Received: 28 June 2023 | Revised: 12 July 2023 | Accepted: 23 August 2023

DOI: 10.1002/pamm.202300067

RESEARCH ARTICLE

Fluid-structure interaction with fully coupled mesh generation

Teresa Schwentner | **Thomas-Peter Fries**

Institute of Structural Analysis, Graz University of Technology, Graz, Austria

Correspondence
Teresa Schwentner, Institute of Structural Analysis, Graz University of Technology, Lessingstraße 25/II, A-8010 Graz, Austria.
Email: schwentner@tugraz.at

Abstract

The modeling and simulation of fluid-structure interaction (FSI) involve the two-way coupling of a fluid flow and a deforming structure. The fluid exerts a load on the structure, the structure deforms, and, hence, the fluid flow changes, which leads to an altered loading on the structure. To accomplish this coupling, the fluid domain has to be updated after each time step, leading to a change in the fluid mesh. Various mesh update methods have been presented where an initial mesh is generated and then updated after each time step. Each method comes with different levels of complexity and has its own advantages and disadvantages. Herein, FSI simulations with fully integrated mesh generation, rather than updates, are proposed. Instead of updating the mesh at each time step, a new mesh for the fluid domain is generated based on the deformation of the solid domain. An advanced structured meshing algorithm, based on a block structure, enables this integrated mesh generation approach. An initial set of coarse-scale conforming blocks is generated by the user, representing the topology and rough initial position of the solid and the fluid. Further information is the exact geometry at the boundary and the deformed position of the fluid-structure interface, plus grading information to obtain a high-quality fluid mesh. Transfinite maps are used to generate elements inside the blocks with any desired resolution and order.

1 | INTRODUCTION

In fluid-structure interaction (FSI), the fluid forces lead to a deformation of the solid. Subsequently, the fluid flow changes, which leads to a coupling of the domains. As a result, the mesh has to be updated at every time step, as seen in Figure 1. Various methods for this update exist, each with different advantages and disadvantages. Especially for higher-order elements, the existing approaches lead to rather poor results or even to invalid meshes, which abort the simulation. Furthermore, choosing and adapting the right method for a specific FSI application can be difficult, often, a lot of fine-tuning is needed.

One of the more common methods is an elasticity analogy method [1], where the mesh is treated as a solid and the deformations for the mesh nodes are calculated by solving the elasticity equations, therefore, also called the pseudo-solid approach. Another popular method is, for example, the spring analogy method [2], where the nodes are connected with

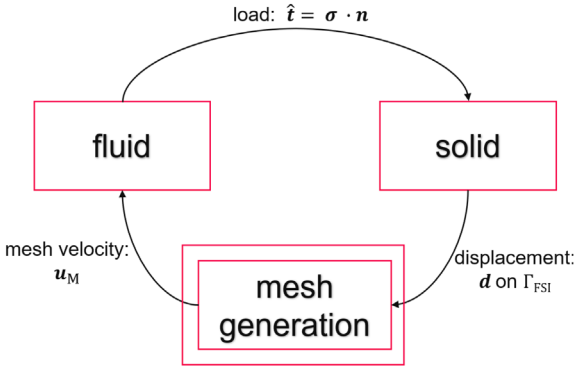

FIGURE 1 The coupled problem for FSI.

springs and the node positions are then calculated based on the spring deformations. Various methods exist, like the bi-harmonic approach[3], the radial basis function interpolation method [4], or the Delaunay interpolation method [5], only to mention a few. Some examples of hybrid approaches, combine different methods are the moving submesh method [6] or the finite macro-element-based method [7, 8]. The disadvantage of most of the mentioned approaches is, that a large system of equations has to be solved, with often one equation per node.

Instead of updating the mesh in every time step and solving a system of equations, herein, we propose to fully couple the mesh generation into the FSI simulation. That is, a new mesh is generated in every time step, using an advanced higher-order block-structured mesh generation method. This leads to a very robust approach, especially for higher-order elements. The mesh generation is based on the topology definition via a block structure, using conforming quads, geometry, and grading assignments to the block edges, and transfinite mapping functions to generate higher-order elements inside the blocks. Based on this approach, any desired order and number of elements may be generated.

The remainder of the article is organized as follows: In Section 2, the governing equations for the FSI problem are described, and in Section 3, the mesh generation is presented. Section 4 describes two numerical examples and conclusions are given in Section 5.

