



MUNICH SCHOOL OF ENGINEERING

TECHNICAL UNIVERSITY OF MUNICH

Bachelor's Thesis in Engineering Science

Chair of Scientific Computing, Department of Informatics

**Partitioned Fluid Structure Interaction:  
Coupling FEniCS and OpenFOAM via  
preCICE**

Richard Hertrich



MUNICH SCHOOL OF ENGINEERING

TECHNICAL UNIVERSITY OF MUNICH

Bachelor's Thesis in Engineering Science

Chair of Scientific Computing, Department of Informatics

**Partitioned Fluid Structure Interaction:  
Coupling FEniCS and OpenFOAM via  
preCICE**

**Partitionierte Fluid Struktur  
Wechselwirkung: Ein gekoppelter Ansatz  
mit FEniCS, OpenFOAM und preCICE**

Author: Richard Hertrich  
Supervisor: Prof. Dr. Hans-Joachim Bungartz  
Advisor: M.Sc. (hons) Benjamin R uth  
Submission Date: September 27, 2019

I confirm that this bachelor's thesis in engineering science is my own work and I have documented all sources and material used.

Munich, September 27, 2019

Richard Hertrich

# Abstract

In partitioned fluid structure interaction, a structure solver and a fluid solver are coupled via boundary conditions at the interface. This thesis presents partitioned FSI simulations using OpenFOAM for the fluid, preCICE as a coupling tool, and a structure solver I developed with the FEM library FEniCS. I extended the preCICE-FEniCS adapter to match the requirements for FSI, such that users can couple FEniCS simulations with vector functions and read conservatively mapped quantities from preCICE to a FEniCS solver. Plus, the adapter now features a mapping between pseudo-3D OpenFOAM and 2D FEniCS.

The setup is tested with two FSI scenarios: An elastic flap in a channel and the FSI3 benchmark. I compare the results to validated partitioned FSI methods and reference results in literature to validate the preCICE-FEniCS adapter and the structure solver.

# Contents

<b>Abstract</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Physical and Numerical Model for Fluid-Structure-Interaction</b>	<b>3</b>
2.1 Continuity Assumption and Balance Laws . . . . .	3
2.2 Structure mechanics . . . . .	4
2.2.1 Constitutive Relations for Linear Elasticity . . . . .	4
2.2.2 Time Integration with the Generalized $\alpha$ -Method . . . . .	5
2.3 Fluid mechanics . . . . .	6
2.4 Implicit Coupling . . . . .	7
2.5 Black Box Surface Coupling . . . . .	8
2.5.1 Conservative and Consistent Mapping . . . . .	8
2.5.2 Mapping Implementations: Nearest-Neighbor and Radial Basis Functions . . . . .	8
<b>3 Software Components</b>	<b>10</b>
3.1 preCICE . . . . .	10
3.2 FEniCS . . . . .	11
3.3 preCICE-FEniCS Adapter . . . . .	11
3.4 OpenFOAM and OpenFOAM Adapter . . . . .	12
<b>4 Implementation</b>	<b>13</b>
4.1 A FEniCS Solver for Linear Elastic Structures . . . . .	13
4.2 New Features in the FEniCS-preCICE Adapter . . . . .	15
4.2.1 Handling Vector Valued Functions . . . . .	15
4.2.2 3D-2D Coupling . . . . .	15
4.2.3 Runtime Improvements by Buffering RBF-Interpolation . . . . .	15
4.2.4 Reading Forces as Point Loads . . . . .	16
4.2.5 Special Treatment of Dirichlet-Boundary Nodes Inside the Cou- pling Boundary . . . . .	17

<b>5</b>	<b>Testing and Validation</b>	<b>18</b>
5.1	Elastic Flap in a Channel . . . . .	18
5.1.1	Setup . . . . .	18
5.1.2	Stability . . . . .	20
5.1.3	Results . . . . .	21
5.2	FSI3 Benchmark - Flow Around a Cylinder with a Flap . . . . .	23
5.2.1	Setup . . . . .	23
5.2.2	Testing the Structure Solver - CSM3 . . . . .	26
5.2.3	FSI3 Results . . . . .	27
<b>6</b>	<b>Conclusions</b>	<b>30</b>
6.1	Future Work . . . . .	30
	<b>List of Figures</b>	<b>32</b>
	<b>List of Tables</b>	<b>33</b>
	<b>Bibliography</b>	<b>34</b>

# 1 Introduction

Cardiovascular flow [14], the collapse of the Tacoma Narrows Bridge [29], the wings of an aircraft in turbulence [10] or the blades of a wind turbine [30], these examples have something in common: A fluid flow in or around a structure leads to forces deforming the structure, such that blood vessels expand when the pressure rises, the bridge shakes under the impact of the wind, the airplane wings bend and the wind turbine blades rotate and twist under the aerodynamic load. The deformation of the structure affects the flow at the same time: The diameter of the artery increases and the fluid domain grows. The twisted wind turbine blade affects the angle of attack and changes the lift. This two-way coupled phenomenon is called fluid-structure interaction (FSI).

Simulations help us to better understand and optimize these kinds of phenomena. However, these simulations are challenging because of the multi-physics nature of the problems. Fluid and structure are described with different models and equations that require coupling to express the interaction. There are different coupling aspects in multi-physics. The physical models can either be weakly or strongly coupled depending on how much they affect each other. In the simulation, we speak of tight coupling if we synchronize the numerical models at all times. A loose coupling scheme might allow a shift of a timestep for the coupling variables [15].

There are two main approaches for strongly coupled FSI simulations: The first approach is a monolithic solver, that includes both the equations for the fluid and the structure and solves them at the same time [13], the second one is the partitioned approach, where one solver solves the governing equations in the fluid domain and an independent solver calculates the displacements of the structure [20]. The monolithic approach is usually tightly coupled [26][3], whereas the partitioned approach can be either loosely coupled with an explicit coupling scheme [11] or tightly using implicit coupling schemes [23].

In this thesis, I follow a partitioned approach with tight coupling of the numerical models. I use the finite element software FEniCS [1] to implement a structure solver,

the computational fluid dynamics (CFD) software OpenFOAM [31] to model the surrounding fluid flow and the multi-physics coupling library preCICE [4] to couple these two. So far, FEniCS and OpenFOAM have been coupled using preCICE for conjugate heat transfer (CHT) by B. R uth [25]. OpenFOAM has been used as a fluid solver for FSI with preCICE and the structural mechanics solver CalculiX by D. Risseuw [23]. In this work I will reuse the existing components as far as possible.

In chapter 2, I explain the theoretical background for FSI simulations focusing on the structural part. In chapter 3, I introduce the software components I used for my FSI simulations. Chapter 4 presents the implementation of a structure solver in FEniCS and the new features of the preCICE-FEniCS adapter for FSI. Chapter 5 contains FSI simulations to test and validate my FSI simulation framework.

To simplify notation in this thesis, I use the Einstein notation and tensor notation. In Einstein notation, a second order tensor is denoted as  $a_{ij}$  and a first order tensor as  $b_j$ . We sum over indices appearing twice in a term, such that the expression

$$c_i = \sum_{j=1}^3 a_{ij} b_j \quad (1.1)$$

reduces to

$$c_i = a_{ij} b_j. \quad (1.2)$$

In tensor notation, all tensors are **bold** symbols. The scalar product of two second order tensors  $\sigma_{ij} \epsilon_{ij}$  is expressed as  $\boldsymbol{\sigma} : \boldsymbol{\epsilon}$ . Time derivatives are denoted with the dot notation where  $\dot{\boldsymbol{x}} = \frac{d\boldsymbol{x}}{dt}$  is the first and  $\ddot{\boldsymbol{x}} = \frac{d^2\boldsymbol{x}}{dt^2}$  the second material time derivative.

