Engineering with Computers (2026) 42:69
https://doi.org/10.1007/s00366-026-02299-0

ORIGINAL ARTICLE

Isogeometric suitable coupling methods for partitioned multiphysics simulation with application to fluid–structure interaction

Jing-Ya Li[1] · Hugo M. Verhelst[2] · Henk den Besten[3] · Matthias Möller[1]

Received: 3 October 2025 / Accepted: 27 February 2026

Abstract

This paper presents spline-based coupling methods for partitioned multiphysics simulations, specifically designed for isogeometric analysis (IGA) based solvers. Traditional vertex-based coupling approaches face significant challenges when applied to IGA solvers, including geometric accuracy issues, interpolation errors, and substantial communication overhead. The methodology draws on the IGA mathematical framework to deliver coupling solutions that preserve the high-order continuity and exact geometric representation of splines. We develop two complementary strategies: (1) a spline-vertex coupling method that enables efficient interaction between IGA and conventional solvers, and (2) a fully isogeometric coupling approach that maximizes accuracy for IGA-to-IGA communication. Both theoretical analysis and extensive numerical experiments demonstrate that our spline-based methods significantly reduce communication overhead compared to traditional approaches while simultaneously enhancing geometric accuracy through exact boundary representation and maintaining higher-order solution continuity across the coupled interfaces. We quantitatively confirm the communication efficiency benefits through systematic measurements of both transfer times and data volumes across various mesh refinement levels, with experimental results closely aligning with our theoretical predictions. Our benchmark studies further demonstrate the geometric fidelity advantages through exact boundary representation, while also highlighting how the inherent mathematical structure of splines naturally preserves solution derivatives across interfaces without requiring additional computation or specialized transfer algorithms. This work not only provides efficient coupling strategies tailored to IGA-based solvers but also establishes a practical bridge between IGA and traditional discretization methods in partitioned multiphysics simulations. By offering viable options for coupling conventional solvers with IGA-based components, our approach enables broader adoption of IGA in established simulation workflows while ensuring accurate and high-performance interface communications.

Keywords Isogeometric analysis · Fluid–-structure interaction · Interface coupling · Splines · Partitioned multiphysics

1 Introduction

Fluid–-structure interaction (FSI) remains a fundamental challenge in multiphysics simulations, governing a wide range of engineering applications, such as the aerodynamic response of aircraft wings, the pulsatile behavior of cardiovascular systems, and the structural resilience of bridges and offshore structures under hydrodynamic loading. These problems exhibit inherently bidirectional coupling, where the fluid exerts pressure on the structure, causing deformations that, in turn, modify the fluid domain, creating a nonlinear and dynamic feedback loop. Ensuring numerical stability, accuracy, and efficiency in such simulations is especially challenging when heterogeneous discretizations

✉ Jing-Ya Li
j.li-9@tudelft.nl

[1] Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628CD Delft, The Netherlands

[2] Department of Civil Engineering and Architecture, University of Pavia, Via Adolfo Ferrata 3, 27100 Pavia, PV, Italy

[3] Maritime and Transport Technology Department, Delft University of Technology, Mekelweg 2, 2628CD Delft, The Netherlands

Published online: 22 March 2026

are utilized for the fluid and structural solvers, requiring robust interface coupling approaches [1–3].

1.1 Partitioned methods for multiphysics problems

FSI problems are typically tackled via one of the two following solution strategies: monolithic and partitioned formulations. The monolithic approach integrates fluid and structural solvers into a single, coupled system, yielding superior numerical stability and convergence behavior [4] but at the cost of flexibility, often necessitating tightly integrated code bases and customized solvers. In contrast, partitioned methods decouple the problem into distinct subdomains, allowing fluid and structural equations to be solved independently and with different solvers [5]. These subproblems communicate through the iterative exchange of boundary data across the interface. The modularity provided by partitioned schemes promotes solver reuse, algorithmic specialization, and compatibility with legacy software [6, 7]. Moreover, partitioned formulations are often favored over monolithic ones because they offer the flexibility to couple pre-existing solvers, facilitating the integration of specialized codes developed for each physical domain [8]. However, this same modularity introduces significant challenges: ensuring numerical stability, enforcing kinematic and dynamic consistency, and preserving accuracy across potentially non-conforming meshes are nontrivial tasks [9–11]. At the same time, a number of mature approaches successfully address these issues on non-matching interfaces, including (dual) mortar methods [12, 13], common-refinement based conservative/consistent transfer [7, 10], Nitsche-type formulations, and black-box mesh morphing with RBFs [14, 15]. We position our contribution relative to these by leveraging spline spaces at the interface to achieve smoothness, accuracy, and compact data exchange.

Partitioned methods involve two main coupling aspects: a temporal and spatial one. Temporal coupling controls how and when data are exchanged between the fluid and structural solvers. This can range from simple staggered approaches to more advanced methods, such as fixed-point or quasi-Newton iterations that improve stability and accuracy [16]. Spatial coupling, on the other hand, deals with how data is transferred across the interface, especially when the fluid and structural meshes do not match. This usually involves interpolation or projection between different meshes [7, 9, 14]. It is difficult to achieve both high accuracy and good computational performance at the same time. The conventional approach of increasing mesh resolution to improve accuracy directly conflicts with the need to minimize communication overhead, leading to diminishing returns as the size of the problem increases. This challenge is further compounded by the need to balance efficient data exchange rates with parallel execution capabilities [17], particularly in demanding applications such as marine and offshore engineering where large-scale flexible FSI simulations are increasingly critical [18].

1.2 Spline-based coupling: a paradigm shift

Isogeometric Analysis (IGA), pioneered by Hughes et al. [19], presents a powerful alternative to traditional finite element approaches by unifying geometry representation and solution approximation. In IGA, the same mathematical basis functions (typically Non-Uniform Rational B-Splines (NURBS) or B-splines) are used to define both the computational domain and the solution field [20]. This seamless integration enables exact geometric representation, even on coarse meshes, and supports higher-order continuity across elements: features especially advantageous for problems dominated by interface behavior.

Partitioned FSI methods have gained widespread adoption over the past decades due to their flexibility in leveraging specialized solvers for different physical domains. However, a persistent challenge in this context lies in the accurate and efficient transfer of data across the fluid–structure interface, particularly when nonmatching spatial discretizations are involved [21]. Spline-based coupling techniques, especially those employing NURBS or B-splines, have shown significant potential to address these issues. For example, Hosters et al. [1] first developed a NURBS-enhanced FSI framework using spline descriptions on both fluid and structure sides, enabling accurate and direct data exchange. Kamensky et al. [22] introduced a stabilized immersogeometric formulation, showing improved stability properties when using collocated constraints across thin, deforming interfaces.

Further developments in spline technology have expanded the scope of isogeometric coupling. Floating IGA (FLIGA), introduced by Hille et al. [23], enables analysis under extreme deformation by adapting basis functions along deformation-dependent directions. Similarly, Rosa et al. [24] proposed a blended IGA-FEM approach for modeling fracture propagation without remeshing, further illustrating the adaptability of spline-based representations to evolving geometries. Guarino et al. [25] presented a novel penalty-based coupling scheme for trimmed non-conformal shell patches, relevant for industrial applications where surface continuity and physical correctness must be preserved.

As the complexity of FSI problems increases, particularly in scenarios involving large structural deformations or compressible flows, robust coupling strategies become even more critical. While spline-based methods offer smooth and accurate interface representations, recent advances in finite element-based coupling have tackled these challenges from a different perspective. Rajanna et al. [26] proposed

a compressible FSI framework using nonmatching finite element meshes, suitable for high-speed flow applications. Klöppel et al. [13] developed a dual mortar formulation to improve force and displacement consistency across nonconforming FEM interfaces. These contributions are important within the FEM paradigm, though they differ fundamentally in how geometry and continuity are handled compared to spline-based methods.

Despite advances in interface accuracy and stability, the issue of computational efficiency remains unresolved, particularly in large-scale, strongly coupled simulations. Data interpolation between nonmatching meshes can become a performance bottleneck. Radial Basis Function (RBF) methods have gained traction as black-box interpolants for multiphysics applications, as seen in the work of Lindner et al. [27] and its parallelized extension by Schneider et al. [15]. However, while RBFs offer flexibility, they may lack the conservation and variational structure desired in high-fidelity IGA settings. De Boer et al. [28] highlighted that consistent interpolation often outperforms conservative approaches in terms of accuracy and stability, particularly for quasi-1D FSI problems. In contrast, spline-based coupling inherently supports both consistency and high-order smoothness through function-space projection, offering a more principled route for interface treatment.

In contrast to existing point-based and spline-based coupling methods that mainly interpolate interface data at discrete points, this paper proposes a fundamentally new approach for partitioned FSI simulations by directly expressing interface fields as smooth, continuous spline functions. The key innovation lies in directly utilizing the inherent advantages of IGA to represent the interface as a continuous spline geometry, rather than as isolated interpolation points. In the proposed framework, both the structural and fluid participants maintain their own spline-defined interface fields, allowing for a consistent and symmetric field-to-field coupling. This shift in paradigm offers several compelling advantages: it preserves geometric fidelity for complex curvilinear boundaries, reduces communication overhead via compact control point exchange, improves numerical stability due to high-order continuity, and inherently handles nonmatching meshes through direct value evaluation, eliminating the need for ad hoc interpolation schemes.

The proposed methodology treats the interface as a mathematically structured entity rather than a numerical artifact. This approach maintains physical consistency while achieving high accuracy with reduced computational cost, effectively addressing the longstanding accuracy-efficiency trade-off in spatial coupling for partitioned FSI systems.

1.3 Research contributions and paper structure

This work introduces key contributions to multiphysics simulation. While the methods are broadly applicable to general multiphysics problems, we focus without loss of generality on fluid–structure interaction as a representative application throughout this study.

- A theoretical framework for interface coupling in partitioned FSI based on IGA, specifically addressing challenges with non-matching meshes while maintaining accuracy and computational efficiency.
- Two coupling strategies: (i) a hybrid approach that combines IGA-based structural solvers with traditional fluid solvers, and (ii) a fully isogeometric method for high-accuracy IGA-to-IGA coupling.
- An analysis of communication overhead, showing that our spline-based method significantly reduces communication cost, especially with finer meshes.
- Verification and validation using standard FSI benchmarks, demonstrating a strong balance between accuracy and efficiency.
- Implementation as an open-source software package integrated into the preCICE multiphysics framework, featuring a modular plug-and-play design compatible.

The remainder of the paper is organized as follows. Section 2 introduces the governing equations used in this study. Section 3 provides an overview of IGA, which serves as the spatial discretization method for the structural domain in this work. Section 4 explains the numerical methods for both fluid and structural solvers and presents our spline-based coupling approach. Section 5 provides verification and validation results to demonstrate the accuracy of our method. The concluding Sect. 8 ends this study, summing up the main results and opening up for further research.

The methodology presented in this work has been implemented in `gsPreCICE`, developed as part of this research. As the first dedicated IGA solver adapter for the preCICE (Precise Code Interaction Coupling Environment) [8] framework, `gsPreCICE` is open-source and officially integrated as a module within the `G+Smo` (Geometry + Simulation Modules) library [29–31], which is a C++ library dedicated to IGA.

2 Governing equations

Partitioned algorithms tackle the distinct challenges posed by fluid dynamics and structural mechanics by separately solving each domain and exchanging data at a shared interface. While such approaches are broadly applicable to a

wide range of multiphysics problems, the scope here is limited to FSI, involving the coupling of incompressible fluid flow with nonlinear elastic structural deformation in multi-dimensional spaces.

In this section, we briefly review the governing equations of incompressible fluid flow and nonlinear elasticity formulated in a total Lagrangian framework used throughout the paper. Clearly, restating these equations here serves two purposes: (a) it establishes a consistent notation that is essential for accurately describing our numerical methods and coupling strategies, and (b) it provides context for subsequent discussions on interface conditions and numerical discretizations.