2 | FLUID-STRUCTURE INTERACTION

The coupled problem in FSI is based on a $\overline{d}$-dimensional flow acting on a structure. The computational domain $\Omega^{\overline{d}} \subset \mathbb{R}^{\overline{d}}$ is subdivided into the solid and fluid domains Ω_{S} and Ω_{F}, respectively, containing the FSI-interface denoted Γ_{FSI}. This problem is solved using a partitioned approach, only briefly described herein.

For the solid problem, a total Lagrangian formulation is used, always referring to the initial configuration. The structure is modeled using the St. Venant solid, with the governing equations [9] given by

$$\varrho_{\mathrm{S}} \ddot{\boldsymbol{d}} - \nabla_{\boldsymbol{X}} \cdot (\mathbf{F}\mathbf{S}) - \boldsymbol{f} = \mathbf{0} \qquad \text{in} \quad \Omega_{\mathrm{S}}, \tag{1}$$

where ϱ_{S} is the solid's density, $\boldsymbol{d}$ are the displacements and $\boldsymbol{f}$ are the volume forces. Furthermore the deformation gradient $\mathbf{F} = \mathbf{I} + \nabla_{\boldsymbol{X}} \boldsymbol{d}$, the second Piola–Kirchhoff stress tensor $\mathbf{S} = \lambda(\mathrm{tr}\mathbf{E})\mathbf{I} + 2\eta\mathbf{E}$ with λ and η being the Lamé parameters and the Green–Lagrange strain tensor $\mathbf{E} = \frac{1}{2}(\mathbf{F}^{\mathrm{T}}\mathbf{F} - \mathbf{I})$ are given.

Instead the fluid domain model is formulated based on an arbitrary Lagrangian–Eulerian (ALE) description enabling moving meshes during the simulation. The governing equations are the instationary, incompressible Navier–Stokes equations [10], consisting of the momentum and the continuity equation. In the pressure-velocity formulation they are

$$\varrho_{\mathrm{F}} (\dot{\boldsymbol{u}} + \bar{\boldsymbol{u}} \cdot \nabla_{\boldsymbol{x}} \boldsymbol{u}) - \nabla_{\boldsymbol{x}} \cdot \boldsymbol{\sigma} - \boldsymbol{f} = \mathbf{0} \qquad \text{in} \quad \Omega_{\mathrm{F}}, \tag{2}$$

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{u} = 0 \qquad \text{in} \quad \Omega_{\mathrm{F}}, \tag{3}$$

where ϱ_{F} is the fluid's density, $\boldsymbol{u}$ are the velocities. $\bar{\boldsymbol{u}} = \boldsymbol{u} - \boldsymbol{u}_{\mathrm{M}}$ are the advective velocities, with $\boldsymbol{u}_{\mathrm{M}}$ being the mesh velocities. The fluid stress tensor $\boldsymbol{\sigma}$ for a Newtonian fluid is given as $\boldsymbol{\sigma}(\boldsymbol{u}, p) = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u})$ with $\boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}(\nabla_{\boldsymbol{x}}\boldsymbol{u} + (\nabla_{\boldsymbol{x}}\boldsymbol{u})^{\mathrm{T}})$, with p being the fluid's pressure and μ being the dynamic viscosity.

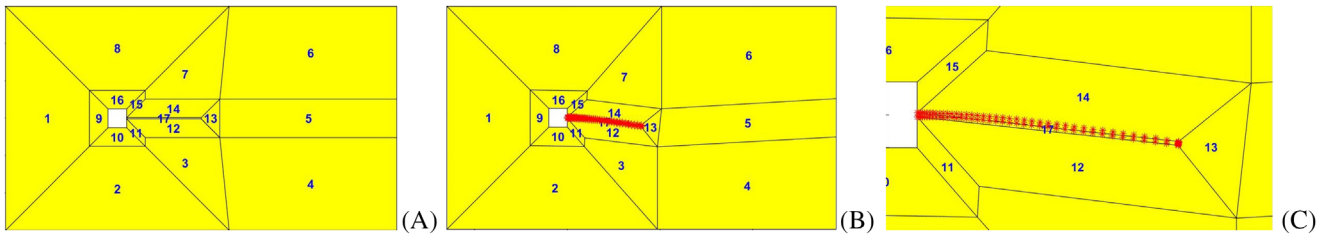

FIGURE 2 Block-structured mesh generation: (A) Block-structure—valid coarse linear mesh; (B) simulated solid displacements on FSI-interface; (C) geometry assigned via node position on FSI-interface.

