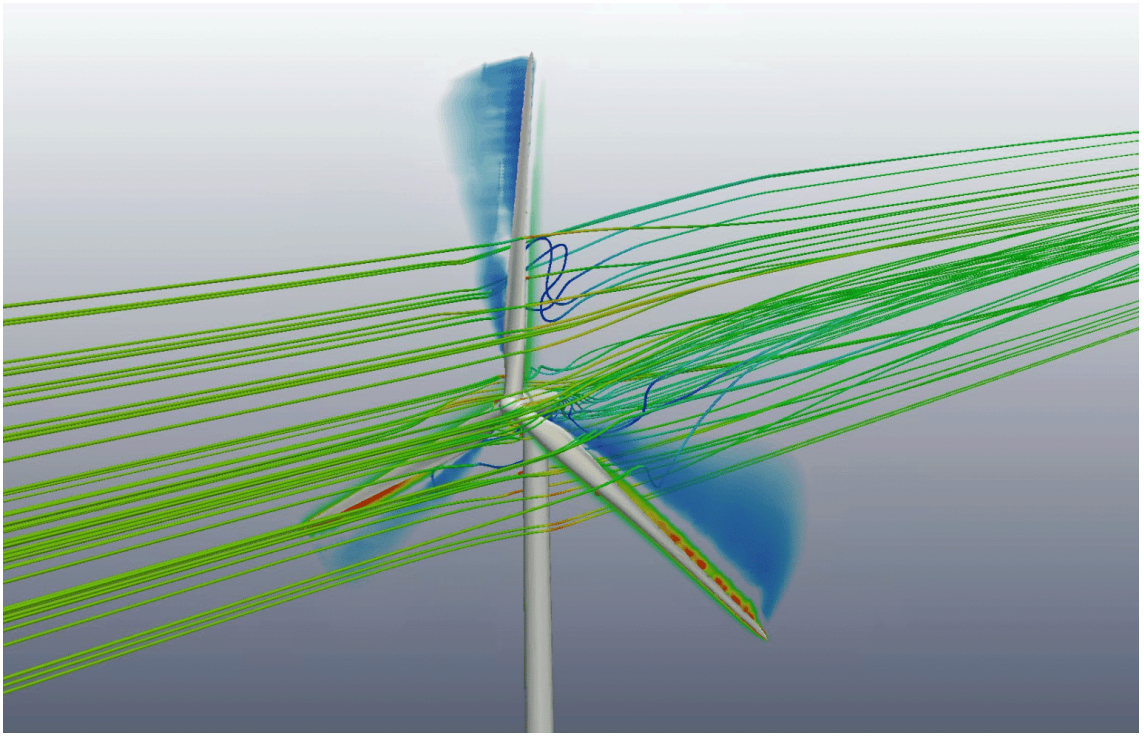
Gecko

Design for IGA-type discretization workflows

Gecko Technical Report 1

DC04 - Juan Ignacio Camarotti



This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101073106
Call: HORIZON-MSCA-2021-DN-01

Funded by the European Union

Deliverable Information sheet

Version	Date	Author	Document history/approvals
0		Alberto Tena	Draft version circulated to partners
1	20.01.2025	Juan I. Camarotti	Adding information
2	23.01.2025	Ricky Aristio	Revising
3	23.01.2025	Juan I. Camarotti	Final document

Executive summary

This report outlines the progress and key findings of ongoing PhD research focused on developing innovative strategies for partitioned multi-disciplinary simulations, particularly in the context of Fluid-Structure Interaction (FSI) problems. The study addresses the challenges of coupling different discretizations in FSI, with an emphasis on advancing data transfer methods and enhancing simulation accuracy across complex technical systems.

A central focus of this research is the development of robust and efficient co-simulation techniques, leveraging Isogeometric B-Rep Analysis (IBRA) for its CAD-based surface representation. By integrating IBRA with other solver types, the study enables flexible combinations of discretization approaches tailored to diverse physical domains. Among the notable contributions is the introduction of advanced data transfer operators, including the Mortar Mapper, designed for seamless coupling of body-fitted and unfitted discretizations.

Additionally, the research introduces a novel solution to the singularity problem encountered when imposing strong boundary conditions on unfitted meshes. This solution also addresses the challenge of mapping a field between unfitted meshes, enabling more accurate and stable simulations in scenarios involving non-conforming discretizations, such as low order FEM discretizations (usually used for the fluid domain) and trimmed NURBS patches.

These methods, implemented within the Kratos-Multiphysics open-source framework, are validated through benchmark problems, demonstrating their potential to improve accuracy and computational efficiency.

This work provides a significant foundation for advancing partitioned simulations in engineering applications. Future efforts will focus on publishing the methodology for boundary condition enforcement on unfitted meshes and conducting comparative studies of mapping operators in FSI simulations. These advancements aim to support the design of innovative solutions in fields such as aerospace, biomechanics, and civil engineering.

Table of contents

List of figures

List of abbreviations

CAD	*Computer-aided Design*
EGM	*Extended Gradient Method*
FEM	*Finite Element Method*
FSI	*Fluid-Structure Interaction*
IBRA	*Isogeometric B-Rep Analysis*
IGA	*Isogeometric Analysis*
LHS	*Left-hand side*
MVQN	*Multi-vector Quasi-Newton*
PDE	*Partial Differential Equation*
RBF	*Radial Basis Functions*

Introduction

This report presents advanced computational methods for multiphysics simulations, focusing on partitioned approaches in Fluid-Structure Interaction (FSI) problems. These methods aim to address critical challenges in coupling different discretization techniques, ensuring robust and accurate simulations that meet the demands of complex industrial applications.

The study begins by contrasting monolithic and partitioned approaches in multiphysics simulations, highlighting the advantages, challenges, and practical implications of each strategy. It then delves into partitioned simulations of FSI problems, underscoring the importance of efficient and accurate data transfer operators in achieving stability and precision.

A comprehensive discussion is provided on key techniques for data mapping at the coupling interface, including Nearest Neighbour, Nearest Element and Mortar Mapper. Special emphasis is placed on the Mortar Mapper, which has been effectively applied to both body-fitted and unfitted discretizations. This operator is recognized as a reliable solution for coupling different discretizations, facilitating seamless integration across domains.

The report also includes a case study on partitioned simulations of IsoGeometric Analysis (IGA) multipatch coupling using the Mortar Mapper, with specific attention to challenges associated with trimming. A novel solution is introduced for mapping fields in unfitted discretization scenarios, addressing singularity issues and enabling simulations between non-conforming meshes. Benchmark results from FSI problems validate the effectiveness of these methods, demonstrating their potential to improve simulation accuracy and computational efficiency.

By advancing methodologies for partitioned multiphysics simulations, this work contributes to a deeper understanding and broader applicability of these techniques in solving complex engineering problems. The findings have significant implications for fields such as aerospace, biomechanics, and civil engineering, where reliable and efficient simulations are critical to innovation and performance.

1. Monolithic vs Partitioned Approaches in Multiphysics Simulations

In multiphysics problem-solving, two primary approaches are widely used: monolithic and partitioned strategies.

Monolithic Approach

The monolithic approach involves solving all governing equations simultaneously within a unified framework. This method leads to a more stable and robust formulation, as all physical interactions are accounted for within a single system of equations.

However, the monolithic strategy often results in a large, complex global system that can be difficult to solve, especially as the size of the problem increases. As the global system grows, the conditioning of the system tends to worsen, making it more computationally demanding. Despite these challenges, the monolithic approach offers high accuracy and robustness, making it well-suited for certain complex multiphysics problems.

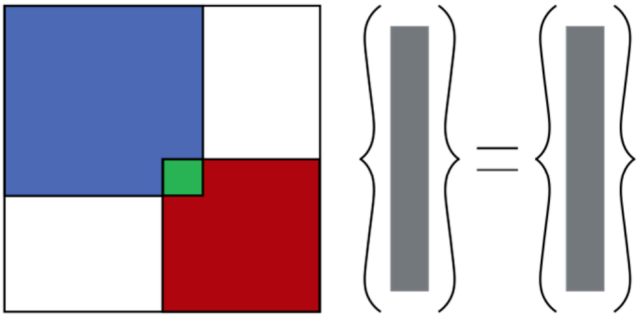


Figure 1: *System Matrix Level of the Monolithic Approach*

Partitioned Approach

In contrast, the partitioned approach separates the problem into distinct subsystems, solving each independently while coupling them through data exchange, often in the form of boundary conditions.

