Fluid-Structure Interaction simulations involving multipatch Isogeometric membrane structures

Antonios Kamariotis*, Armagan Berkcan Erman*, Bodhinanda Chandra*,
Jun-Hyoung Kwon*, Rishith Ellath Meethal*

Abstract

Membrane structures have been widely used in many kinds of engineering purposes and architectural styles due to their special load carrying behavior. They are, however, very sensitive to changes in environmental conditions and susceptible to flow-induced effects as a consequence of their very low mass. Since the flow-induced excitation can sometimes be severely large, it is important to assess and analyze this multi-physics interaction phenomenon through a numerical simulation. Coming with this motivation, a surface-coupled, partitioned Fluid-Structure Interaction (FSI) scheme considering Aitken relaxation was employed. In the current project, the communication scheme between existing structural and fluid solver is implemented through EMPIRE, a framework which has been developed by the Chair of Structural Analysis at the Technical University of Munich to perform co-simulations between different tools for FSI simulations. Here, the membrane structure is discretized using IGA control points, while the fluid domain is discretized into a finite number of control volumes in an open-source software. In order to validate our works, two comparisons of test cases are presented. A good agreement between the numerical results demonstrates the success of the implemented approach.

Keywords: Fluid-structure interaction, Isogeometric analysis, membrane structures, partitioned approach

1 Introduction

The term "*Multiphysics*" stands for a computational discipline that studies multiple interacting physical phenomena in computer simulations. For instance, the fluid-structure interaction (FSI) is a study of mutual influence between a fluid in rest or in flow and a flexible or moving structure. In order to understand such complicated phenomena, analytical solutions to the model equations are impossible to obtain, whereas laboratory experiments are limited in scope and scale; thus in order to investigate the fundamental physics involved in the interaction, numerical simulations may be employed. Methods to efficiently and accurately model the behavior of these complex systems are indeed in great economic and environmental demands. In the context of membrane structures, which are widely used for civil engineering designs, FSI simulation is essential to predict the flow-induced vibration caused by the surrounding fluid flow as it sometimes can be considerably large. Coming with this motivation, in the current project, FSI simulations of multipatch membrane structures are investigated.

For the purpose of our structural analysis, Isogeometric analysis (IGA) is employed to discretize the structural behavior of our membrane structures. IGA is a recently introduced computational method which integrates classical finite element analysis into NURBS-based CAD design tools. The essential idea of IGA is that it reuses the basis functions used for the geometric modeling in CAD as an approximation of the solution field. Due to the advantageous properties of NURBS, one can enrich the solution space by

* Graduate Student, Computational Mechanics Program, Department of Civil, Geo, and Environmental Engineering, Technical University of Munich

specific refinement strategies while keeping the exact and smooth original geometry. By doing this, the entire process of meshing can be omitted and the two models for design and analysis can be merged into one. Consequently, the time and labor works used for meshing can be significantly reduced.

In the current project, the Isogeometric concept is applied to the FSI simulations which adopt the staggered two-way strongly coupled scheme. This means that the structure and fluid part should have their own independent solver and the communication between these two solvers should be implemented in an iterative manner. To do that, first, the pressure computed from the fluid solver is being transferred to the structure, where the displacement of the structure is being computed through a nonlinear structural solver. Afterwards, the computed displacement field is transferred back to the fluid solver as a response to the initially given pressure forces. Here, the discretization of the structures in IGA consists of single patch or multipatch membranes. In the multipatch case, the coupling of deformation between patches is important and unavoidable. In this project, the Nitsche and the Penalty method were adopted for the coupling. Meanwhile, in the fluid solver, the finite volume method is used under the framework of OpenFOAM. More specific aspects of the FSI and the implementation will be dealt in more detail later in section 2 and 3. Following those, section 4 highlights several numerical tests to validate our FSI communication scheme. In this chapter, we are comparing our numerical results for both single patch and multi-patch membranes considering different types of patch coupling methods. From those comparisons, we could get a very good agreement in simulating two different numerical examples. Finally, section 5 concludes and gives a summary of this report as well as discusses on directions for future work.

2 Underlying Theory

2.1 Structural Analysis

As mentioned earlier, the membrane structures in the current project are confronted using IGA. The main idea of this approximation method is to reuse the basis functions used for the geometric description in CAD (e.g. NURBS) as the shape functions of the geometric and the solution field. The following subsections will describe some theoretical details used in formulating the structural solver.

2.1.1 Geometric Description

NURBS The Non-Uniform Rational B-Splines (NURBS) functions are used for the description of our structural geometry. The term "*non-uniform*" in NURBS refers to the knot vector which is generally non-uniform, while the term "*rational*" refers to the basis functions used for approximating curves or surfaces from the given data points (normally called as control points). Compared to its predecessor methods, the Bezier and the B-Spline geometry, that use piece-wise polynomials, the NURBS geometry use piece-wise rational functions as it has to consider the partition of unity after introducing additional weightings w in its control points. The significant advantage of the rational basis functions is that they allow an exact representation of conic sections, which includes circles and ellipses [12]. We can define a NURBS surface and its basis functions as:

$$\mathbf{S}(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{ij}^{pq}(\xi,\eta)\mathbf{P}_{ij} \tag{1}$$

$$R_{ij}^{pq}(\xi,\eta) = \frac{N_{i,p}(\xi)N_{j,q}(\eta)w_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,p}(\xi)N_{j,q}(\eta)w_{ij}} \tag{2}$$

where, $N_{i,p}$ is the ordinary B-Spline basis functions at point i with polynomial degree p. In Figure 1, which is retrieved from [12], depending on the weight of each point, the NURBS curve or surface is pulled or pushed from the corresponding control point. If all control weights are equal, the rational functions reduce to the normal B-Spline functions. This means that a B-Spline is a special case of NURBS with equal control weights.

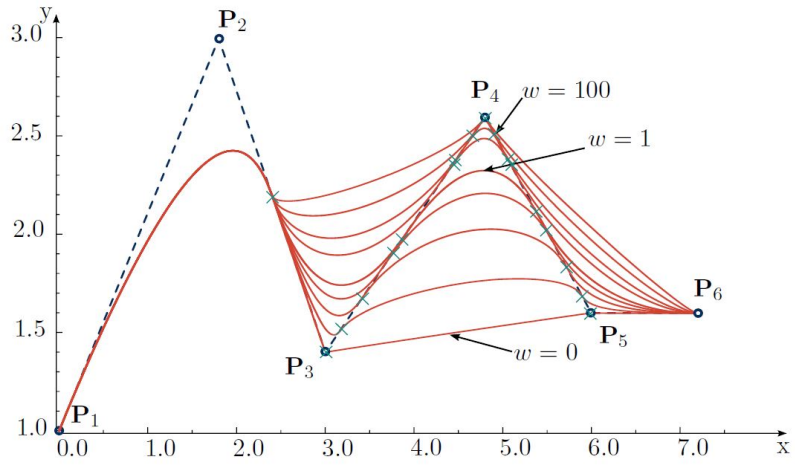

Figure 1: NURBS curve with variable weighting of control point $\mathbf{P}_4$ [12]

Knot Vector The parametric space in NURBS is defined by the so-called knot vector $\Xi = [\xi_1, \xi_2, \ldots, \xi_{(n+p+1)}]$. Based on the knot vector and the polynomial degree, the B-Spline basis functions which are used for NURBS approximation are computed using a recursive algorithm such as the Cox-de Boor method. It starts for $p = 0$ with:

$$N_{i,0}(\xi) = \begin{cases} 1 & \xi_i \leq \xi < \xi_{i+1} \\ 0 & otherwise \end{cases} \tag{3}$$

For $p \geq 1$ it is,

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{4}$$

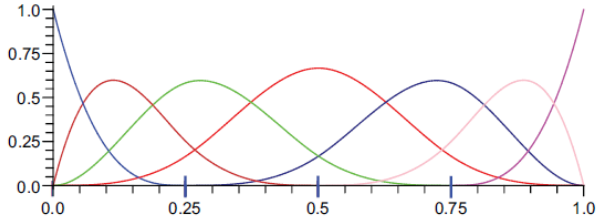

Figure 2: p=3, basis functions with open knot vector Ξ=[0,0,0,0,0.25,0.5,0.75,1,1,1,1] [5]

Refinement There are basically two ways of refining a NURBS curve or surface, namely knot insertion (or h-refinement) and polynomial order elevation (or p-refinement). The knot insertion (h-refinement) is when the knot spans are divided into smaller ones by inserting new knots. As a consequence, at this point the continuity is reduced from C^{∞} to C^{p-k} if the new knots introduced in the middle of an existing knot span; here k denotes the knot multiplicity. Meanwhile, the order elevation (p-refinement) keeps the number of knot intervals remains unchanged but the polynomial degree of the basis functions is

increased. While increasing the order, existing knots' multiplicity is increased so that the continuity at these points remains unchanged. A very important feature of refinement in IGA is that it does not change the pre-refined geometry.

2.1.2 Structural Analysis with IGA

In conventional FEA, geometric information can be lost in the process of meshing. It is clear that the CAD model and the finite element model almost never coincide in real applications even for very fine meshes, which by further, leads to error due to discretization. Whereas, in IGA, due to the advantageous properties of B-Splines or NURBS, one can keep the exact and smooth geometry when refining. Also, the process of mesh generation, which generally time and labor consuming, can be skipped. The figures below represent the comparison between the standard FEM and IGA.