## 2 Physical and Numerical Model for Fluid-Structure-Interaction

### 2.1 Continuity Assumption and Balance Laws

The models for computational mechanics for both fluids and structures come from continuum mechanics. In continuum mechanics, we assume the medium to have continuous material properties like density or viscosity. This continuum assumption simplifies the "real world" since materials consist of discontinuous matter on the molecular level. While molecular dynamics is used for simulation on nanoscale, it is impossible to apply it on macroscale due to limited computing power [18]. However, continuum mechanics yields highly accurate results for macroscopic geometries where the length-scales in the simulation are orders greater than the mean free path length [16].

The concept of continuity enables us to describe the medium with continuous functions over its domain, allowing us to define its motion and properties. These functions can be derived, integrated and used in the balance laws. For a deforming continuum, these equations can be formulated in either a Eulerian point of view, describing points by their current location  $x_i$ , or a Lagrangian point of view with respect to their reference coordinates  $X_i$ . The balance laws in the Eulerian frame are the balance of momentum

$$\frac{\partial \sigma_{ij}}{\partial x_j} + \rho b_i = \rho \ddot{u}_i, \quad (2.1)$$

balance of mass

$$\dot{\rho} + \rho \frac{\partial v_i}{\partial x_i} = 0, \quad (2.2)$$

and balance of angular momentum

$$\sigma_{ij} = \sigma_{ji}, \quad (2.3)$$

with stress tensor  $\sigma_{ij}$ , body forces  $b_i$ , density  $\rho$ , displacements  $u_i$ , and velocity  $v_i$ . The balance equations are connected to the equations of motion by constitutive relations. Constitutive relations differ for solids and fluids, whereas the balance laws apply for all materials satisfying the continuity condition.

## 2.2 Structure mechanics

### 2.2.1 Constitutive Relations for Linear Elasticity

For the structure simulation, we follow the approach of J. Bleyer's tutorial for elastodynamics with a linear elastic material, a material that depends on only three parameters: Density  $\rho$ , and the two Lamé-parameters  $\mu$  and  $\lambda$  [2]. Linear-elastic means that deformations  $u_i$  are small and we use the linearized strain-tensor

$$\epsilon_{ij} = \frac{1}{2} \left( \frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right). \quad (2.4)$$

While the density affects dynamic aspects, the Lamé parameters relate the stresses  $\sigma$  and strains  $\epsilon$  with the constitutive relation

$$\sigma_{ij} = \lambda \delta_{ij} \epsilon_{kk} + 2\mu \epsilon_{ij} \quad (2.5)$$

where  $\delta_{ij}$  denotes the Kronecker delta. The Young's modulus  $E$  and Poisson's ratio  $\nu$  of a material can be converted to the Lamé parameters using

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad (2.6)$$

and

$$\mu = \frac{E}{2(1+\nu)}. \quad (2.7)$$

We calculate  $\mathbf{u}$  using the Finite Element Method (FEM) [17]. First, we multiply the balance of momentum (Equation 2.1) with a test function  $\mathbf{v} \in \mathbf{V}$  and integrate by parts over the domain  $\Omega$  to obtain the weak form

$$\int_{\Omega} \rho \ddot{\mathbf{u}} \cdot \mathbf{v} dx + \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{v}) dx = \int_{\Omega} \rho \mathbf{b} \cdot \mathbf{v} dx + \int_{\partial\Omega} (\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot \mathbf{v} ds \quad \forall \mathbf{v} \in \mathbf{V} \quad (2.8)$$

where the function space  $V$  is chosen according to the Dirichlet boundary conditions. We discard the first integral on the right hand side of the equation since there are no bodyforces  $\mathbf{b}$  in my FSI simulations. The second integral describes traction forces. Here, we substitute  $\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t}$ .  $\mathbf{t}$  is the traction vector and it contains the loads from the fluid. Then, we express the weak form (Equation 2.8) in terms of the displacements  $\mathbf{u}$  by applying the constitutive relation (Equation 2.5) and the definition of the strain tensor  $\boldsymbol{\epsilon}$  (Equation 2.4). We sort Equation 2.8 into functions of displacements  $\mathbf{u}$  and accelerations  $\ddot{\mathbf{u}}$  and obtain the form

$$\mathbf{m}(\ddot{\mathbf{u}}, \mathbf{v}) + \mathbf{k}(\mathbf{u}, \mathbf{v}) = \mathbf{L}(\mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{V}. \quad (2.9)$$

with

$$\mathbf{m}(\ddot{\mathbf{u}}, \mathbf{v}) = \int_{\Omega} \rho \ddot{\mathbf{u}} \cdot \mathbf{v} dx, \quad (2.10)$$

$$\mathbf{k}(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{v}) dx \quad (2.11)$$

and

$$\mathbf{L}(\mathbf{t}, \mathbf{v}) = \int_{\partial\Omega} \mathbf{t} \cdot \mathbf{v} ds \quad (2.12)$$

### 2.2.2 Time Integration with the Generalized $\alpha$ -Method

Following Bleyer [2], we discretize Equation 2.9 with constant time steps and perform time integration with the generalized  $\alpha$ -method [8]. The generalized  $\alpha$ -method is widely used in structural dynamics due to its second-order accuracy, A-stability and parameter controlled high frequency dissipation. The idea is to discretize Equation 2.9 in time and solve the equation for an intermediate point in time.

The discretized form reads

$$\mathbf{m}(\ddot{\mathbf{u}}_{n+1-\alpha_m}, \mathbf{v}) + \mathbf{k}(\mathbf{u}_{n+1-\alpha_f}, \mathbf{v}) = \mathbf{L}(\mathbf{t}_{n+1-\alpha_f}, \mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{V} \quad (2.13)$$

where the notation  $\mathbf{u}_{n+1-\alpha}$  is the linear interpolation of a function  $\mathbf{u}$  at  $t_{n+1-\alpha}$ :

$$\mathbf{u}_{n+1-\alpha} = (1 - \alpha)\mathbf{u}_{n+1} + \alpha\mathbf{u}_n. \quad (2.14)$$

We use the approximations

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t \dot{\mathbf{u}}_n + \frac{\Delta t^2}{2} ((1 - 2\beta)\ddot{\mathbf{u}}_n + 2\beta\ddot{\mathbf{u}}_{n+1}), \quad (2.15)$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \Delta t ((1 - \gamma)\ddot{\mathbf{u}}_n + \gamma\ddot{\mathbf{u}}_{n+1}), \quad (2.16)$$

and

$$\ddot{\mathbf{u}}_{n+1} = \frac{1}{\beta\Delta t^2} (\mathbf{u}_{n+1} - \mathbf{u}_n - \Delta t \dot{\mathbf{u}}_n) - \frac{1 - 2\beta}{2\beta} \ddot{\mathbf{u}}_n \quad (2.17)$$

to express Equation 2.13 in terms of  $\mathbf{u}_{n+1}$  and known quantities at  $t_n$ . Equation 2.13 is the form that we use to describe our problem in FEniCS. FEniCS automatically rearranges Equation 2.13 to an equation of the form

$$\bar{\mathbf{K}}(\mathbf{u}_{n+1}, \mathbf{v}) = F(\mathbf{v}, \mathbf{L}(t_n), \mathbf{L}(t_{n+1}), \mathbf{u}_n, \dot{\mathbf{u}}_n, \ddot{\mathbf{u}}_n) \quad \forall \mathbf{v} \in \mathbf{V} \quad (2.18)$$

with the bilinear form  $\bar{\mathbf{K}}$  and the linear form  $F$ , such that only  $\bar{\mathbf{K}}$  depends on the unknowns  $\mathbf{u}_{n+1}$ .