2.1 Incompressible flow

The fluid dynamics within the domain are governed by the Navier–Stokes equations for incompressible flow, outlined as follows:

$$\rho^f \left(\frac{\partial \mathrm{u}^f}{\partial t} + (\mathrm{u}^f \cdot \nabla)\mathrm{u}^f \right) = -\nabla p^f + \nabla \cdot (\mu^f \nabla \mathrm{u}^f) + \mathrm{f}^f \quad \text{in } \Omega^f \times [0,T], \tag{1a}$$

$$\nabla \cdot \mathrm{u}^f = 0 \quad \text{in } \Omega^f \times [0,T], \tag{1b}$$

$$\mathrm{u}^f = \mathrm{g}^f \quad \text{on } \Gamma^f_D \times [0,T], \tag{1c}$$

$$(\mu^f \nabla \mathrm{u}^f - p^f \mathrm{I}) \cdot \mathrm{n}^f = \mathrm{h}^f \quad \text{on } \Gamma^f_N \times [0,T]. \tag{1d}$$

Within these equations, ρ^f, μ^f, and f^f represent the fluid's density, dynamic viscosity, and external body force vector, respectively. Γ^f_D and Γ^f_N denote the Dirichlet and Neumann boundaries of the fluid domain Ω^f. This set of equations, (1a) through (1d), is utilized to solve for the fluid's velocity field $\mathrm{u}^f(\mathrm{x},t)$ and pressure $p^f(\mathrm{x},t)$ at any given time $t \in [0,T]$.

2.2 Structure deformation

The solid displacement field, $\mathrm{d}^s(\mathrm{x},t)$, which responds to external forces, is governed by Newton's second law of motion, as expressed below:

$$\frac{d^2\mathrm{d}^s}{dt^2} - \frac{1}{\rho^s_0} \boldsymbol{\nabla}_0 \cdot \left(\mathrm{S}^s \mathrm{F}^T \right) = \mathrm{b}^s \quad \text{in } \Omega^s_0 \times [0,T] \tag{2a}$$

Here, ρ^s_0 represents the solid's density and b^s denotes the external body force acting on it. This equation is derived from a total Lagrangian framework and is hence formulated with respect to the unstrained reference configuration Ω^s_0. All quantities and operators in this formulation are referenced with a subscript '0' to indicate this baseline state. Unlike the Cauchy stress tensor T^s, the internal stress here is expressed using the second Piola-Kirchhoff stress tensor $\mathrm{S}^s = \det(\mathrm{F})\mathrm{F}^{-1}\mathrm{T}^s\mathrm{F}^{-T}$, where F is the deformation gradient. In this work, we adopt the Saint-Venant-Kirchhoff (SVK) material model with a linear stress–strain relation $\mathrm{S} = \mathbb{C} : \mathrm{E}$. The overall nonlinear structural response arises from geometric nonlinearity in the Green-Lagrange strain $\mathrm{E} = \frac{1}{2}(\mathrm{F}^T\mathrm{F} - \mathrm{I})$, which is essential for large deformations. The problem formulation is completed with initial zero displacement conditions and appropriate boundary constraints on $\partial\Omega^s$. The boundary $\partial\Omega^s_0$ is partitioned into two complementary regions: a Dirichlet boundary Γ^s_D where displacement is prescribed, and a Neumann boundary Γ^s_N where traction is prescribed. Here, n_0 denotes the unit outward normal vector to the reference configuration, g^s is the prescribed displacement boundary vector, and h^s is the prescribed traction boundary vector. The boundary conditions are expressed as:

$$\mathrm{d}^s = \mathrm{g}^s \quad \text{on } \Gamma^s_D \times [0,T] \tag{3a}$$

$$\mathrm{F}\mathrm{S}^s\mathrm{n}_0 = \mathrm{h}^s \quad \text{on } \Gamma^s_N \times [0,T] \tag{3b}$$

2.3 Coupling conditions

In the partitioned approach, the fluid problem and the structure problem are solved independently, potentially by different solvers. For consistent coupling across the shared interface $\Gamma^{fs} = \Gamma^f \cap \Gamma^s$, the interface solutions of both subproblems must fulfill kinematic continuity requirements:

$$\mathrm{d}^f = \mathrm{d}^s \quad \text{on } \Gamma^{fs} \times [0,T] \tag{4a}$$

$$\mathrm{u}^f = \mathrm{u}^s \quad \text{on } \Gamma^{fs} \times [0,T] \tag{4b}$$

where d^f and u^f represent the displacement and velocity of the moving fluid boundary, while d^s and u^s are those of the structure.

The implied accelerations are equivalent too. And the dynamic continuity for the equilibrium of stress:

$$\mathrm{T}^f \cdot \mathrm{n}^f = -\mathrm{T}^s \cdot \mathrm{n}^s \quad \text{on } \Gamma^{fs} \times [0,T] \tag{5}$$

where n^f and n^s are the interface unit normal vectors pointing outwards from the corresponding domains. Note that $\mathrm{n}^f = -\mathrm{n}^s$ on the interface. While the kinematic constraints guarantee displacement and velocity continuity at the interface, the dynamic equilibrium condition maintains stress balance according to Newton's third law (action-reaction principle). Together, these interface conditions preserve the

physical consistency required for accurate fluid–structure interaction modeling [3].

In our partitioned framework, the fluid solver computes tractions $\mathrm{t}^f = \mathrm{T}^f \cdot \mathrm{n}^f$ on the current (deformed) interface. Since the structural formulation employs a total Lagrangian approach, these tractions must be pulled back to the reference configuration. Using Nanson's formula ($\mathrm{n}\, da = J\mathrm{F}^{-T}\mathrm{n}_0\, dA$), the equivalent traction on the reference configuration becomes:

$$\mathrm{t}_0^s = J^s(\mathrm{F}^s)^{-1} \cdot \mathrm{t}^f \quad \text{on } \Gamma_0^{fs} \tag{6}$$

where $J^s = \det(\mathrm{F}^s)$ is the Jacobian determinant of the structural deformation gradient. This transformation ensures that the resultant force is preserved between configurations, i.e., $\int_{\Gamma^{fs}} \mathrm{t}^f\, da = \int_{\Gamma_0^{fs}} \mathrm{t}_0^s\, dA$, thereby maintaining physical consistency for large deformations.

3 Isogeometric analysis overview

In this work, we adopt a partitioned coupling approach, treating fluid and structure as separate subproblems that are solved independently. For the structural subproblem, we employ IGA, which offers both geometric exactness and higher continuity in the solution space.

IGA, introduced by Hughes et al. [19], enhances geometric accuracy and promotes closer integration between CAD models and numerical analysis by using the same basis functions for geometry representation and solution approximation. Typically, CAD geometries are represented by NURBS or b-splines, defined by a set of control points and associated rational basis functions. A NURBS surface can be mathematically expressed as:

$$\bar{\mathrm{x}}(\boldsymbol{\xi}) = \sum_{i=1}^{n_c} R_i^p(\boldsymbol{\xi})\mathrm{x}_i, \quad \text{with } \boldsymbol{\xi} = (\xi^1, \xi^2), \tag{7}$$

where n_c is the number of control points, $\mathrm{x}_i \in \mathbb{R}^d$ represents the control points, $R_i^p(\boldsymbol{\xi})$ are rational basis functions of polynomial degree p defined on the parameter domain $\boldsymbol{\Xi} \subset \mathbb{R}^d$, and $d \in \{2, 3\}$ is the spatial dimension.

IGA provides an ideal framework for our coupling methods because it offers smooth and continuous representations of solution fields across the interface, which are critical for stable and accurate multiphysics coupling. In addition, it enables precise geometric descriptions, which help avoid approximation errors in curved or evolving interfaces.

4 Numerical methods

In partitioned FSI simulations, the fluid and structure subproblems are solved using separate solvers, each tailored to its respective physical field. This modular strategy offers considerable flexibility: established solvers can be reused or further developed independently, without requiring changes to the overall framework [8].

At the interface between fluid and structure, the two solvers often use different meshes and sometimes even different types of numerical methods. Because of this, we need a reliable way to transfer data (like forces and displacements) between them in a way that is accurate and physically consistent. One of the main contributions of this paper is a spline-based coupling method, which is especially effective when combined with IGA. This method helps maintain smooth and consistent data transfer across nonmatching meshes.

Building on this motivation, the remainder of this section focuses on spatial coupling strategies that bridge the fluid and structural solvers, with particular emphasis on the proposed spline-based methods that support the consistency and efficiency of the partitioned framework.

4.1 Evolution of spatial coupling schemes: from vertex-based to spline-based methods

Spatial coupling schemes are critical components in partitioned FSI simulations, determining the manner and accuracy of data exchange at the interface. We consider three representative approaches: vertex-vertex, spline-vertex, and spline-spline coupling, that range from conventional point-based methods to the proposed spline-based formulations.

4.1.1 Vertex-based coupling method: traditional approach and its limitations

The vertex-based coupling method facilitates communication between an IGA-based structural solver and traditional solvers based on discretization techniques such as finite volume (FV) or finite element methods (FEM). In this approach, the data exchanged between solvers is defined at discrete points along the interface, typically mesh vertices or quadrature points (Fig. 1).

The vertex-based coupling procedure is performed in three main steps:

(a) *Data mapping for the discrete forces:* Vertex-based coupling requires discrete stress components at each fluid node along the FSI interface Γ^{fs}. To transfer these stresses between nonmatching meshes, a range of mesh-to-mesh interpolation techniques are available,

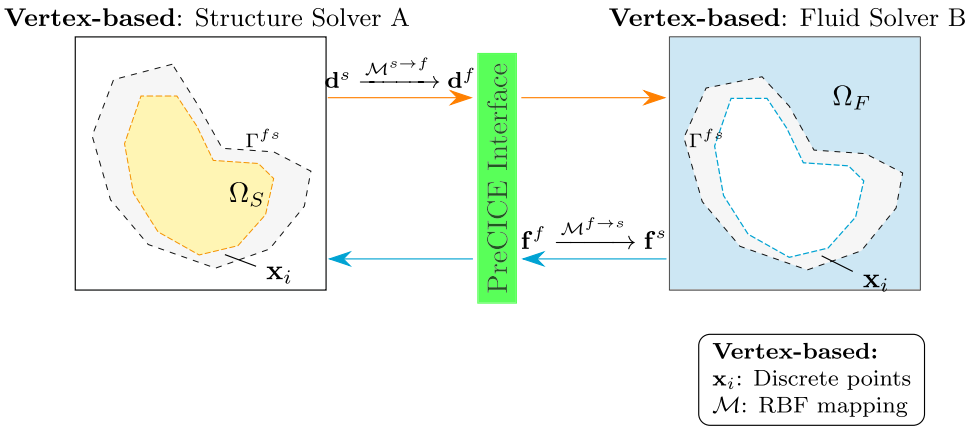


Fig. 1 Illustration of vertex-based solvers coupled with IGA-based solvers, with information exchanged at quadrature points/vertices. Shaded region (in grey) depicts the FSI interface. Arrows indicate the direction of data exchange across the interface

including nearest-neighbor, linear projection, and mortar methods. In this work, we use RBF interpolation, implemented via the preCICE library [15, 27], due to its flexibility, smoothness, and robustness for scattered, non-conforming interface meshes. Compared to nearest-neighbor or linear methods, RBF interpolation offers higher accuracy and smoother field transfer without requiring mesh alignment or auxiliary mesh constructions [15]. This enables accurate interpolation of stress values from the fluid mesh Γ^f to the structural mesh Γ^s. Each interface mesh is defined by a set of vertices $\{\mathrm{x}_i \in \mathbb{R}^d \,|\, i = 1, \ldots, N\}$ in $d \in \{2, 3\}$ spatial dimensions. Let $\{\boldsymbol{\sigma}_i \in \mathbb{R}^m \,|\, i = 1, \ldots, N\}$ denote the stress tensor or vector values (with $m = d$ or $m = d(d+1)/2$ depending on tensor representation) at the fluid mesh vertices. For RBF interpolation, each stress component is treated as a scalar field. Given the k-th stress component $\{\sigma_{i,k} \in \mathbb{R} \,|\, i = 1, \ldots, N\}$ at the input vertices, the interpolant $\mathcal{I}_{\sigma_k} : \mathbb{R}^d \to \mathbb{R}$ for the k-th component is defined as:

$$\mathcal{I}_{\sigma_k}(\mathrm{x}) = \sum_{i=1}^{N} \lambda_{i,k}\, \phi\left(\|\mathrm{x} - \mathrm{x}_i\|\right) + \beta_{0,k} + \boldsymbol{\beta}_{l,k}^T \mathrm{x}, \tag{8}$$

where N is the number of input vertices, $\lambda_{i,k} \in \mathbb{R}$ are the RBF coefficients, $\phi : \mathbb{R} \to \mathbb{R}$ is the radial basis function, and $\beta_{0,k} \in \mathbb{R}$, $\boldsymbol{\beta}_{l,k} \in \mathbb{R}^d$ are coefficients of a linear polynomial ensuring consistency for linear stress fields. Specifically, we employ Compact Polynomial Radial Basis Functions (Compact Poly-RBFs), which provide excellent smoothness properties with C^6 continuity. The C^6 continuity of these basis functions ensures that higher-order derivatives are continuous, thereby avoiding spurious oscillations that can occur with globally supported RBF kernels. This property is particularly important for transferring sensitive quantities like stress components, where ill-conditioned interpolation matrices can produce non-physical oscillations. The interpolation conditions are enforced as:

$$\mathcal{I}_{\sigma_k}(\mathrm{x}_i) = \sigma_{i,k}, \quad i = 1, \ldots, N, \tag{9}$$

with regularization constraints to guarantee uniqueness:

$$\sum_{i=1}^{N} \lambda_{i,k} = 0 \quad \text{and} \quad \sum_{i=1}^{N} \lambda_{i,k} \mathrm{x}_i = 0. \tag{10}$$

(b) *Transfer of forces onto the structure:* The process is repeated for all stress components via preCICE. The interpolant $\mathcal{I}$ can yield a matrix $\mathcal{M}^{f\to s}$ defining a linear mapping of the input forces f^f from the fluid mesh to forces f^s in the structural mesh:

$$\mathrm{f}^s = \mathcal{M}^{f\to s} \mathrm{f}^f \quad \text{with} \quad \mathrm{f}^f = \begin{pmatrix} \mathrm{f}^f(\mathrm{x}_1) \\ \mathrm{f}^f(\mathrm{x}_2) \\ \vdots \\ \mathrm{f}^f(\mathrm{x}_N) \end{pmatrix} \tag{11}$$

When the row-sum of $\mathcal{M}^{f\to s}$ is equal to one, the interpolation is consistent, which means that constant values are interpolated exactly. On the other hand, when the column-sum of $\mathcal{M}^{f\to s}$ equals one the interpolation is conservative.