This approach allows for the reuse of existing single-field solvers, a key advantage in the co-simulation philosophy, where different physical domains are solved independently. The partitioned method is modular, flexible, and scalable, enabling the integration of different solvers for various disciplines. However, it is often less robust than the monolithic approach, as the coupling between subsystems may introduce stability challenges, particularly in highly coupled or stiff problems. Despite this, partitioned approaches are more flexible and allow for easier integration of specialized solvers, making them valuable for complex multidisciplinary simulations.

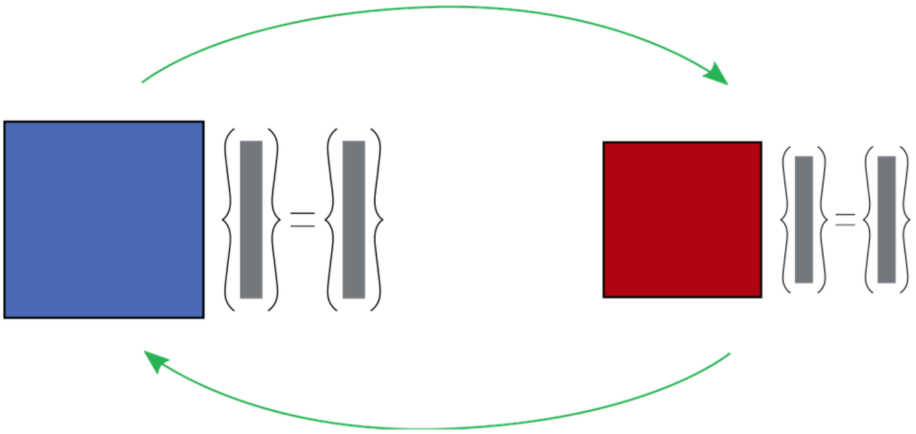


Figure 2: *System Matrix Level of the Partitioned Approach*

2. Partitioned Simulation of FSI Problems

Fluid-structure interaction (FSI) simulations are essential for accurately modeling and predicting the behavior of systems where fluids and structures interact dynamically. It plays a critical role in various engineering fields, including aerospace, biomechanics, and civil engineering, where structural integrity and performance depend on fluid forces. By capturing the mutual influence between fluids and structures, FSI enables the design of safer and more efficient systems, such as aircraft wings, blood flow in arteries, and bridges subjected to wind forces. Without FSI analysis, critical phenomena like vibrations, instabilities, and failure risks may be overlooked, leading to inaccurate predictions and potential structural failures.

FSI, like any multiphysics problem, can be solved using either a monolithic or partitioned approach. Monolithic methods solve the fluid and structural equations simultaneously, ensuring strong coupling but requiring complex solver development. The Partitioned method, on the other hand, is a computational approach where the fluid and structural domains are solved independently using specialized solvers, which are coupled to exchange key data at their interface.

The structural solver provides the displacement field, defining the motion and deformation of the structure, which is then used by the fluid solver to update the fluid domain and enforce the moving boundary conditions. Conversely, the fluid solver calculates the pressure field (and sometimes shear stresses), which acts as the load applied to the structure. This bidirectional exchange of displacements and pressure fields ensures the mutual influence of the fluid and structure, enabling the accurate simulation of their interaction.

Partitioned FSI offers flexibility and modularity by integrating existing solvers without significant modification, but it relies heavily on effective data transfer and coupling algorithms to maintain stability and accuracy, particularly in cases involving strong interactions or large deformations. The data transfer between the fluid and structural domains is accomplished through various data transfer operators, which ensure that critical information, such as displacement and pressure fields, is exchanged correctly. These operators are vital for ensuring the robustness of the coupling, and their efficiency will be addressed shortly.

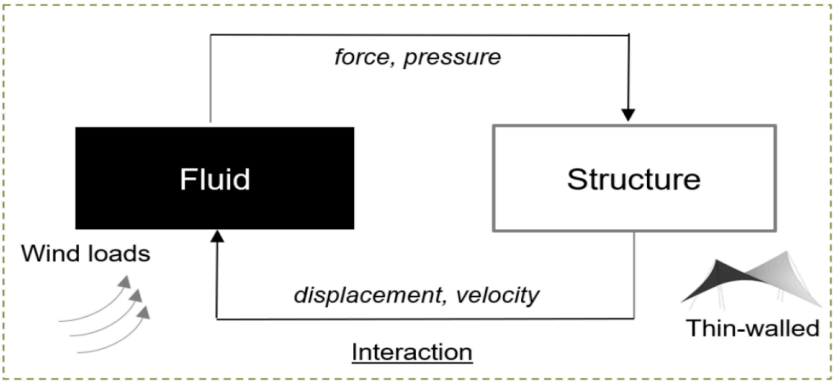


Figure 3: *Data Transfer in Partitioned FSI Simulations*

3. Fundamental Concepts of CoSimulation

CoSimulation is inherently a partitioned approach. In this context, multiple simulation tools or solvers, each specializing in a specific physical phenomenon, are coupled together to model complex systems involving interactions between different physical domains. Unlike the monolithic approach, where all phenomena are solved together within a single system, CoSimulation allows for the independent solution of each phenomenon.

These individual solvers are then coupled through specialized methods and tools to ensure accurate interaction between the domains. This partitioned nature of CoSimulation enables the reuse of existing solvers, offering flexibility and reducing development time, though it may require careful attention to ensure stability and accuracy, especially in cases involving strong interactions.

3.1 Building Blocks of CoSimulation

CoSimulation using the partitioned approach consists of several fundamental building blocks that are essential for ensuring effective and stable interactions between different solvers. These building blocks are synchronization, data exchange, and data transfer, as illustrated in Figure 4.

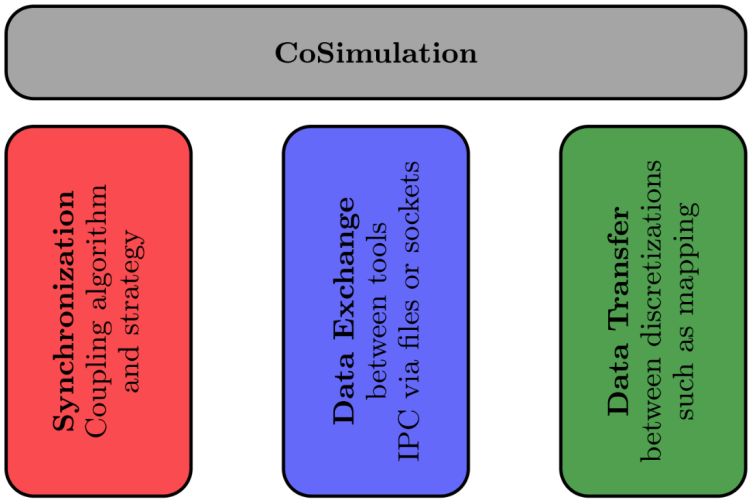


Figure 4: *Building Blocks of CoSimulation (Ref. [1])*

1) **Synchronization and Solution Techniques:** The first building block of CoSimulation, shown in Figure 4, is synchronization, which encompasses the coupling algorithm and strategy. This block determines the sequence in which the participating tools are applied, as well as the incorporation of additional components, such as relaxation techniques. The selection of these procedures greatly impacts the robustness, accuracy, performance, efficiency, and stability of the coupled simulation, making it crucial to choose them based on the specific problem being addressed.
2) **Data Exchange and Data Transfer:** Each coupling partner employs its own solution technique, chosen based on what best suits the specific application. Consequently, each tool handles and stores data in its own format. For CoSimulation, it is essential to transfer data between coupling partners. Often, this requires data exchange.

Additionally, the data must be converted from one format to another, a process referred to as data transfer in this work, as shown in column three of Figure 4. The flow of information proceeds from the origin to the destination, as illustrated in Figure 5.

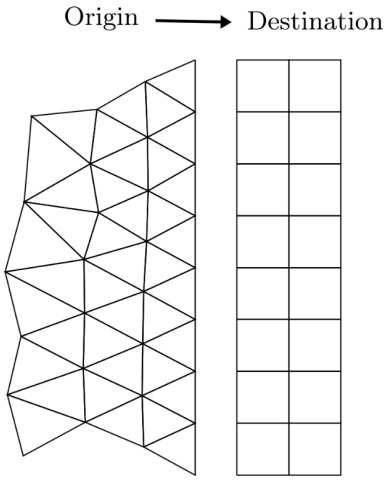


Figure 5: *Flow of information during data transfer between solution techniques, from origin to destination, from the origin to the destination (Ref. [1])*

4. Data Transfer Operators for CoSimulation

Data transfer operators, such as mapping operators, are fundamental components of partitioned solution strategies within CoSimulation frameworks. A variety of techniques are available for transferring data between different domains, including methods like Nearest Element, Nearest Neighbour, RBF Interpolation, and Mortar Mapping. It is important to emphasize that maintaining the non-intrusiveness of existing single-field codes is a critical requirement for co-simulation approaches. We will now discuss these data transfer operators, with particular focus on the mortar mapper.