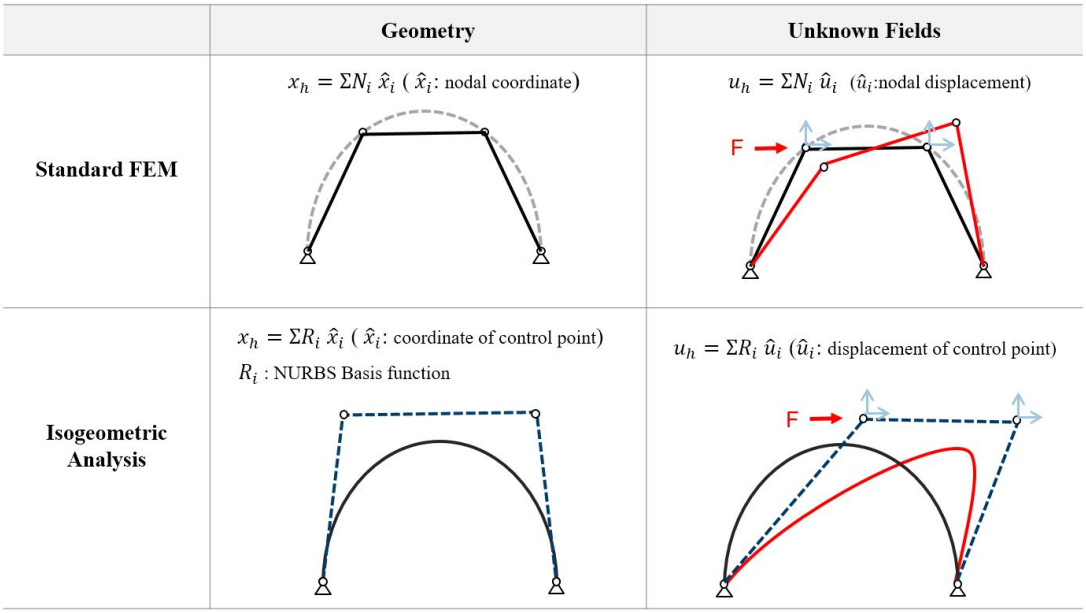

Figure 3: Comparison between "standard" FEM and IGA [12]

In conventional FEM, the unknown fields, e.g. displacements, can be approximated by the shape functions and the nodal unknown values. In IGA, however, external forces are applied on the control points and the displacement field can be approximated with NURBS basis functions considering the control points' displacements. Moreover, the elements in IGA are defined as the knot spans in the parameter space, and thus, an additional integration domain has to be generated in order to perform Gauss integration using Gauss points. While each element knot span range from two different arbitrary parametric coordinates, the Gauss domain will always range from $u, v \in [-1, 1]$. In short, these two transformations need to be combined to finally perform the Gauss integration. Figure 4 shows these two transformations graphically.

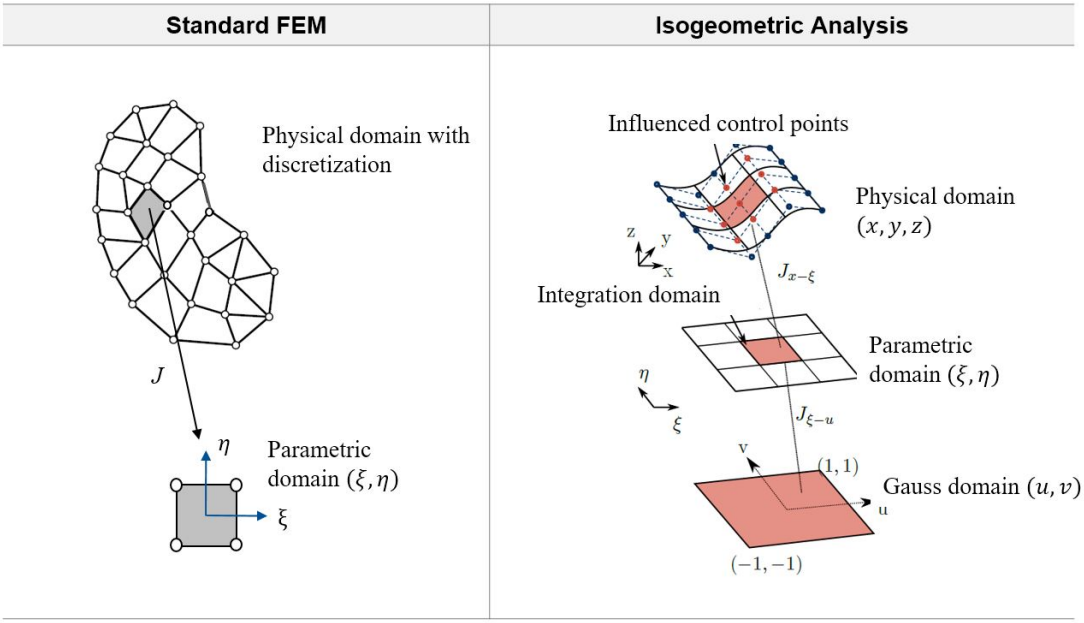

Figure 4: Difference in integration scheme between FEM and IGA [12]

To explain it more precisely, we look into the 3D linear elastic case as an example. As mentioned earlier, deformation of geometry based on NURBS can be computed by rational functions and deformation of control points.

$$\begin{Bmatrix} u^1 \\ u^2 \\ u^3 \end{Bmatrix} = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{o} R_{ijk} \begin{Bmatrix} d_{ijk}^1 \\ d_{ijk}^2 \\ d_{ijk}^3 \end{Bmatrix} \tag{5}$$

where u is the deformation field and d is the deformation of control points. Since we are in 3D domain, it has 3 degrees of freedom on each control point. Also m, n, and o represent the number of control points in x, y, and z-direction. The equation can be also written in the matrix form as,

$$\begin{Bmatrix} u^1 \\ u^2 \\ u^3 \end{Bmatrix} = \begin{bmatrix} R_1 & 0 & 0 & R_2 & 0 & 0 & \cdots & R_{nmo} & 0 & 0 \\ 0 & R_1 & 0 & 0 & R_2 & 0 & \cdots & 0 & R_{nmo} & 0 \\ 0 & 0 & R_1 & 0 & 0 & R_2 & \cdots & 0 & 0 & R_{nmo} \end{bmatrix} \begin{Bmatrix} d_1^1 \\ d_1^2 \\ d_1^3 \\ \vdots \\ d_{nmo}^1 \\ d_{nmo}^2 \\ d_{nmo}^3 \end{Bmatrix} \tag{6}$$

Similar with the standard FEM, the stiffness matrix $\boldsymbol{K}$ can be computed as,

$$\boldsymbol{K} = \int_{\Omega} \boldsymbol{B}^T \boldsymbol{C} \boldsymbol{B} \mathrm{d}\Omega = \int_{\xi=0}^{1}\int_{\eta=0}^{1} \boldsymbol{B}^T(\xi,\eta)\boldsymbol{C}\boldsymbol{B}(\xi,\eta)||\boldsymbol{J}||\mathrm{d}\xi\mathrm{d}\eta \tag{7}$$

where ξ, η and ζ represent the parametric domain knot vectors and $||\boldsymbol{J}||$ is the determinant of Jacobian matrix between physical domain x to parametric space ξ.

In IGA, likewise standard FEM, in order to solve the weak form, Gauss quadrature and Jacobian are necessary for integration. However, as described in Figure 4 above, IGA has one additional transformation since IGA elements are already defined in the parameter space with knot vector. Here, the Jacobian to

map between physical and parameter space is shown as,

$$\boldsymbol{J}_{x-\xi} = \begin{bmatrix} \frac{dx}{d\xi} & \frac{dx}{d\eta} \\ \frac{dy}{d\xi} & \frac{dy}{d\eta} \\ \frac{dz}{d\xi} & \frac{dz}{d\eta} \end{bmatrix} \tag{8}$$

The other mapping from parametric space to Gauss domain is also shown as,

$$\boldsymbol{J}_{\xi-u} = \begin{bmatrix} \frac{d\xi}{du} & \frac{d\xi}{dv} \\ \frac{d\eta}{du} & \frac{d\eta}{dv} \end{bmatrix} \tag{9}$$

Combining these two transformations allows us to finally perform Gauss integration to obtain the stiffness matrix as,

$$\boldsymbol{K} = \sum_{\xi}\sum_{\eta}\sum_{i=1}^{\#u_{GP}}\sum_{j=1}^{\#v_{GP}} \boldsymbol{B}^T(\xi(u_i),\eta(v_j))\boldsymbol{C}\boldsymbol{B}(\xi(u_i),\eta(v_j))||\boldsymbol{J}_{x-\eta}||||\boldsymbol{J}_{\eta-u}||w_i w_j \tag{10}$$

where, w is the weight factor of Gauss quadrature.

2.1.3 Multipatch coupling

While using IGA for the structural analysis of complex geometric models, a single NURBS patch will not be enough to represent the whole geometries with arbitrary complexity, and thus, multiple patches have to be considered. However, these multi-patches do not necessarily join conformingly at their common boundaries. So the problem finally boils down to the solution of a boundary value problem in more than one subdomains. For the coupling and the solution of boundary value problems in multiple subdomains some domains decomposition methods are employed, e.g. the Penalty, Lagrange Multiplier, Augmented Lagrange Multiplier, Nitsche methods, etc. Figure 5 taken from [1], shows a decomposed boundary value problem in two subdomains and shows the Dirichlet and Neumann Interface conditions that need to be satisfied on the boundary of the two subdomains.