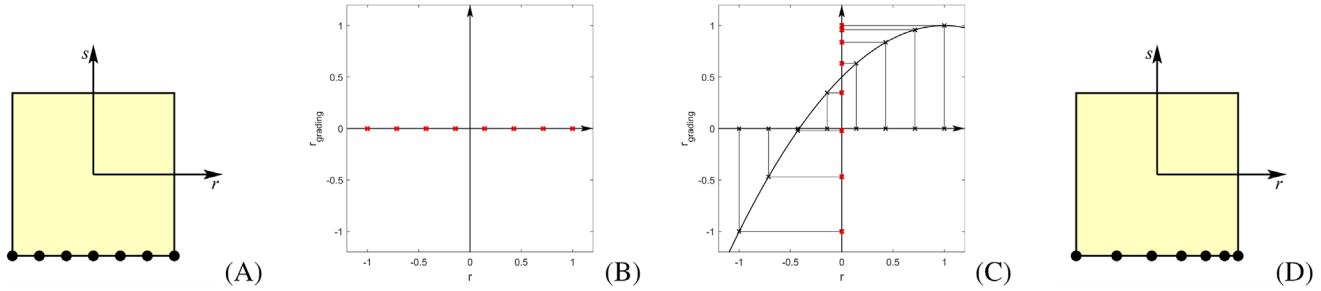

FIGURE 3 Grading assigned to a block edge: (A) Equally distributed nodes on a reference element; (B) reference nodes plotted on the graph; (C) evaluated quadratic function for the reference nodes; and (D) new quadratic distribution of the nodes on the element edge.

On the moving, hence, time-dependent FSI-interface, Γ_{FSI}, geometrical consistency and mechanical equilibrium is required. Further boundary conditions are omitted for brevity. The spatial discretization is done via a finite element framework using Taylor–Hood elements and, for the temporal discretization, a finite difference method is used. The weak forms of the equations, serving as the basis for the finite element treatment, are omitted here for brevity as well.

3 | HIGHER-ORDER BLOCK-STRUCTURED MESH GENERATION

In the FSI-problem, the instationary Navier–Stokes equations are formulated using the ALE description, naturally enabling a consistent consideration of the moving mesh in the fluid domain. For the proposed inherent and frequent mesh generation, an advanced higher-order block-structured mesh generation is used, consisting of the topology definition via a block-structure, geometry and grading assigned to the block edges and transfinite maps to generate nodes inside the blocks. The mesh velocity is needed to calculate the advective velocity in Equation (2). As the number and the order of the elements do not change throughout the simulation, the calculation of the mesh velocity is simple and no search or projection algorithm is required. Next, the individual steps are described in more detail.

As a first step, the topology is defined via a block-structure using conforming (bi-linear) quads. The coordinates and the node connectivity are user input, and defined so that the block-structure leads to a valid coarse and linear mesh, allowing the occurring deformation of the FSI-interface without producing invalid blocks, exemplified in Figure 2A. The initial geometry information is then assigned to the block edges using explicit or implicit definitions, for example, higher-order edge elements, NURBS, B-splines, function evaluations, or isolines in the frame of the level-set method [11, 12]. After the first time step of the simulation, the deformation of the FSI-interface results directly from the displacement field of the solid, as shown in Figure 2B. This means that the simulated solid displacements serve as the new geometry representation on the block edges corresponding to the FSI-interface Γ_{FSI}, exemplified in Figure 2C. On all other edges, the same geometry information as in the initial mesh is prescribed.

Next, grading on the edges to resolve boundary layers is assigned. The steps to generate the desired grading are presented in Figure 3. As an example, a quadratic grading on the lower edge of a 2D quad-element toward one of the vertices is considered here. Starting points are equally distributed nodes on the lower edge of the reference element, shown in Figure 3A, which are plotted on the abscissa of Figure 3B. A quadratic function is then used to evaluate the graded new positions shown on the ordinate, see Figure 3C. The new distribution on the ordinate is then mapped back to the reference element in Figure 3D. Other than quadratic grading functions may be used alternatively, enabling different grading options, for

