconstruction of the diaphragm on a half-open position of the valve is shown in Figure 8. In particular, a slice of the half of mesh on the plane x-y is presented. The four different states of the mesh during the process are shown at each subfigure: the main mesh Ω_M, the portion of the mesh inside the activated domain Ω_A^{in}, the triangulated mesh on $\Omega_{\mathcal{X}}$, and the mesh of the activated domain $\Omega_A = \Omega_A^{\text{in}} \cup \Omega_{\mathcal{X}}$.

An inside view of the mesh is shown in Figure 9. A view of a fully open, half-open, and closed diaphragm valve is presented in Figure 9A–C, respectively. Note that one layer of elements remains in the closed position since no flow is possible through a single layer. The mesh is an unstructured tetrahedral mesh with a refinement on the region where the diaphragm moves. In particular, element sizes of $h = 0.5$ and $h = 1$ mm are used in the fine and coarse mesh zones, respectively. For comparison with the Poiseuille mesh convergence study, we compute the dimensionless element size

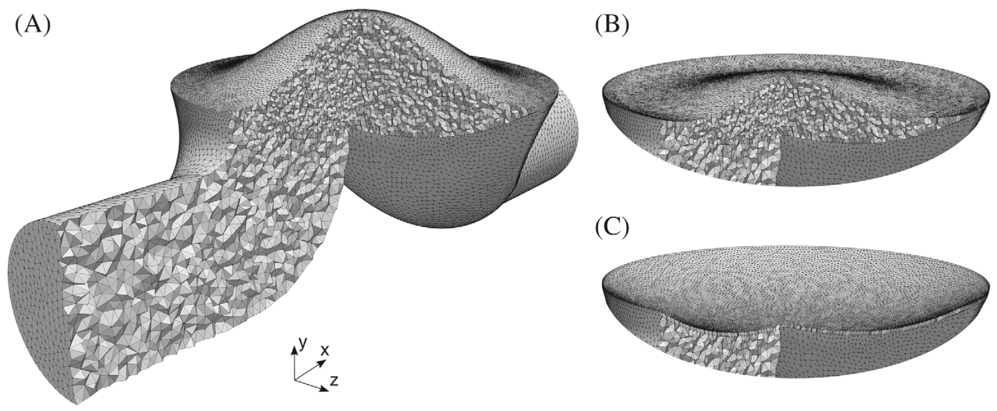

FIGURE 9 Diaphragm valve: Inside view of the mesh at three different opening positions of the valve, (A) fully open, (B) half-open, and (C) closed.

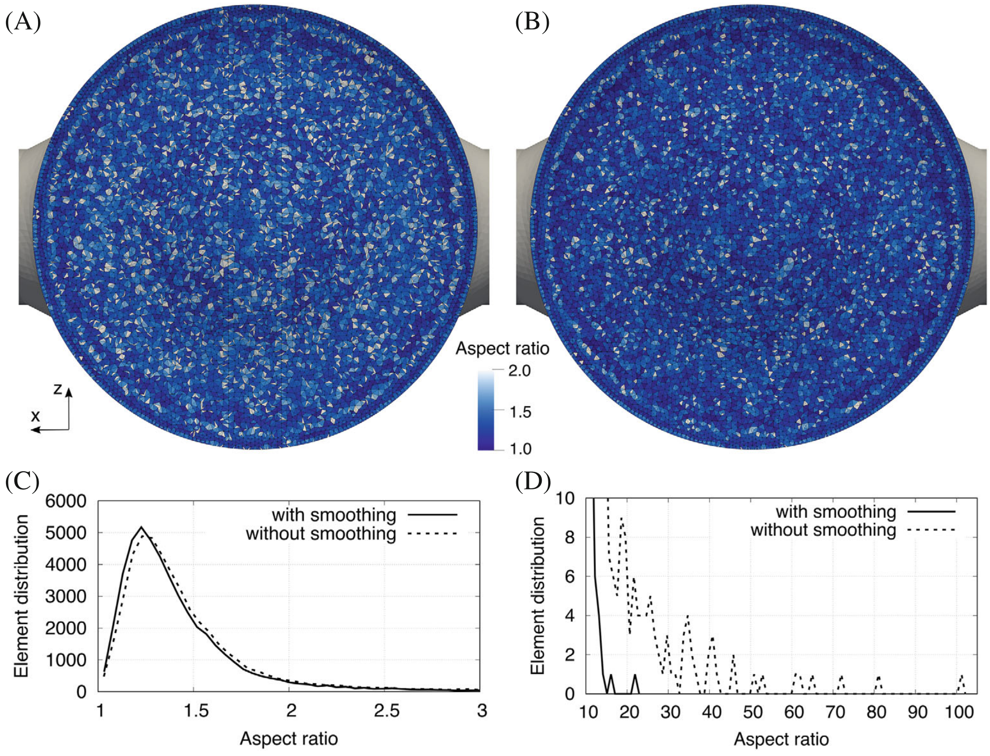

FIGURE 10 Diaphragm valve: Effect of the smoothing technique on the mesh quality of the triangulated mesh $\Omega_{\mathcal{X}}$. Analysis of the half-open positions of the valve; top view of mesh (A) without smoothing and (B) with smoothing; histogram of the mesh element quality at (C) small aspect ratios and (D) large aspect ratios.