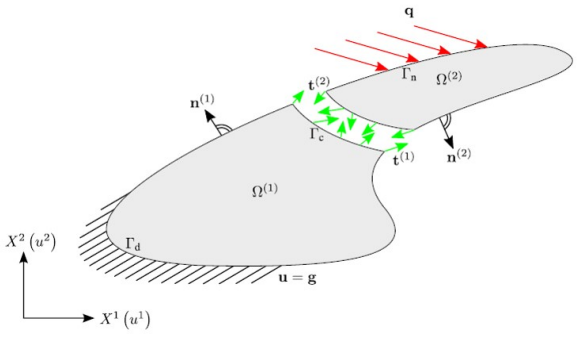

Figure 5: Decomposed Boundary Value problem, $u^{(1)} - u^{(2)} = 0 and t^{(1)} + t^{(2)} = 0$ on Γ_c [1]

The initial discrete system of equations before the coupling is written as,

$$\begin{bmatrix} \boldsymbol{K}^{(1)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}^{(2)} \end{bmatrix} \hat{\boldsymbol{u}} = \begin{bmatrix} \hat{\boldsymbol{F}}^{(1)} \\ \hat{\boldsymbol{F}}^{(2)} \end{bmatrix} \tag{11}$$

where, $\boldsymbol{K}$ and $\boldsymbol{F}$ are the stiffness matrix and the load vector at each subdomain, respectively. Using the

Penalty method as an example, the discrete system of equation now looks like,

$$\begin{bmatrix} \boldsymbol{K}^{(1)} + \boldsymbol{K}_p^{(1)} & \boldsymbol{C}_p^{(1)} \\ \boldsymbol{C}_p^{(2)} & \boldsymbol{K}^{(2)} + \boldsymbol{K}_p^{(2)} \end{bmatrix} \hat{\boldsymbol{u}} = \begin{bmatrix} \hat{\boldsymbol{F}}^{(1)} \\ \hat{\boldsymbol{F}}^{(2)} \end{bmatrix} \tag{12}$$

where $\boldsymbol{K}_p$ is the penalty stiffness matrices and $\boldsymbol{C}_p$ is the penalty coupling matrices. The other methods mentioned above will also produce a new form of the discrete system of equations with the right coupling terms of each method showing up on the left-hand side matrix. In this project, the Penalty method and the Nitsche method are the two approaches used for the analysis. The detailed equation and implementation of the Nitsche method have been clearly described by [1].

2.2 Computational Fluid Dynamics

2.2.1 Governing Equations

The governing equations of the fluid flow, the continuity and the Navier-Stokes equation for incompressible Newtonian fluids, are represented as,

$$\nabla \cdot \boldsymbol{v} = 0 \tag{13}$$

$$\frac{D\boldsymbol{v}}{Dt} = \frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{v} = -\frac{1}{\rho_0}\nabla P + \nu \nabla^2 \boldsymbol{v} + \boldsymbol{g} \tag{14}$$

where $\boldsymbol{v}$ is the velocity field, P is the pressure, ρ is the density, and ν is the molecular kinematic viscosity of the fluid, respectively. Here, $\boldsymbol{g}$ and t indicate the gravitational acceleration and time. In the most general incompressible flow approach, the density is assumed to be constant, with its initial value ρ_0. When dealing with the turbulent flows, the turbulent stress term which can be derived from the nonlinear convective term in (14) needs to be modeled. However, for the sake of simplicity, in the current project, the flow is considered as laminar, where it does not use any turbulence models.

2.2.2 Finite Volume Method in OpenFOAM

In the current project, the fluid domain and its governing equation are discretized in Finite Volume approach and the simulation is conducted in OpenFOAM [3], a widely used open-source Computational Fluid Dynamics (CFD) software which is based on C++ frameworks. The popularity of the Finite Volume Method (FVM) in CFD stems from the high flexibility it offers as a numerical method, where the discretization is carried out directly in the physical space without the necessity of transformation of physical and computational coordinate system. Moreover, its adoption of a collocated arrangement [9] made it suitable for solving flows in complex geometries. These developments have expanded the applicability of the FVM to encompass a wide range of fluid flow simulations in various applications while retaining the simplicity of its mathematical formulation. Another important aspect of the FVM is that its numerics mirrors the physics and the conservation principles it models, such as the integral property of the governing equations, and the characteristics of the terms it discretizes [7].

In the FVM manner, the conservation equation (13) and (14) should be writen in its weak or integral

form as,

$$\int_{\Gamma} \boldsymbol{v} \cdot \boldsymbol{n} d\Gamma = 0 \tag{15}$$

$$\underbrace{\frac{\partial}{\partial t}\left(\rho \int_{\Omega} \boldsymbol{v} d\Omega\right)}_{\text{transient term}} + \underbrace{\rho \int_{\Gamma} \boldsymbol{v}(\boldsymbol{v} \cdot \boldsymbol{n}) d\Gamma}_{\text{convective term}} = \underbrace{-\int_{\Gamma} P \boldsymbol{n} d\Gamma}_{\text{pressure term}} + \underbrace{\int_{\Gamma} \mu(\nabla \boldsymbol{v}) \cdot \boldsymbol{n} d\Gamma}_{\text{diffusive term}} + \underbrace{\rho \boldsymbol{g} \Omega}_{\text{source term}} \tag{16}$$

Here, $\int_{\Omega}$ and $\int_{\Gamma}$ are the volume and surface integrals over volume Ω and surface Γ, respectively. Notice that the transformation of integral form from volume integral and surface integral and vice versa can be performed through the use of the Gauss theorem. The following procedure necessary is then to approximate these volumes and faces integrals in each discrete element which can be evaluated in various approach taking into account the order of accuracy, boundary conditions, integration schemes, and, most importantly, computational FVM mesh (whether structured or unstructured, or whether staggered or collocated). Several possible approximations and interpolation methods and its implementation in OpenFOAM are clearly explained and described in [7].

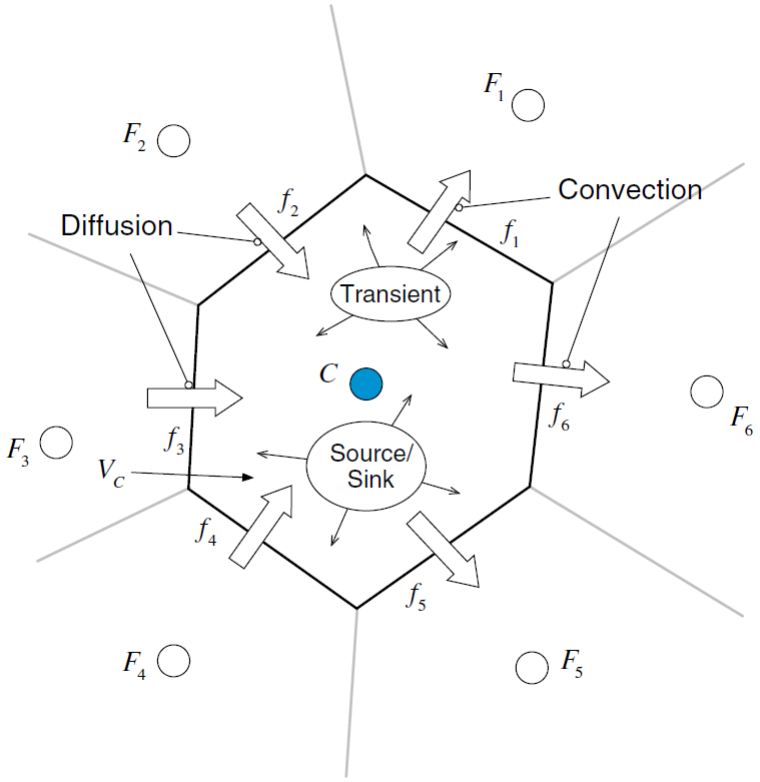

Figure 6: Conservation in a discrete Finite Volume element [7]

2.2.3 Arbitrary Lagrangian Eulerian

Arbitrary-Lagrangian-Eulerian (ALE) schemes are characterized by a moving computational mesh. At each time step, the new position of all the nodes has to be recomputed according to a prescribed mesh velocity, denoted as $\boldsymbol{v}_M$, which can be chosen arbitrarily. The aim of ALE is simply to reduce the numerical dissipation errors due to the convective terms [4], hence to capture contact discontinuities sharply and to precisely identify and track the FSI interfaces which change according to the structural deformation. The

fundamental ALE equation is written as,

$$\frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + (\boldsymbol{v} - \boldsymbol{v}_M)\frac{\partial \phi}{\partial t} \tag{17}$$

which is valid for any scalar quantity ϕ, or for any component of a vector function [10]. Notice the mesh velocity can be chosen as close as possible to the local fluid velocity $\boldsymbol{v}_M = \boldsymbol{v}$ (as it is in the purely Lagrangian framework), but it can also be set to zero $\boldsymbol{v}_M = \boldsymbol{0}$ (to reproduce the Eulerian case).

$$(\text{Lagrangian Case :})\ \frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} \tag{18}$$

$$(\text{Eulerian Case :})\ \frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + \boldsymbol{v}\frac{\partial \phi}{\partial t} \tag{19}$$

Applying ALE into our governing equation, we can write the conservation of mass and momentum in their weak form as,

$$\left.\frac{\partial}{\partial t}\right|_{\boldsymbol{\chi}} \int_\Omega \rho d\Omega + \int_\Gamma (\boldsymbol{v} - \boldsymbol{v}_M)\cdot \boldsymbol{n} d\Gamma = 0 \tag{20}$$

$$\left.\frac{\partial}{\partial t}\right|_{\boldsymbol{\chi}} \left(\int_\Omega \rho \boldsymbol{v} d\Omega\right) + \int_\Gamma \rho \boldsymbol{v}((\boldsymbol{v} - \boldsymbol{v}_M)\cdot \boldsymbol{n}) d\Gamma = -\int_\Gamma P\boldsymbol{n} d\Gamma + \int_\Gamma \mu(\nabla \boldsymbol{v})\cdot \boldsymbol{n} d\Gamma + \rho \boldsymbol{g}\Omega \tag{21}$$

where, to be precise, $\boldsymbol{\chi}$ subscript in the time derivative indicates local derivative with the reference mesh coordinate $\boldsymbol{\chi}$ held fixed. The following figures show the deformation of fluid meshes within the ALE scheme following the structural deformation in the bottom edge. At the right figure, we can see that the fluid meshes have been deformed through interpolation in such a way according to the prescribed boundary deformation and velocity.