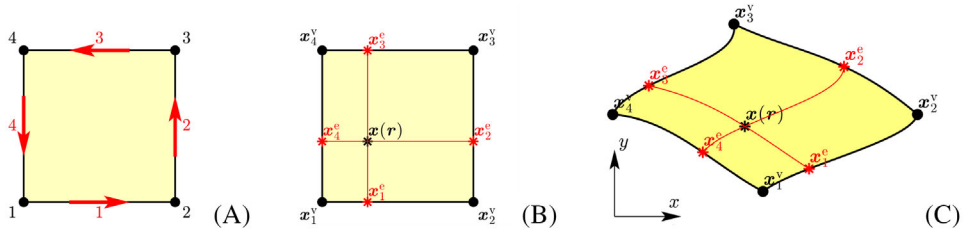

FIGURE 4 Transfinite maps: (A) Vertex and edge numbering; (B) coordinate data on vertices and edges for $\boldsymbol{x}(\boldsymbol{r})$; (C) element and data in a Cartesian coordinate system; [15].

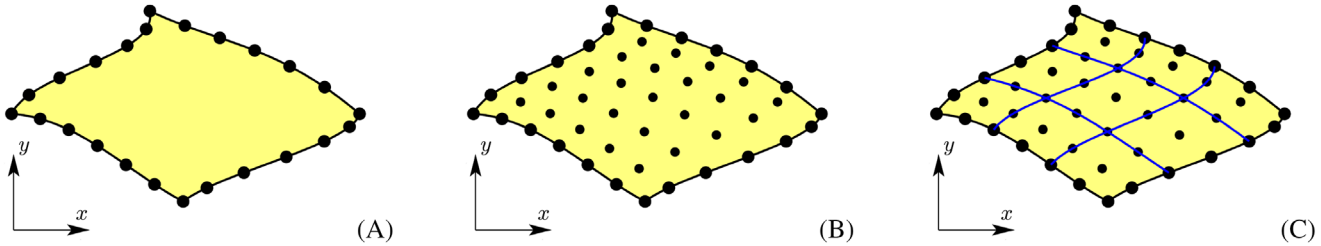

FIGURE 5 Higher-order submeshes: (A) Reference element with nodes on the edges; (B) reference element with generated nodes via transfinite mapping function; (C) element and data in a Cartesian coordinate system.

example, with weaker and stronger grading or grading in the other or both directions of the edge. After accomplishing these steps, the nodes on the edges of the blocks are defined, based on the desired number and order of the elements. As a next step, the nodes inside the blocks are generated using transfinite maps.

3.1 | Transfinite maps

This step enables the generation of the nodes inside the blocks based on the edge nodes, using transfinite mapping functions. The concept is based on the blending-function method [13, 14] and also outlined for 3D cases in [15].

Starting point is a reference element in the (r, s)-coordinate system with numbered edges and vertices, as seen in Figure 4A. The usual bi-linear shape functions $N_i(\boldsymbol{r})$ are assigned to the reference element, resulting in

$$N_1(\boldsymbol{r}) = \frac{1}{4}(1-r)\cdot(1-s), \quad N_2(\boldsymbol{r}) = \frac{1}{4}(1+r)\cdot(1-s),$$
$$N_3(\boldsymbol{r}) = \frac{1}{4}(1+r)\cdot(1+s), \quad N_4(\boldsymbol{r}) = \frac{1}{4}(1-r)\cdot(1+s).$$

Based on the shape functions, every edge is now assigned a ramp function $R_i(\mathbf{r})$

$$R_1 = N_1 + N_2, \quad R_2 = N_2 + N_3, \quad R_3 = N_3 + N_4, \quad R_4 = N_4 + N_1,$$

being 1 on the corresponding edge.

For a point $\boldsymbol{x}(\boldsymbol{r})$, seen in Figure 4B, coordinates on the related edges $\boldsymbol{x}_i^{\mathrm{e}} \in \mathbb{R}^2$ and vertices $\boldsymbol{x}_i^{\mathrm{v}} \in \mathbb{R}^2$ are given. In order to generate, the position $\boldsymbol{x}(\boldsymbol{r})$ in terms of the given vertex and edge data, the following transfinite map is used,

$$\boldsymbol{x}(\boldsymbol{r}) = \sum_{i=1}^{4} R_i(\boldsymbol{r}) \cdot \boldsymbol{x}_i^{\mathrm{e}} - \sum_{i=1}^{4} N_i(\boldsymbol{r}) \cdot \boldsymbol{x}_i^{\mathrm{v}}. \tag{4}$$