We use two different choices for the parameters: First,  $\alpha_m = 0.2$  and  $\alpha_f = 0.4$ , which is a popular choice in structure dynamics, and second,  $\alpha_m = \alpha_f = 0$ , such that the generalized  $\alpha$ -method equals the Newmark- $\beta$ -method [21].  $\gamma$  and  $\beta$  are defined as  $\gamma = \frac{1}{2} + \alpha_m - \alpha_f$ , and  $\beta = \frac{1}{4} \left( \gamma + \frac{1}{2} \right)^2$ . In section 5.2, I explain why I used 2 different sets of parameters. Finally, we solve Equation 2.18 using Finite Elements in space for every timestep.

## 2.3 Fluid mechanics

For modeling fluids, I use a Newtonian fluid following the FSI3 benchmark by S. Turek and J. Hron in their benchmarking proposal [26]. Since my work is focused on the structure and I use the fluid setups from D. Risseuw's Master Thesis [23], no specific knowledge in fluid mechanics is required to understand the following chapters. The fluid simulation is treated as a "black box" that reads mesh deformations  $\mathbf{u}$  and writes forces  $\mathbf{t}$  on the boundary. Readers interested in more details can find those in the references.

## 2.4 Implicit Coupling

For FSI simulations, the equations for the fluid and the structure are coupled. In the partitioned approach, we have two subsystems that read boundary conditions from each other. The fluid solver  $F$  reads displacements  $\mathbf{u}$  as input and calculates forces  $\mathbf{t}$  as a response:

$$F(\mathbf{u}) = \mathbf{t} \quad (2.19)$$

The solid solver  $S$  calculates its response accordingly:

$$S(\mathbf{t}) = \mathbf{u} \quad (2.20)$$

Explicit coupling is the simplest way to implement this coupling. In explicit coupling, at least one solver calculates its response for the next timestep  $t_{n+1}$  with the input from  $t_n$ :

$$\mathbf{t}_{n+1} = F(\mathbf{u}_n) \quad (2.21)$$

The other solver either uses the input  $\mathbf{t}_{n+1}$  in a serial staggered scheme or  $\mathbf{t}_n$  in a parallel staggered scheme to compute  $\mathbf{u}_{n+1}$ . Explicit coupling is a loose coupling scheme, since there is a shift between the coupling variables  $\mathbf{t}$  and  $\mathbf{u}$ . Explicit coupling only needs one iteration per timestep and is therefore computationally cheap. However, it suffers from instabilities, because the added mass effect is not taken into account properly. The added mass effect describes the additional inertia in a system when a solid is accelerated in a fluid. It occurs because the solid has to move the surrounding fluid. With implicit coupling, we can represent the added mass effect accurately and avoid these instabilities [5].

In implicit coupling, we approximate

$$\mathbf{t}_{n+1} = F(\mathbf{u}_{n+1}), \quad \mathbf{u}_{n+1} = S(\mathbf{t}_{n+1}) \quad (2.22)$$

iteratively until the convergence criteria are satisfied. Convergence can be accelerated by a postprocessing method that takes previous iterations into account. For postprocessing, I chose the Interface Quasi-Newton method (IQN-ILS) since it performed best in [9].

preCICE offers ready to use implementations of various coupling methods. Users only specify the coupling scheme, postprocessing method and convergence limits. More details on implicit coupling with the Interface Quasi-Newton Method and other cou-

pling methods can be found in B. Gatzhammer's PhD Thesis [12] and B. Uekermann's PhD Thesis [27].

## 2.5 Black Box Surface Coupling

In partitioned FSI, structure and fluid share a common surface. This surface is represented in both the structure mesh and the fluid mesh. On this surface, both solvers receive a boundary condition and write a boundary condition for the other solver. This boundary condition is communicated between the solvers by a vector at each node of their mesh. If the meshes match at the interface, this nodal data can be directly transferred to the other solver. However, if the meshes do not match, the nodal data has to be mapped from one mesh to the other.

### 2.5.1 Conservative and Consistent Mapping

In the following, I restrict myself to the example present in my FSI simulations: Conservative mapping of forces from the fluid to the structure mesh and consistent mapping of displacements from the structure to the fluid mesh. For more details on conservative and consistent mapping, see [12].

How the nodal data is mapped to the other mesh depends on what it resembles. Conservative mapping applies to vectors that resemble an integral over a part of the coupling surface. Forces are mapped conservatively since the force at a node resembles the integral over the pressure and shear stress around this node. Every part of the coupling surface belongs to the integration domain of a node. The magnitude of the force vector increases for a coarser mesh, since there are less nodes and the integral for a single node is performed over a bigger area. Consistent mapping applies to quantities that are independent from the mesh fineness, such as displacements. The vectors at the nodes resemble samples of a continuous function defined on the whole surface.

### 2.5.2 Mapping Implementations: Nearest-Neighbor and Radial Basis Functions

There are different schemes that can be applied to both conservative and consistent mapping. Details on various mesh mapping schemes are in [27]. Again, I restrict myself to force and displacement mapping in FSI.

The simplest mapping scheme is nearest-neighbor mapping. For conservative mapping of forces from fluid to structure, the force on a node of the solid mesh is the sum over the forces of all fluid nodes that are closer to this solid node than to any other solid node. In consistent mapping of displacement from structure to fluid, the displacement of a fluid node equals the displacement of the nearest structure node. This way of mapping displacements leads to a ragged fluid boundary if the fluid mesh is finer than the structure mesh, since it is a zero-order interpolation.

To prevent this, I use a mapping scheme based on radial basis functions (RBF) with thin plate splines [27]. RBF mapping interpolates the data between the nodes such that a smooth fluid boundary is reconstructed. RBF mapping is sensitive for instabilities in certain cases [19]. In subsection 5.1.2, these instabilities are discussed concerning the extrapolation of displacements. However, it interpolates with first order and only needs the same nodal data as nearest-neighbor mapping. Therefore, its use is beneficial over other higher order mapping schemes that need connectivity information of the meshes like nearest projection. Figure 2.1 compares the meshes of the same simulation with nearest-neighbor mapping on the left and RBF-mapping on the right.

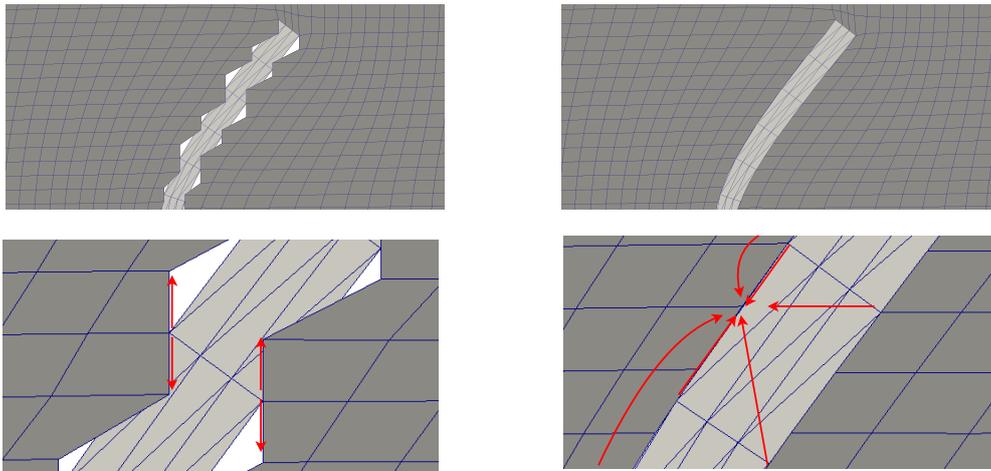


Figure 2.1: Nearest-neighbor mapping results in a ragged fluid domain for coarse structure meshes (left), whereas the RBF-mapping (right) leads to a smooth interface. On the bottom, arrows indicate which structure nodes influence the fluid node. For nearest-neighbor mapping, it is just 1 structure node for each fluid node, whereas all the surrounding structure nodes affect the position of a fluid node in RBF-mapping.