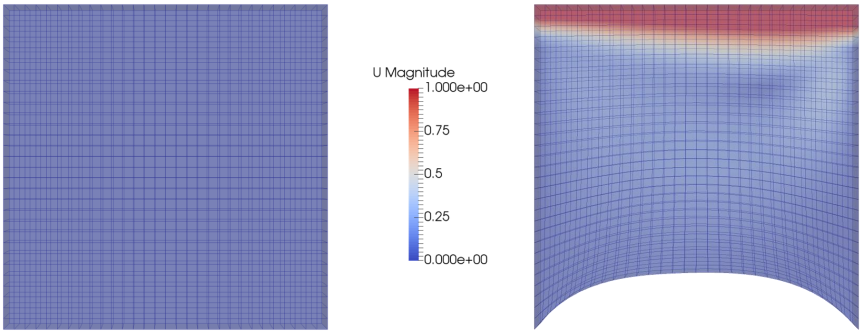

Figure 7: Computational grids in OpenFOAM: (left) initial FV mesh; (right) ALE mesh at t=8s

2.3 Fluid-Structure Interaction (FSI)

Fluid-structure interactions (FSI) are a crucial consideration in the design of various engineering structures since the accurate understanding and prediction of impact loads by fluid flow can provide an important information on the structure safety assessment, in particular, membrane structures, which are extremely lightweight and sensitive to flow-induced deformations and vibrations. A FSI-simulation of light-weight structures subjected to wind requires the appropriate combination of different physical and numerical disciplines to account for the relevant factors. In order to meet all of these requirements, a partitioned FSI approach is chosen among other possible approaches. The coupling scheme and algorithm between the structural solver and the fluid solver described earlier will be discussed further in the following subsections.

2.3.1 Basic Ideas and Types of FSI Simulations

The basic idea of FSI scheme is to connect the mechanical behavior between two completely different systems, the structure and the fluid, which have different mathematical models and physical behavior. In addition to that, FSI scheme also acts as an interface between them, which generally provides a way to connect the two distinct solvers or rules the flow of information from one system to another. In FSI, generally, two different approaches can be used to obtain the solution of coupled problems. The *monolithic* approach solve the governing equations both of the flow and the structure simultaneously, within a single solver. This approach involves a larger and more complex tangent matrix which normally requires suitable preconditioners to invert. As a result, the monolithic FSI problem will converge as good as the approximation of the tangent matrix allows, which by further, resulting in a very robust solver if it can be done effectively. Meanwhile, another possibility is to keep the fluid and the structure solvers separated as two distinct solvers; this approach is known as *staggered* or *partitioned* approach, which is widely used in FSI since it can preserve each software modularity as it couples an existing, different, possibly more efficient flow and structural solvers that have been developed specifically for each governing equations. However, the development of stable and accurate coupling algorithm is generally required in the partitioned simulation as the coupling condition used will strongly influence the stability of the results.

Furthermore, the partitioned methods are divided into *one-way* and *two-way* coupling. The one-way coupling methods provide a possibility to reduce the computational effort of FSI by only passing the fluid pressure force acting at the structure to the structure solver. On the other hand, the more intuitive two-way coupling methods transfer the displacement of the structure back to the fluid solver as an additional step of the one-way coupling scheme. Two-way coupling scheme is further divided into *weakly* and *strongly* coupled methods where the convergence at the boundary between structure and fluid is not considered and considered, respectively. The differences and comparisons of the mentioned coupling methods in several FSI examples can be found in more details at the following literatures: [11] [8] [2].

2.3.2 FSI Coupling with Dynamic Relaxation Scheme

In the current project, the staggered two-way strongly coupled FSI scheme is chosen considering dynamic relaxation scheme. The FSI interface is implemented to first transfer the fluid pressure force acting on the structure to the structure solver. The force transferred will further cause deformation on the structure and this deformation is then sent back to the fluid solver as a response to the given force. This process of transferring and receiving information, namely force and displacement, is then continued in an iterative loop until certain convergence criteria are satisfied. Here, the convergence criteria must preserve the continuity of velocity and position (or the Dirichlet conditions) as well as the equilibrium of forces (the Neumann condition) at the boundary between the structure and the fluid domain. Ideally, one can write,

$$\begin{aligned} \boldsymbol{x}_{s\Gamma} &= \boldsymbol{x}_{f\Gamma}\,, \\ \boldsymbol{v}_{s\Gamma} &= \boldsymbol{v}_{f\Gamma}\,, \\ \boldsymbol{t}_{s\Gamma} &= \boldsymbol{t}_{f\Gamma}\,. \end{aligned} \tag{22}$$

where, $\boldsymbol{x}_\Gamma$, $\boldsymbol{v}_\Gamma$, and $\boldsymbol{t}_\Gamma$ are the position, velocity, and the traction vectors at the boundary, respectively, and the subscript s and f indicate the partitioned structure and fluid parts. Therefore, the coupling loop between the structure and the fluid parts will always continue until the convergence of displacements $\boldsymbol{u}$ and forces $\boldsymbol{F}$ are reached, up to certain tolerance, or until the maximum number of iterations is reached.

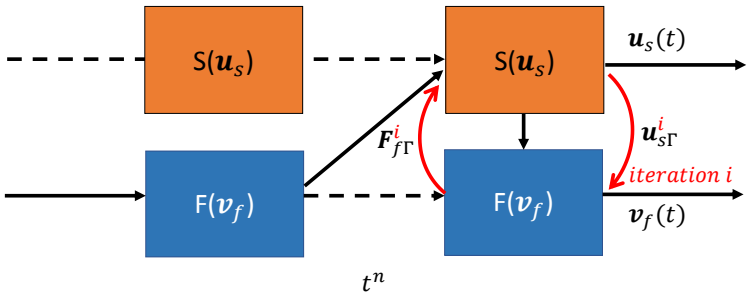

Figure 8: Solution algorithm for two-way coupling at every time step

In order to improve convergence of the FSI iteration, the relaxation of the displacement passed from the structure to fluid is commonly applied. The relaxation of the FSI cycle at each sub-iteration i at a certain time step is defined as:

$$\boldsymbol{u}_{f\Gamma}^{i+1} = \hat{\boldsymbol{u}}_{s\Gamma}^{i+1} = \omega^i \boldsymbol{u}_{s\Gamma}^{i+1} + (1-\omega^i)\boldsymbol{u}_{s\Gamma}^{i}\,, \tag{23}$$

where $\boldsymbol{u}_\Gamma^i$ is the displacement at iteration i and ω^i is the relaxation parameter. Choosing fix value of ω^i for all iterations leads to fix point methods. In this method, the relaxation ω has to be set empirically at the beginning of the convergence iteration. Notice that if $\omega = 1$, the system is not relaxed, since it passes directly $\boldsymbol{u}_{f\Gamma}^{i+1} = \boldsymbol{u}_{s\Gamma}^{i+1}$. For under-relaxation cases, ω is chosen in the range of $0 < \omega < 1$. This will lead into a more robust iteration yet less efficient. In the current project, the Aitken relaxation technique is used. The method can be simply implemented by dynamically modifying the relaxation factor ω^i, taking into account the residual value at each iteration, instead of keeping it constant as the original fix point method would give. The relaxation factor and the residual function can be defined as,

$$\omega^i = \omega^{i-1}\frac{(\boldsymbol{r}_\Gamma^{i-1})^T(\boldsymbol{r}_\Gamma^{i-1} - \boldsymbol{r}_\Gamma^{i})}{||\boldsymbol{r}_\Gamma^{i-1} - \boldsymbol{r}_\Gamma^{i}||^2}\,, \tag{24}$$

$$\boldsymbol{r}_\Gamma^i = \boldsymbol{u}_{s\Gamma}^i - \boldsymbol{u}_{f\Gamma}^i\,. \tag{25}$$

For the first iteration, the Aitken step cannot be performed as additional information required. Thus, for the very first time step, an initial value of ω needs to be set manually. Note that, the Aitken formula requires at least one prior FSI cycle, before relaxation step to be possible.

2.3.3 Mortar Method Between Fluid and Structure Domain

As what has been explained in the previous chapters, in the current project, the fluid domain is discretized in FVM while the structure domain is discretized using IGA. The difference in discretization creates a non-conforming interface meshes, and thus, requires a specific mapping technique to accurately transfer the data field during FSI between the FV meshes, which are body-fitted to the FSI interface, and the Isogeometric control points, which are not necessarily lying on the surface interface. In order to project such field in non-matching discretizations, the Mortar method, which is based on Lagrange multiplier method, is used in our FSI simulations. This method has been widely used in recent researches related to contact mechanics and interface coupling as it offers numerous numerical advantages, such as sound theoretical framework and mathematical analysis, applicable to many different single-field and multi-field problems, and proven satisfaction of stability [6].