4.1 Nearest Neighbour (Closest Point)

This mapper is perhaps one of the most straightforward mapping techniques: each point in the destination identifies its closest counterpart in the origin based on geometry and adopts its value.

The key benefits of this mapper are its simplicity in implementation and reliability. It only requires point clouds as input, rather than complete meshes, and the transfer matrix H consists solely of ones and zeros, making it highly memory-efficient. This mapper is particularly effective in distributed systems, as it only needs to search for neighboring points within different partitions.

On the downside, its accuracy is relatively low, especially with highly irregular meshes, potentially resulting in a step-like pattern on the destination if the discretizations vary too much. When applied to matching meshes, this method can be considered a special case of the nearest neighbor technique, enabling various optimizations like customized search settings.

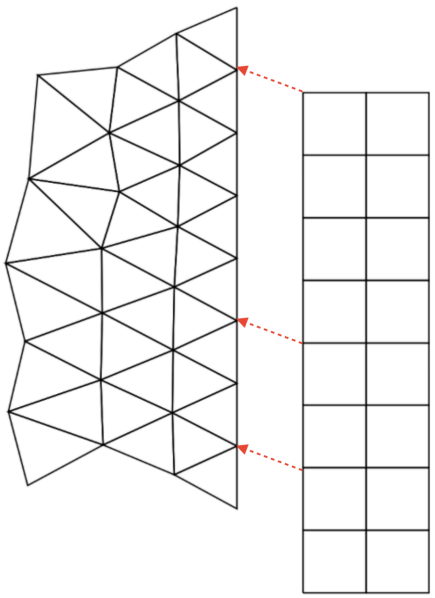


Figure 6: *Nearest Neighbour Mapper*

4.2 Nearest Element

This mapper utilizes the geometry of the origin interface by interpolating values through its shape functions, as described in [2]. Each point in the destination identifies the closest elements in the origin and projects onto them. The element with a valid projection and the shortest projected distance is selected, which is why this method is also known as the closest projection mapper. The assigned value at the destination is then computed by interpolating with the shape functions of the chosen element. Consequently, the transfer matrix H contains the shape function values evaluated at the projection points.

The main advantages of this mapper are its relatively simple implementation and its ability to interpolate values smoothly. However, a key drawback is that it requires elements and their shape functions as input. Since the method relies on projections, these can sometimes fail in practical applications, necessitating additional handling to enhance robustness.

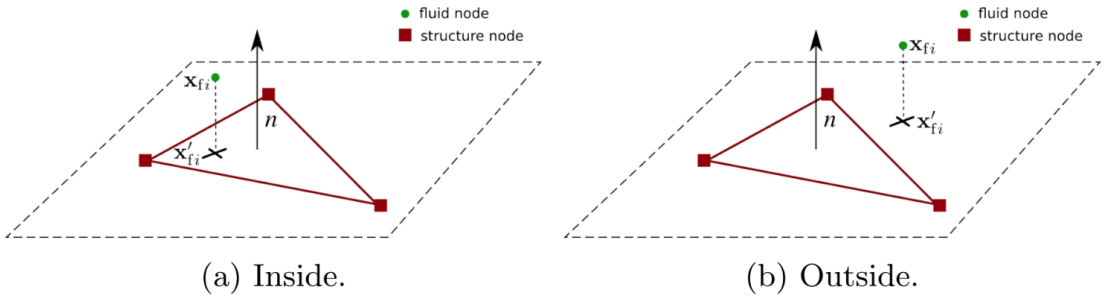


Figure 6: *Projection to the origin nearest element (dark red) in nearest element interpolation (Ref. [2])*

4.3 The Mortar Mapper for Body-Fitted and Unfitted Discretizations

One of the most commonly used data transfer operators in CoSimulation approaches is the "Mortar Mapper", which has been the primary focus of our work.

Let $\phi_O(x)$ be a known origin field and $\phi_D(x)$ the destination field. The Mortar Mapper can be derived from the minimization of the following functional $\varphi(\phi_D(x))$ (minimization of the L2-Norm Error between the origin and destination field):

$$\psi(\phi_D) = \frac{1}{2}\int_\Gamma \left(\phi_D(\boldsymbol{x}) - \phi_O(\boldsymbol{x})\right)^2 d\Gamma$$

As in any FEM-like approach, both fields are represented as a linear combination of nodal values and their associated shape functions. To minimize the functional, its stationarity with respect to the nodal values of the destination field is imposed. This leads to a minimization problem that can be expressed in the following matrix form:

$$\frac{\partial \varphi(\phi_D)}{\partial \phi_{Dj}} = 0 \Rightarrow \int_{\Gamma_D} N_D \, N_D^T \, d\Gamma_D \, \mathbf{\Phi_D} = \int_{\Gamma_{O\to D}} N_D \, N_O^T \, d\Gamma_{OD} \, \mathbf{\Phi_O}$$

$$\mathbf{M_{DD}\Phi_D} = \mathbf{M_{D0}\Phi_O} \Rightarrow \textit{Linear system to solve}$$

The primary challenge in the mortar mapper is determining the integration domain Γ_{OD}, which represents the overlapping region between a destination element and a neighboring origin element. The following subsections will outline the procedure for defining this integration domain across various discretization scenarios.

After implementing this mapper in the Kratos Multiphysics framework for various discretization techniques, including FEM and IGA, we began applying this data transfer operator to increasingly complex scenarios. Initially, we focused on partitioned multipatch coupling within the IGA context, followed by its application to fluid-structure interaction problems.

4.3.1 1D FEM-FEM Interface

In the 1D FEM-FEM case, each node from the destination domain is projected onto the origin, forming a new intersecting domain. Integration points are then positioned within each resulting interval. Graphically:

$$proj_{\,d\to o}(\xi,\eta) \Rightarrow (\xi,\eta)_d \xrightarrow{map} (x,y,z)_d \xrightarrow{project} (x,y,z)_o \xrightarrow{map} (\xi,\eta)_o$$

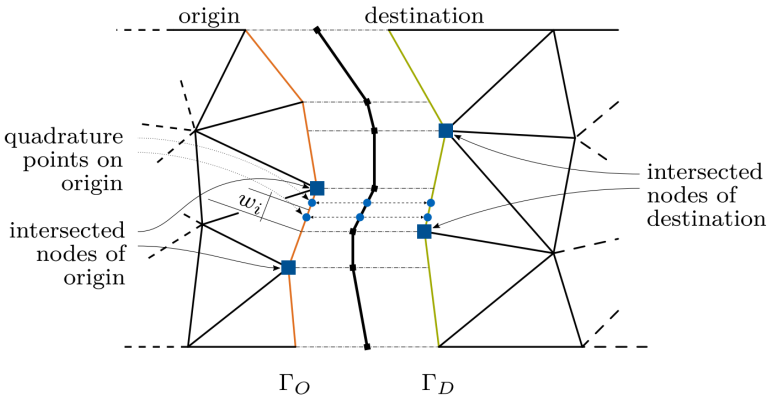
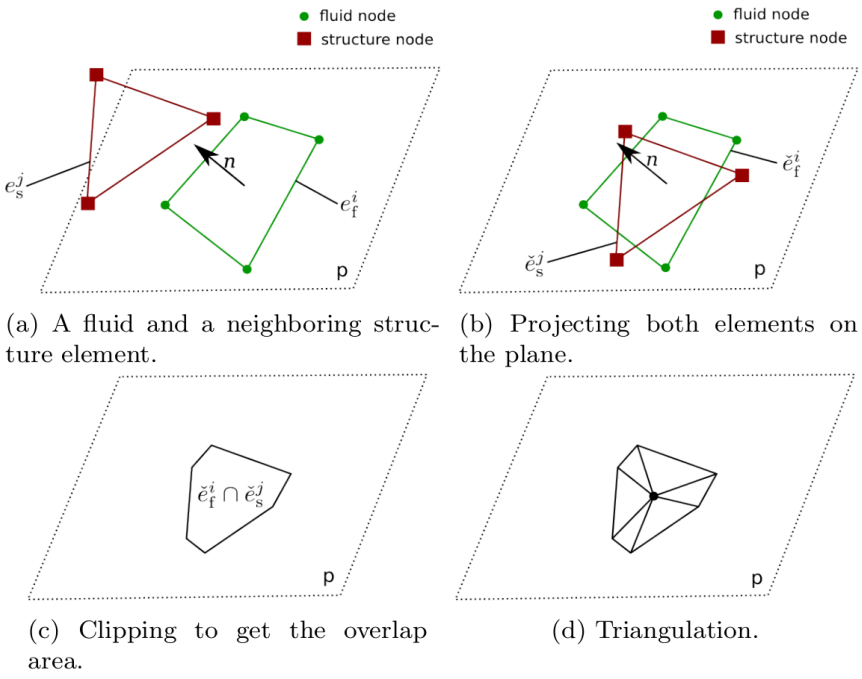