## 3 Software Components

### 3.1 preCICE

The multi-physics coupling library preCICE [4] provides the tools to couple single physics solvers for multi-physics simulations. It supports schemes for mapping between different meshes and timesteps and is mainly used for surface coupling. preCICE is available on GitHub<sup>1</sup> under the LGPL 3.0 Licence<sup>2</sup>.

An advantage of coupling with preCICE is that users can add new coupled components to their simulation with few changes in the original. Plus, the components of a simulation are exchangeable if users want to test different solver combinations. The solvers are treated as "black-boxes": preCICE only observes the data that is sent by the solvers and manipulates their input without knowing their internal models.

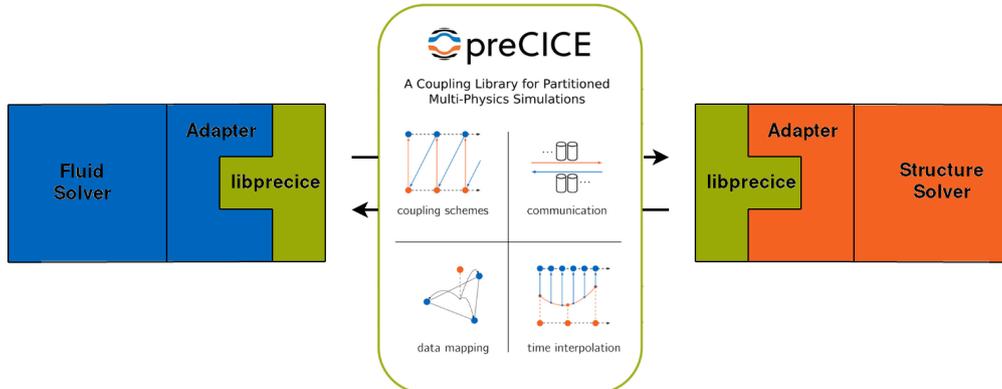


Figure 3.1: General scheme of a coupled FSI simulation using preCICE. In my case, I use OpenFOAM as fluid solver and a FEniCS structure solver with their respective adapters. Adapted from [22]

An FSI simulation with a FEniCS structure solver and OpenFOAM consists of five

<sup>1</sup><https://github.com/precice/precice>

<sup>2</sup><https://opensource.org/licenses/lgpl-3.0.html>

components as seen in Figure 3.1. On the left hand side is the fluid solver OpenFOAM, connected to preCICE with the OpenFOAM-adapter. On the right, the preCICE-FEniCS adapter connects the structure solver FEniCS with preCICE.

An adapter converts the coupling data from the representation in preCICE to a representation that the solver uses internally and vice versa. There are ready-to-use adapters for many well known solvers like OpenFOAM, CalculiX, deal.II or SU<sup>2</sup> [4][28]. For other solvers, an adapter can be implemented "in a few weeks" [22]. preCICE provides a C++, Python, Fortran and C API. Adapters are usually written in the language of the solver and use the corresponding preCICE API.

## 3.2 FEniCS

FEniCS<sup>3</sup> is an open-source project for solving partial differential equations (PDEs) with the Finite Element Method (FEM). Well-documented tutorials and the high-level Python interface make it beginner friendly, while powerful features also attract experts. A key feature is the handling of abstract weak forms, such that scientific models are quickly translated to a FEM simulation. For a simple FEniCS simulation, users specify a mesh, a function space for test and trial functions, the weak form of the PDE, and boundary conditions. Time dependent simulations require the user to implement an additional timestepping scheme.

## 3.3 preCICE-FEniCS Adapter

The previously available FEniCS-preCICE adapter was implemented by B. R uth. He simulated conjugate heat transfer with OpenFOAM and FEniCS [25]. The adapter transforms boundary conditions from a representation that FEniCS understands to the representation preCICE needs and vice versa. To write to preCICE, the adapter evaluates the coupling function at the coupling nodes and writes those values in a NumPy<sup>4</sup>-array to preCICE. The read direction is more sophisticated for a Neumann boundary condition such as forces. The forces  $\mathbf{t} = \boldsymbol{\sigma} \cdot \mathbf{n}$  are a part of the weak form (Equation 2.18). In FEniCS, the weak form is an abstract expression. The adapter creates an expression from the nodal values with a subclass of FEniCS' UserExpression. The

---

<sup>3</sup><https://fenicsproject.org/>

<sup>4</sup><https://numpy.org/>

evaluation function of the `UserExpression` is overloaded with an RBF-interpolation of the nodal values using `SciPy`<sup>5</sup>. This expression is then added to the weak form.

The extension of the `preCICE-FEniCS Adapter` for FSI was a main goal of my thesis. The new functionality enables users to couple vector valued functions in `FEniCS` with `preCICE`, using only five function calls in their simulation script (see Figure 4.1). Additionally, the new adapter can read quantities that were mapped conservatively (see section 2.5 for conservative and consistent mapping). I describe the new features and improvements in detail in section 4.2. The version I used is on `GitHub`<sup>6</sup>.

## 3.4 OpenFOAM and OpenFOAM Adapter

`OpenFOAM`<sup>7</sup> is a popular open-source software using the Finite Volume Method for numerical simulations. It started with computational fluid dynamics and was extended with solvers for electromagnetics, heat transfer and other applications. L. Cheung already coupled `OpenFOAM` and `preCICE` for conjugate heat transfer in her Master's Thesis in 2016 [6] and G. Chourdakis developed a general `OpenFOAM-preCICE` adapter in his Master's Thesis [7] based on [6]. D. Risseuw simulated wind turbines (FSI) with `preCICE`, `OpenFAOM` and `CalculiX` in 2019 [23], where he also ran the benchmark scenario from Turek and Hron [26] to validate his models. The benchmark scenario and an elastic flap are hosted as ready-to-use tutorials for FSI with `preCICE` on `GitHub`<sup>8</sup>, such that I could extract the fluid case and use it for my FSI simulations with `FEniCS` instead of `CalculiX` with only minor changes in the fluid setup. I used `OpenFOAM 5.x` for my simulations.

---

<sup>5</sup><https://www.scipy.org/>

<sup>6</sup><https://github.com/precice/fenics-adapter/tree/6939ad4daa7365a9a9fa0972980fa8186bb94603>

<sup>7</sup><https://openfoam.org>

<sup>8</sup><https://github.com/precice/>

# 4 Implementation

## 4.1 A FEniCS Solver for Linear Elastic Structures

FEniCS is a platform to solve PDEs that lets users implement their own solvers. To solve the equations from section 2.2, I derived a 2D solver from Bleyers 3D-transient elastodynamics tutorial [2]. My simulation script contains the solver and the specific cases described in chapter 5 in one file. The two cases are basically beams that differ in length, width, and the location of the boundary conditions, so the script can be used in both cases with minor modifications.

I chose continuous Galerkin elements with second order Lagrange polynomials on a mesh of triangles. The boundary is split up into a clamped Dirichlet boundary and a Neumann boundary where the coupling takes place. On the Neumann boundary, we write displacements and read forces. Figure 4.1 visualizes the control flow with focus on the coupling. In the beginning, the adapter is initialized with the corresponding functions. Then, the read field is incorporated into the simulation as a boundary condition. This can be done in two different ways as I will explain in subsection 4.2.4.