In the current implementation, the generation of mapping matrix by Mortar method is performed in EMPIRE. First, the fluid and the structure send the information of their discretized meshes and control

points into EMPIRE at the beginning of the simulation. Then the mapping between them in the FSI surface Γ_c is generated by the following equation, which is simply based on weighted residual method,

$$\sum_{j=1}^{n_{FV}^{(e)}} \int_{\Gamma_c^{(e)}} N_i^{(e)} N_j^{(e)} d\Gamma \hat{\phi}_j^{FV} = \sum_{j=1}^{n_{IGA}^{(e)}} \int_{\Gamma_c^{(e)}} N_i^{(e)} R_j^{(e)} d\Gamma \hat{\phi}_j^{IGA}. \tag{26}$$

where $\hat{\phi}_j^{FV}$ and $\hat{\phi}_j^{IGA}$ stand for the degrees of freedom of the finite volume surface and the Isogeometric surface. Here, the shape function of the FV mesh and the NURBS control points are denoted as N_j and R_j, respectively, while N_i is the shape function used to discretized the Lagrange multiplier in Mortar method. By computing equation (26) numerically, we can come up with our Mortar matrices to map the transferred fields at each FSI iteration,

$$\boldsymbol{M}_{\Gamma}^{FV} \hat{\boldsymbol{\phi}}^{FV} = \boldsymbol{M}_{\Gamma}^{IGA} \hat{\boldsymbol{\phi}}^{IGA}. \tag{27}$$

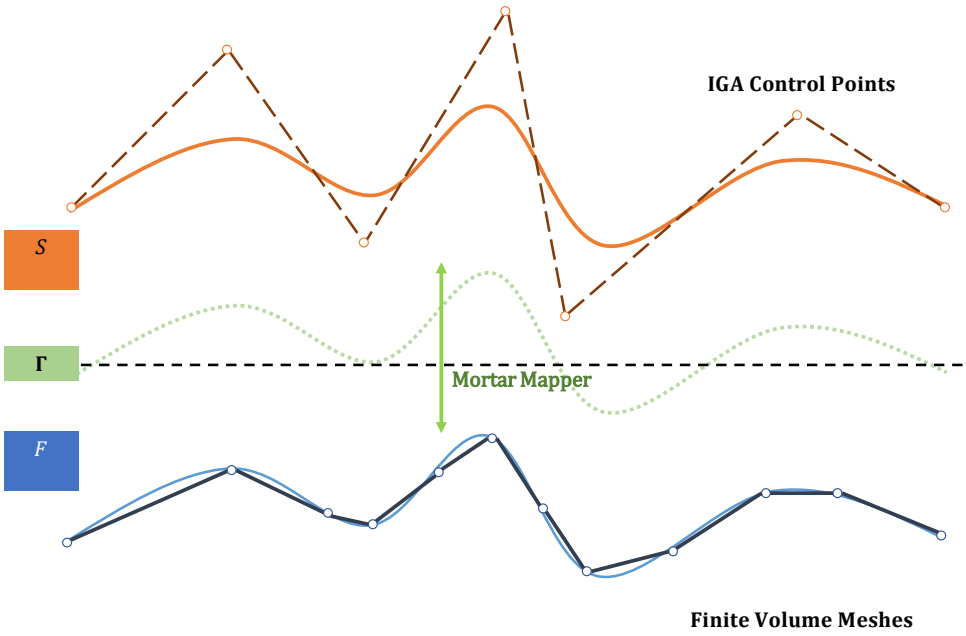

Figure 9: Surface projection from Finite Volume Meshes onto Isogeometric Control Points and vice versa using Mortar mapper

3 Implementation

3.1 Simulation Framework

FSI problems come under the category of co-simulation. In a co-simulation, different subsystems of a given problem are modeled and simulated in a distributed manner. Each subsystem is modeled differently and data exchanged between them should be implemented in a designated sequence. In the FSI type of co-simulation, we have two different subsystems, the fluid and the structural parts. It is a very common approach to use specialized software tools for different subsystems particularly while adopting the staggered approach. The benefit of this approach is that it can preserve each software modularity and allow each solver to develop separately for specific use of applications. In the examples we are dealing with, we use a Matlab-implemented structural solver for the Computational Structural Mechanics and an Open-source

software, OpenFOAM, for the Computational Fluid Dynamics part.

In our co-simulation framework, the Matlab structural solver solves transient non-linear analysis of Isogeometric membranes and OpenFOAM solves the control volume cells of the fluid part. However, as we need to use the results from one simulation as a necessary input for the other simulation in each FSI iterative step, an interface to communicate between these two simulations is also needed. This interface is used to find solutions for the following questions:

- What data need to be transferred?
- When the data has to be transferred?
- How to map the data fields for nonconforming discretization?

Here, the EMPIRE framework, which has been developed excessively by the Chair of Structural Analysis at the Technical University of Munich, is used as an interface between the structural and fluid parts. It provides the functionality required to address the aforementioned questions. EMPIRE achieves this by its message passing interface (MPI) functions which can connect different programming languages (e.g. C/C++, Python, Matlab, etc.). It also provides several useful features, like the Aitken relaxation technique and the field mapper between non-matching meshes, those which are useful for performing FSI co-simulation.

3.2 Sequence Call

The sequence call or the communication steps used in our FSI simulation can be explained by the following Figure 10. The communication steps at each time step can be divided into three different stages; initialization, static step and transient analysis.

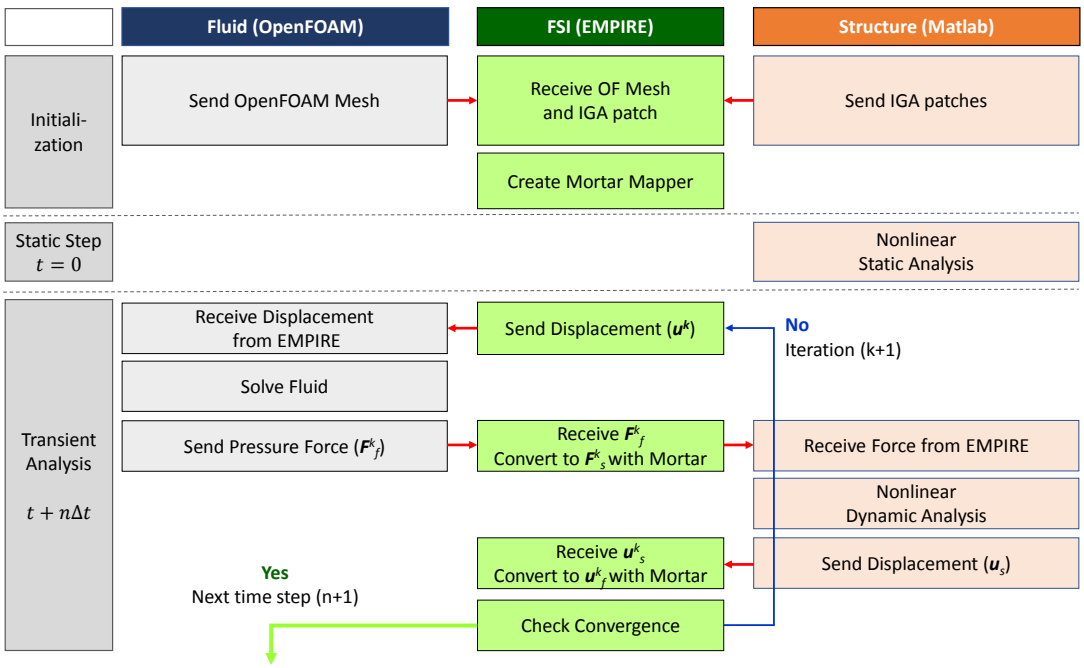

Figure 10: Sequence call in the implemented FSI framework

Initialization In the initialization stage, each subsystem is modeled in each respective software. Then, before the simulation start, the OpenFOAM sends the information of its control volume mesh details to EMPIRE, while Matlab sending its IGA patches information at the send time. EMPIRE then creates a mapper by using the Mortar method using the mesh and IGA patch data since those data are nonconforming at the FSI surface. This mapper will be constantly the same and can be used throughout the simulation.

Static Step Static step is performed at time $t = 0$. In the example simulations, this step is only required for the structural part, i.e. in Matlab. It is done by the nonlinear static analysis to compute the static forces present in the structure in its equilibrium state.

Transient Analysis At each time step t, the transient analysis starts by sending an initial guess of displacement field to the fluid solver in OpenFOAM. OpenFOAM then receives this information and solve the Navier-Stokes equation with the received displacement information as one of its boundary condition. This will result in a fluid pressure field distribution which is computed by solving the pressure Poisson equation. This pressure field is then sent to Matlab as pressure forces via EMPIRE. Following that, EMPIRE receives the pressure forces F_f^k and convert to F_s^k. The conversion of data fields from Finite Volume meshes to IGA control points is done using the Mortar mapper created at the initialization stage. Matlab then performs a non-linear dynamic analysis using the forces received from EMPIRE as its Neumann boundary condition. The analysis gives the response structural displacement field at IGA control points. Correspondingly, the response displacement is sent back to EMPIRE where the convergence check under certain tolerance is performed. If the convergence criteria are not satisfied, the next iterative step will be started by converting and transferring the response displacement back to the fluid solver after applying the Aitken dynamic relaxation. Meanwhile, if the convergence criteria are fulfilled, we can move to the next time step t^{n+1} and performs the same procedure. The fix-point iteration performed at each time step is the basis of the strongly coupled FSI scheme.

4 Numerical Results

4.1 Shear-Driven Cavity Benchmark

The shear-driven cavity is a very common benchmark problem for incompressible viscous flow simulation. In this section, shear-driven cavity benchmark will be investigated in order to test and validate the coupling scheme between the Isogeometric membrane structure and the laminar fluid flow. The objective of this numerical test is to check whether our coupling scheme can produce similar results in both single and multi-patch cases compared to the results obtained in Carat++. The visual description of the test setup is given as following Figure 11.