(c) **Transfer of deformation onto the fluid:** The computed structural deformation has to be transferred back to the fluid in order to account for the mesh deformation. Once again, this procedure is straightforward due to the use of the RBF method in preCICE. The deformation is computed at every quadrature point of the structural element:

$$\mathrm{d}^f = \mathcal{M}^{s\to f}\mathrm{d}^s \quad \text{with} \quad \mathrm{d}^s = \begin{pmatrix} \mathrm{d}^s(\mathrm{x}_1) \\ \mathrm{d}^s(\mathrm{x}_2) \\ \vdots \\ \mathrm{d}^s(\mathrm{x}_n) \end{pmatrix} \tag{12}$$

where d^s represents the structural displacements at discrete structural points x_i, and d^f represents the fluid mesh displacements at discrete fluid mesh points. $\mathcal{M}^{s\to f}$ represents the interpolation operator, which maps the structural displacement onto the fluid mesh on the IGA mesh cell level.

Although vertex-based methods are widely used due to their simplicity and compatibility with standard finite element frameworks, they face several significant challenges when dealing with complex interface dynamics [10, 11]:

1. Geometric accuracy issues, especially for complex or curved interfaces [32].
2. Interpolation errors that can lead to numerical inaccuracies in force and displacement transfer processes.
3. Significant increase in communication overhead as interface meshes are refined.

These limitations motivate the development of more accurate and efficient spline-based coupling methods, which are introduced in this paper.

4.2 Spline-based coupling method

The spline-based coupling method leverages analytical spline functions to represent the interface, providing a more efficient and accurate approach for data exchange between solvers. Unlike vertex-based methods that operate on discrete points, spline-based coupling utilizes a continuous mathematical description of both interface geometry and field variables.

We propose two types of spline-based coupling methods:

- Spline-vertex coupling method: one solver is IGA-based.
- IGA-IGA coupling method: both solvers are IGA-based.

4.2.1 Spline-vertex coupling method

We first consider the spline-vertex coupling method, which is designed for scenarios in which only one side of the interface (typically the structure) is modeled using IGA, while the opposing side (e.g., the fluid) employs a standard finite element or finite volume discretization. In this case, the interface geometry is represented using a spline on the IGA side, and data is transferred between the spline surface and the vertex points of the non-IGA mesh using projection or interpolation techniques (Fig. 2).

The coupling procedure consists of two key steps:

- **(a) Force transfer (fluid to structure):** Force data from the fluid solver is transferred to the structure using a RBF interpolation method:

$$\mathrm{f}^s = \mathcal{M}^{f\to s}\mathrm{f}^f \quad \text{with} \quad \mathrm{f}^f = \begin{pmatrix} \mathrm{f}^f(\mathrm{x}_1) \\ \mathrm{f}^f(\mathrm{x}_2) \\ \vdots \\ \mathrm{f}^f(\mathrm{x}_N) \end{pmatrix} \tag{13}$$

This step maps the discrete fluid force data to the structural solver's representation.

- **(b) Displacement transfer (structure to fluid):** The structural displacement field, computed in terms of control point displacements d_i^s, is evaluated at any interface point using:

$$\mathrm{d}^s(\boldsymbol{\xi}) = \sum_{i=1}^{n_c} R_i^p(\boldsymbol{\xi})\mathrm{d}_i^s \tag{14}$$

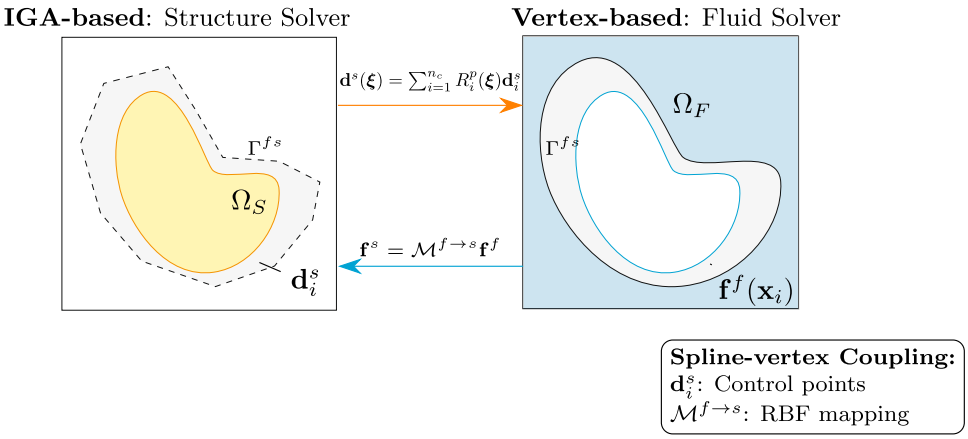


Fig. 2 Spline-based coupling between IGA-based structural solver and vertex-based fluid solver, with interface data exchanged via continuous spline representation

The fluid solver samples this continuous field at required interface points to update its mesh.

4.2.2 IGA-IGA communication: optimal spline-based coupling

When both fluid and structural domains use isogeometric analysis or support spline representation (of solution/geometry), the IGA-IGA communication approach fully leverages spline representations on both sides of the interface, representing the most efficient coupling strategy within the spline-based family (Fig. 3).

The communication procedure involves:

(a) *Interface parameterization:* Both domains ideally share a common parametric representation for direct control point data exchange. When parameterizations differ, reparameterization through knot vector exchange and interpolation creates conforming interface splines. This mapping is computed once during initialization for stationary interfaces.

(b) *Stress transfer via control points:* Forces are directly expressed in terms of control point values. When reparameterization in step (a) introduces new nodes, force and stress information is evaluated at these newly inserted nodes to enable accurate data transfer:

$$\mathrm{f}_c^s = \mathcal{T}_{f\to s}\mathrm{f}_c^f \tag{15}$$

where f_c^s and f_c^f are the control point force vectors for the structural and fluid domains, respectively, and $\mathcal{T}_{f\to s}$ is the transformation operator established during reparameterization.

(c) *Displacement transfer via control points:* Similarly, structural displacements computed at control points transfer directly or through transformation:

$$\mathrm{d}_c^f = \mathcal{T}_{s\to f}\mathrm{d}_c^s \tag{16}$$

where d_c^s and d_c^f are the control point displacement vectors for the structural and fluid domains, respectively, and $\mathcal{T}_{s\to f}$ is the transformation operator mapping between the two spline spaces.

This approach provides minimal communication overhead, exact geometry representation, preservation of higher-order solution properties, enhanced numerical stability, and efficient handling of complex geometries. We will further discuss the advantage of reduced communication overhead in Sect. 5.

5 Coupling methods verification and validation

In this section, we present a comprehensive verification and validation of the spatial coupling methods described previously. The verification procedure employs both the constant load and nonlinear load test cases to assess the accuracy, efficiency, and stability of the proposed coupling approaches.

We first consider a vertical beam that is clamped at its base and subjected to a constant distributed load along its free edge. The setup ensures that the force transmission is uniform and can be analytically validated.

The problem setup, including geometry and loading conditions, is shown in Fig. 4. We compare the tip displacement computed by the standalone structural solver (reference solution) to the coupled solver solutions using vertex-based and spline-based coupling methods.

The structural solver computes the initial beam deformation under the applied load. The force distribution is mapped to the fluid mesh using the vertex-based and spline-based methods described earlier. The reconstructed forces are transferred back to the structure to verify accuracy in communication. Finally, the resulting tip displacement is compared with the theoretical solution, which is the monolithic application of the force along the beam within one solver.

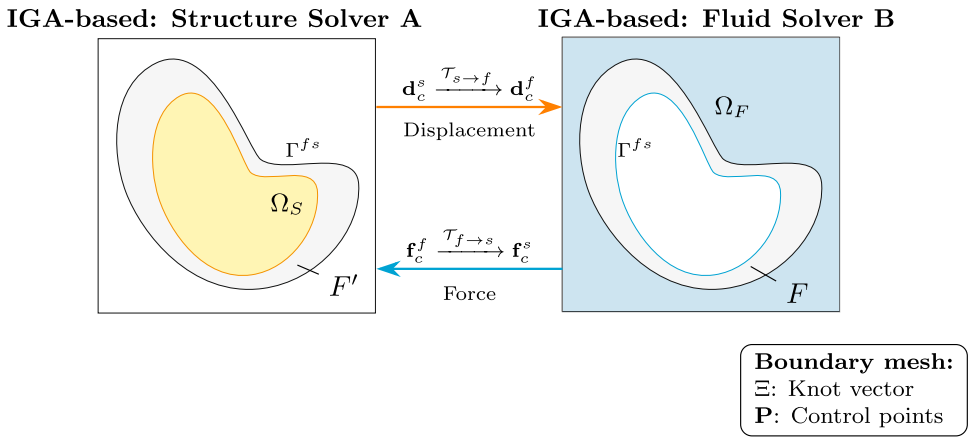


Fig. 3 IGA-IGA communication with spline representations on both fluid and structural domains, maximizing the benefits of isogeometric analysis

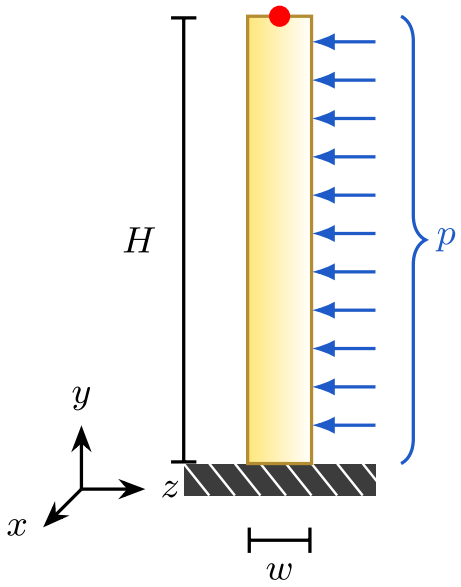


Fig. 4 Constant load test setup for the vertical beam

Parameter	Value
Width (W)	0.50 m
Height (H)	1.00 m
Thickness (t)	0.05 m
Density (ρ)	$3000\,\mathrm{kg/m^3}$
Distributed Load (p)	$5 \times 10^3 \mathbf{e}_z$ Pa

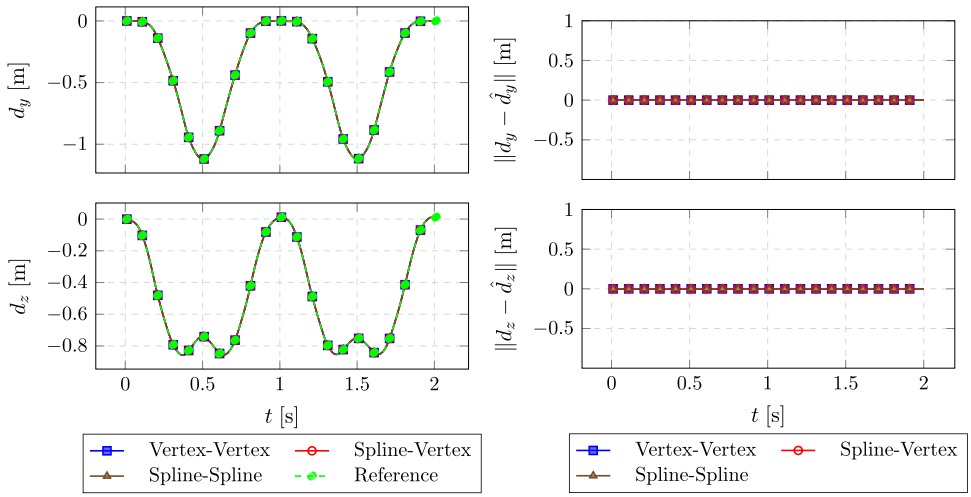


Fig. 5 Left: Comparison of tip displacement between reference solution and different coupling methods. Right: Relative L^2 error of different coupling methods compared to the reference solution

All results remain below machine precision ($\sim 10^{-15}$)

Figure 5 compares the vertical beam tip displacements obtained via different coupling strategies against the monolithic reference solution (left), and the corresponding L^2 -norm errors in the y- and z-displacement directions (right). All spline-based methods (spline-vertex and spline-spline) yield errors below machine precision ($\sim 10^{-15}$), indicating perfect agreement with the reference. The vertex-vertex method achieves the same level of accuracy in this constant load scenario. This demonstrates the consistency of all tested coupling approaches.