The timestep is defined in the preCICE-configuration file and read via the adapter during initialization. For implicit coupling, we compute several iterations for one timestep. Until convergence, the solver updates the forces and writes displacements by calling the `advance()` function of the preCICE-FEniCS adapter in every iteration. The solver advances to the next timestep if preCICE marks the current timestep as converged. Once the coupling is completed, the solver postprocesses the results.

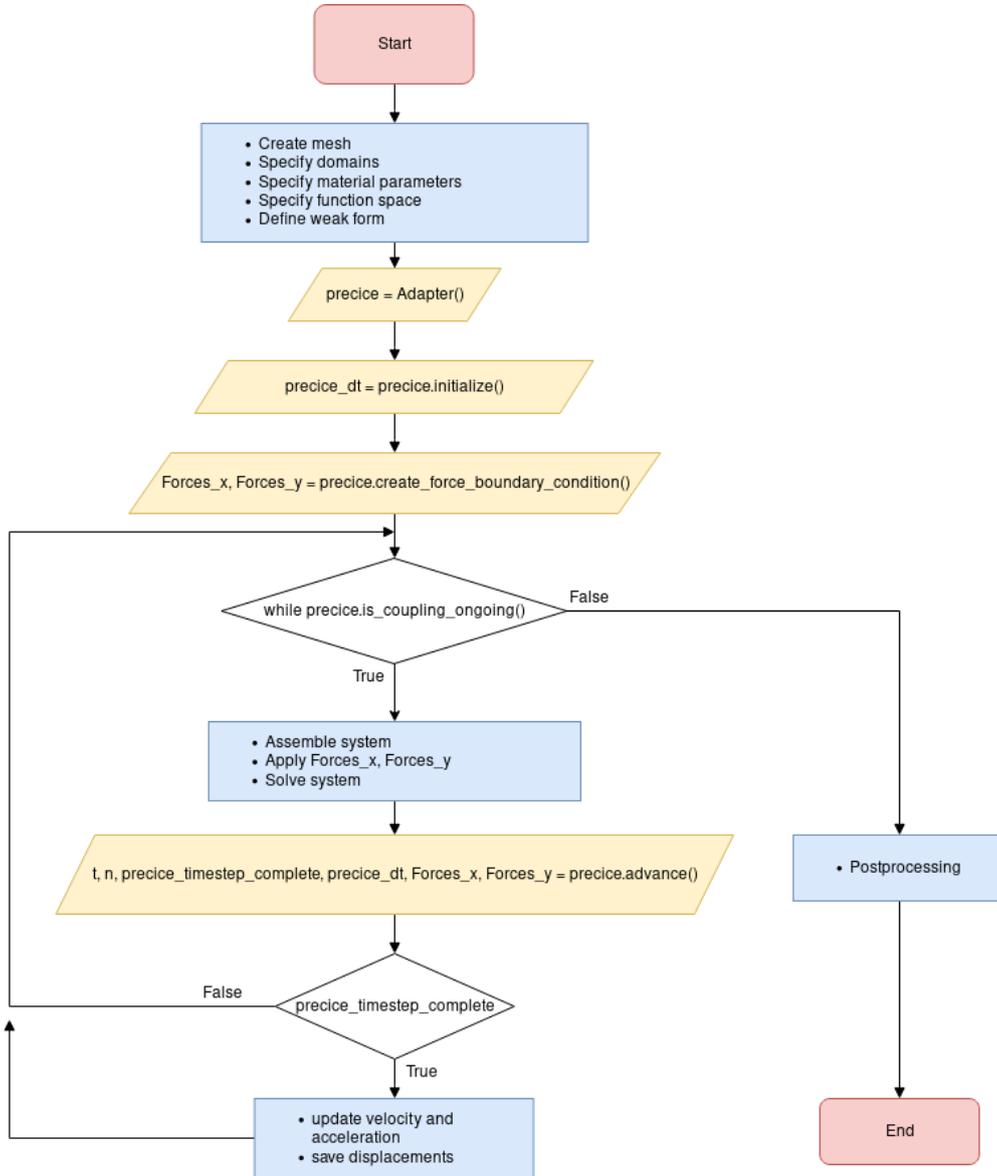


Figure 4.1: Overview over the structure solver. Parallelograms are methods of the preCICE-FEniCS Adapter. Boxes show a summary of the solver's actions.

## 4.2 New Features in the FEniCS-preCICE Adapter

### 4.2.1 Handling Vector Valued Functions

So far, the official preCICE-FEniCS adapter featured scalar valued functions only. In FSI however, we couple vector functions. In preparatory work, Rafal Kulaga coupled two FEniCS structure solvers in a static split beam. Scalar and vector data need different preCICE functions, so I implemented wrappers that automatically call the correct preCICE function to read or write the coupled quantity. These wrappers are called in the `initialize()` and `advance()` step (see Figure 4.1).

### 4.2.2 3D-2D Coupling

OpenFOAM provides 3D solvers for 3D meshes only. However, users can simulate 2D scenarios with a mesh that has only one layer in the third dimension and solve for the flow field only in the two actual dimensions. We call such a setup pseudo-3D.

In FEniCS, users choose the dimensions of their simulation themselves. I solve my 2D structure simulation with a 2D solver for minimal computational effort. The preCICE-FEniCS adapter maps between a 2D simulation in FEniCS and pseudo-3D OpenFOAM to allow coupling. For preCICE, the simulation is 3D, although the z-component is ignored in the RBF-mapping (see section 2.4 or [27]) of forces and displacements.

The 2D-3D mapping is needed for three sets of data: Forces, displacements and the positions of the nodes. For the forces, the adapter deletes the z-components, whereas it appends zeros for the displacements and positions in the z-dimension. The forces additionally have to be scaled with  $1/\Delta d$ , where  $\Delta d$  is the thickness of the fluid domain, since the forces are proportional to  $\Delta d$  and the stiffness of the 2D structure in FEniCS corresponds to a structure with unit thickness in the z-direction. I chose  $\Delta d = 1$  such that the scaling is unnecessary.

### 4.2.3 Runtime Improvements by Buffering RBF-Interpolation

Test simulations ran much longer than expected. OpenFOAM seemed to wait for FEniCS much longer than vice versa, although the structure has less degrees of freedom. Profiling showed that the preCICE-FEniCS adapter spends most of the time computing RBF-interpolations. Whenever the read function was evaluated, a new interpolant was

initialized on the whole mesh. The runtime decreased by more than 80% when the interpolation is buffered for reuse. The two lower bars in Figure 4.2 show the runtimes before and after the optimization for an elastic flap in a channel.

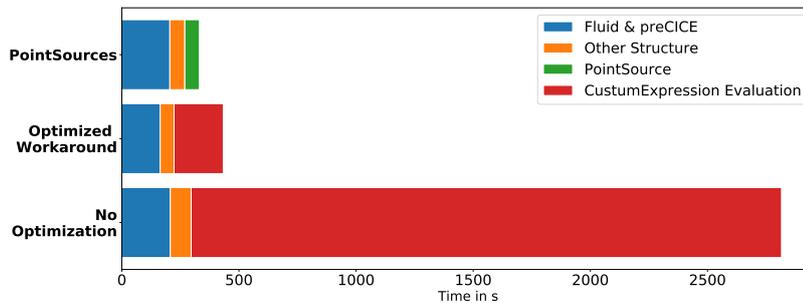


Figure 4.2: Runtimes of the simulation of an elastic flap with different versions of the preCICE-FEniCS adapter. The top bar shows the approach with PointSources as presented in subsection 4.2.4, the other bars compare the approach with a CustomExpression before and after optimization.