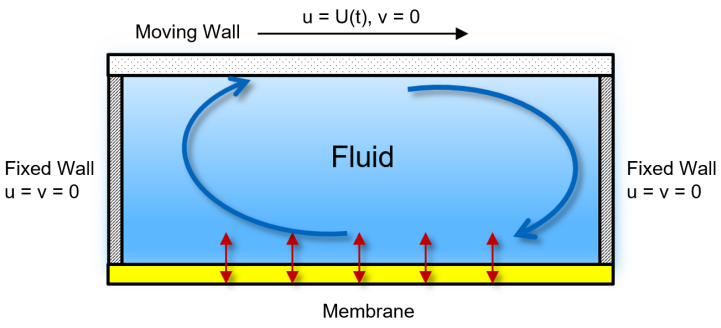

Figure 11: Shear-driven cavity problem for FSI

As can be seen from the Figure 11, the modeled geometry consists of 2 fixed walls at which the no-slip boundary condition is applied and a moving wall at which the tangential velocity $u = U(t)$ is applied. Isogeometric membrane structure is placed at the bottom of the geometry. The interaction between the membrane and the fluid will be investigated, particularly, since the deformation of the membrane is considerably large and it also contributes to the evolution of the fluid velocity and pressure field distribution. For the following cavity benchmark setup cases, the material parameters of the membrane structure are given in the following Table 1.

E $[N/m^2]$	ν [-]	t [m]	σ_0 $[N/m^2]$	ρ $[kg/m^3]$
2.5×10^5	0.0	0.002	1	500

Table 1: Material parameters of the membrane structure for Cavity benchmark case

Time integration for the transient analysis of the structure is performed by using the Bossak Scheme, while the Newton-Raphson Method is used for the nonlinear iteration. In the fluid domain, solutions of the algebraic equation system are obtained using DIC (Simplified diagonal-based incomplete Cholesky) preconditioner and PCG (Preconditioned conjugate gradient) solver implemented in OpenFOAM.

4.1.1 Singlepatch Membrane

In the first model, the structure is only discretized as one single patch with one continuous parametric coordinate. Boundary conditions are imposed in such a way that membrane only deforms in the y-direction, perpendicular to the surface. The patch is assigned to a polynomial degree $p = 1$ and $q = 1$ with corresponding open knot vector [0,0,1,1] and [0,0,1,1] initially in ξ- and η-direction, respectively. Then k-refinement is performed by applying different degree elevation and knot insertion (p-refinement and then h-refinement) to the whole patch. The structure support conditions and the top-view of the geometry can be seen in the Figure 12 below.

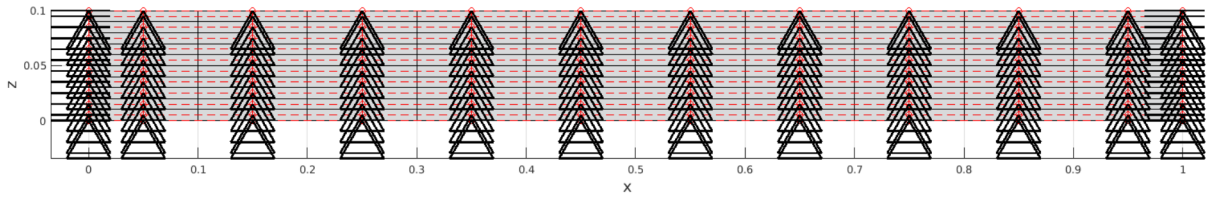

Figure 12: Membrane with a singlepatch

As described in Figure 11, the top wall moves with a velocity $U(t)$, which is changing in time. The x-component of the velocity is given as $u = U(t) = 1 - cos(\frac{2\pi t}{5})$. Moreover, the fluid has a viscosity value of $\nu = 0.01\ m^2/s$ and density of $\rho = 1\ kg/m^3$. The analysis domain and the initial condition of the system can be seen in Figure 13. In the numerical analysis, a 8-second real-time simulation has been investigated with Δt $= 0.01s$.

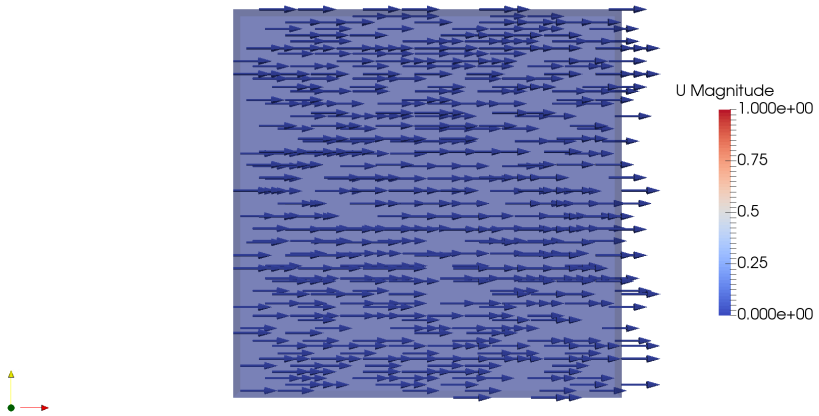

Figure 13: Initial condition

In order to validate our implementation, the results obtained in the simulation is compared to the Carat++. Corresponding results in real time t = 2 s, 4 s, 6 s and 8 s were visualized in ParaView for velocity and pressure field distribution. The comparison of the obtained velocity fields is presented in contour and in glyph vector in Figure 14. Movement of the top wall results in the formation of vortices in the flow. These vortices, furthermore, influence the change of pressure distribution in the fluid domain and, by further, results in the deformation of the membrane structure.

From the presented figures, we can see that the implemented Matlab results match almost perfectly with the Carat++ results. The Fluid velocity field pattern and its magnitudes agree with the reference solution as well as the membrane deformation at the bottom part of the analysis domain. The results of the structural deformation are also observed further by using GiD. After 800 time steps, which corresponds to 8-second real-time simulation, the maximum membrane deformation reaches approximately 0.175m which is also very close to the reference solution. Figure 15 below shows the structural deformation distribution in tested single-NURBS-patch at time step 800.

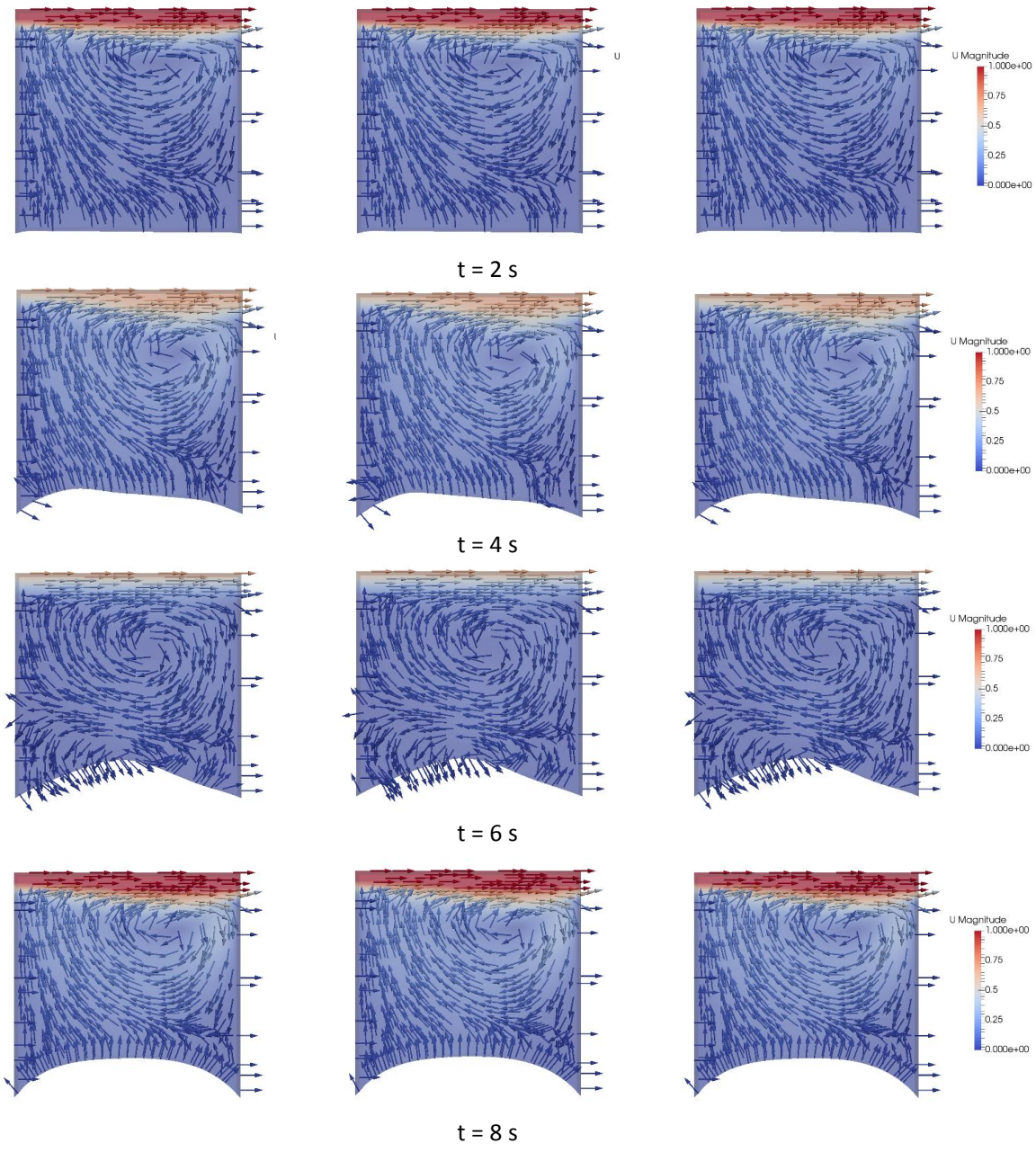

Figure 14: Comparison of results between Carat++ reference code (left) and implemented code: single patch (middle) and multi patch (right)