To further validate the accuracy and robustness of the coupling methods, we designed a nonlinear load test case. In this scenario, the vertical beam is subjected to a nonlinear load $\hat{p}(\xi) = -\xi \mathrm{e}_z$ that varies with parameter ξ.

Figure 6 illustrates the nonlinear load test setup. In this test, we compare the reference solution (direct calculation by a monolithic solver) with solutions obtained using different coupling methods (vertex-based and spline-based) under identical load conditions.

Figure 7 shows the tip displacement over time under a nonlinear load. The spline-spline based, spline-vertex based, and vertex-vertex based coupling methods yield consistent results under complex loading conditions.

6 Communication overhead comparison and analysis

In partitioned multiphysics simulations, communication overhead is a critical metric for evaluating coupling method efficiency. We conducted both theoretical analysis and experimental validation of communication overhead

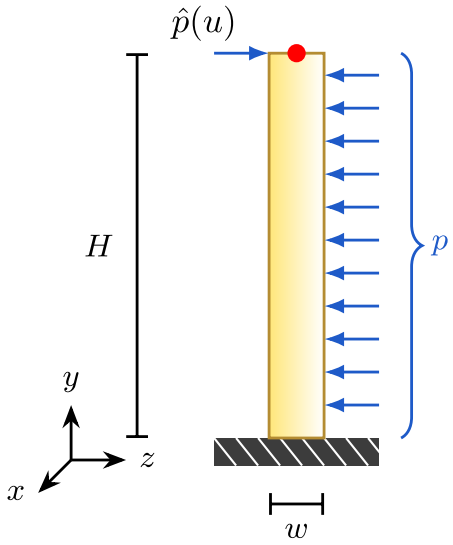


Fig. 6 Nonlinear load test setup for the vertical beam. The nonlinear load $\hat{p}(\xi)$ varies linearly with parameter ξ, where ξ is the normalized coordinate along the beam height, with $\xi = 0$ at the fixed bottom end and $\xi = 1$ at the free top end

Parameter	Value
Width (W)	0.50 m
Height (H)	1.00 m
Thickness (t)	0.05 m
Density (ρ)	3000 kg/m^3
Distributed Load (p)	$5 \times 10^3 \mathbf{e}_z$ Pa
Nonlinear Load ($\hat{p}(\xi)$)	$-\xi \mathbf{e}_z$

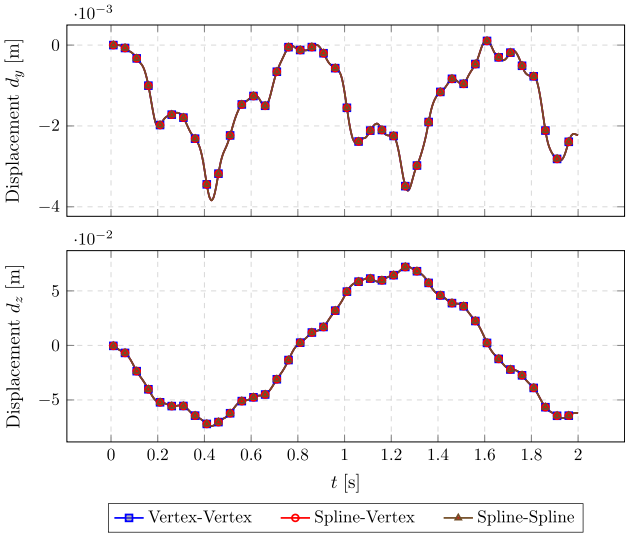


Fig. 7 Tip displacement versus time under nonlinear loading

for different coupling methods to assess the advantages of spline-based coupling methods compared to traditional vertex-based approaches.

6.1 Theoretical Communication Overhead

As described in Sect. 4.2, the communication overhead for spline-based coupling methods primarily depends on the number of control points and parametric representation, while vertex-based methods see overhead increase exponentially with mesh refinement and quadrature point quantities. Figure 8 illustrates this theoretical difference.

For vertex-based coupling using traditional Gauss quadrature with $p+1$ quadrature points per parametric direction per element, the communication overhead O is proportional to the number of interface elements n_{element}, the polynomial degree p, the number of timesteps N_t, and the parametric dimension $\hat{d}$:

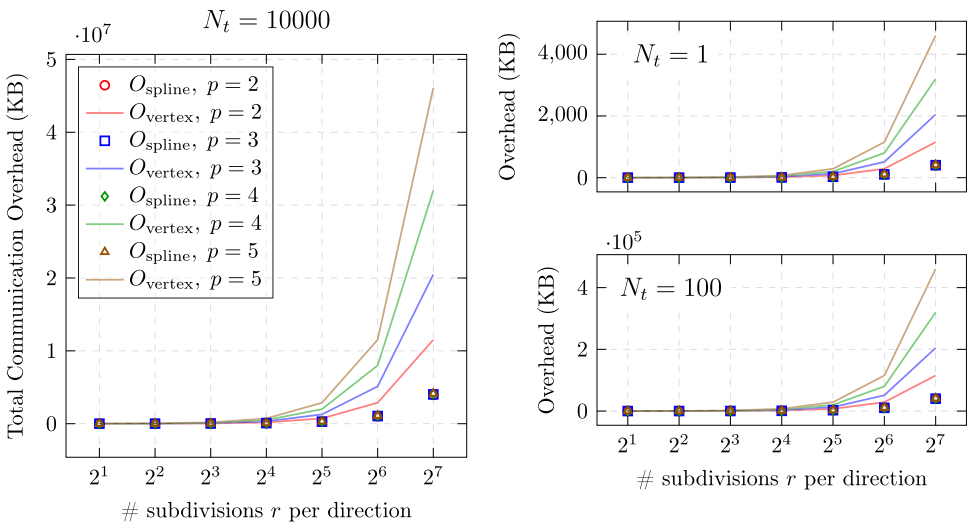


Fig. 8 Comparison of the communication overhead between vertex-based (FEM) and spline-based (IGA) coupling. The horizontal axis represents the number of parameter subdivisions per direction, serving as an equivalent indicator of the unique knots in IGA while reflecting mesh resolution in FEM. The vertical axis shows the total communication overhead in kilobytes

$$O_{\text{quad}} = N_t \cdot d \cdot n_{\text{element}} \cdot (p+1)^{\hat{d}} \tag{17}$$

For spline-based coupling, the total communication overhead is primarily governed by the number of control points, the physical field dimension d, and the polynomial degrees p_i along each parametric direction. Unlike mesh-based coupling (which requires per-node communication), spline-based coupling grows at a much slower rate:

$$O_{\text{spline}} = N_t \cdot d \cdot \left(\prod_{i=1}^{\hat{d}} n_i \right) + \sum_{i=1}^{\hat{d}} (n_i + p_i + 1) \tag{18}$$

This expression consists of two main components:

- $N_t \cdot d \cdot \left(\prod_{i=1}^{\hat{d}} n_i \right)$: the total number of scalar values exchanged during time integration. Here, n_i denotes the number of control points along the i-th parametric direction (with $\hat{d}$ such directions in total), and d represents the number of coupled physical fields.
- $\sum_{i=1}^{\hat{d}} (n_i + p_i + 1)$: the cumulative size of the knot vectors in each parametric direction. This geometric data is exchanged once during initialization and remains unchanged throughout the simulation.

We use preCICE [8] to mediate the exchange of field data (e.g., displacement, velocity, pressure) between solvers. On the interface, both the geometry and field values are represented using B-splines or NURBS. The spline representation allows us to transmit compact and structured data, where:

- The knot vector (of length $\sum_{i=1}^{d} (n_i + p_i + 1)$) is transferred at the beginning of the simulation.
- The field values are exchanged at every coupling step through the control points.

This comparison highlights a major advantage of spline-based coupling. As shown in Fig. 8, vertex-based methods lead to an exponential increase in communication overhead as mesh refinement and polynomial degree grow, making them computationally expensive for high-fidelity simulations. In contrast, spline-based methods maintain significantly lower communication costs, as their overhead scales more efficiently with refinement.

The analysis shows that as the mesh becomes more refined, the difference in communication overhead between vertex-based and spline-based methods will grow significantly.

6.2 Experimental validation results

The experimental results confirm our theoretical predictions about communication overhead in different coupling methods. We measured both the actual data transfer volume during the initial timestep ($N_t = 1$) and total communication time per dimension over $N_t = 10000$ timestep in a vertical beam test case across various mesh refinement levels.

Our instrumentation specifically targeted only the coupling-related data transfer functions, ensuring that the measured values reflect purely the communication overhead of different coupling methods without contamination from other computational processes.

In the case of spline-based coupling, additional overhead arises from transmitting knot vectors alongside control point data. To support tensor-product B-spline representations in multiple parametric directions, knot vectors are embedded into a unified matrix structure with `NaN` padding. Since the knot vectors are defined separately for each parametric direction, we embed them into a unified matrix where each row corresponds to one direction. `NaN` values are used to fill the remaining entries. This ensures consistency in data formatting and facilitates reliable transfer and storage.

Here we consider a parametric dimension $\hat{d} = 2$ and a physical dimension $d = 3$. The tensor-product B-spline space is defined by two univariate knot vectors associated with each parametric direction:

$$\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n_\xi + p_\xi + 1}\} \subset \mathbb{R},$$
$$\mathscr{H} = \{\eta_1, \eta_2, \ldots, \eta_{n_\eta + p_\eta + 1}\} \subset \mathbb{R},$$

where each knot vector is a non-decreasing sequence of real numbers. Here, n_ξ and n_η denote the number of basis functions in the ξ- and η-directions, respectively, and p_ξ, p_η are the corresponding polynomial degrees.

In the implementation, while arranging knot vectors into a unified matrix format to ensure consistent data transmission, we inevitably introduce some overhead.