Figure 7: *Definition of the integration domain for the 1D FEM-FEM interface (Ref. [3])*

4.3.2 2D FEM-FEM Interface

In the 2D FEM-FEM interface case, each element from the destination domain is projected onto the origin elements, and their intersection is determined to obtain the overlap area. The resulting geometry is typically a polygon, which is then triangulated. Integration points are subsequently placed within each triangle. Graphically:



(a) A fluid and a neighboring structure element.

(b) Projecting both elements on the plane.

(c) Clipping to get the overlap area.

(d) Triangulation.

Figure 8: *Definition of the integration domain for the 2D FEM-FEM interface (Ref. [2])*

4.3.3 1D IBRA-IBRA or IBRA-FEM Interface

The process of determining the intersective integration domain in the 1D Isogeometric B-Rep Analysis (IBRA) mortar mapper consists of three main steps. First, the knots from the parameter space of the slave patch (destination) are projected onto the parameter space of the master patch (origin) to establish a geometric relationship between the two. Next, integration points are defined within the parameter space of the master patch, ensuring proper numerical integration over the overlap region. Finally, these integration points are mapped back from the master domain's parameter space to the parameter space of the slave domain, effectively defining the intersective integration domain needed for accurate coupling.

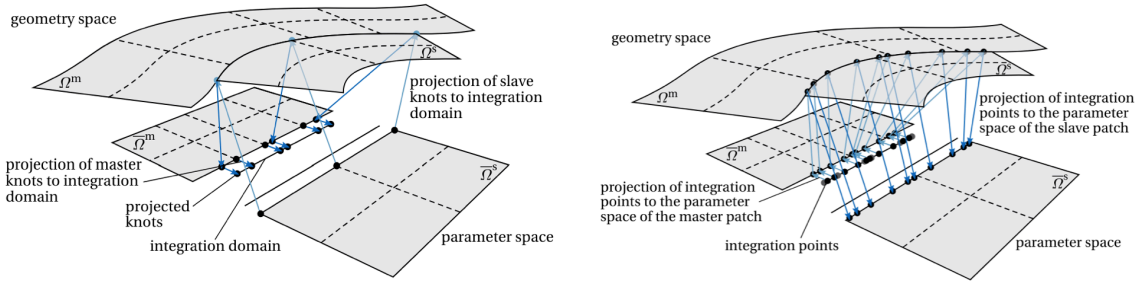


Figure 9: *Definition of the integration domain for the 1D IBRA-IBRA interface (Ref. [5])*

4.3.4 2D IBRA-FEM Interface

A particularly relevant scenario for 3D Fluid-Structure Interaction (FSI) simulations arises when the structural domain is discretized using a high-order NURBS multipatch surface, while the fluid domain is represented by a low-order finite element or finite volume mesh. In such cases, it is crucial to define an appropriate integration domain between the origin (fluid domain) and the destination (structural domain) for the mortar mapper. This process involves several steps to ensure accurate coupling between the two domains.

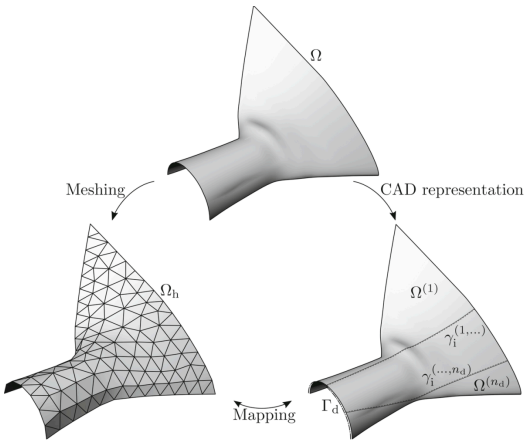


Figure 10: *Low order FEM discretization (fluid) and CAD surface (structure) (Ref. [4])*

First, the finite element nodes from the fluid domain are projected onto the parameter space of the NURBS patch, and the corresponding elements are reconstructed. For elements that are only partially projected into the patch, the finite element edges are clipped using the tessellated trimming curves that define the patch's boundary, with the help of some clipping algorithm. Next, the projected finite elements are clipped with the knot lines of the patch's parametric space, followed by a simple triangulation of the resulting geometry. Finally, integration points are placed in the master domain (fluid domain) and then projected back onto the destination finite elements (structural domain) to complete the coupling process. These steps ensure that the integration domain is well-defined for the mortar mapper, enabling an effective exchange of data between the fluid and structural solvers in FSI simulations.

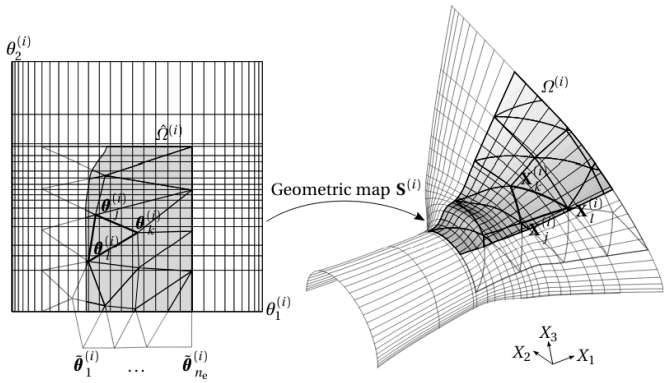
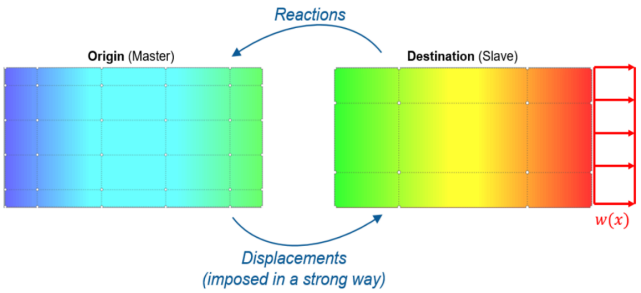


Figure 11: *Projection of the low order element to the CAD surface (Ref. [4])*

5. Partitioned Simulation of IGA Multipatch Coupling Using the Mortar Mapper

After integrating this mapper into the Kratos Multiphysics framework [6] for various discretization techniques, such as FEM and IGA, we started applying this data transfer operator to progressively more complex scenarios. Our initial focus was on partitioned multipatch coupling within the IGA framework.

The algorithm for this partitioned simulation of multipatch coupling is illustrated in the following figure.



Algorithm 1 Partitioned Multipatch Coupling Algorithm

1: **Solve** the origin domain.
2: **Map the displacements** from the origin domain to the destination domain.
3: **Impose the mapped displacements** as strong Dirichlet boundary conditions (BCs) and **solve** the destination domain.
4: **Map the reactions** from the destination domain back to the origin domain.
5: **Compute the residual** and check for convergence.
6: **while** convergence at the interface is not achieved **do**
7: Repeat Steps 1–5.
8: **end while**
9: Advance in time.

Figure 12: *Partitioned Multipatch Coupling Algorithm*

To utilize this partitioned framework for multipatch coupling, a simplified benchmark was designed to study various combinations of discretization strategies, including IGA, FEM, body-fitted (untrimmed IGA), and unfitted (trimmed IGA) configurations. This analysis aimed to evaluate the advantages, drawbacks, and potential challenges associated with these different discretization approaches.

In general, discretization techniques can be classified into two main categories: body-fitted and unfitted discretization. Before exploring various examples, we would like to review the difference between body-fitted and unfitted discretizations.