$h^* = h/D$ with respect to the valve diameter D. Here, the fine dimensionless element size is $h^* = 0.025$. This value is in the middle of all element sizes studied in the Poiseuille flow test case. The mesh consists of 781,356 elements and 147,792 nodes. The mesh decomposition is computed with ParMetis[56] and includes 336 partitions, of which 30 belong to Ω_{add}.

A smoothing step was mentioned when explaining the surface reconstruction process in Section 3. Here, a brief discussion of the effect of using a smoothing technique on the mesh quality of the triangulated part on $\Omega_{\mathcal{X}}$ is presented. As an example, the half-open valve position results are shown in Figure 10. There is no unique way to measure the mesh quality. In this work, the tetrahedron quality is quantified by the aspect ratio, defined as the ratio between the longest edge and the shortest height. A top view of the mesh showing the element aspect ratio without and with smoothing is presented in Figure 10A, B, respectively. When both figures are compared, the mesh quality improvement is visible when

TABLE 2 Mesh quality over time; average aspect ratio and average of maximum values.

Aspect ratio on $\Omega_{\mathcal{X}}$	Average	Average of max.
Without smoothing	1.65	87.8
With smoothing	1.51	21.4
Aspect ratio on Ω_M	1.33	4.62

smoothing is applied. Furthermore, it is even more evident by coloring a range of small aspect ratios. Figure 10C, D show the histogram of the mesh element quality for different aspect ratios. The histogram of small aspect ratios confirms the visible improvement mentioned before. Although a slight difference is shown for small aspect ratios, there is a considerable effect on large aspect ratios. While the number of elements suddenly dropped at an aspect ratio close to 20 when smoothing is used, without using smoothing, the values continue until 100. Thus, the number of elements with large aspect ratios can be drastically reduced by using a smoothing technique. It is important to mention that even though only the half-open valve position is presented in this example, the same behavior is seen at any other valve opening position. The average aspect ratios over time among all valve opening positions are shown in Table 2. In particular, for elements on the triangulated mesh $\Omega_{\mathcal{X}}$, the average aspect ratio and the average of the maximum aspect ratios are shown. For comparison reasons, the average values of the main mesh Ω_M are also presented. The average values show the same pattern as the example introduced before: a considerable reduction of the maximum aspect ratios can be achieved by using a smoothing technique. However, when they are compared with the values of the main mesh, higher aspect ratios are obtained. Despite this, the aspect ratios still lie within an acceptable range of values.

4.2.1 | Steady simulation

For this section, the steady version of the Navier–Stokes equations (1) and (2) are solved on a static mesh. The SR-VR method is applied to the original mesh to conform Γ_{vr} to a fixed position at a particular opening position of the valve. The same mesh is used in all cases and only the position of Γ_{vr} changes.

As a means of comparison, we relate the fixed-position valve to piping systems. These are subject to energy losses of two kinds, friction loss and minor loss. The first one is due to the fluid friction along the pipe, and the second is due to pipe components. In particular, the minor energy loss in a valve is defined by the loss coefficient K:[57]

$$K = \frac{\Delta p}{\frac{1}{2}\rho V^2}, \tag{6}$$

where Δp, V, and ρ are the pressure drop, the average velocity in the pipe, and the fluid density, respectively.

In Figure 11, the loss coefficient K and the flow rate (%) is shown for different valve openings. Each dot corresponds to a static simulation with a fixed valve opening. The flow rate is computed as a percentage of the fully open valve flow rate. Note that for openings less than 10%, the flow rate decays to values close to zero. We notice that the element size has a significant influence at low openings. Further simulations with finer meshes should be considered for studying openings less than 10%. Regarding the loss coefficient, it drastically increases at openings of less than 40%. High loss coefficient values at openings of less than 30% are not shown.

4.2.2 | Transient simulation

For the transient simulation, the movement of the diaphragm is included. The SR-VR method is applied on Γ_{vr} to ensure boundary conformity during the closing of the valve. Five different closing times are considered, that is, 0.5, 1, 2, 5, and 10 s.