An example is seen in Figure 4C. Based on this function, the nodes inside the blocks can now be generated. In Figure 5A, the nodes on the edges related to the desired number and order of the elements inside the block are shown. Based on these nodes and the transfinite map in Equation (4) the inner nodes are generated and associated with elements via a

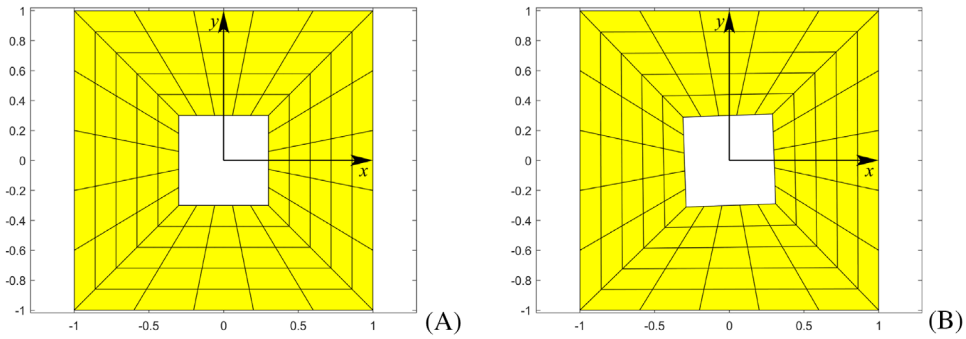

FIGURE 6 L_2-projection: (A) Initial configuration; (B) configuration for simulation: square cavity rotated by 2°.

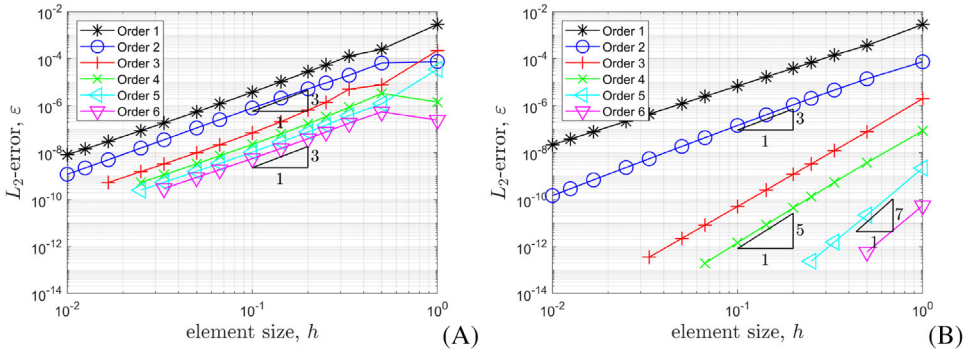

FIGURE 7 Convergence study—L_2-error of L_2-projection: (A) pseudo-solid approach; (B) mesh generation approach.

connectivity matrix, as depicted in Figure 5C. Generating these submeshes for all blocks and connecting them to one mesh, leads to a fast mesh generation approach and allows meshes of any desired number and order of elements.

4 | NUMERICAL RESULTS

Two numerical examples are considered herein, further examples are omitted for brevity. The first one is an L_2-projection on a domain with re-entrant angles, comparing the mesh generation and the pseudo-solid approach and highlighting the advantages of the proposed approach. The second example, a full FSI-simulation for a flap attached to a rigid block in a fluid flow, is a classic benchmark problem. The results for the mesh updating and mesh generation approaches are compared again.

4.1 | Study on interpolation properties of mesh update and mesh generation schemes

The initial domain is depicted in Figure 6A and an initial mesh is generated. The square cavity is rotated by $\alpha = 2°$ around the origin, comparing the mesh generation approach and the pseudo-solid approach, where a simple linear elastic material model is used. An L_2-projection of the function $f(x, y) = c \cdot \sin(f + m \cdot x - e \cdot y) + g \cdot \cos(l + w \cdot x + k \cdot y) + (a - x) \cdot (b - y)$, with $a, b, c, e, f, g, k, l, m, w \in [0, 1]$ on the deformed domain, is executed.