#### 4.2.4 Reading Forces as Point Loads

Despite the new runtime improvements from subsection 4.2.3, the read-pipeline presented in section 3.3 is not optimal for FSI with OpenFOAM. OpenFOAM writes point loads with conservative mapping between nodes, whereas the preCICE-FEniCS adapter expects the read-field to be a distributed load as we read it from consistent mapping (section 2.4). I solved this issue with 2 different approaches: First, a workaround that can handle simple geometries and second, a more general solution that treats the forces as point loads.

The workaround is valid for structures, where every edge on the fluid structure interface has the same length  $h$ . In my test cases, the structure is a rectangle. We can easily choose the coupling boundary to consist of edges of length  $h$ . If the solver scales the forces with  $\frac{1}{h}$ , they can be treated like a force density and the read pipeline from section 3.3 with the optimization from subsection 4.2.3 is a valid approach.

It might not always be possible to choose a mesh with constant edge length and the adapter should support coupling on arbitrary meshes. In a new approach, the forces are treated as point loads on the nodes instead of a continuous function. FEniCS offers functionality for point loads with the PointSource feature. The adapter creates a list

of `PointSources` and returns it to the solver in every `advance()` call. Then, they are applied to the assembled system.

`PointSources` are only implemented for scalar function spaces, such that every point load is split up into its  $x$ - and  $y$ -component and a `PointSource` is created for each component using its corresponding subspace. Plus, their magnitude cannot be modified and they can only be applied to an assembled system and not to the weak form. This forces the solver to explicitly assemble the system and apply the forces without preprocessing. In the generalized  $\alpha$ -method (see subsection 2.2.2), the solver is supposed to apply the forces at  $t_{n+1-\alpha_f}$ , an interpolation of the forces at  $t_n$  and  $t_{n+1}$ . This interpolation is not easily computed from a list of `PointSources`, so the solver applies the forces of  $t_{n+1}$  instead, as it receives those forces in implicit coupling. This impacts the convergence for  $\alpha_f > 0$  and is discussed with an example in section 5.2. The `PointSource` approach lowers the runtime even further, as seen in Figure 4.2.

#### 4.2.5 Special Treatment of Dirichlet-Boundary Nodes Inside the Coupling Boundary

At the fluid-solid interface there might be nodes that belong to the Dirichlet boundary of the solid, where the displacement is zero. For the fluid solver, these nodes belong to the coupling boundary, so the solver expects to read valid displacements and it writes forces on that point. However, applying this force as a `PointSource` on the Dirichlet boundary of the structure leads to instabilities. We also cannot exclude these points from the structure's coupling domain, since `preCICE` then extrapolates displacements of surrounding nodes. This extrapolation shifts the fluid domain such that fluid and structure overlap or a gap appears.

In section 5.1, I provide examples for these issues. Figure 5.3 shows instabilities that occur with a `PointSource` on the Dirichlet boundary and Figure 5.2 shows the nonphysical overlap and gap. I solved this issue in the `preCICE-FEniCS` adapter: The adapter receives the Dirichlet boundary domain additionally to the coupling domain. It only creates a `PointSource` for nodes of the coupling domain that are not inside the Dirichlet boundary at the same time.

## 5 Testing and Validation

I chose 2 different FSI scenarios to test and validate the simulation software. In the first, I analyze the flow through a channel with an elastic flap qualitatively. The simulation runs in less than 6 minutes and shows whether the displacement mapping is performed correctly. Plus, I provide details to the challenges from section 4.2 and measure the difference between the two approaches to read forces (see subsection 4.2.4). This case is a quick way to check whether the framework gives reasonable results. In the second case, I simulate the well known FSI3 benchmark [26], a simulation in which the flow around a cylinder with an elastic flap causes the flap to oscillate. While this simulation takes much longer to compute (2-10 hours depending on the timestep), it allows a quantitative comparison to reference results from validated setups.

### 5.1 Elastic Flap in a Channel

#### 5.1.1 Setup

In this simulation I calculate the flow through a channel with an elastic flap attached to the ground. I use my solver from section 4.1 with the geometry described in Figure 5.1 and Table 5.1. The fluid enters the domain on the left with a constant velocity of  $v_i = 10 \frac{m}{s}$ . I chose the parameters (Table 5.1) such that the elastic flap bends significantly under the load of the flow.

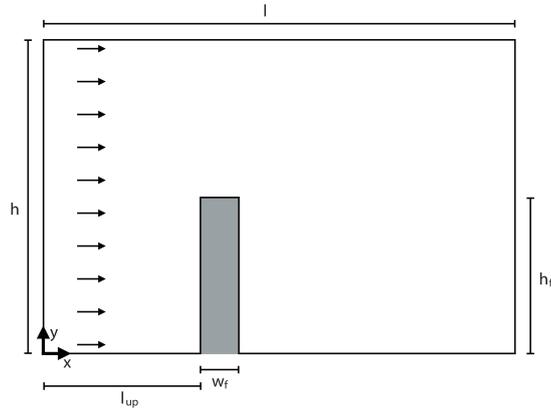


Figure 5.1: Setup of the channel with an elastic flap. The dimensions and material parameters are listed in Table 5.1. Taken from [24]

Geometry Parameter	Value	Unit
$l$	6	$m$
$h$	4	$m$
$l_{up}$	2.95	$m$
$w_f$	0.1	$m$
$h_f$	1	$m$
Material Parameter Structure	Value	Unit
$\rho_s$	1000	$\frac{kg}{m^3}$
$E$	400000	$\frac{kg}{ms^2}$
$\nu$	0.3	
Material Parameter Fluid	Value	Unit
$\rho_f$	1000	$\frac{kg}{ms^2}$
$\nu_f$	0.001	$\frac{m^2}{s}$
Simulation Parameter	Value	Unit
$\Delta t$	0.01	$s$
$\alpha_m, \alpha_f$	0	
# Cells Fluid Domain	2790	
# Elements Solid Domain	640	

Table 5.1: Parameters of the elastic flap testcase.

### 5.1.2 Stability

In section 4.2, I mentioned issues at the intersection of the structure's Dirichlet boundary and the coupling boundary. For the elastic flap case, this intersection consists of two points: The left and right bottom node. Figure 5.2 shows exemplary how the fluid mesh deforms if these points are excluded from the structure's coupling domain.

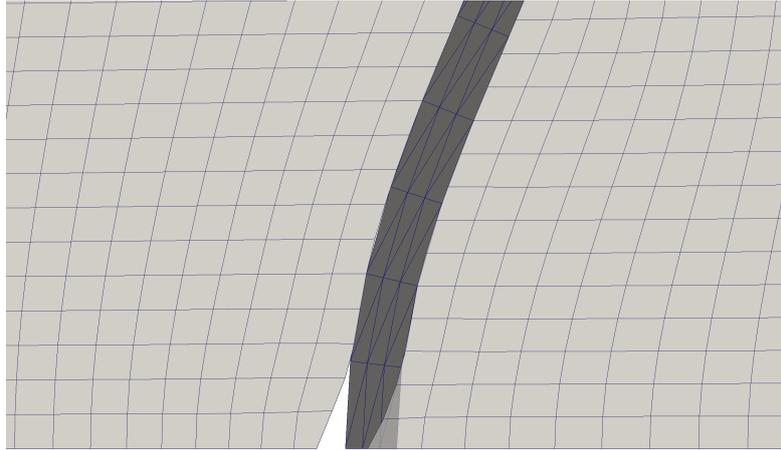


Figure 5.2: A shift of the fluid domain near the structure's Dirichlet boundary occurs if the bottom nodes of the structure don't belong to the coupling boundary and the RBF-mapping extrapolates the displacements for the fluid nodes.