The results of the inflow, outflow, and the average flow rate are shown in Figure 12. As a reference, the results of the static valve are also plotted. The average flow rate is computed from the inlet and outlet flow rate values. The movement of the diaphragm pushes fluid with it; this additional flow rate in the system causes variations in the flow rates at the inlet and outlet. In particular, these variations can produce backflow at the inlet. The fastest closing time, that is, 0.5 s, shows

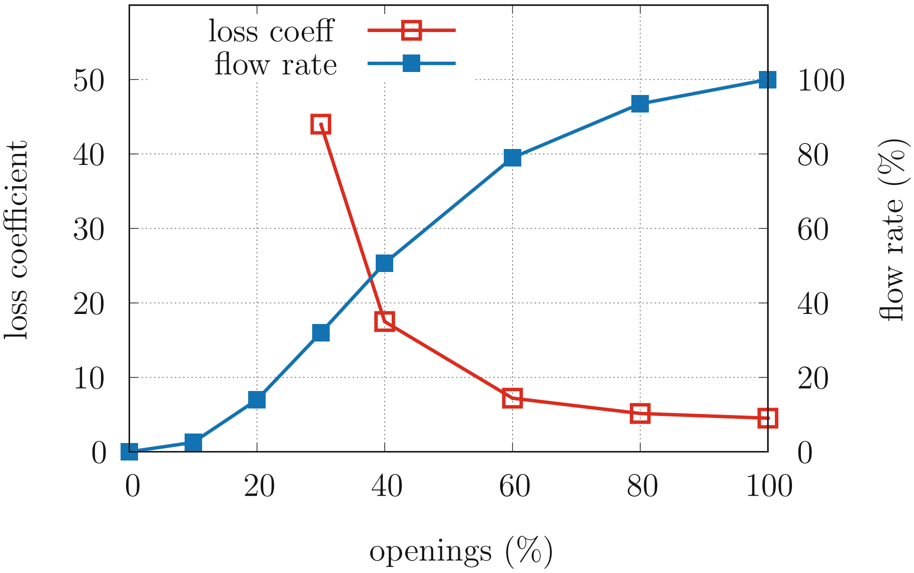

FIGURE 11 Steady diaphragm valve: Loss coefficient and flow rate (%) at different opening.

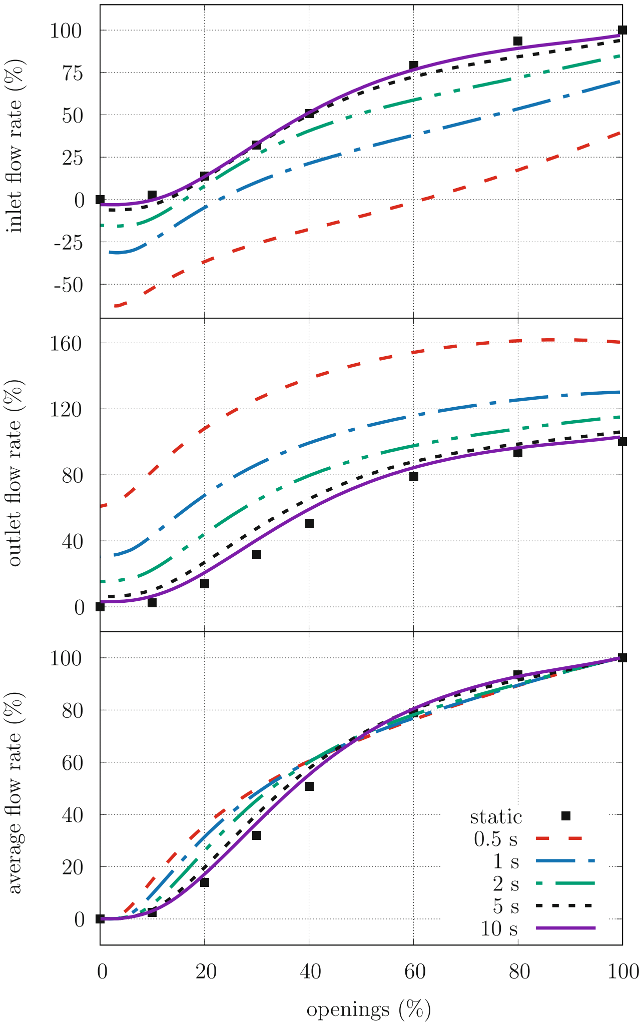

FIGURE 12 Unsteady diaphragm valve: Inflow, outflow, and average flow rate (%) at different opening position of the valve. Results of different closing times are plotted with lines, and results of the static simulation are plotted with dots.