The L_2-error of this L_2-projection is calculated and the convergence plots are shown in Figure 7A for the pseudo-solid approach and in Figure 7B for the mesh generation approach for element order one to six and various resolutions. The convergence rates for linear and quadratic elements are optimal for both approaches. For higher-order elements, only the mesh generation approach leads to optimal convergence rates, whereas, for the pseudo-solid approach, the convergence rate does not increase with an increasing order of the elements. For the pseudo-solid approach, the elasticity equations are solved. As can be seen, for a domain with re-entrant angles, singularities may occur in the pseudo-solid approach which then spoils the accuracy and the convergence rates of the resulting meshes.

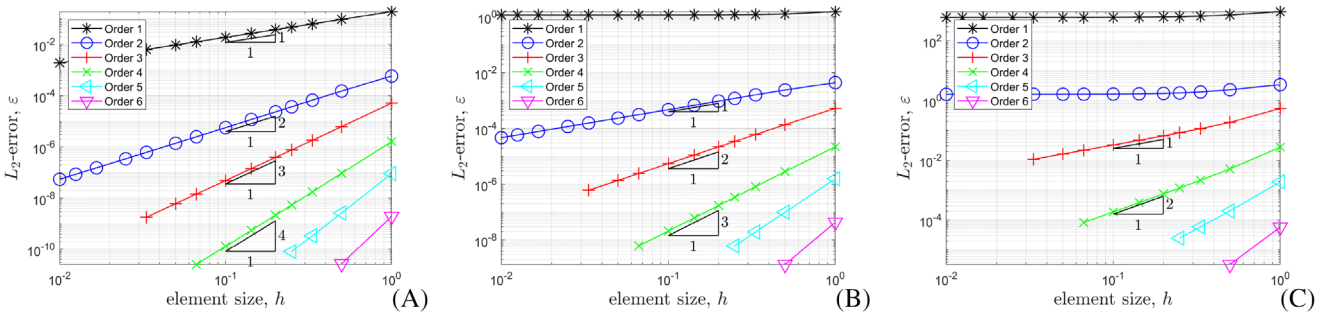

FIGURE 8 Convergence study—L_2-error of post-processed derivatives of L_2-projection: (A) first derivative; (B) second derivative; (C) third derivative.