Instabilities also occur if the preCICE-FEniCS adapter does not filter for Dirichlet nodes in the coupling domain. In this case the coupled simulation crashes in the first timestep. A structure-only simulation reveals unphysical deformations of Dirichlet nodes as seen in Figure 5.3 when `PointSources` are applied along the left side of the beam.

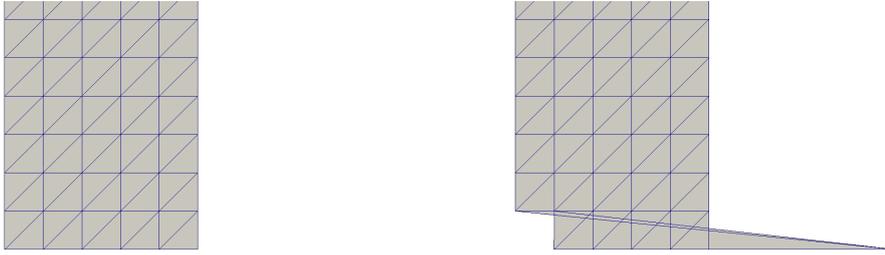


Figure 5.3: Lower part of the structure mesh before (left) and after (right) the first timestep: The node on the bottom left is pulled all the way through the beam by a PointSource. This happens because the node is also part of the Dirichlet boundary.

### 5.1.3 Results

The simulation ran in about 5 minutes. The flap bends to the right under the load of the fluid as seen in Figure 5.6. At 1.8 seconds the flap reached its maximum deformation. Then it swings back to an almost vertical position. The RBF mapping of displacements results in a smooth fluid mesh at the interface. A snapshot of the simulation at 5 s is shown in Figure 5.4.

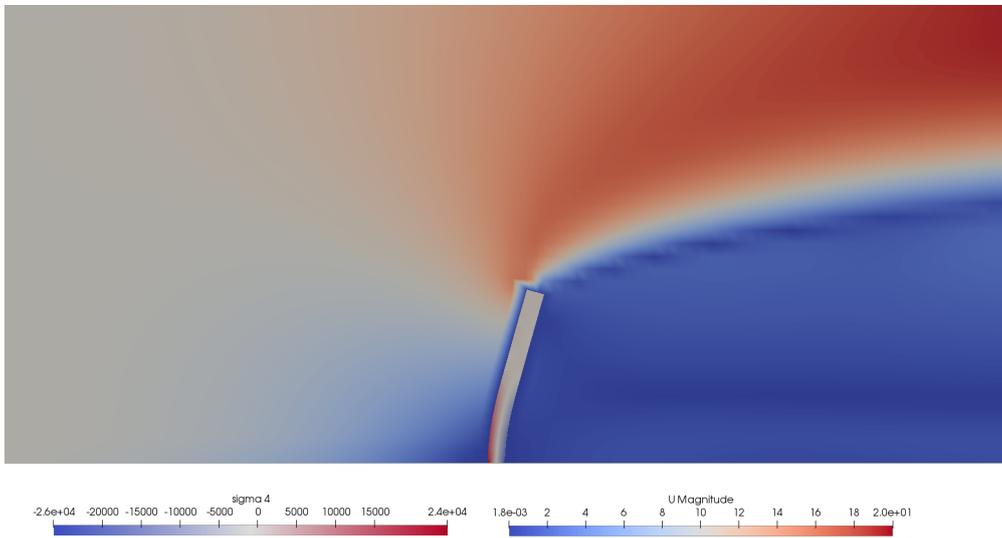


Figure 5.4: Velocity magnitude and normal stresses at 5 s in the elastic flap case. The highest stresses occur around the left and right bottom nodes.

### Workaround vs. PointSources

In subsection 4.2.4 I presented two approaches to incorporate the forces into the structure solver. Figure 5.5 compares them for an otherwise equivalent simulation. The absolute difference is less than  $0.01\text{ m}$  throughout the whole simulation. The relative difference is less than 1 % except for the first timesteps, where very small displacements introduce a high relative difference despite a very small absolute difference.

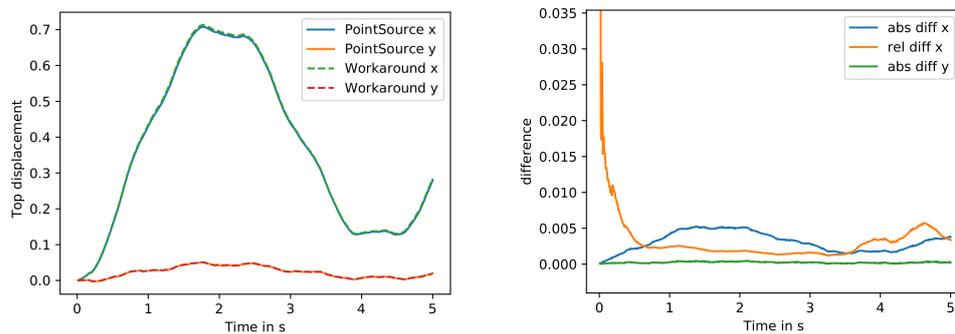


Figure 5.5: Displacement (left) and difference (right) of the point (0,1) in the elastic flap simulation for the workaround and PointSource approach.

### Comparison to a Linear and Nonlinear Structure Simulated by CalculiX

The same simulation is on preCICE's GitHub<sup>1</sup> with CalculiX<sup>2</sup> as structure solver. The simulation uses a nonlinear hyperelastic St. Venant-Kirchhoff model that is also valid for larger deformations. The material can be changed to a linear elastic material, such that the inaccuracy of the linear elastic material can be evaluated. In Figure 5.6, I compare the displacement of the top of the flap for my FEniCS solver with a linear and a nonlinear CalculiX solver for the same fluid case. In the x-direction, the solvers show similar results, though the linear solvers overestimate the displacements. In the y-direction, both linear solvers fail to simulate the deformation properly.

<sup>1</sup><https://github.com/precice/precice/wiki/Tutorial-for-FSI-with-OpenFOAM-and-CalculiX>

<sup>2</sup><http://www.calculix.de/>

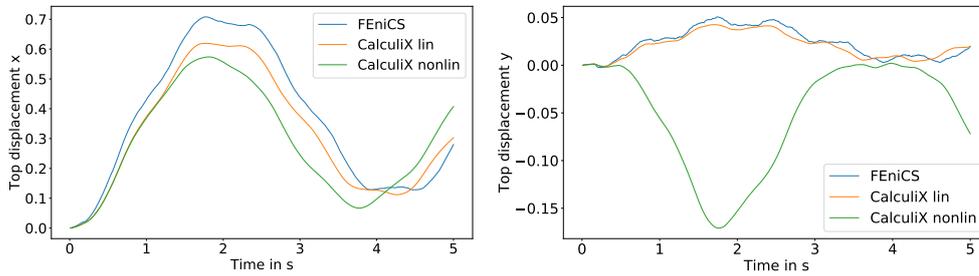


Figure 5.6: Displacement of the point (0,1) in  $m$  in the elastic flap simulation in  $x$ -direction (left) and  $y$ -direction (right) from 0 to 5 s for linear elastic FEniCS, a linear elastic CalculiX and a nonlinear CalculiX Solver.

## 5.2 FSI3 Benchmark - Flow Around a Cylinder with a Flap

In 2007 S. Turek and J. Hron proposed three FSI benchmark simulations to test FSI solvers [26]. FSI1 converges to a steady state whereas FSI2 and FSI3 are time-dependent and result in a periodic oscillation of the structure. I chose to simulate the FSI3 case because it shows large deformations and the added mass effect [32]. There are minor differences between my simulation and the original FSI3. The original simulation features a St. Venant-Kirchhoff material whereas mine features a linear-elastic material. Plus, my inlet has a constant velocity instead of a parabolic velocity profile for practical reasons.