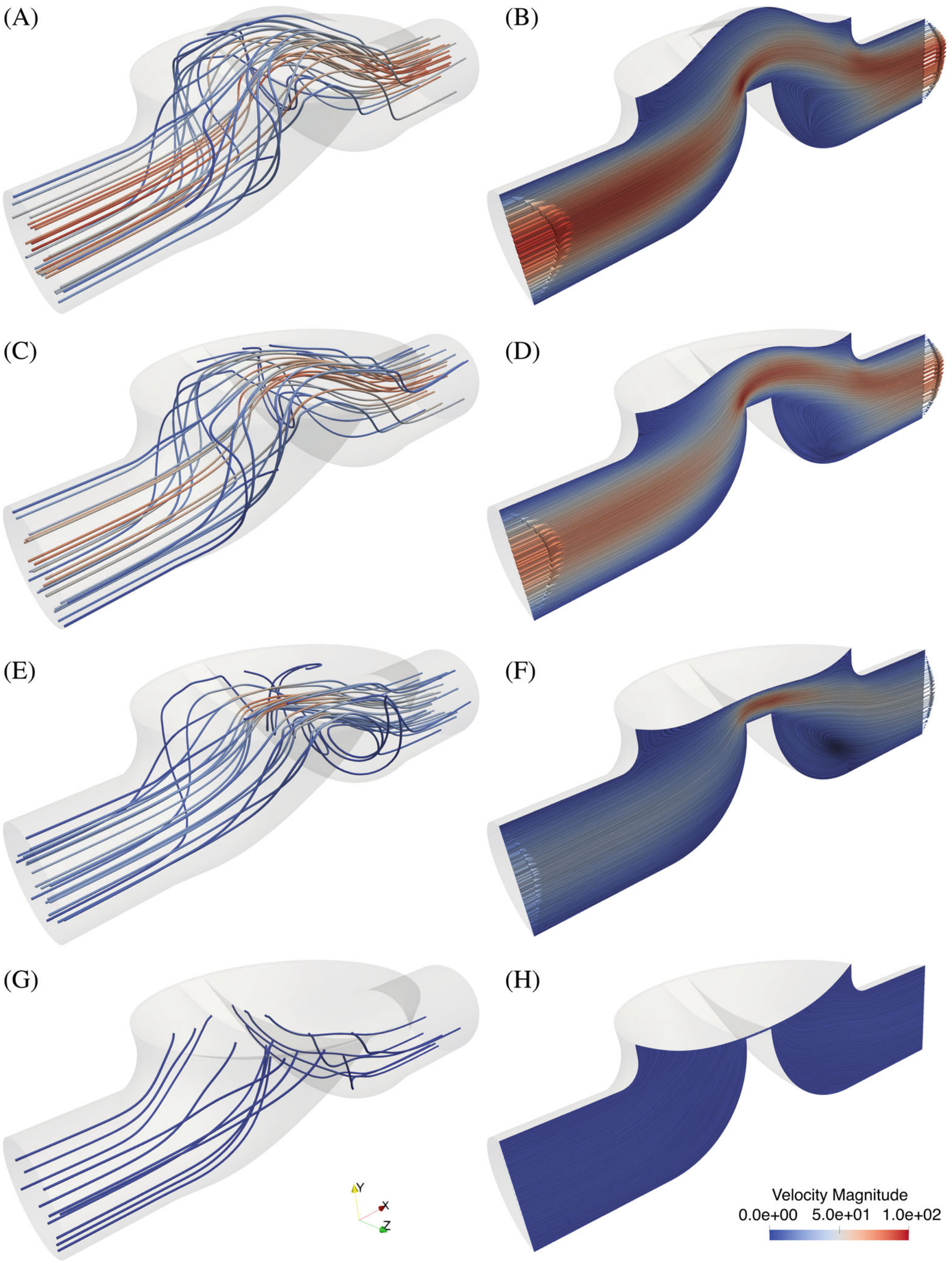

FIGURE 13 Unsteady diaphragm valve: The 3D streamlines (left) and the midsection velocity field (right) are shown for a closing time of 10 s (slow) at four different opening positions: (A,B) 90%, (C,D) 60%, (E,F) 30%, and (G,H) closed.

the most significant influence on the inlet and outlet flow rates, exhibiting a variation of 60% for a fully open valve. In contrast, a closing time of 10 s is closer to the static solution. Even though more significant variations are found at the inlet and outlet flow rates, the average flow rate shows only small ones. Overall, the same flow rate behavior can be considered.

In Figure 13, the 3D streamlines and the mid-section velocity field for a closing time of 10 s at different opening positions of the valve are shown. The plot is enhanced by streamlines, and the velocity profile at the inlet and outlet is also plotted. In general, all the inlet velocity profiles are parabolic with different maximum velocities depending on the opening position of the valve, except for the fully closed valve, where a small backflow component is found. The streamlines show that the main flow comes from the inlet, and the influence of the moving diaphragm is low.

The 3D streamlines and the mid-section velocity field are also shown in Figure 14, but for a closing time of 0.5 s. An essential difference in the flow field is found for this fast-closing valve. The streamlines show that the flow rate caused