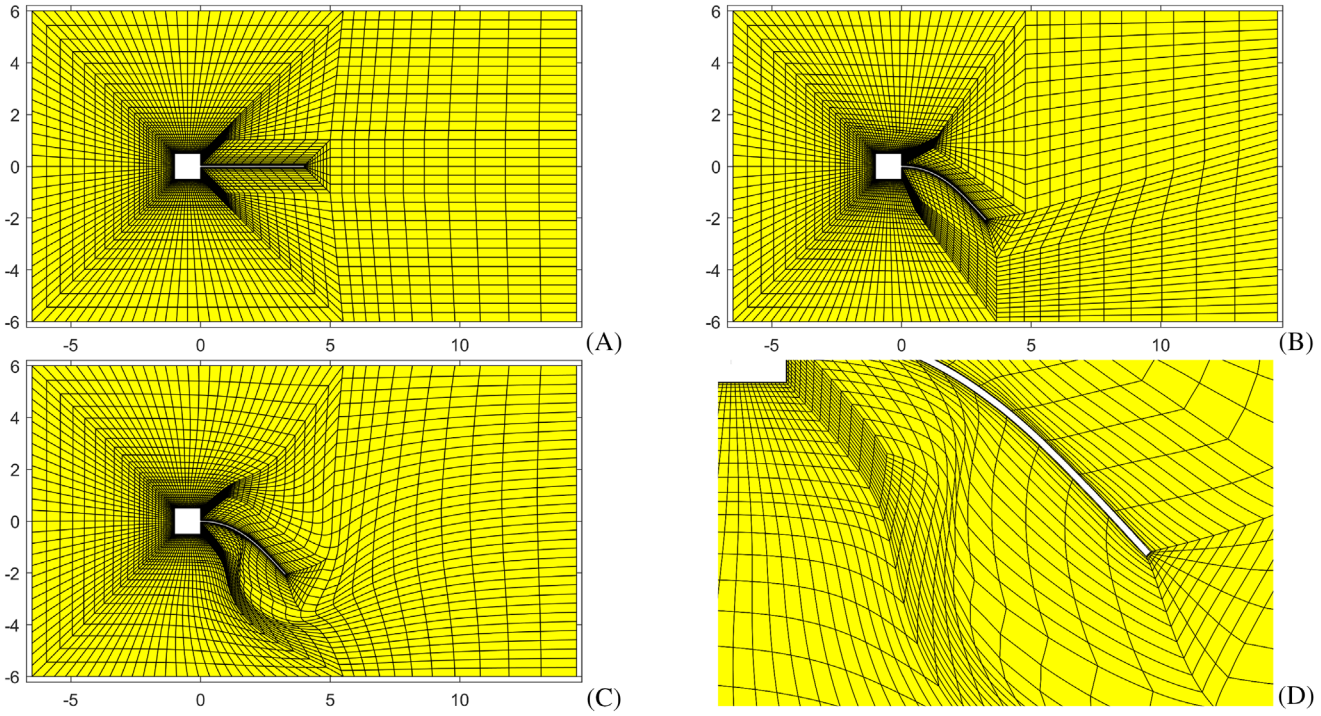

FIGURE 9 Flap: (A) initial mesh; (B) mesh after 1640 time steps for the coupled mesh generation approach; (C) mesh after 1640 time steps for the pseudo-solid approach; (D) zoom in on distorted elements of the pseudo-solid approach.

In a post-processing step, the derivatives of the L_2-projection are calculated, resulting in optimal convergence rates for the mesh generation approach for all derivatives, as seen in Figure 8 for the first, second, and third derivatives. This feature of optimal convergence rates in the proposed mesh generation highlights the advantages of the proposed approach, especially for higher-order elements. The optimal convergence for derivatives is particularly beneficial as derivatives are often needed, for example, to calculate stresses.

4.2 | FSI-application with moving flap behind square cylinder

As a benchmark problem in FSI, the flap behind a rigid block is considered using mesh generation and the pseudo-solid approach, and, later on, comparing mesh metrics. The full problem description may be found in [16] and is omitted here as the focus is on the treatment of the meshing herein.

A mesh for the initial domain is produced and shown in Figure 9A, and used as a starting mesh for both approaches. For the pseudo-solid approach, the linear elasticity equations are solved, where on the FSI-interface Γ_{FSI} the displacements of the solid and zero-displacements on all other boundaries are prescribed. For that approach, significant fine-tuning

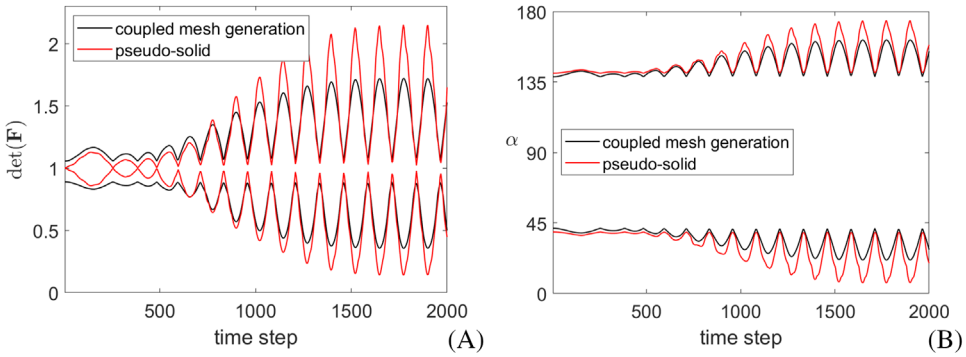

FIGURE 10 Mesh metrics: (A) maximal and minimal determinant of the deformation gradient; (B) maximal and minimal angle between neighboring elements.

of the pseudo-Young's moduli for all elements, depending on the size and position, is needed, to generate valid meshes throughout the simulation. In contrast, for the mesh generation approach, one needs to focus on providing a valid block-structure which is much simpler.

In Figure 9B the mesh after 1640 time steps for the mesh generation approach is depicted and in Figure 9C the mesh for the pseudo-solid approach at the same time step is shown. For the second mesh, squeezed elements, especially around the flap are arising and are shown in Figure 9D. To compare the two meshes with mesh metrics, the maximal and minimal determinant of the deformation gradient in the whole mesh over time is calculated, as well as the maximal and minimal angle between neighboring elements, shown in Figures 10A and B, respectively. Both metrics confirm that the mesh generation approach leads to significantly better results.

5 | CONCLUSIONS

We propose to inherently couple the mesh generation and the FSI-simulation. Instead of using mesh-update algorithms, a new mesh is generated after each time step, using the concept of advanced higher-order block-structured mesh generation. The mesh generation is based on a topology definition via a block-structure using conforming quads, geometry, and grading assigned to the block edges and the transfinite mapping function to generate nodes inside the blocks resulting in any desired number and order of the elements.