Since the lengths of knot vectors can differ between parametric directions, `NaN` values are used as padding to maintain consistent matrix dimensions:

$$\text{knotMatrix} = \begin{bmatrix} \xi_1 & \xi_2 & \ldots & \xi_{n_\xi + p_\xi + 1} & \text{NaN} & \ldots & \text{NaN} \\ \text{NaN} & \ldots & \text{NaN} & \eta_1 & \eta_2 & \ldots & \eta_{n_\eta + p_\eta + 1} \end{bmatrix}$$

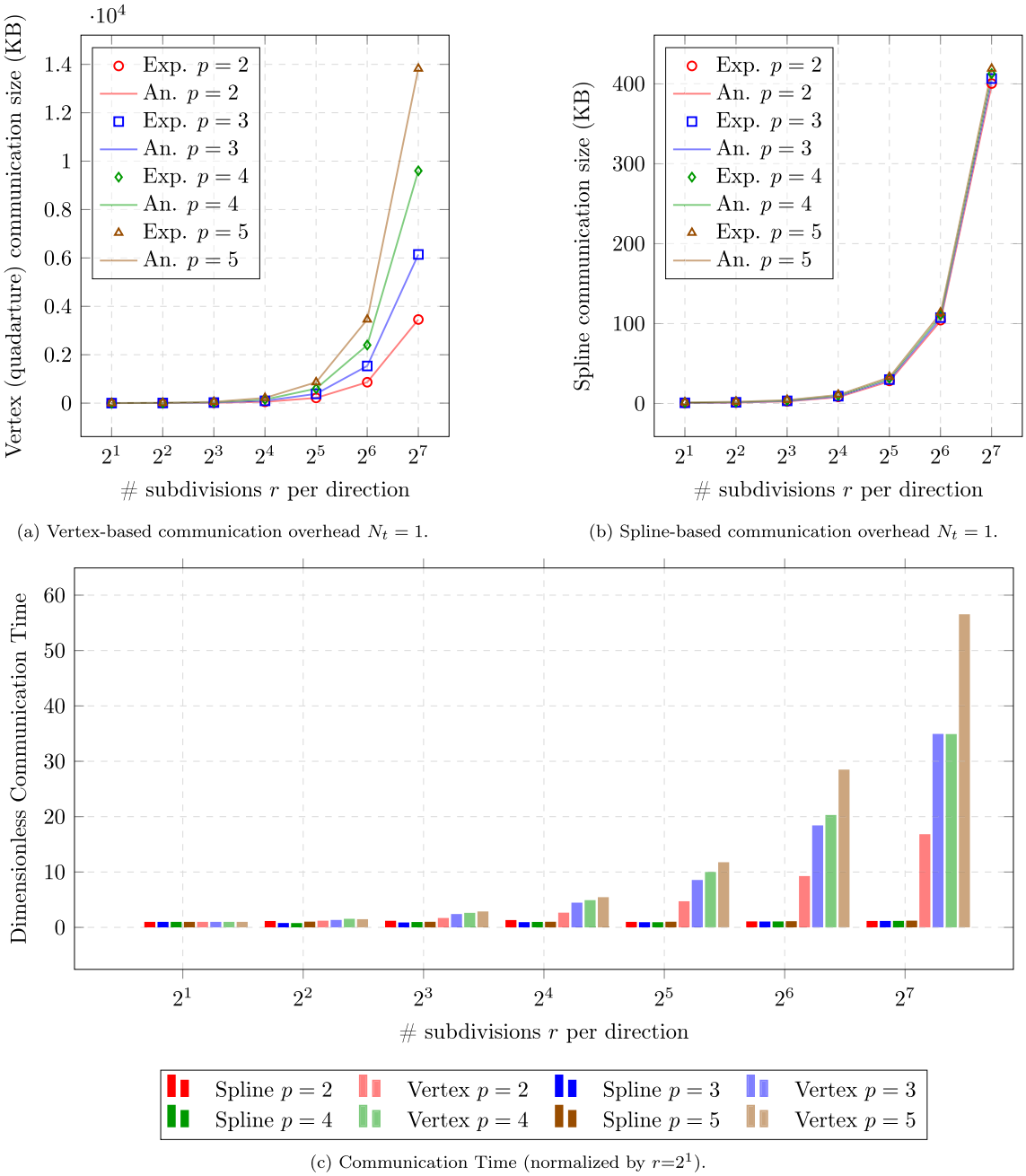
In the numerical experiment, we consider the symmetric case where $n_\xi = n_\eta = n$ and $p_\xi = p_\eta = p$, so that both knot vectors Ξ and $\mathscr{H}$ have the same length, namely $n + p + 1$. To embed them into a unified matrix for transfer, we allocate a matrix of size $2 \times (2n + 2p + 2)$, where each row stores one knot vector padded with `NaN` values in the

complementary half. Consequently, the total number of `NaN` entries equals $2(n + p + 1)$.

All timing data were collected on a single compute node of the DelftBlue supercomputer [33]. Given that runtime performance can be affected by external processes and background system noise, all measurements were repeated ten times in an isolated environment with minimal external interference. The values reported represent the averaged results, ensuring reliability and minimizing the influence of transient fluctuations.

Figure 9 presents a comprehensive analysis of communication overhead for different coupling methods through four complementary perspectives. In Fig. 9a

Figure 9b provides a similar data volume comparison for spline-based coupling, where both theoretical and experimental results confirm the significantly reduced growth rate in communication overhead as mesh resolution increases. Finally, Fig. 9c translates these data volume differences into actual communication time measurements, validating that the theoretical efficiency gains of spline-based coupling



(a) Vertex-based communication overhead $N_t = 1$.

(b) Spline-based communication overhead $N_t = 1$.

(c) Communication Time (normalized by r=2^1).

Fig. 9 Experimental (Exp.) and theoretical (An.) comparison of communication overhead with respect to the number of parameter subdivisions r per direction, with for $r \in \{1, 2, ..., 7\}$

Table 1 Absolute discrepancy (KB) between measured and analytical communication sizes at $r = 7$, with corresponding analytical number of NaN-padded entries

p	Absolute discrepancy (KB)	# Measured NaN entries	# Analytical NaN entries*
2	2.078125	266.00	266
3	2.1096875	270.04	270
4	2.140375	273.97	274
5	2.1721875	278.04	278

*Computed as $2(n + p + 1)$, where $n = 2^r$ is the number of basis functions per parametric direction and p is the spline degree

translate to meaningful performance improvements in practice, with timing differences becoming increasingly pronounced at finer mesh resolutions.

As shown in Fig. 9b

Table 1 supports this observation by listing the absolute differences between experimental and analytical spline-based communication overheads at $r = 7$ for various polynomial degrees p, corresponding to the results shown in Fig. 9b

$$\varepsilon_{\text{absolute}} = |O_{\text{exp}} - O_{\text{an}}|, \tag{19}$$

where O_{exp} denotes the measured (experimental) communication overhead in kilobytes, and O_{an} is the corresponding analytical estimate based on Eq. 18.

Key observations from comparing the experimental results (Fig. 9) with theoretical results (Fig. 8) include:

1. The communication overhead for Vertex-based coupling methods increases rapidly with mesh refinement in practice, which is consistent with the theoretical prediction.
2. Spline-based coupling methods (including spline-vertex and IGA-IGA approaches) demonstrate significantly lower growth rates in communication overhead, confirming the theoretical advantages.
3. The experimental curves closely match the theoretical predictions, although the absolute values differ due to implementation details and overhead from the coupling library.

These results confirm the communication efficiency advantages of spline-based coupling methods, particularly when dealing with high-resolution interface meshes. The close agreement between theoretical predictions and experimental measurements validates our communication overhead model and provides a solid foundation for estimating performance in larger, more complex simulations.

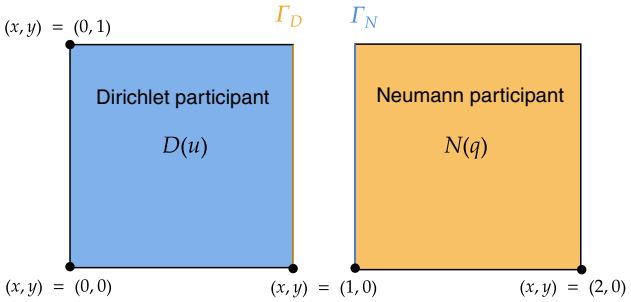


Fig. 10 Partitioned heat conduction problem setup

7 Benchmark problems

So far we have demonstrated the superior communication efficiency of spline-based coupling methods in solvers that support spline representation, while also illustrating how vertex-based coupling methods can effectively interface with solvers utilizing alternative discretization techniques through the preCICE library. In this section, we extend our analysis to validate the accuracy and robustness of the proposed methods through a series of carefully selected benchmark problems that progressively increase in complexity and physical relevance. The solid domain is modeled using the G+Smo library [31], which provides native support for IGA, while the fluid domain is solved using OpenFOAM[34], a widely used finite volume-based CFD framework.

7.1 Partitioned heat conduction

In this example, we employed the spline-based communication method to simulate heat transfer across partitioned domains. Beyond the previously demonstrated advantages in communication efficiency and information preservation, this application reveals two additional benefits particularly relevant to FSI. Let's first start with the benchmark setup.

The partitioned heat conduction setup, shown in Fig. 10, consists of two square regions with different thermal properties. The governing equation is the dimensionless, transient heat equation in two spatial dimensions:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \nabla^2 u + f \quad \text{in } \Omega = [0,2] \times [0,1], \quad t \in [0,T], \\ u &= g \qquad\qquad\qquad\quad \text{on } \partial\Omega, \quad t \in [0,T], \end{aligned} \tag{20}$$

where $u(x, y, t)$ is the temperature field, f is a source term, and g denotes prescribed boundary data.

To validate the numerical method, we adopt a manufactured solution of the form

$$u_{\text{exact}}(x, y, t) = 1 + x^2 + \alpha y^2 + \beta t, \tag{21}$$

with constants $\alpha, \beta \in \mathbb{R}$. Substituting u_{exact} into (20) yields the corresponding source term f and Dirichlet boundary condition g.

Following the formulation in [35], the domain Ω is partitioned into two non-overlapping subdomains:

$$\Omega_D = [0,1] \times [0,1], \qquad \Omega_N = [1,2] \times [0,1],$$

such that $\Omega = \Omega_D \cup \Omega_N$. These subdomains are coupled along the shared interface

$$\Gamma = \Omega_D \cap \Omega_N = \{1\} \times [0,1],$$

as shown in Fig. 10. This configuration is referred to as the partitioned heat equation.

During the coupling procedure, either the temperature u or the normal component of the heat flux

$$q_n := \nabla u \cdot \mathrm{n}$$

is exchanged across Γ. Specifically, one subdomain applies a Dirichlet condition

$$u = u_\Gamma \quad \text{on } \Gamma,$$

while the other enforces a Neumann condition

$$\nabla u \cdot \mathrm{n} = q_\Gamma \quad \text{on } \Gamma,$$

where u_Γ and q_Γ represent the interface temperature and heat flux, respectively.

Naturally non-conforming support: One of the main advantages of the proposed coupling strategy is its natural ability to handle non-conforming discretizations at the coupling interface. Even when the mesh is geometrically discontinuous across the interface, the use of spline-based representations allows us to preserve functional smoothness. In the current implementation, non-conforming meshes are assumed to be nested, meaning that the control points or knot vectors of one mesh are fully contained within those of the other.

As illustrated in Fig. 11, the Dirichlet subdomain Ω_D is discretized with a coarser mesh refined once in Fig. 11a and twice in Fig. 11c and Fig. 11e

Inherited derivatives via spline representation: One of the distinctive advantages of using spline-based representations for coupling is that the derivatives of the solution field are inherently preserved and transferred across the interface. Since Γ_D is represented as a spline function, its smoothness and continuity naturally extend to its derivatives. As a result, when exchanging data across non-conforming interfaces, both the field values and their gradients are consistently evaluated from the same spline representation, without the need for additional reconstruction or projection.

As illustrated in Fig. 12, the spline-based temperature field received from the Ω_D participant on the interface Γ_D is differentiated to evaluate the heat flux. These computed flux values are then compared against the exact analytical solution. The red curve presents the relative error at each time step on a logarithmic scale, while the blue dashed line denotes the mean error across all time steps, which remains consistently around 6.7×10^{-15}. This result demonstrates the high-fidelity derivative that can be accurately evaluated from the spline field without needing to transfer the derivative separately.

7.2 Perpendicular flap

The perpendicular flap benchmark is a widely used FSI test case [8] provided by preCICE to evaluate the performance and accuracy of partitioned coupling strategies between fluid and solid solvers. This benchmark is particularly useful for assessing solver interoperability and conducting a mesh convergence study.

We model a three-dimensional flow that interacts with an elastic flap fixed perpendicularly to the channel floor. The physical parameters for both the fluid and solid domains are detailed in Fig. 13. We employ vertex-vertex coupling with data exchange at quadrature points. The structure uses IGA discretization (single spline patch) while the fluid mesh is a structured grid with local refinement near the flap interface. Convergence results for structural refinement levels $r \in \{1,2,3\}$ and polynomial orders $p = 3,4$ are shown in Fig. 14.

Figure 15 illustrates the time evolution of the flap tip displacement under fluid loading with time step size. The flap exhibits characteristic oscillatory behavior due to the interplay between fluid pressure and structural elasticity.

In Fig. 15, the dark blue curve corresponds to G+Smo, which is the only solver in this comparison based on IGA. It shows strong agreement with other well-established solvers in both the amplitude and phase of the oscillatory motion of the flap tip. This comparison is based on the tip displacement in the x-direction, using a time step size of $\Delta t = 0.01\,\mathrm{s}$.