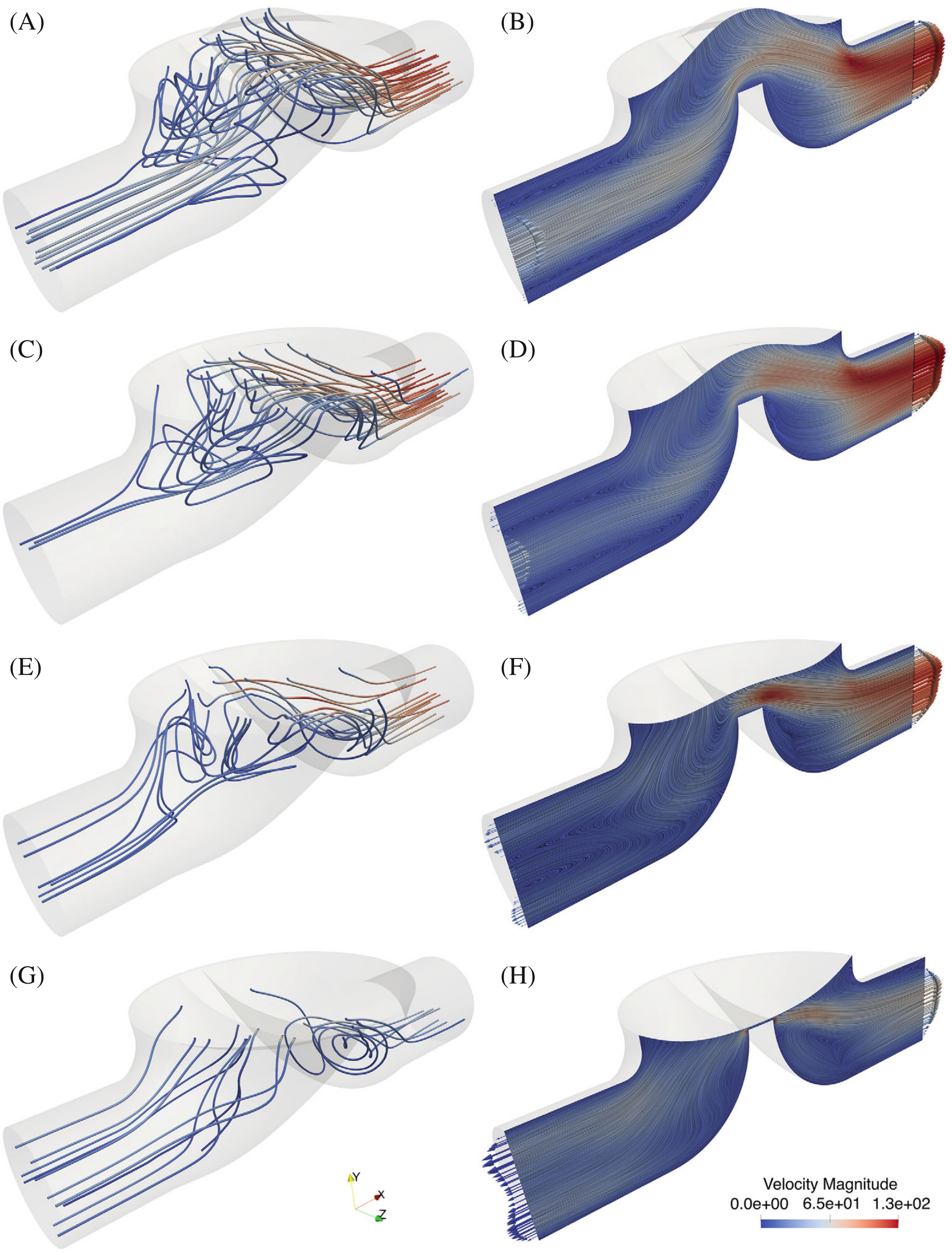

FIGURE 14 Unsteady diaphragm valve: The 3D streamlines (left) and the midsection velocity field (right) are shown for a closing time of 0.5 s (fast) at four different opening positions: (A,B) 90%, (C,D) 60%, (E,F) 30%, and (G,H) closed.

by the diaphragm is relatively high; for instance, the streamlines of Figure 14A,C,E that reach the outlet mainly come from the diaphragm. In addition, backflow is already present at a 90% opening of the valve, and it is increasing over time, finding a maximum value just at the moment of the fully closed position.

5 | CONCLUSIONS

We have presented a boundary-conforming mesh update method that handles challenging problems with large boundary displacements and opening or closing motions. This method, called the SR-VR mesh update method, uses the virtual region concept to handle an activated and deactivated part of the mesh. The boundary conformity between these two domains is enforced by a surface reconstruction process based on the Steiner constrained Delaunay triangulation.

We showed the robustness of the proposed method in two numerical examples. First, we validated the results in a variation of a benchmark Poiseuille flow. Then, we successfully simulated the flow inside of a closing diaphragm valve. In particular, we could address the different challenges presented in this industrial application by using the SR-VR method, for instance, the large displacement of the diaphragm, the complex diaphragm geometry, and the closing of the valve. Results of the diaphragm valve, like the flow rate and the loss coefficient, were presented at different opening positions and closing times. In addition, the proposed method was applied to the same mesh in all simulations at different conditions, avoiding the generation of meshes for each particular case.

Regarding parallelization, the current implementation does not depend on how the different domains are partitioned. However, the communication after a connectivity update can be improved by minimizing the number of processors sharing the triangulated domain. We also notice that by performing the mesh decomposition only once, the partitioning slightly deteriorates with a new connectivity. Repartitioning the mesh might solve this issue, but would incur an additional cost.