The generation of a suitable block-structure prior to the simulation is a topic on its own. For the moderately complex geometries common in many FSI simulations, this is not a problem and a broad range of applications may be covered. Future work may include the application of the approach in biomedical FSI simulations [15, 17]. The inherent coupling of FSI schemes with a mesh generator adds a level of complexity in the software design but leads to a very robust and versatile FSI framework, even enabling the use of higher-order meshes.

ORCID

Teresa Schwentner https://orcid.org/0009-0004-7947-2808
Thomas-Peter Fries https://orcid.org/0000-0003-1210-1557

REFERENCES

1. Stein, K., Tezduyar, T., & Benney, R. (2003). Mesh moving techniques for fluid-structure interactions with large displacements. *Journal of Applied Mechanics*, *70*, 58–63.
2. Batina, J. T. (1990). Unsteady Euler airfoil solutions using unstructured dynamic meshes. *AIAA Journal*, *28*, 1381–1388.
3. Helenbrook, B. T. (2003). Mesh deformation using the biharmonic operator. *International Journal for Numerical Methods in Engineering*, *56*(7), 1007–1021.
4. de Boer, A., van der Schoot, M. S., & Bijl, H. (2007). Mesh deformation based on radial basis function interpolation. *Computers & Structures*, *85*(11–14), 784–795.
5. Liu, X., Qin, N., & Xia, H. (2006). Fast dynamic grid deformation based on Delaunay graph mapping. *Journal of Computational Physics*, *211*(2), 405–423.

6. Lefrançois, E. (2008). A simple mesh deformation technique for fluid-structure interaction based on a submesh approach. *International Journal for Numerical Methods in Engineering*, *75*(9), 1085–1101.
7. Bartels, R. E. (2005). *Finite macro-element mesh deformation in a structured multi-block Navier-Stokes code* ((Report No. NASA/TM-2005-213789). National Aeronautics and Space Administration. https://ntrs.nasa.gov/citations/20050204125.
8. Ko, J. H., Park, S. H., Park, H. C., & Byun, D. (2010). Finite macro-element-based volume grid deformation for large moving boundary problems. *International Journal for Numerical Methods in Biomedical Engineering*, *26*(12), 1656–1673.
9. Bathe, K. J. (1996). *Finite Element Procedures*. Prentice-Hall.
10. Hughes, T. J. R., Liu, W. K., & Zimmermann, T. K. (1981). Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, *21*(3), 329–349.
11. Fries, T. P., Omerović, S., Schöllhammer, D., & Steidl J., (2017). Higher-order meshing of implicit geometries - part I: Integration and interpolation in cut elements. *Computer Methods in Applied Mechanics and Engineering*, *313*, 759–784.
12. Omerović, S., & Fries, T. P. (2017). Conformal higher-order remeshing schemes for implicitly defined interface problems. *International Journal for Numerical Methods in Engineering*, *109*(6), 763–789.
13. Gordon, W. J., & Hall, C. A. (1973). Transfinite element methods: Blending-function interpolation over arbitrary curved element domains. *Numerische Mathematik*, *21*, 109–129.
14. Gordon, W. J., & Hall, C. A. (1973). Construction of curvilinear co-ordinate systems and applications to mesh generation. *International Journal for Numerical Methods in Engineering*, *7*(4), 461–477.
15. Bošnjak, D., Pepe, A., Schussnig, R., Schmalstieg, D., & Fries, T. P. (2023). Higher-order block-structured hex meshing of tubular structures. *Engineering with Computers*, 1–21.
16. Wall, W. A. & Ramm, E. (1998). Fluid-Structure Interaction Based upon a Stabilized (ALE) Finite Element Method. *4th World Congress on Computational Mechanics - New Trends and Applications* (pp. 1–20). Buenos Aires, Argentina, (CIMNE, Barcelona).
17. Schussnig, R., Pacheco, D. R. Q., & Fries, T. P. (2022). Efficient split-step schemes for fluid-structure interaction involving incompressible generalised Newtonian flows. *Computers & Structures*, *206*, 106718.

How to cite this article: Schwentner, T., & Fries, T.-P. (2023). Fluid-structure interaction with fully coupled mesh generation. *Proceedings in Applied Mathematics and Mechanics*, *23,* e202300067. https://doi.org/10.1002/pamm.202300067