To demonstrate applicability to large-scale systems, we extended the benchmark to a fluid domain of $20\mathrm{m}$ width containing 20 perpendicular flaps, each spaced at a center-to-center distance of $1.05\mathrm{m}$. All other parameters are consistent with the perpendicular flap benchmark 13. This configuration employs vertex-based coupling. Such multi-flap systems represent practical applications, including arrays of flow control systems or flexible marine structures [18], where decomposition of the system into subdomains for efficient parallel computing across numerous interface

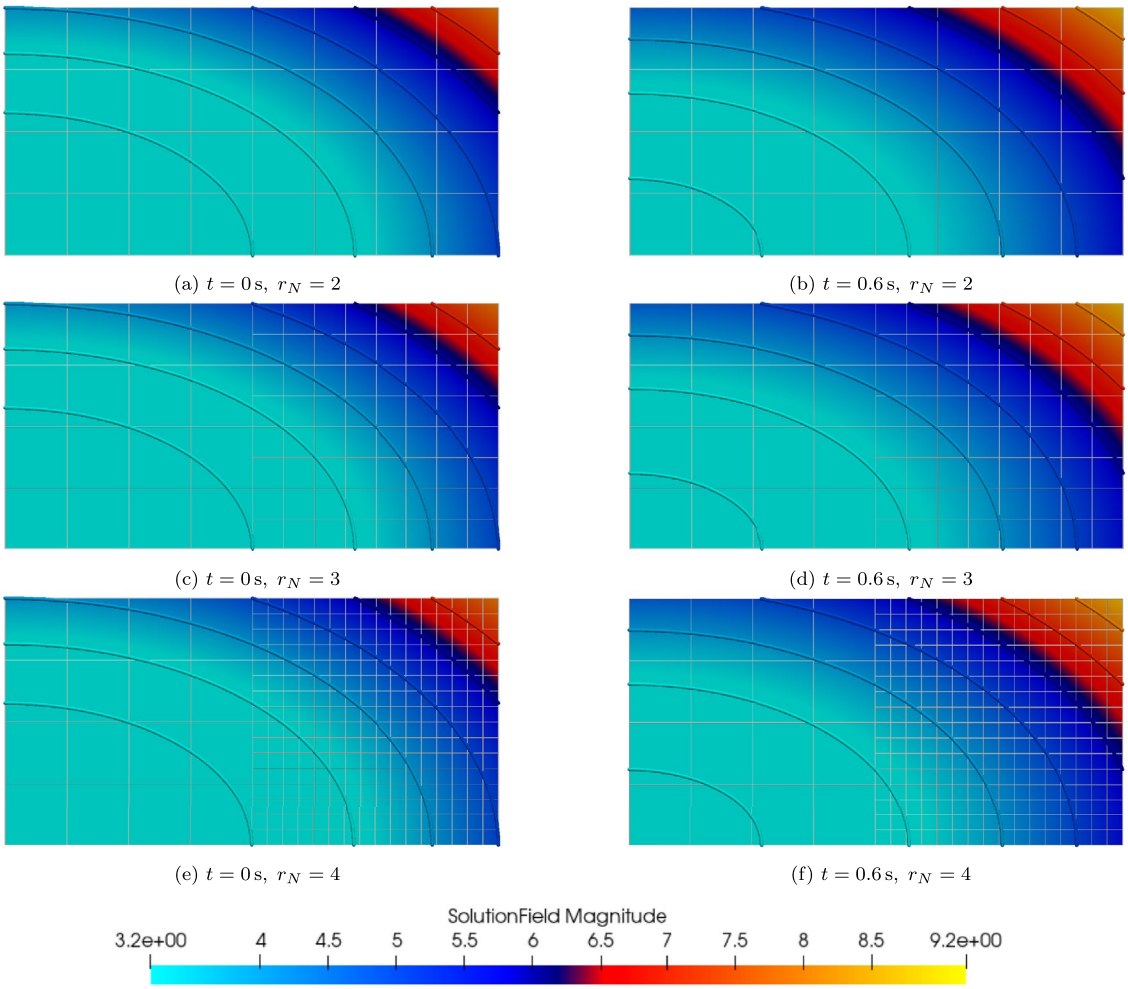


Fig. 11 Temperature field of a partitioned spline-mesh domain with varying Ω_N refinement levels r_N

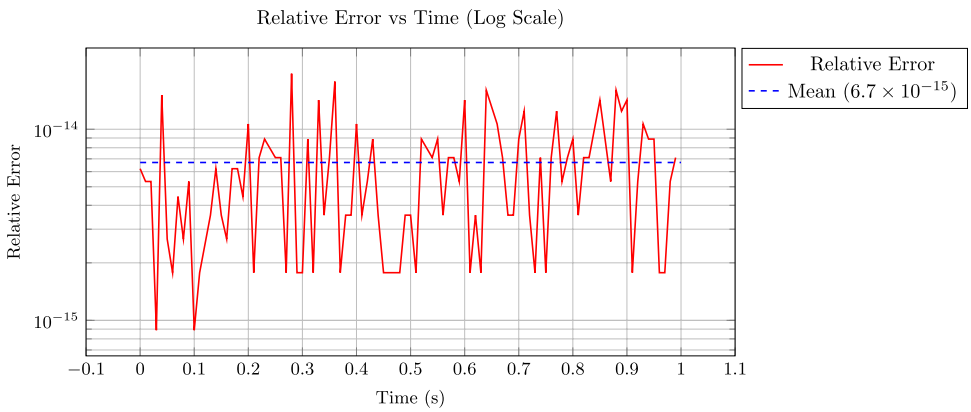
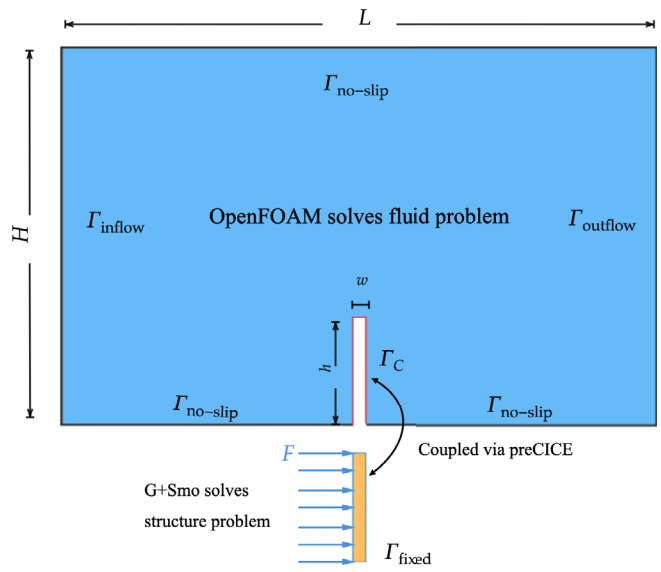


Fig. 12 Validation of derivative transfer via spline representation, relative L^2-error of the heat flux remains at machine precision level

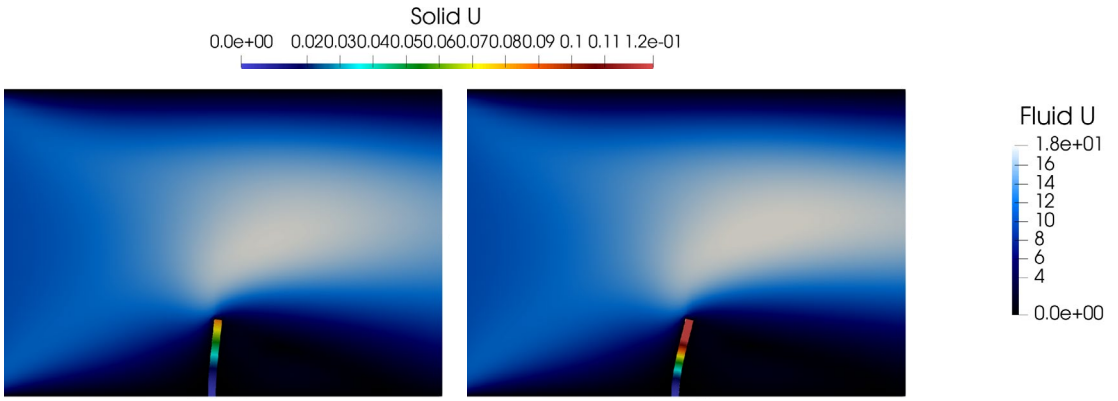
components becomes critical for overall performance (Fig. 16).

The benchmark has also been extended to a pure IGA setting, where both the fluid and structural solvers are based on G+Smo (specifically, the `gsIncompressibleFlow` [36] and `gsElasticity` modules) and the information is exchanged through the spline-based communication method in a partitioned way coupled through preCICE. In this fully IGA framework, the fluid domain motion is handled using an Arbitrary Lagrangian–Eulerian (ALE) formulation. Specifically, we employ the Tangential Incremental Nonlinear Elasticity (TINE) method [37] to compute the mesh deformation, ensuring robustness under large interface displacements. For detailed comparison and advantages of this



Dimensions		
L	6.00	m
H	4.00	m
l	2.95	m
h	1.00	m
w	0.10	m
Fluid		
Ma_∞	0.01	
p_∞	101325	Pa
T_∞	288.15	K
Solid		
E	$4 \cdot 10^6$	N/m^2
ν_s	$3 \cdot 10^{-1}$	
ρ_s	$3 \cdot 10^3$	kg/m^3

Fig. 13 Setup and parameters of the perpendicular flap test case. The original image is taken from the preCICE paper [8]



(a) t = 0.3 s (b) t = 0.7 s

Fig. 14 Coupled fluid–structure interaction at selected time instances: fluid velocity magnitude (background) and structural pressure distribution (foreground) at $t = 0.3\,\text{s}$ and $t = 0.7\,\text{s}$

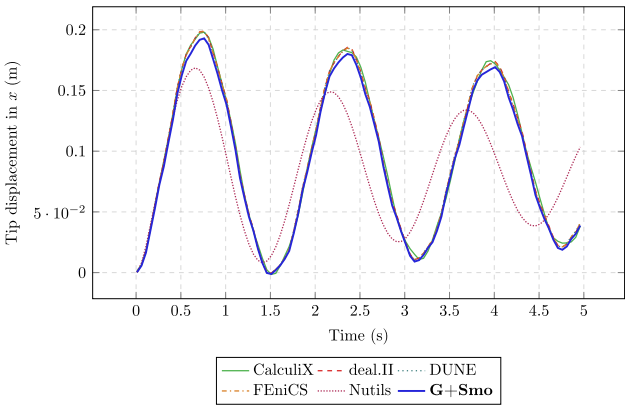
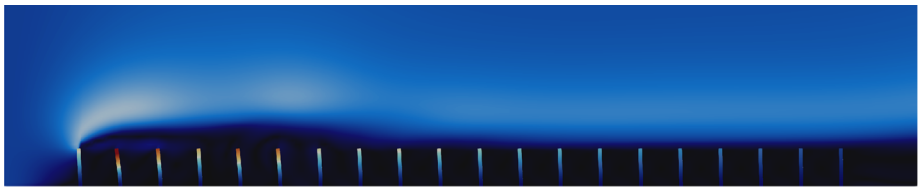


Fig. 15 Displacement of the perpendicular flap over time under fluid loading

method, the authors kindly ask the readers refer to the paper by A. Shamanskiy [37] for more details. Figure 17 presents the simulation results at $t = 0.3\,\text{s}$ and $t = 0.7\,\text{s}$, depicting both the fluid velocity field and the isogeometric mesh preserved by the TINE algorithm.

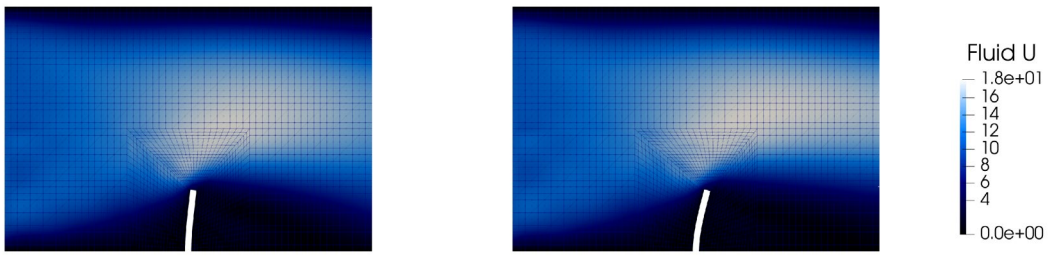
To verify the proposed pure IGA coupling method, we benchmark the G+Smo-G+Smo framework against the hybrid OpenFOAM(v2404)-G+Smo configuration. Figure 18a presents the tip displacement comparison, showing that the pure IGA coupling yields results consistent with the vertex-based hybrid approach. Furthermore, Fig. 18b highlights the efficiency of the IGA-based method, which demonstrates significantly lower communication overhead compared to the vertex-based strategy.