In comparison with the precursor methods of this work,[29,30] the SR-VR method is a significant improvement. Even though the method's complexity increases by including connectivity changes, the SR-VR method was shown to be reliable for handling large topological changes. With SR-VR, more complex virtual boundary geometries are achieved, and options to control the mesh quality are added. The latter result in minimizing the chances of the failure of the algorithm and therefore, increasing its robustness.

Finally, we can conclude that the presented mesh update method can be applied in flow problems involving large boundary displacements and topology changes. We plan to extend the application of the SR-VR method to problems where the motion is not prescribed, like free-surface or FSI problems. Generally, for this kind of problems, the mesh update method is included in a strong coupling loop with the governing equations. Hence, the SR-VR method is used several times per time step. A way to avoid this is by using a vertex-displacement-based method in the coupling loop and performing the SR-VR method only when required. In future work, we will discuss the challenges of this kind of problem.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support provided by the National Agency for Research and Development (ANID) and the German Academic Exchange Service (DAAD) through the scholarship CONICYT PFCHA/DOCTORADO ACUERDO BILATERAL CONICYT-DAAD/2018—62180006. We would also like to acknowledge the computing time granted by the JARA Vergabegremium and provided on the JARA Partition part of the supercomputer CLAIX at RWTH Aachen University. This work is also funded by the Federal Ministry of Education and Research (BMBF) and the state of North Rhine-Westphalia as part of the NHR program. Open Access funding enabled and organized by Projekt DEAL.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Felipe A. González https://orcid.org/0000-0001-8638-7864
Stefanie Elgeti https://orcid.org/0000-0002-4474-1666
Marek Behr https://orcid.org/0000-0003-4257-8276

REFERENCES

1. Elgeti S, Sauerland H. Deforming fluid domains within the finite element method: five mesh-based tracking methods in comparison. *Arch Comput Methods Eng*. 2016;23(2):323-361. doi:10.1007/s11831-015-9143-2
2. Osher S, Sethian JA. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J Comput Phys*. 1988;79(1):12-49. doi:10.1016/0021-9991(88)90002-2
3. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *J Comput Phys*. 1981;39(1):201-225. doi:10.1016/0021-9991(81)90145-5
4. McKee S, Tomé MF, Ferreira VG, et al. The MAC method. *Comput Fluids*. 2008;37(8):907-930. doi:10.1016/j.compfluid.2007.10.006
5. Emmerich H. *The Diffuse Interface Approach in Materials Science: Thermodynamic Concepts and Applications of Phase-Field Models*. Vol 73. Springer Science & Business Media; 2003. doi:10.1007/3-540-36409-9
6. Mittal R, Iaccarino G. Immersed boundary methods. *Annu Rev Fluid Mech*. 2005;37(1):239-261. doi:10.1146/annurev.fluid.37.061903.175743
7. Hughes TJ, Liu WK, Zimmermann TK. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Comput Methods Appl Mech Eng*. 1981;29(3):329-349. doi:10.1016/0045-7825(81)90049-9