Body-fitted discretization

A body-fitted discretization refers to a numerical discretization technique where the computational mesh conforms to the geometry of the domain. The mesh elements align with the boundaries of the physical object, ensuring accurate representation of the geometry. In the case of FEM body-fitted meshes, considering that the shape functions are interpolatory, the boundary conditions are imposed in a strong manner. However, in IGA, where the shape functions are

typically non-interpolatory (e.g., NURBS), boundary conditions are generally imposed weakly even when the discretization is body-fitted (untrimmed). The following figure represents an example of a body-fitted discretization of a geometry.

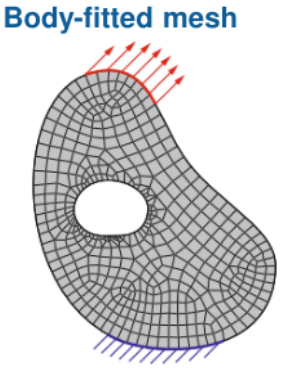


Figure 13: *Body-fitted discretization of an arbitrary geometry*

Unfitted discretization

An unfitted discretization (also called immersed or embedded methods) is a technique where the computational mesh does not conform to the geometry. Instead, the geometry is embedded into the background mesh, and boundary conditions are typically imposed weakly, meaning they are enforced through different methods such as penalty methods, Lagrange multipliers, or Nitsche's method.

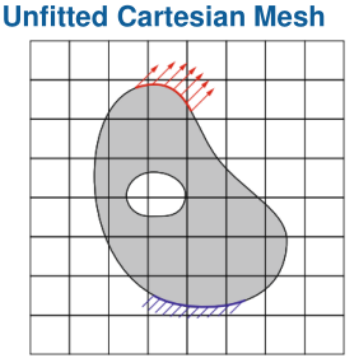


Figure 14: *Unfitted discretization of an arbitrary geometry*

Now, we would like to introduce the proposed example, which is essentially a two-patch plate under membrane action:

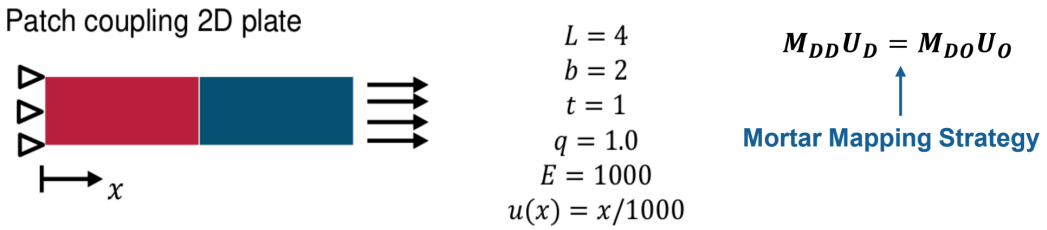


Figure 15: *Problem Definition of the Proposed Benchmark*

In this case, the analytical solution is represented by a linear displacement field in the x-direction:

$$u(x) = \frac{x}{1000}[m] \rightarrow u(x = 4) = 0.004[m]$$

Now, we will proceed to analyze the different scenarios in detail, with particular focus on the numerical results obtained from the partitioned simulation and the well-posedness of the mapping matrix (indicated by the condition number of the right-hand side mapping matrix), which is essential for the data transfer between the meshes.

Before proceeding, it is important to note that IBRA meshes [7] with trimming essentially represent an unfitted discretization, as the shape functions used in the analysis are primarily those defining the NURBS surface.

5.1 Case 1: IBRA Untrimmed - IBRA Untrimmed

In this scenario, when the discretizations of both the origin and destination domains are body-fitted, the data transfer problem is well-defined (the matrix M_{DD} is not singular), allowing us to accurately compute the nodal values in the destination domain.

The following figure presents the color map for the displacement field obtained from the partitioned simulation, with the a continuous displacement field and the displacement at the tip aligning as expected with the analytical solution:

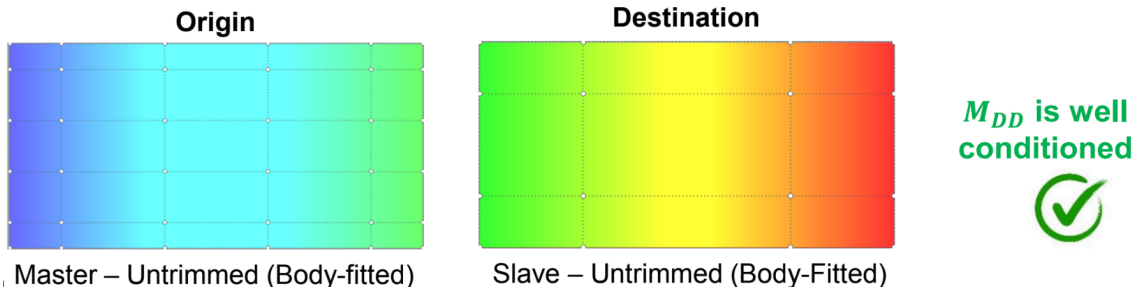


Figure 16: *Displacement field u(x) for the Case 1*

5.2 Case 2: IBRA Trimmed - IBRA Untrimmed

In this second scenario, where the origin domain is unfitted, and the destination domain is body-fitted, the data transfer problem remains well-defined (the matrix M_{DD} is not singular).

The following figure presents the color map for the displacement field obtained from the partitioned simulation, with the a continuous displacement field and displacement at the tip aligning as expected with the analytical solution:

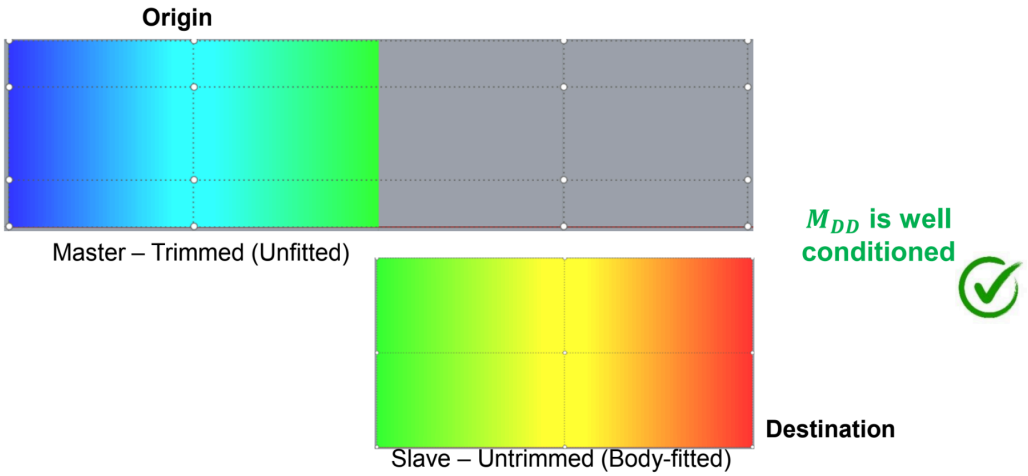


Figure 17: Displacement field u(x) for the Case 2

5.3 Case 3: IBRA Trimmed - IBRA Trimmed

In this final scenario, where the origin domain is body-fitted and the destination domain is unfitted, the matrix M_{DD} becomes singular, rendering the problem ill-defined.