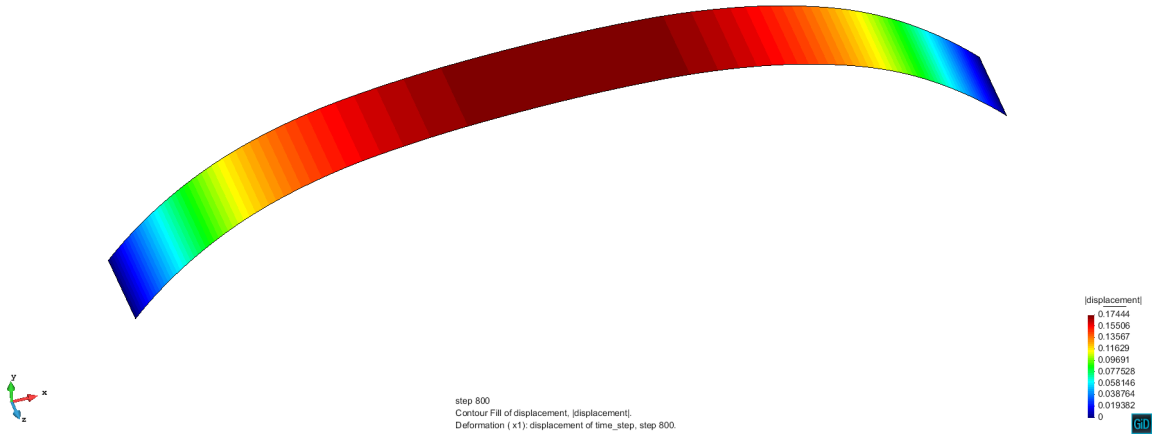

Figure 15: Deformation of the membrane observed in GiD at t=8 s

4.1.2 Multipatch Membrane

In this section, membrane structure consists of 3 different patches is investigated. The multipatch coupling between these patches is performed using a Nitsche type formulation. Each patch has the same material properties as given in Table 1 above. Moreover, the fluid properties are set to be exactly the same including the applied initial and boundary conditions in the setup.

At the beginning, each patch has been assigned to a polynomial degree in ξ-direction with p = 1 and an initial open knot vector [0,0,1,1] as well as in η-direction q = 1 with an open knot vector [0,0,1,1]. Then the k-refinement is also applied to each patch by applying a different degree of elevation and knot insertion. Due to different refinement condition applied for each patch, the number of control points and knot vectors may vary between patches in the end. The geometry and the corresponding support after k-refinement can be seen in Figure 16.

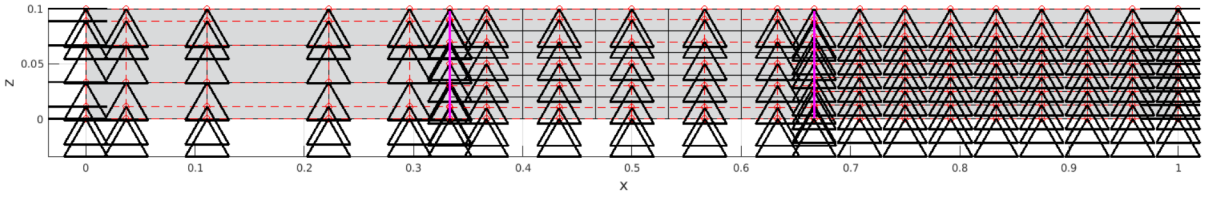

Figure 16: Membrane with 3 patches

Results of the numerical simulation are given in the Figure 14 above. Similarly, the velocity field and pressure distribution obtained from the multipatch geometry matches with the single patch results, as well as the Carat++ reference results mentioned in the previous section. The exact comparison between the single patch and the multipatch membrane structure at the mid-point can be easily seen in Figure 17. In this graph, the deformation magnitude of the middle point of the membrane is measured and plotted at every time step.

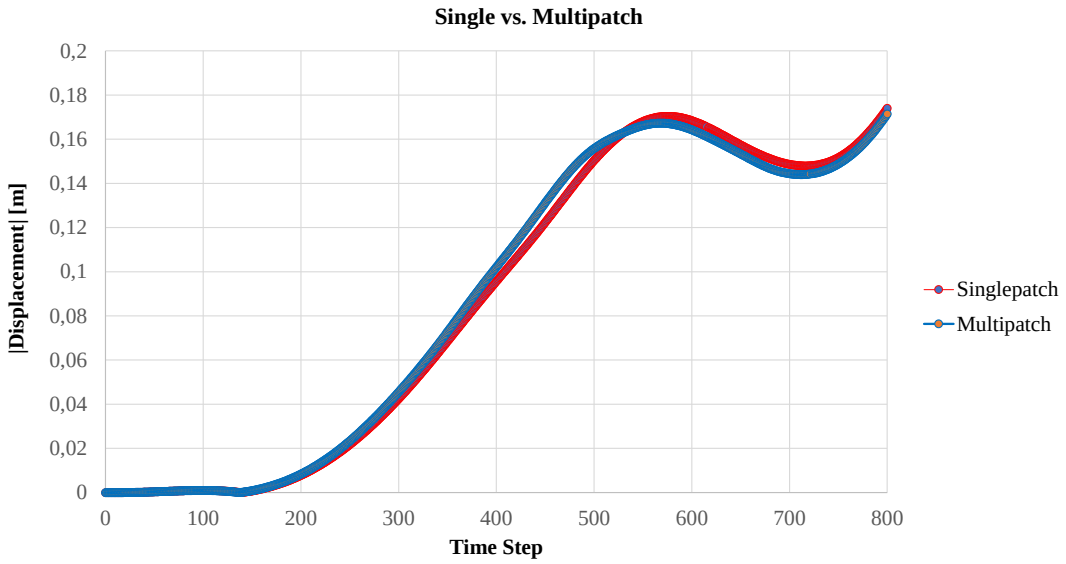

Figure 17: Comparison of the mid-point deformation in singlepatch and multipatch membrane

4.2 Four Point Tent

In this section, FSI simulation of a four point tent structure will be investigated by using single and multipatch geometries. Geometry of the tent is described in the Figure 18 while material parameters for structure is given in the following Table 2. Moreover, the viscosity and density of the fluid used in the analysis are given as $\nu = 10^{-5}\ m^2/s$ and $\rho = 1.225 \times 10^{-6}\ kg/m^3$, respectively.

$E\ [N/m^2]$	ν [-]	t [m]	$\sigma_0\ [N/m^2]$	$\rho\ [kg/m^3]$
8×10^8	0.4	0.001	3×10^6	500

Table 2: Material parameters of the membrane structure for Four Point Tent case

Analysis domain consists of a half sphere bounding box which has an inlet, outlet and bottom wall assigned. The tensile tent structure is placed closer to the inlet part of the spherical bounding box in order to avoid flow reversal and to preserve mass-conservation in the outlet wall. In this numerical analysis, two-way FSI coupling of the tent and the fluid is investigated by creating two interaction surfaces that correspond to the top and the bottom surfaces of the tent. These two FSI interface surfaces simply connected the data transfer between the upper and lower surface of the tent with the surrounding fluid. This consequently means that two separate Mortar mapping need to be applied in order to simulate the interaction in these two interfaces. Bounding box and the position of the four point tent can be seen illustratively in Figure 19.

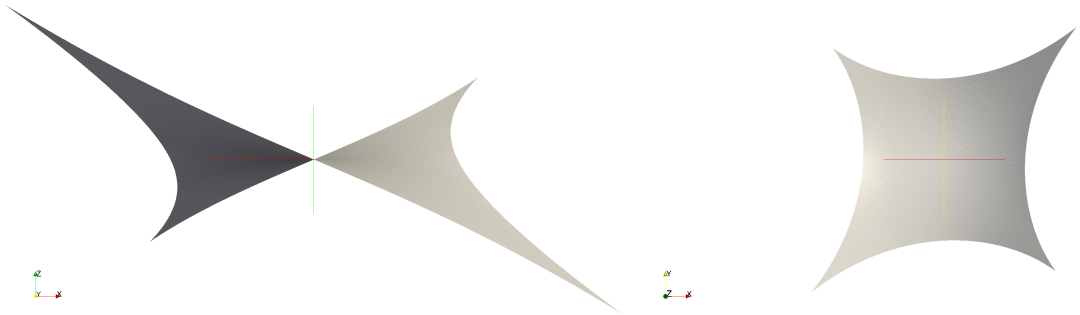

Figure 18: Side and top view of the four point tent geometry

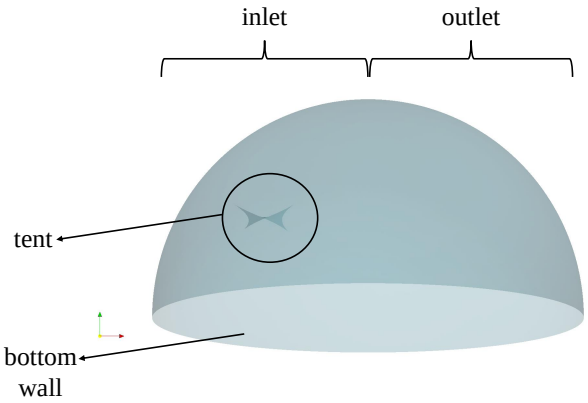

Figure 19: Half spherical bounding box for FSI problem

Similar to the previous case, single and multipatch membrane discretization will be investigated and compared for this relatively complex tent geometry. Moreover, for the multipatch geometry, different patch coupling methods such as the Penalty and Nitsche method will be compared in order to understand the performance of the methods and to determine the optimum method for this study. Before the FSI analysis is started, the initial shape of the tent should be preliminary form-founded by using the form finding algorithm implemented by the Chair of Structural Analysis. Using the previously obtained data provided by the chair, we can simply generate our structural geometry considering the initial polynomial degrees and knot vectors given in the data.

4.2.1 Singlepatch

Tent geometry is firstly modeled as a single patch membrane as seen in Figure 20. Constraints are then applied on the four support points and the control points used for NURBS discretization can be seen as the red circles around the surface. The maximum dimension of the tent is given as $20 \times 20 \times 10$ m.