(a) Perpendicular flap system at t = 0.14 s



(b) Perpendicular flap system at t = 0.44 s

Fig. 16 Multi-field simulation of a system with 20 perpendicular flaps under fluid loading



(a) t = 0.3 s (b) t = 0.7 s

Fig. 17 Visual simulation results of IGA-IGA coupling: fluid velocity magnitude and isogeometric mesh at $t = 0.3\,\mathrm{s}$ and $t = 0.7\,\mathrm{s}$

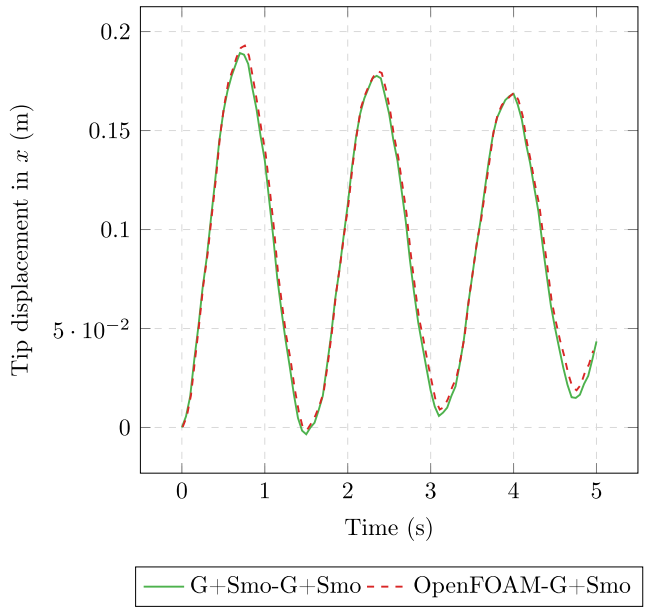
7.3 Turek Hron FSI benchmark

The third benchmark involves two-dimensional flow-induced oscillations of an elastic structure mounted behind a rigid cylinder, commonly referred to as the Turek-Hron FSI3 benchmark. This configuration is widely used to evaluate the performance of FSI solvers, particularly under challenging conditions characterized by strong coupling effects and significant added-mass phenomena.

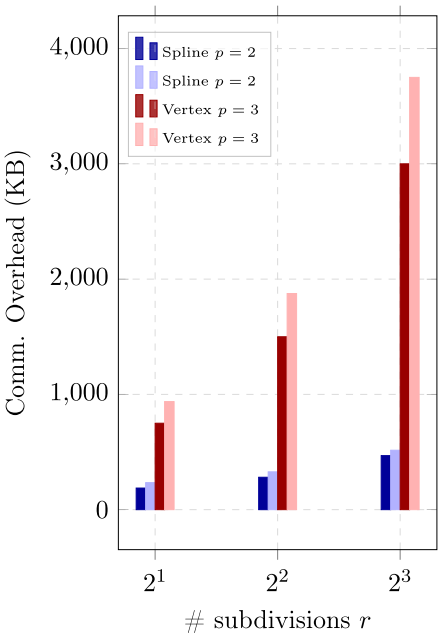
The geometry of the Turek-Hron FSI3 benchmark is illustrated in Fig. 19. The configuration consists of a flexible beam attached to the downstream side of a fixed circular cylinder, located inside a rectangular channel of height $0.41\,\mathrm{m}$ and length $2.5\,\mathrm{m}$. The diameter of the cylinder is $0.1\,\mathrm{m}$, and the beam has a length of $0.35\,\mathrm{m}$ and a thickness of $0.02\,\mathrm{m}$. The inflow boundary Γ_{in} is prescribed with a parabolic velocity profile, $v_{\mathrm{in}}(y) = 1.5\bar{v}\frac{4y(0.41-y)}{0.41^2}$, where $\bar{v}$ is the average inlet velocity. No-slip boundary conditions are applied along the top and bottom channel walls, denoted by $\Gamma_{\text{no-slip}}$. The average inlet velocity $\bar{v} = 2\,m/s$.

The fluid is assumed to be Newtonian and incompressible, with a density of $\rho_f = 1000\,\mathrm{kg/m^3}$ and a kinematic viscosity of $\nu_f = 10^{-4}\,\mathrm{m^2/s}$, resulting in a Reynolds number of $\mathrm{Re} = 200$. The fluid domain is discretized using an OpenFOAM structured mesh generated with blockMesh, composed of hexahedral cells organized in 18 blocks, consisting of approximately 46k cells. The structure is modelled using the Saint-Venant–Kirchhoff material with a Young's modulus of $E = 5.6\,\mathrm{MPa}$ and Poisson's ratio $\nu_s = 0.4$. The solid density is taken as $\rho_s = 1000\,\mathrm{kg/m^3}$, which matches the fluid density to ensure neutral buoyancy. Coupling employs the vertex-vertex strategy with data exchange at interface quadrature points. On the structural side, we use IGA discretization with spline bases of degree $p = 2$ and uniform refinements $r \in \{3, 4\}$ at the interface; on the fluid side, the structured hexahedral mesh is refined near the cylinder and beam.

The left plot in Fig. 20 shows the time evolution of the vertical displacement at the tip of the elastic beam. The results obtained using G+Smo with two refinement levels ($r = 3$ and $r = 4$, both with degree $p = 2$) are compared



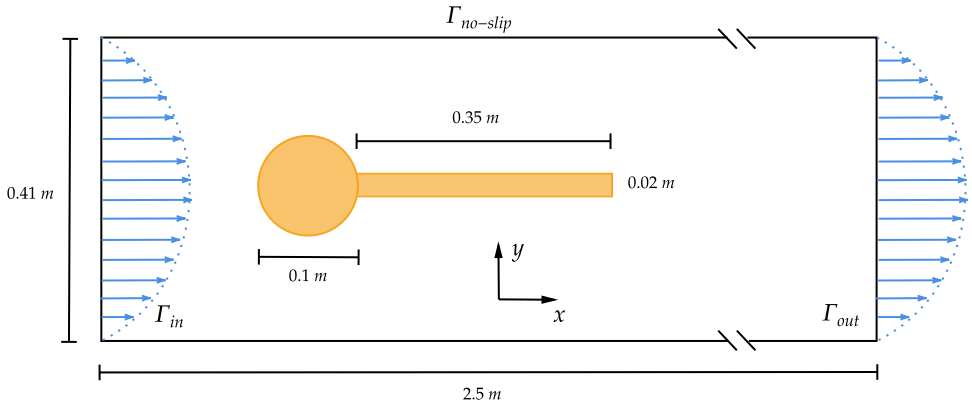
(a) Tip displacement comparison between G+Smo-G+Smo and OpenFOAM-G+Smo.



(b) Communication overhead comparison.

Fig. 18 Perpendicular flap benchmark in pure IGA setting, information exchanged through the spline based communication method in a partitioned way coupled through preCICE

Fig. 19 Turek Hron FSI benchmark setup



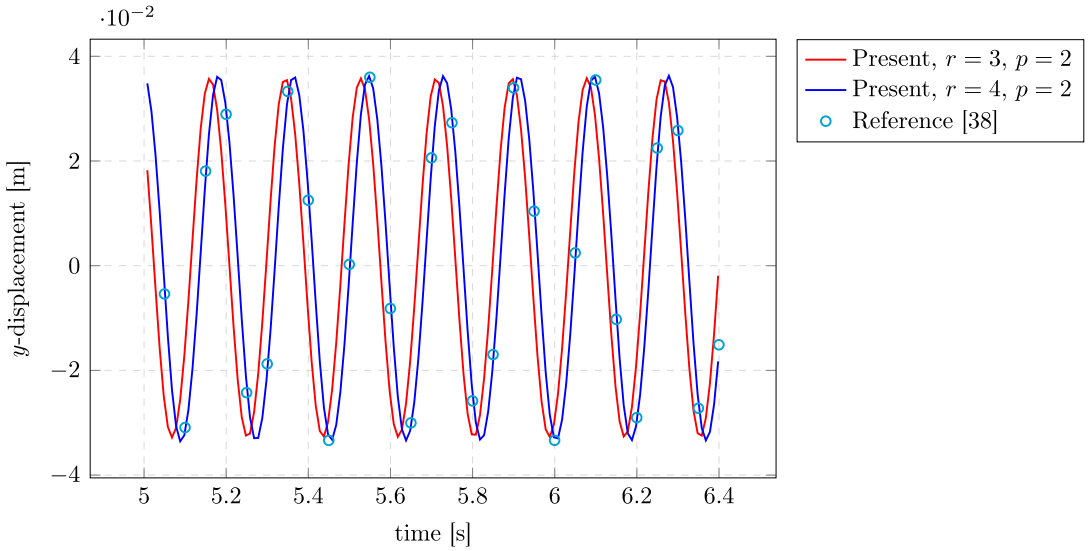
against reference data. The numerical solution from the finer mesh ($r = 4$) closely follows the reference in both amplitude and phase, while the coarser mesh ($r = 3$) slightly underestimates the peak displacements but still captures the oscillation frequency well.

This strong agreement demonstrates that our coupling framework for the IGA-based structure solver is capable of accurately resolving complex FSI dynamics. It highlights the method's ability to perform high-fidelity and efficient coupling within libraries that support splines, while also remaining compatible with solvers that work without spline-based boundary representations.

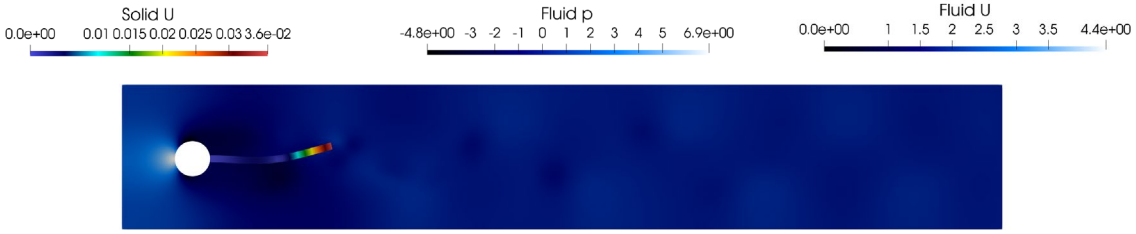
8 Conclusion

In this work, we have put together a seamless, start-to-finish spline coupling workflow for partitioned multiphysics simulations, with a particular focus on fluid–structure interaction problems. Our approach demonstrates that splines not only serve as mathematical representations for geometry and solution fields, but also function effectively as an efficient data exchange format between different solvers.

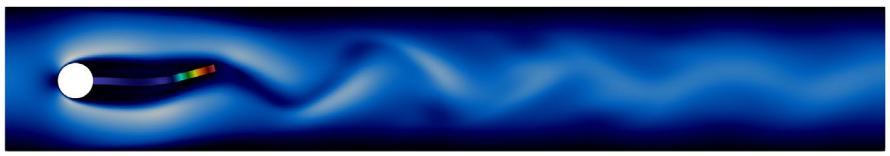
1. Development of spline-based coupling methods for enhanced representation of interface fields.
2. Comprehensive analysis of communication overhead, demonstrating the efficiency of spline-based methods over traditional vertex-based approaches. Our theoretical model and experimental validation show that



(a) y-displacement of the flap tip with respect to time on the interval $t \in [5.0, 6.4]$. The reference data [38] is represented by blue marker, and the data obtained by the present model is represented by blue and red lines.



(b) Pressure field for the mesh with $r = 4$, $p = 2$ at time $t = 5.9s$. The color bars are provided above.



(c) Velocity field for the mesh with $r = 4$, $p = 2$ at time $t = 5.9s$. The color bars are provided above.

Fig. 20 Simulation results for the Turek-Hron FSI3 benchmark using G+Smo coupled with OpenFOAM. (a) Vertical displacement of the beam tip over time. (b) Beam displacement and fluid pressure distribution. (c) Beam displacement and fluid velocity magnitude for mesh $r = 4$ with degree $p = 2$ at time step $t = 5.9$ s

communication overhead for vertex-based coupling grows rapidly with mesh refinement, while spline-based methods demonstrate significantly lower growth rates.

3. Implementation of robust coupling strategies that naturally handle non-conforming but nested meshes and preserve solution derivatives across interfaces. The spline-based approach eliminates the need for complex interpolation schemes when meshes do not match at the interface, and inherently preserves both solution values and their gradients, as demonstrated in our heat conduction benchmark where derivative information was accurately maintained without requiring separate transfer operations.

The proposed methods are verified using vertical beam test cases under both constant and nonlinear loading, and validated through established FSI benchmarks, including the perpendicular flap and Turek-Hron configurations. Although all benchmarks are two-dimensional, this restriction does not limit the generality of our findings. The primary goal is to assess the coupling strategy and its communication behavior, both of which extend naturally to three-dimensional settings.