8. Huerta A, Liu WK. Viscous flow with large free surface motion. *Comput Methods Appl Mech Eng*. 1988;69(3):277-324. doi:10.1016/0045-7825(88)90044-8
9. Donea J, Huerta A, Ponthot JP, Rodríguez-Ferran A. Arbitrary Lagrangian–Eulerian methods. In: Stein E, Borst R, Hughes TJR, eds. *Encyclopedia of Computational Mechanics*. Vol 14. John Wiley & Sons, Ltd; 2004. doi:10.1002/0470091355.ecm009
10. Tezduyar T, Behr M, Liou J. A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space-time procedure: I. The concept and the preliminary numerical tests. *Comput Methods Appl Mech Eng*. 1992;94(3):339-351. doi:10.1016/0045-7825(92)90059-S
11. Tezduyar T, Behr M, Mittal S, Liou J. A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space-time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. *Comput Methods Appl Mech Eng*. 1992;94(3):353-371. doi:10.1016/0045-7825(92)90060-W
12. de Boer A, van der Schoot M, Bijl H. Mesh deformation based on radial basis function interpolation. *Comput Struct*. 2007;85(11):784-795. doi:10.1016/j.compstruc.2007.01.013
13. Luke E, Collins E, Blades E. A fast mesh deformation method using explicit interpolation. *J Comput Phys*. 2012;231(2):586-601. doi:10.1016/j.jcp.2011.09.021
14. Löhner R, Yang C. Improved ALE mesh velocities for moving bodies. *Commun Numer Methods Eng*. 1996;12(10):599-608. doi:10.1002/(SICI)1099-0887(199610)12:10<599::AID-CNM1>3.0.CO;2-Q
15. Farhat C, Degand C, Koobus B, Lesoinne M. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Comput Methods Appl Mech Eng*. 1998;163(1-4):231-245. doi:10.1016/S0045-7825(98)00016-4
16. Degand C, Farhat C. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Comput Struct*. 2002;80(3-4):305-316. doi:10.1016/S0045-7949(02)00002-0
17. Bottasso CL, Detomi D, Serra R. The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes. *Comput Methods Appl Mech Eng*. 2005;194(39-41):4244-4264. doi:10.1016/j.cma.2004.08.014
18. Masud A, Bhanabhagvanwala M, Khurram RA. An adaptive mesh rezoning scheme for moving boundary flows and fluid–structure interaction. *Comput Fluids*. 2007;36(1):77-91. doi:10.1016/j.compfluid.2005.07.013
19. Kanchi H, Masud A. A 3D adaptive mesh moving scheme. *Int J Numer Methods Fluids*. 2007;54(6-8):923-944. doi:10.1002/fld.1512
20. Shamanskiy A, Simeon B. Mesh moving techniques in fluid-structure interaction: robustness, accumulated distortion and computational efficiency. *Comput Mech*. 2021;67(2):583-600. doi:10.1007/s00466-020-01950-x
21. Johnson AA, Tezduyar TE. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Comput Methods Appl Mech Eng*. 1994;119(1-2):73-94. doi:10.1016/0045-7825(94)00077-8
22. Takizawa K, Tezduyar TE, Boben J, Kostov N, Boswell C, Buscher A. Fluid–structure interaction modeling of clusters of spacecraft parachutes with modified geometric porosity. *Comput Mech*. 2013;52(6):1351-1364. doi:10.1007/s00466-013-0880-5
23. Baum J, Luo H, Loehner R. A new ALE adaptive unstructured methodology for the simulation of moving bodies. Proceedings of the 32nd Aerospace Sciences Meeting and Exhibit; 1994. 10.2514/6.1994-414
24. Hassan O, Sørensen K, Morgan K, Weatherill N. A method for time accurate turbulent compressible fluid flow simulation with moving boundary components employing local remeshing. *Int J Numer Methods Fluids*. 2007;53(8):1243-1266. doi:10.1002/fld.1255
25. Saksono P, Dettmer W, Perić D. An adaptive remeshing strategy for flows with moving boundaries and fluid–structure interaction. *Int J Numer Methods Eng*. 2007;71(9):1009-1050. doi:10.1002/nme.1971
26. Behr M, Tezduyar T. The shear-slip mesh update method. *Comput Methods Appl Mech Eng*. 1999;174(3):261-274. doi:10.1016/S0045-7825(98)00299-0
27. Behr M, Arora D. Shear-slip mesh update method: implementation and applications. *Comput Methods Biomech Biomed Eng*. 2003;6(2):113-123. doi:10.1080/1025584031000091650
28. Pauli L, Behr M. On stabilized space-time FEM for anisotropic meshes: incompressible Navier–Stokes equations and applications to blood flow in medical devices. *Int J Numer Methods Fluids*. 2017;85(3):189-209. doi:10.1002/fld.4378
29. Key F, Pauli L, Elgeti S. The virtual ring shear-slip mesh update method. *Comput Fluids*. 2018;172:352-361. doi:10.1016/j.compfluid.2018.04.006
30. Hilger D, Hosters N, Key F, Elgeti S, Behr M. A novel approach to fluid-structure interaction simulations involving large translation and contact. In: van Brummelen H, Vuik C, Möller M, Verhoosel C, Simeon B, Jüttler B, eds. *Conference on Isogeometric Analysis and Applications*. Springer; 2018:39-56.
31. Bazilevs Y, Hughes T. NURBS-based isogeometric analysis for the computation of flows about rotating components. *Comput Mech*. 2008;43(1):143-150. doi:10.1007/s00466-008-0277-z
32. Takizawa K, Tezduyar TE, Mochizuki H, et al. Space–time VMS method for flow computations with slip interfaces (ST-SI). *Math Models Methods Appl Sci*. 2015;25(12):2377-2406. doi:10.1142/S0218202515400126
33. Helmig J, Behr M, Elgeti S. Boundary-conforming finite element methods for twin-screw extruders: unsteady-temperature-dependent-non-Newtonian simulations. *Comput Fluids*. 2019;190:322-336. doi:10.1016/j.compfluid.2019.06.028
34. Helmig J, Key F, Behr M, Elgeti S. Combining boundary-conforming finite element meshes on moving domains using a sliding mesh approach. *Int J Numer Methods Fluids*. 2021;93(4):1053-1073. doi:10.1002/fld.4919
35. Hinz J, Helmig J, Möller M, Elgeti S. Boundary-conforming finite element methods for twin-screw extruders using spline-based parameterization techniques. *Comput Methods Appl Mech Eng*. 2020;361:112740. doi:10.1016/j.cma.2019.112740
36. Takizawa K, Tezduyar TE, Buscher A, Asada S. Space–time interface-tracking with topology change (ST-TC). *Comput Mech*. 2014;54(4):955-971. doi:10.1007/s00466-013-0935-7