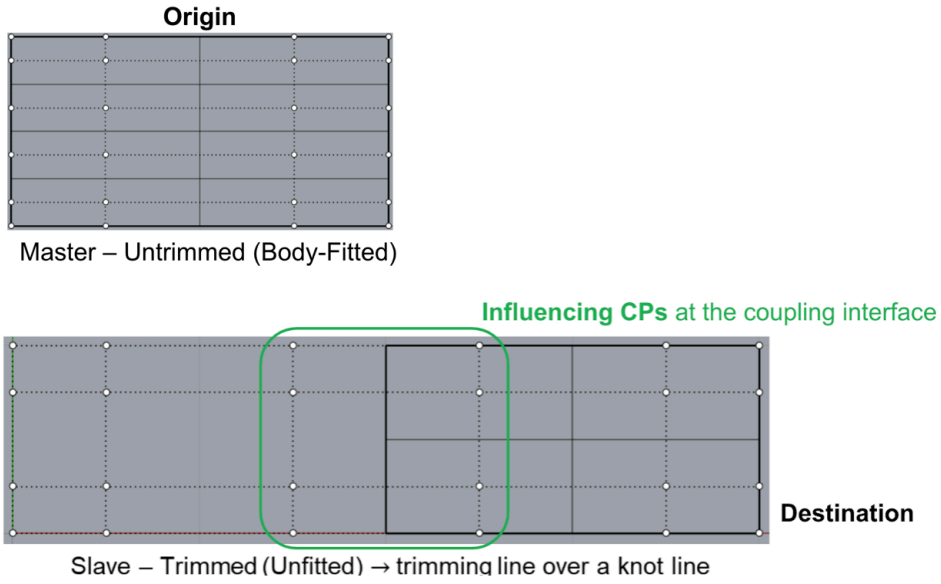


Figure 18: Case 3

In this scenario, the mapping matrix M_{DD} that needs to be inverted is singular. Specifically, when the trimming line in the destination domain aligns with a knot line, identical columns and rows can be observed, leading to redundancy.

$$M_{DD} = \int_{\Gamma_D} N_D \, N_D^T \, d\Gamma_D$$

$$\mathbf{M_{DD}} = \begin{bmatrix} 0.025 & 0.025 & 0.0145833 & 0.0145833 & 0.00208333 & 0.00208333 & 0 & 0 \\ 0.025 & 0.025 & 0.0145833 & 0.0145833 & 0.00208333 & 0.00208333 & 0 & 0 \\ 0.0145833 & 0.0145833 & 0.0416667 & 0.0416667 & 0.025 & 0.025 & 0.00208333 & 0.00208333 \\ 0.0145833 & 0.0145833 & 0.0416667 & 0.0416667 & 0.025 & 0.025 & 0.00208333 & 0.00208333 \\ 0.00208333 & 0.00208333 & 0.025 & 0.025 & 0.0416667 & 0.0416667 & 0.0145833 & 0.0145833 \\ 0.00208333 & 0.00208333 & 0.025 & 0.025 & 0.0416667 & 0.0416667 & 0.0145833 & 0.0145833 \\ 0 & 0 & 0.00208333 & 0.00208333 & 0.0145833 & 0.0145833 & 0.025 & 0.025 \\ 0 & 0 & 0.00208333 & 0.00208333 & 0.0145833 & 0.0145833 & 0.025 & 0.025 \end{bmatrix}$$

Mathematically, this implies the existence of infinitely many possible solutions for the destination nodal values that satisfy the original minimization problem.

Motivated by the challenges associated with mapping a field between unfitted discretizations, a novel algorithm was developed to address this issue within the context of mapping problems. This approach evolved into a general framework for strongly imposing Dirichlet boundary conditions in unfitted problems, which will be presented in an upcoming publication.

5.3.1 Physical Interpretation of the Singularity for Unfitted Discretizations

Consider a simplified example involving an unfitted linear FEM discretization for a bar structure. In this case, the goal is to impose a Dirichlet boundary condition at the left end of the bar (middle of the first element). The governing equation and boundary conditions for this problem are as follows:

$$\frac{d}{dx}\left(EA\frac{du}{dx}\right) = -p(x) \qquad u_{\Gamma_D} = \overline{u} \qquad EA\frac{du}{dx} = F$$

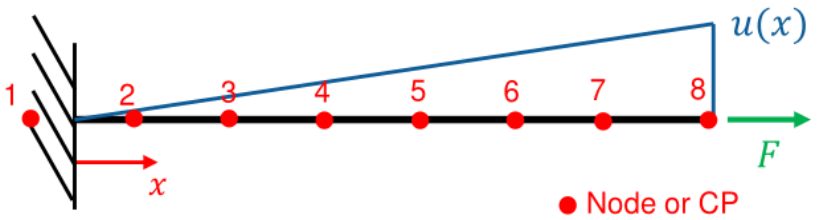


Figure 19: Illustration of the simple bar structure

In this particular problem, $p(x) = 0$ and $u_{\Gamma_D} = 0$. It is possible to interpret the imposition of Dirichlet boundary conditions as the minimization of the following functional which represents the L2-norm error between the desired value at the boundary and the field $u(x)$:

$$\varphi(u) = \int_{\Gamma_D} (u(x) - \overline{u})^2 \, d\Gamma \rightarrow \frac{d\varphi}{du_i} = 0 \rightarrow \int_{\Gamma_D} N N^T d\Gamma = \int_{\Gamma_D} N \, \overline{u}(\mathbf{x}) \, d\Gamma$$

In this simple case, Γ_D is a single point. Finally, we obtain:

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \longrightarrow \textbf{Underdetermined System (infinite solutions)}$$

Focusing on the first element, it is clear that there are infinitely many ways to assign values to u_1 and u_2 so that the boundary condition is satisfied.

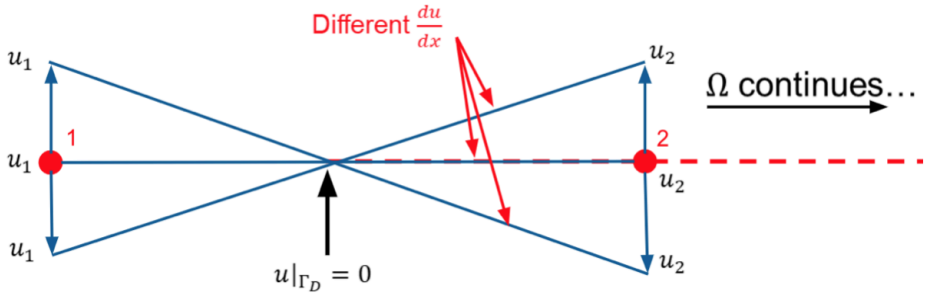


Figure 20: *Infinite possibilities for the imposition of the BC*

The concept of imposition of strong Dirichlet boundary conditions as a minimization problem is generalized. As already mentioned above, the Dirichlet boundary condition can be interpreted as the minimization of the $L2$ -norm error between the field and the imposed value. Back to the two different scenarios depending on the discretization technique, in body-fitted discretization, with this approach, the linear system is well-posed, and solving it directly ensures that the solution exactly matches the Dirichlet boundary condition at the specified points, in the unfitted approach, the resulting linear system is ill-posed, leading to infinitely many solutions (exactly the same as the bar structure case presented before).

The key aspect here is that the infinite solutions differ only in the gradient of the field between u_{Γ_D} and the second node. This raises the question: what if this issue is addressed by enforcing an additional constraint on the gradient within the first element (the trimmed element)?

6. The Extended Gradient Method and its Algorithm

The main idea of the proposed extended gradient method (EGM) is to enable the strong imposition of Dirichlet BCs on unfitted meshes without altering the underlying weak form of the PDE, resulting in a non-intrusive method. In this new method, a novel functional to define strong Dirichlet boundary conditions on unfitted meshes is introduced. The functional incorporates two key components: the first term is exactly equal to the original formulation, and the second term is the regularization or stabilization term. This term ensures consistency of the normal gradient within trimmed elements. The gradient is an *approximation* (e.g., interpolation) of the actual normal gradient on these elements at the iteration k. In the first iteration, the gradient is assigned to zero. The newly proposed functional is expressed as follows:

$$\varphi(u) = \int_{\Gamma_D} (u(x) - \overline{u})^2 \, d\Gamma + Stabilization\ Term$$

The algorithm below summarizes the procedure of EGM:

Algorithm 2 The Extended Gradient Method

1: **Classify elements** in the background mesh as active, inactive, or intersected.
2: **Assemble the system of equations** for the physical problem, considering contributions only from active elements.
3: **Initialize:** Set $error = 1.0$ and define the tolerance $\epsilon > 0$ (tipically $\epsilon \approx 10^{-6}$)
4: **while** $error > \epsilon$ **and** $iteration < max$ **do**
5: **Solve the auxiliary problem:**

$$LHS\ \phi_{dir}^{k+1} = RHS^k \tag{1}$$

6: **Impose the coefficients** obtained from the auxiliary problem as strong Dirichlet boundary conditions (BCs).
7: **Solve the original problem** for the degrees of freedom with the updated BCs.
8: **Compute the relative error** between iterations:

$$error = \frac{\|\phi^{k+1} - \phi^k\|_2}{\|\phi^k\|_2} \tag{2}$$

9: Increment iteration counter.
10: **end while**
11: Advance in time.