Future work will focus on extending the proposed methods to more complex multiphysics problems, including those involving large deformations, topology changes, multiple coupled fields, and higher Reynolds number flows. While the current benchmarks demonstrate the effectiveness of spline-based coupling for established FSI test cases, future work will address higher Reynolds number regimes where steeper interface gradients and more complex flow physics pose additional challenges for interface data transfer. The

smooth, high-order nature of spline representations should provide advantages in such demanding applications. In addition, we plan to implement adaptive local refinement using Truncated Hierarchical B-splines (THB-splines) [39], which enable targeted mesh refinement in regions with high solution gradients while preserving the advantages of spline-based coupling in 3D geometries. Additionally, we intend to adapt our coupling methodology to Kirchhoff–Love shell structures [40, 41], which are prevalent in many FSI applications such as aerospace and biomedical engineering. Another promising direction is the development of IGA-based mesh transfer techniques specifically optimized for FSI applications, which could further improve the handling of moving and deforming interfaces [42]. Finally, we will continue optimizing our implementation to support large-scale, industrial-grade simulations.

Acknowledgements We would like to express our gratitude to Jun.-Prof. Benjamin Uekermann and Gerasimos Chourdakis from the preCICE team for their technical support and valuable insights regarding the coupling interface implementation. We also appreciate Dr. Marin Lauber for the fruitful discussions on integrating spline-based coupling methods in Waterlily.jl.

Author Contributions J. Li: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. H.M. Verhelst: Writing – review & editing, Software, Investigation, Conceptualization, Supervision. J.H. Den Besten: Writing – review, Supervision, Project administration, Funding acquisition. M. Möller: Writing –review, Software, Supervision, Project administration, Funding acquisition, Conceptualization.

Funding This research has been conducted as a part of the project FlexFloat with project number 19002 of the research programme Open Technology which is (partly) financed by the Dutch Research Council (NWO), Netherlands. In addition, HMV is grateful for the financial support from the Italian Ministry of University and Research (MUR) through the PRIN projects COSMIC (no. 2022A79M75) and ASTICE (no. 202292JW3F), with the contribution of the European union – Next Generation EU.

Data Availability The proposed coupling methods are implemented as part of the G+Smo (Geometry + Simulation Modules) repository. We used preCICE to interface with the fluid solver OpenFOAM and the structure solver G+Smo.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

1. Make M, Spenke T, Hosters N, Behr M (2022) Spline-based space-time finite element approach for fluid-structure interaction problems with a focus on fully enclosed domains. https://doi.org/10.48550/arXiv.2203.16152. ArXiv: 2203.16152
2. Hosters N, Helmig J, Stavrev A, Behr M, Elgeti S (2018) Fluid–structure interaction with nurbs-based coupling. Comput Methods Appl Mech Eng 332:520–539. https://doi.org/10.1016/j.cma.2018.01.003
3. Bazilevs Y, Takizawa K, Tezduyar TE (2013) Computational fluid-structure interaction: methods and applications. Wiley
4. Fan J, Liao H, Ke R, Kucukal E, Gurkan UA, Shen X, Lu J, Li B (2018) A monolithic lagrangian meshfree scheme for fluid-structure interaction problems within the otm framework. Comput Methods Appl Mech Eng 337:198–219. https://doi.org/10.1016/j.cma.2018.03.031
5. Ben el Hadj Ali N, Legay A, Piperno S (2019) Partitioned coupling schemes for fluid-structure interaction: a review of explicit and semi-implicit approaches. Appl Mech Rev 71(4):040801
6. Farhat C, Lesoinne M, Tallec P (1998) Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. Comput Methods Appl Mech Eng 157:95–114. https://doi.org/10.1016/S0045-7825(97)00216-8
7. Li Y, Law Y, Joshi V, Jaiman R (2018) A 3d common-refinement method for non-matching meshes in partitioned variational fluid-structure analysis. J Comput Phys 374:163–187. https://doi.org/10.1016/j.jcp.2018.05.023
8. Chourdakis G, Davis K, Rodenberg B, Schulte M, Simonis F, Uekermann B, Abrams G, Bungartz H, Yau LC, Desai I, Eder K, Hertrich R, Lindner F, Rusch A, Sashko D, Schneider D, Totounferoush A, Volland D, Vollmer P, Koseomur OZ (2011) precice v2: a sustainable and user-friendly coupling library. Open Res Eur 1(114):9 https://doi.org/10.12688/openreseurope.14445.1. arxiv: 2109.14470
9. Jaiman R, Jiao X, Geubelle P, Loth E (2005) Assessment of conservative load transfer for fluid–solid interface with non-matching meshes. Int J Numer Meth Eng 64:2014–2038. https://doi.org/10.1002/NME.1434
10. Jiao X, Heath M (2004) Common-refinement-based data transfer between non-matching meshes in multiphysics simulations. Int J Numer Meth Eng 61:2402–2427. https://doi.org/10.1002/NME.1147
11. Kim H (2010) A new coupling strategy for fluid–solid interaction problems by using the interface element method. Int J Numer Meth Eng 81:403–428. https://doi.org/10.1002/NME.2698
12. Klöppel T, Popp A, Küttler U, Wall W (2011) Fluid–structure interaction for non-conforming interfaces based on a dual mortar formulation. Comput Methods Appl Mech Eng 200:3111–3126. https://doi.org/10.1016/J.CMA.2011.06.006
13. Klöppel T, Popp A, Küttler U, Wall W (2011) Fluid–structure interaction for non-conforming interfaces based on a dual mortar formulation. Comput Methods Appl Mech Eng 200:3111–3126. https://doi.org/10.1016/J.CMA.2011.06.006
14. Rendall T, Allen C (2008) Unified fluid–structure interpolation and mesh motion using radial basis functions. Int J Numer Meth Eng 74:1519–1559. https://doi.org/10.1002/nme.2219

15. Schneider D, Schrader T, Uekermann B (2023) Data-parallel radial-basis function interpolation in precice. In: Proceedings of COUPLED
16. Jaiman R, Geubelle P, Loth E, Jiao X (2011) Transient fluid–structure interaction with non-matching spatial and temporal discretizations. Comput Fluids 50:120–135. https://doi.org/10.1016/J.COMPFLUID.2011.07.001
17. Mehl M, Uekermann B, Bijl H, Blom D, Gatzhammer B, van Zuijlen A (2016) Parallel coupling numerics for partitioned fluid-structure interaction simulations. Comput Math Appl 71(4):869–891. https://doi.org/10.1016/j.camwa.2015.12.025
18. Tavakoli S, Singh M, Hosseinzadeh S, Zhengyu H, Shao Y, Wang S, Huang L, Grammatikopoulos A, Li YP, Khojasteh D, Liu J, Dolatshah A, Cheng H, Hirdaris S (2025) A review of flexible fluid-structure interactions in the ocean: progress, challenges, and future directions. Ocean Eng 342:122545. https://doi.org/10.1016/j.oceaneng.2025.122545
19. Hughes TJR, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: cad, finite elements, nurbs, exact geometry and mesh refinement. Comput Methods Appl Mech Eng 194(39):4135–4195. https://doi.org/10.1016/j.cma.2004.10.008
20. Cottrell JA, Hughes TJ, Bazilevs Y (2009) Isogeometric analysis: toward integration of CAD and FEA. Wiley
21. Malenica S, Derbanne Q (2014) Hydro-structural issues in the design of ultra large container ships. Int J Naval Archit Ocean Eng 6(4):983–999. https://doi.org/10.2478/IJNAOE-2013-0226
22. Kamensky D, Evans JA, Hsu MC (2015) Stability and conservation properties of collocated constraints in immersogeometric fluid-thin structure interaction analysis. In: Communications in Computational Physics, volume 18. Global Science Press, pp. 1147–1180. https://doi.org/10.4208/cicp.150115.170415s
23. Hille HC, Kumar S, Lorenzis LD (2022) Floating isogeometric analysis. Comput Methods Appl Mech Eng 392:114684. https://doi.org/10.1016/j.cma.2022.114684
24. Rosa RJR, Coda HB, Sanches RAK (2022) Blended isogeometric-finite element analysis for large displacements linear elastic fracture mechanics. Comput Methods Appl Mech Eng 392:114622. https://doi.org/10.1016/j.cma.2022.114622. (**ISSN 00457825.**)
25. Guarino G, Antolin P, Milazzo A, Buffa A (2024) An interior penalty coupling strategy for isogeometric non-conformal kirchhoff-love shell patches. Eng Comput 40(5):3031–3057. https://doi.org/10.1007/s00366-024-01965-5
26. Rajanna M, Johnson EL, Liu N, Korobenko A, Bazilevs Y, Hsu M (2022) Fluid-structure interaction modeling with nonmatching interface discretizations for compressible flow problems: computational framework and validation study. Math Models Methods Appl Sci. https://doi.org/10.1142/s0218202522500592
27. Lindner F, Mehl M, Uekermann B (2017) Radial basis function interpolation for black-box multi-physics simulations. In: Technical Report, UPC. https://upcommons.upc.edu/handle/2117/190255
28. de Boer A, Zuijlen AH, Bijl H (2008) Comparison of conservative and consistent approaches for the coupling of non-matching meshes. Comput Methods Appl Mech Eng 197(49–50):4284–4297. https://doi.org/10.1016/j.cma.2008.05.001
29. Mantzaflaris A (2019) An overview of geometry plus simulation modules. In International Conference on Mathematical Aspects of Computer and Information Sciences. Springer, pp. 453–456
30. Jüttler B, Langer U, Mantzaflaris A, Moore SE, Zulehner W (2014) Geometry+ simulation modules Implementing isogeometric analysis. PAMM 14(1):961–962
31. Mantzaflaris A, Verhelst H, Möller M, Karampatzakis C, Takacs S, Imperatore S, Weinmüller P, Vogl J, Li J, Ji Y, Matucci M, ALimkilde, Lee J, Scholz F, Mokriš D, Sogn J, Shamanskiy A, Vollebregt E, Partow A, rschneckenleitner, Rick, Kohl N, Zwar J, DaHofa, roeltielen24, Bressan A, andre caldas, PWeinmueller, Weiner H, Farahat A (2025) Geometry + simulation modules (g+smo) - v25.01.0. https://doi.org/10.5281/zenodo.15231873
32. Spenke T, Hosters N, Behr M (2020) A multi-vector interface quasi-newton method with linear complexity for partitioned fluid-structure interaction. Comput Methods Appl Mech Eng 361:112810. https://doi.org/10.1016/j.cma.2019.112810
33. Delft High Performance Computing Centre (DHPC) (2024) DelftBlue Supercomputer (Phase 2). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2
34. Weller HG, Tabor G, Jasak H, Fureby C (1998) A tensorial approach to computational continuum mechanics using object orientated techniques. Comput Phys 12(6):620–631. https://doi.org/10.1063/1.168744
35. Rüth B, Uekermann B, Mehl M, Birken P, Monge A, Bungartz H (2021) Quasi-newton waveform iteration for partitioned surface-coupled multiphysics applications. Int J Numer Meth Eng 122(19):5236–5257. https://doi.org/10.1002/nme.6443
36. Honnerová H, Bastl B, Turnerová E (2025) gsIncompressibleFlow: incompressible flow module for G+Smo. https://github.com/gismo/gsIncompressibleFlow. Accessed: 2024
37. Shamanskiy A, Simeon B (2021) Mesh moving techniques in fluid–structure interaction: robustness, accumulated distortion and computational efficiency. Comput Mech 67:583–600. https://doi.org/10.1007/s00466-020-01950-x
38. Turek S, Hron J (2006) Featflow - fsi benchmark tests. https://wwwold.mathematik.tu-dortmund.de/~featflow/en/benchmarks/cfdbenchmarking/fsi_benchmark/fsi_tests/fsi_fsi_tests.html. Accessed: 2025-05-13
39. Giannelli C, Jüttler B, Kleiss SK, Mantzaflaris A, Simeon B, Špeh J (2016) Thb-splines: an effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis. Comput Methods Appl Mech Eng 299:337–365. https://doi.org/10.1016/j.cma.2015.11.002
40. Verhelst HM (2024) Isogeometric analysis of wrinkling. In: Dissertation (tu delft), Delft University of Technology
41. Kiendl J, Bletzinger KU (2009) Linhard J, Wüchner, R Isogeometric shell analysis with kirchhoff-love elements. Comput Methods Appl Mech Eng 198(49):3902–3914. https://doi.org/10.1016/j.cma.2009.08.013
42. Ji Y, Ying-Ying Yu, Wang M-Y, Zhu C-G (2021) Constructing high-quality planar nurbs parameterization for isogeometric analysis by adjustment control points and weights. J Comput Appl Math 396:113615. https://doi.org/10.1016/j.cam.2021.113615