37. Terahara T, Takizawa K, Tezduyar TE, Bazilevs Y, Hsu MC. Heart valve isogeometric sequentially-coupled FSI analysis with the space–time topology change method. *Comput Mech*. 2020;65(4):1167-1187. doi:10.1007/s00466-019-01813-0
38. Tezduyar TE, Sathe S, Stein K, Aureli L. Modeling of fluid-structure interactions with the space-time techniques. In: Bungartz HJ, Schäfer M, eds. *Fluid-Structure Interaction*. Springer; 2006:50-81. doi:10.1007/3-540-34596-5_3
39. Dobrzynski C, Frey P. Anisotropic Delaunay mesh adaptation for unsteady simulations. Proceedings of the 17th international Meshing Roundtable; 2008:177–194; Springer. 10.1007/978-3-540-87921-3_11
40. Compere G, Remacle JF, Jansson J, Hoffman J. A mesh adaptation framework for dealing with large deforming meshes. *Int J Numer Methods Eng*. 2010;82(7):843-867. doi:10.1002/nme.2788
41. Alauzet F. A changing-topology moving mesh technique for large displacements. *Eng Comput*. 2014;30(2):175-200. doi:10.1007/s00366-013-0340-z
42. Barral N, Alauzet F. Three-dimensional CFD simulations with large displacement of the geometries using a connectivity-change moving mesh approach. *Eng Comput*. 2019;35(2):397-422. doi:10.1007/s00366-018-0607-5
43. Daldoul W, Hachem E, Mesri Y. A 'R-to-H' mesh adaptation approach for moving immersed complex geometries using parallel computers. *Int J Comput Fluid Dyn*. 2020;34(7-8):597-609. doi:10.1080/10618562.2020.1783441
44. Oñate E, Idelsohn SR, Del Pin F, Aubry R. The particle finite element method—an overview. *Int J Comput Methods*. 2004;1(02):267-307. doi:10.1142/S0219876204000204
45. Cremonesi M, Franci A, Idelsohn S, Oñate E. A state of the art review of the particle finite element method (PFEM). *Arch Comput Methods Eng*. 2020;27(5):1709-1735. doi:10.1007/s11831-020-09468-4
46. Hughes TJ, Hulbert GM. Space-time finite element methods for elastodynamics: formulations and error estimates. *Comput Methods Appl Mech Eng*. 1988;66(3):339-363. doi:10.1016/0045-7825(88)90006-0
47. Von Danwitz M, Karyofylli V, Hosters N, Behr M. Simplex space-time meshes in compressible flow simulations. *Int J Numer Methods Fluids*. 2019;91(1):29-48. doi:10.1002/fld.4743
48. Von Danwitz M, Antony P, Key F, Hosters N, Behr M. Four-dimensional elastically deformed simplex space-time meshes for domains with time-variant topology. *Int J Numer Methods Fluids*. 2021;93(12):3490-3506. doi:10.1002/fld.5042
49. Behr M, Tezduyar T. Finite element solution strategies for large-scale flow simulations. *Comput Methods Appl Mech Eng*. 1994;112(1):3-24. doi:10.1016/0045-7825(94)90016-7
50. Surazhsky V, Gotsman C. High quality compatible triangulations. *Eng Comput*. 2004;20(2):147-156. doi:10.1007/s00366-004-0282-6
51. Loriot S, Rouxel-Labbé M, Tournois J, Yaz IO. Polygon mesh processing. *CGAL User and Reference Manual*. 5.3.1 ed. CGAL Editorial Board; 2021. https://doc.cgal.org/5.3.1/Manual/packages.html#PkgPolygonMeshProcessing
52. Shewchuk JR. General-dimensional constrained Delaunay and constrained regular triangulations, I: Combinatorial properties. *Twentieth Anniversary Volume*. Springer; 2009:1-58. doi:10.1007/978-0-387-87363-3_28
53. Si H, Gärtner K. 3D boundary recovery by constrained Delaunay tetrahedralization. *Int J Numer Methods Eng*. 2011;85(11):1341-1364. doi:10.1002/nme.3016
54. Si H. TetGen, a Delaunay-based quality tetrahedral mesh generator. *ACM Trans Math Softw*. 2015;41(2):1-36. doi:10.1145/2629697
55. Liu Y, Lu L, Zhu K. Numerical analysis of the diaphragm valve throttling characteristics. *Processes*. 2019;7(10):671. doi:10.3390/pr7100671
56. Karypis G, Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J Sci Comput*. 1998;20(1):359-392. doi:10.1137/S1064827595287997
57. White FM, Majdalani J. *Viscous Fluid Flow*. Vol 3. McGraw-Hill; 2006.

How to cite this article: González FA, Elgeti S, Behr M. The surface-reconstruction virtual-region mesh update method for problems with topology changes. *Int J Numer Methods Eng*. 2023;124(9):2050-2067. doi: 10.1002/nme.7200