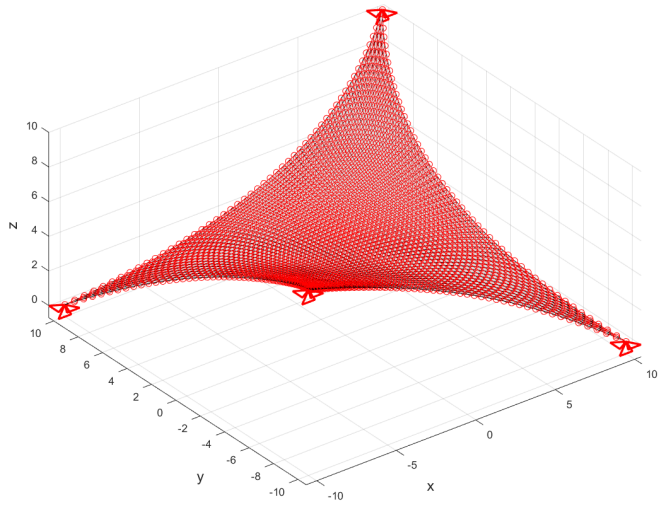

Figure 20: Four Point Tent modeled as single patch membrane.

4.2.2 Multipatch

Successively, the tent is also modeled with 3 patches with the same support and material properties as the earlier case. This can be seen clearly in Figure 21 where the pink-color lines indicate the boundary between the patches. The coupling between these three patches was performed by using Nitsche and Penalty methods.

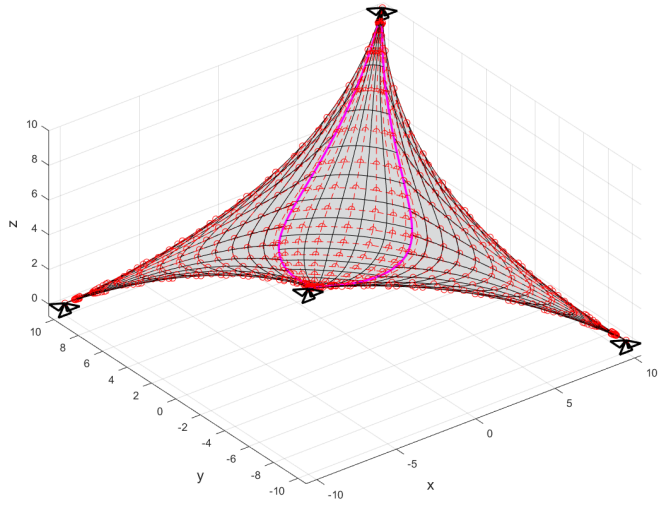

Figure 21: Modeling with multipatch.

As given in Figure 22 the performance of these coupling methods were investigated by comparing the magnitude of displacement field for over 18 time steps. Single patch result is also included as a reference solution for the aforementioned multi-patch results. Both methods show fairly accurate results compared to the reference solution, though the Nitsche method performs better in accuracy compared to the Penalty method. However, since the magnitude of the displacement is very close to zero, it can be said that both

methods for this time interval can be used to analyze the four-point tent structure.

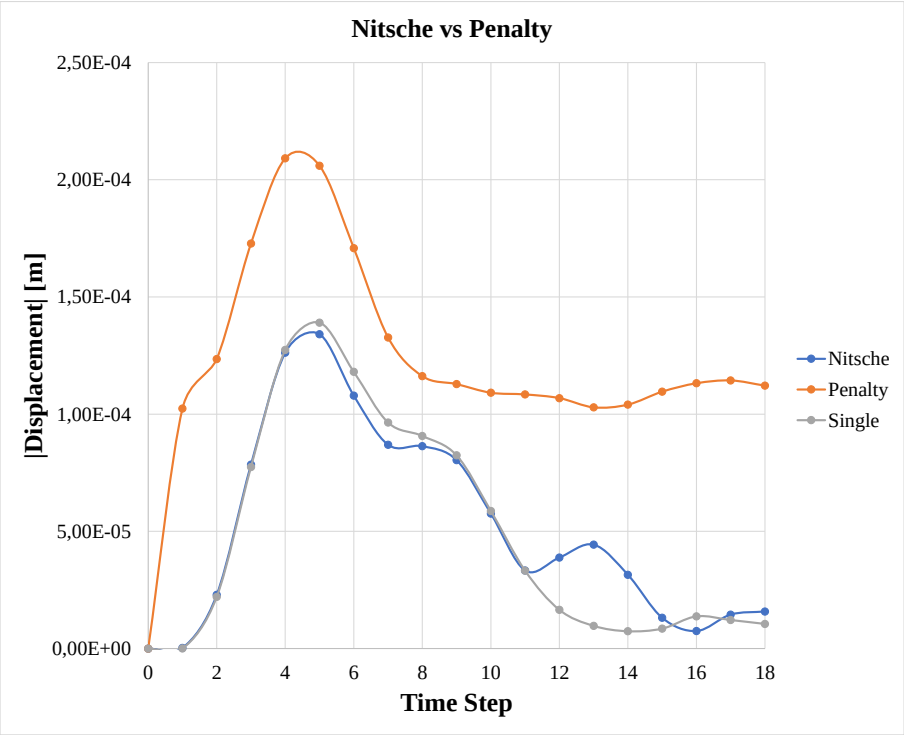

Figure 22: Comparison of different discretization and coupling methods.

5 Conclusions

In this project, a communication scheme is implemented for staggered-strongly coupled FSI simulation involving multipatch membrane structures. The implementation details through Empire to connect the structural and the fluid solver are mostly discussed in chapter 2 and 3. By using the current communication scheme, we can simulate the flow-induced deformation of membrane structures by both single and multipatch discretizations. This communication scheme allowed us to correctly transfer any data field in the interface from fluid to structure and vice versa effortlessly, resulting in an accurate coupling technique which can simulate the FSI phenomena with arbitrarily complex geometry.

The validation tests of our staggered coupling scheme were conducted after the mentioned sequence call is implemented. In our first validation test, a simple shear-driven cavity flow with flexible membrane wall was tested to check the performance of our scheme in comparison with existing simulation, which is performed in Carat++. The comparative results with Carat++ results affirm that the implemented scheme can represent an accurate evolution of structural deformation as well as fluid's velocity and pressure distribution. We also conducted additional numerical tests with Four Point Tent model to validate the coupling scheme between patches by using different DDM (Domain Decomposition Method), i.e. the Penalty and the Nitsche methods. The tent's deformation, which is represented by three separate patches, showed a very good agreement compared to the simulation performed assuming a single patch membrane; particularly can be seen from the comparison of the deformation magnitude showed by Figure 21. By

seeing the obtained results, we believe that the implemented codes can simulate cases with more complex geometry, such as the analysis of roof structure, which involves even more complex multipatch FSI and in a larger scale of simulation.

In further studies, rough changes in the magnitude of the displacement field, especially for the four-point tent case, need to be investigated more in depth in order to understand the numerical problem which causes the solution diverges after a certain number of time steps. Furthermore, material properties for both fluid and structure as well as the coupling parameters in the FSI interface need to be investigated more carefully. Since the Mortar mapper may results in errors during the analysis due to the non-matching geometries defined for FSI problem, correct projection properties, such as the maximum projection distance, the number of refinement for initial guess, the maximum distance for projected points on different patches, should be considered. The optimal configuration of these numerical parameters is extremely important to achieve a more accurate and robust FSI projection in the interface.

References

[1] Andreas Apostolatos et al.
"A Nitsche-type formulation and comparison of the most common domain decomposition methods in isogeometric analysis".
In: *International Journal for Numerical Methods in Engineering* 97.7 (2014), pp. 473–504.

[2] Friedrich-Karl Benra et al.
"A comparison of one-way and two-way coupling methods for numerical analysis of fluid-structure interactions".
In: *Journal of applied mathematics* 2011 (2011).

[3] The OpenFOAM Foundation.
OpenFOAM.
URL: `https://openfoam.org/`.

[4] Elena Gaburro, Michael Dumbser, and Manuel J Castro.
"Direct Arbitrary-Lagrangian-Eulerian finite volume schemes on moving nonconforming unstructured meshes".
In: *arXiv preprint arXiv:1602.01703* (2016).

[5] Josef M Kiendl.
Isogeometric analysis and shape optimal design of shell structures.
Shaker, 2011.

[6] Thomas Klöppel et al.
"Fluid–structure interaction for non-conforming interfaces based on a dual mortar formulation".
In: *Computer Methods in Applied Mechanics and Engineering* 200.45 (2011), pp. 3111–3126.

[7] F Moukalled, L Mangani, M Darwish, et al.
"The finite volume method in computational fluid dynamics".
In: (2016).

[8] Artur K Pozarlik and Jim BW Kok.
"Numerical investigation of one-and two-way fluid-structure interaction in combustion systems".
In: (2007).

[9] CM Rhie and WL Chow.
"Numerical study of the turbulent flow past an airfoil with trailing edge separation".
In: *AIAA Journal(ISSN 0001-1452)* 21 (1983), pp. 1525–1532.

[10] R Rossi and P Dadvand.
Lecture notes in The Finite Element Method for Fluid-Structure Interaction with open source software - Arbitrary Lagrangian Eulerian (ALE) methods.
2015.

[11] Jean-Marc Vassen et al.
"Strong coupling algorithm to solve fluid-structure-interaction problems with a staggered approach".
In: *7th European Symposium on Aerothermodynamics.*
Vol. 692.
2011,
P. 128.

[12] Roland Wüchner, Michael Breitenberger, and Anna Bauer.
Isogeometric Structural Analysis and Design.
2016.