The advantages and disadvantages of this method are summarized below:

- ☐ (+) Non-intrusive approach: Enables the application of Dirichlet boundary conditions in unfitted discretizations without modifying the underlying weak form of the PDE.
- ☐ (+) Improved conditioning of the LHS: Results in better conditioning of the left-hand side (LHS) matrix.
- ☐ (+) No Small Cut-Cell Instability.
- ☐ (+) Discretization-independent: Works seamlessly with different discretization methods and is highly generalizable.
- ☐ (+) Physics-agnostic: Can be applied to a wide range of physical problems without requiring problem-specific modifications.
- ☐ (+) More computationally efficient for explicit dynamics problems.

- ☐ (-) Iterative nature: The method requires solving the system iteratively, making it less computationally efficient compared to direct approaches used in intrusive methods.
- ☐ (-) Less computationally efficient than intrusive methods.
- ☐ (-) Challenges in higher-order schemes: In higher-order methods, such as Isogeometric Analysis (IGA), achieving an accurate approximation of the gradient within trimmed elements can be more challenging.

Due to confidentiality concerns, as this proposed method has not yet been published, this is the only information that can be provided at this time. Once the publication is released, all details will be shared.

7. Fluid-Structure Interaction Benchmarks

Research was conducted on 2D and 3D FSI benchmarks, employing FEM discretization for the fluid mesh and IBRA discretization for the structural elements. A body-fitted mortar mapper was utilized to transfer data between the non-conforming meshes. The coupling process was handled using a partitioned Gauss-Seidel algorithm, with an MVQN convergence accelerator applied to speed up the convergence.

Presented below are color graphs illustrating the velocity and displacement fields from the simulations conducted.

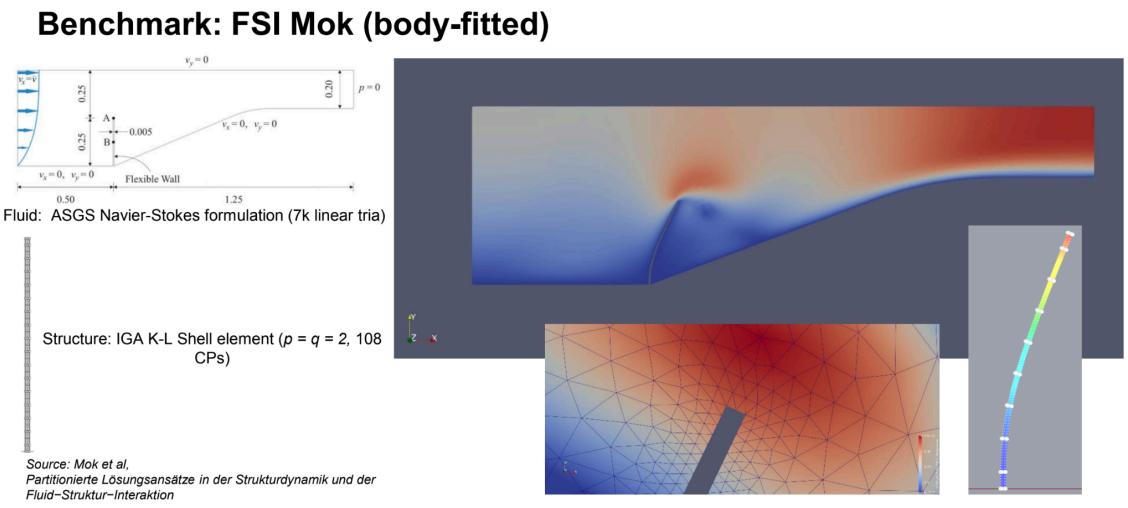


Figure 21: *FSI Mok Benchmark*

The following plot illustrates the x-displacements at two specific points within the structure:

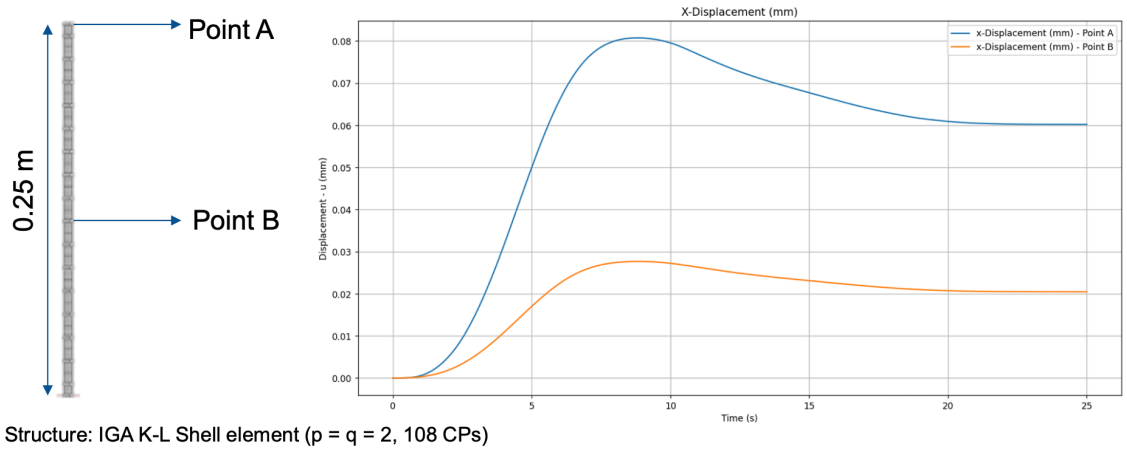


Figure 22: *x-displacements at two specific positions in the structure*

These values were compared with those from Kratos examples and the Mok paper, demonstrating good agreement:

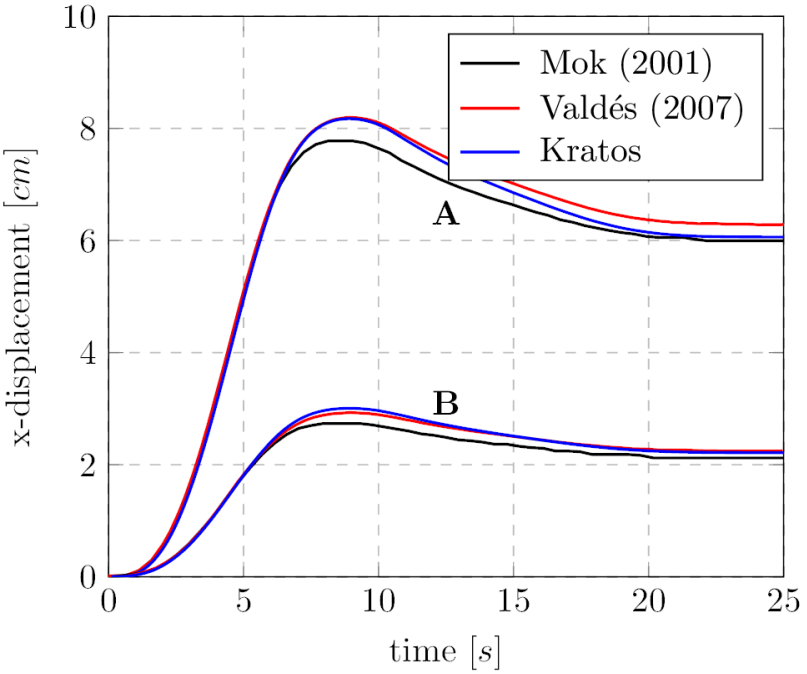


Figure 23: *Comparison of results for the Mok Benchmark across different sources*

The final two figures present the results obtained for the FSI Turek and Lid-Driven Cavity benchmarks:

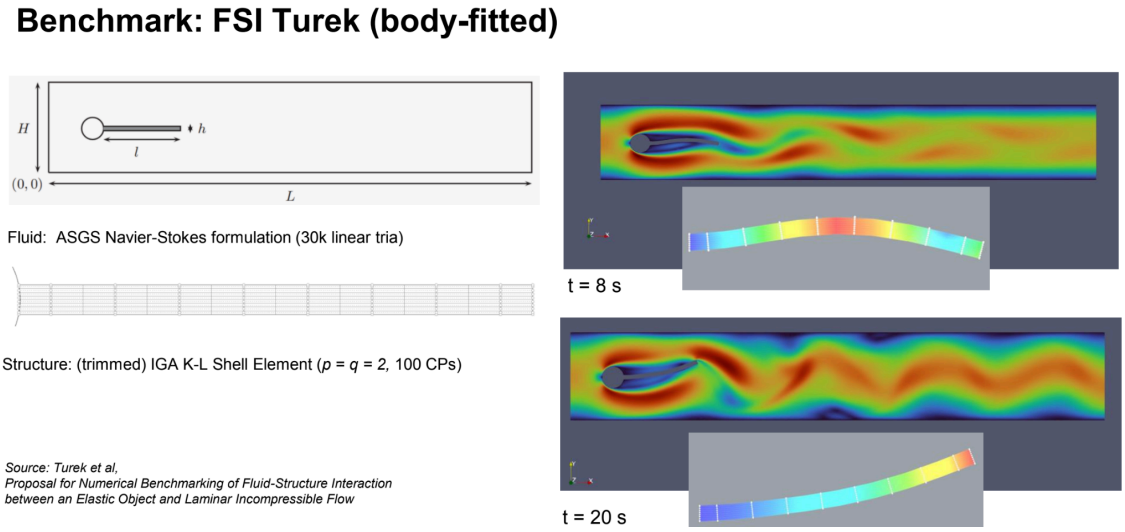
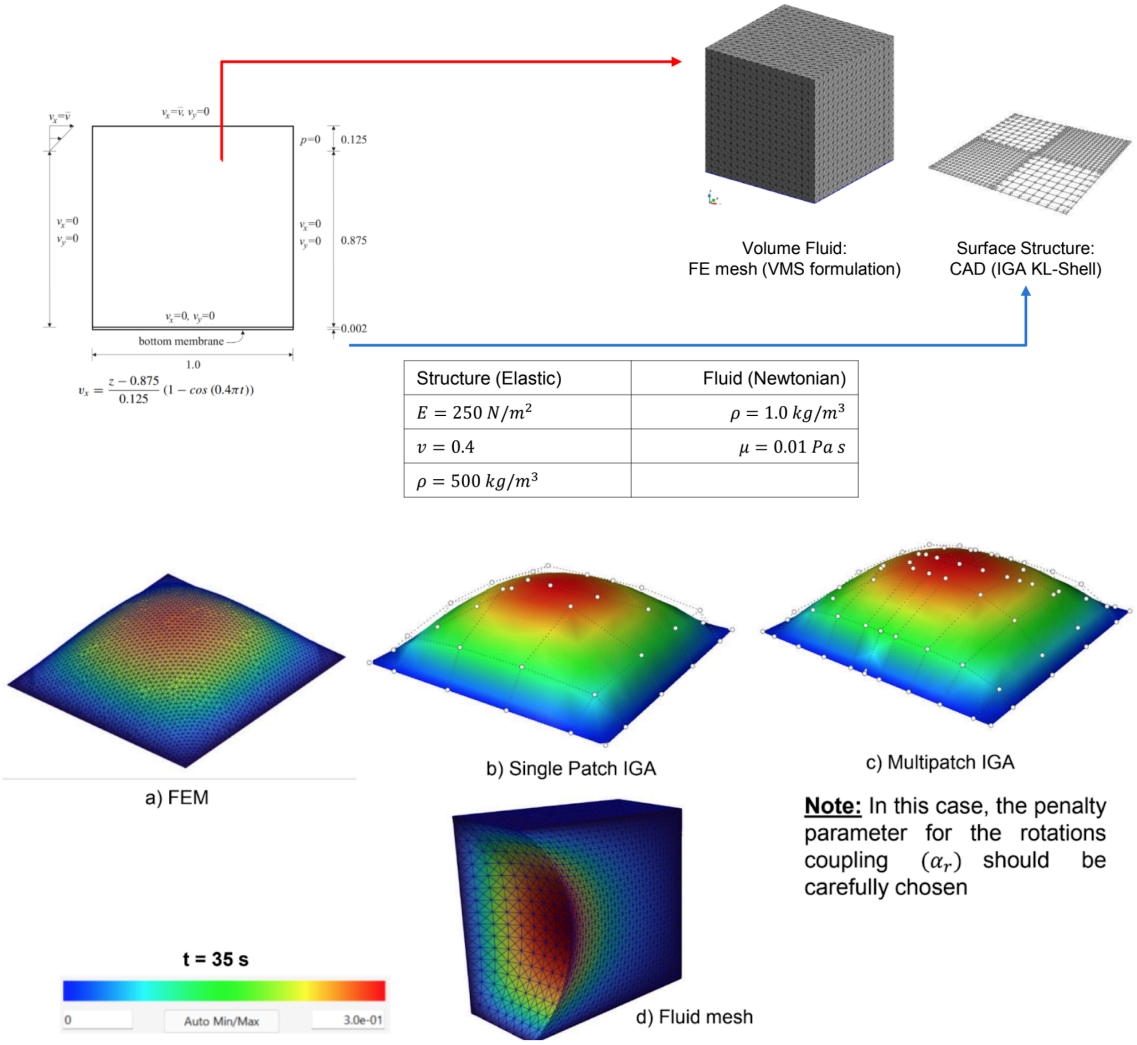


Figure 24: *FSI Turek Benchmark*

Benchmark: 3D lid driven cavity (body-fitted)



Structure (Elastic)	Fluid (Newtonian)
$E = 250\ N/m^2$	$\rho = 1.0\ kg/m^3$
$v = 0.4$	$\mu = 0.01\ Pa\ s$
$\rho = 500\ kg/m^3$	

Note: In this case, the penalty parameter for the rotations coupling (α_r) should be carefully chosen

Figure 25: *FSI 3D Lid Driven Cavity Benchmark*

8. Conclusions

This report highlights the progress made in developing flexible and efficient strategies for partitioned multi-disciplinary simulations, with a specific focus on Fluid-Structure Interaction (FSI) challenges. Key contributions include the implementation of advanced data transfer operators such as the Mortar Mapper and the exploration of robust coupling schemes that effectively integrate various discretization techniques, including Isogeometric B-Rep Analysis (IBRA) and Finite Element Methods (FEM).

By embedding these methods within the Kratos-Multiphysics framework, the research ensures accessibility and extensibility for the broader scientific community. Notably, the proposed solution to the singularity problem when imposing strong boundary conditions on unfitted meshes addresses a critical gap in enabling partitioned simulations between unfitted domains, a significant step forward in multi-physics simulations.

Future work will focus on publishing this new methodology for imposing strong boundary conditions on unfitted meshes. Additionally, research will be conducted to compare the performance of different mapping operators in the context of FSI simulations. These efforts aim to further refine the methodologies, improve computational efficiency, and expand their applicability to more complex engineering problems like the ones encountered in fields such as aerospace engineering, civil engineering or biomechanics.

This work paves the way for more accurate and versatile simulations in multi-disciplinary fields, offering impactful advancements in computational engineering.

9. REFERENCES

[1] Bucher, P. “CoSimulation and Mapping for Large Scale Problems”. Dissertation. Technical University of Munich, 2023.

[2] T. Wang. “Development of Co-Simulation Environment and Mapping Algorithms”. Dissertation. Technical University of Munich, 2016.

[3] Wilson, P., Teschemacher, T., Bucher, P., & Wüchner, R. (2021). Non-conforming FEM-FEM coupling approaches and their application to dynamic structural analysis. Engineering Structures, 241, 112342.

[4] Apostolatos, A., Emiroğlu, A., Shayegan, S., Péan, F., Bletzinger, K. U., & Wüchner, R. (2021). An isogeometric b-rep mortar-based mapping method for non-matching grids in fluid-structure interaction. Advanced Modeling and Simulation in Engineering Sciences, 8(1), 9.

[5] Bauer, A. M. CAD-integrated isogeometric analysis and design of lightweight structures. Doctoral dissertation. Technische Universität München, 2020.

[6] Dadvand, P., Rossi, R., & Oñate, E. (2010). An object-oriented environment for developing finite element codes for multi-disciplinary applications. Archives of computational methods in engineering, 17, 253-297.

[7] Teschemacher, T., Bauer, A. M., Oberbichler, T., Breitenberger, M., Rossi, R., Wüchner, R., & Bletzinger, K. U. (2018). Realization of CAD-integrated shell simulation based on isogeometric B-Rep analysis. Advanced Modeling and Simulation in Engineering Sciences, 5, 1-54.

[8] Mok, D. P. (2001). Partitionierte Lösungsansätze in der Strukturdynamik und der Fluid-Struktur-Interaktion.

[9] Turek, S., Hron, J., Razzaq, M., Wobker, H., & Schäfer, M. (2010). Numerical benchmarking of fluid-structure interaction: A comparison of different discretization and solution approaches (pp. 413-424). Springer Berlin Heidelberg.

[10] Valdés Vázquez, J. G. (2007). Nonlinear Analysis of Orthotropic Membrane and Shell Structures Including Fluid-Structure Interaction.