### 5.2.1 Setup

In this simulation, the fluid enters the computational domain from the left. The velocity is linearly ramped up to  $v_i = 2 \frac{m}{s}$  in the first two seconds. For the outlet on the right, the pressure  $p_{out}$  is set to zero. The other boundaries of the fluid domain (top, bottom, cylinder and flap) are no-slip walls.

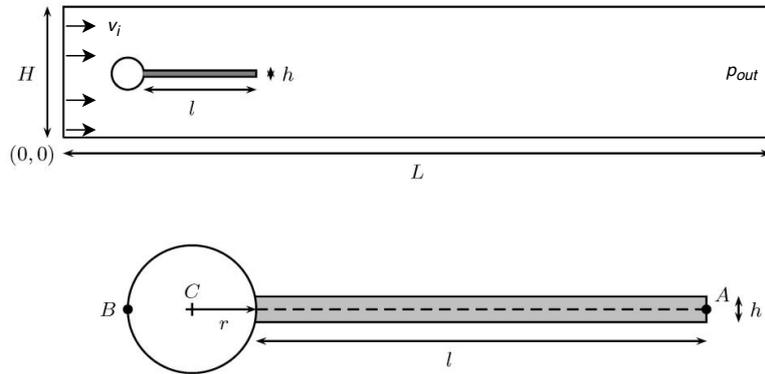


Figure 5.7: Setup of the FSI3 Benchmark Scenario. Parameters are specified in Table 5.2. Adapted from [23].

The schematic setup is shown in Figure 5.7. Geometric and material parameters are defined in Table 5.2 and Table 5.3. The cylinder is a fixed rigid body and therefore excluded from the computational domain. The elastic flap consists of linear elastic material (2.2) and is attached to the cylinder on its left end.

Geometry Parameter	Value	Unit
$H$	0.41	$m$
$L$	2.5	$m$
$l$	0.35	$m$
$h$	0.02	$m$
$r$	0.05	$m$
Points	Coordinates	Unit
$A$	(0.6, 0.2)	(m, m)
$B$	(0.15, 0.2)	(m, m)
$C$	(0.2, 0.2)	(m, m)
Boundary conditions	Value	Unit
$v_i$	2	$\frac{m}{s}$
$p_{out}$	0	$Pa$

Table 5.2: Geometry of the FSI3 Simulation.

Material Parameter Structure	Value	Unit
$\rho_s$	1000	$\frac{kg}{m^3}$
$E$	5600000	$\frac{kg}{ms^2}$
$\nu$	0.4	
Material Parameter Fluid	Value	Unit
$\rho_f$	1000	$\frac{kg}{m^3}$
$\nu_f$	0.001	$\frac{m^2}{s}$

Table 5.3: Material parameters in the FSI3 Simulation.

I run the simulation with two different parameter sets for the generalized  $\alpha$ -method:  $\alpha_m = 0.2$ ,  $\alpha_f = 0.4$  and  $\alpha_f = \alpha_m = 0$ . The first set corresponds to optimal high frequency dissipation [2], however the setup fails to evaluate the forces at  $t_{n+1-\alpha_f}$ , such that they are evaluated at  $t_{n+1}$  instead. The second set refers to the Newmark-method [21], where the forces are correctly evaluated at  $t_{n+1}$ . For each, I test two different timesteps:  $\Delta t = 0.001s$  and  $\Delta t = 0.005s$ .

The fluid mesh consists of 6306 cells. The structure mesh is coarser at the interface along the x-direction and finer in the y-direction. It contains 160 triangular elements and is illustrated in Figure 5.8 along with the surrounding fluid mesh. On the solid-fluid interface, discrete loads are applied on every boundary node of the structure mesh with PointSources as explained in subsection 4.2.4.

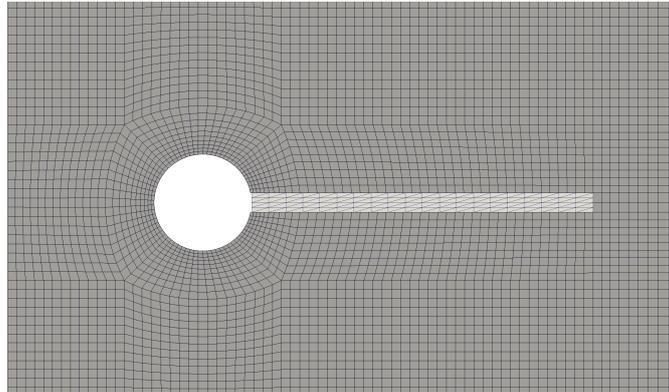


Figure 5.8: Mesh of and around the flap for the FSI3 simulation.

### 5.2.2 Testing the Structure Solver - CSM3

Turek and Hron suggest three tests for the structure solver, where only gravity is applied on the structure and there is no coupling between fluid and structure [26]. I simulate their third test CSM3 since it is the only time-dependent case. The material and geometry parameters are the same as for the FSI3 (see Table 5.3 and Table 5.2) except for the Young's modulus  $E$  which is  $1.4 \text{ MPa}$  instead of  $5.6 \text{ MPa}$ . For this test, I consider the first term on the right hand side of Equation 2.8 with  $\mathbf{b} = (0, -2\frac{m}{s^2})$  instead of the second term. In FEniCS, I add  $\int_{\Omega} \rho \mathbf{b} \cdot \mathbf{v} dx$  to the weak form and omit the coupling forces.

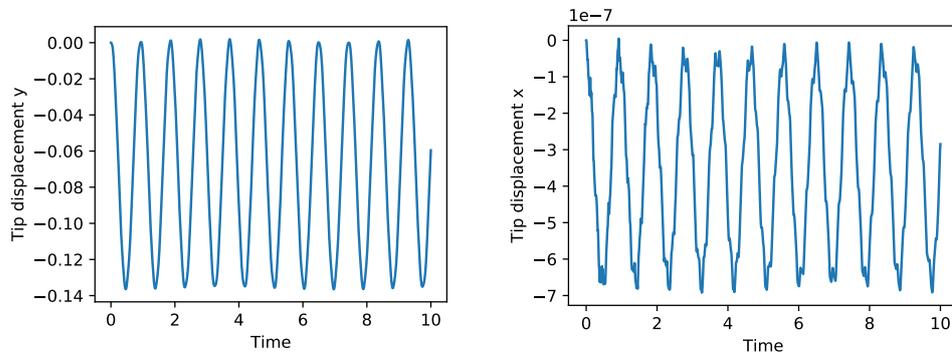


Figure 5.9: Displacement (in  $m$ ) of point A in the CSM3 test in y-direction (left) and x-direction (right) from 0 to 10 s.

Quantity	My Results	Turek & Hron	Abs. variance	Rel. var.
Frequency [Hz]	1.076	1.09	0.014	1.3 %
$\Delta y(A)$ [mm]	$-67.3 \pm 69.2$	$-63.6 \pm 65.2$	$1.9 \pm 4.0$	6 %
$\Delta x(A)$ [mm]	$-0.000035 \pm 0.000035$	$-14.3 \pm 14.3$	$-14.3 \pm 14.3$	100 %

Table 5.4: Comparison of my partitioned CSM3 with FEniCS and OpenFOAM to reference simulations.

Figure 5.9 shows the deflection of the point A under the gravitational loading. The vertical deflections oscillate periodically, whereas the horizontal deflections are very small (max  $-6.92 \times 10^{-7} m$ ) and jagged. In Table 5.4, I compare characteristic values from my CSM3 simulation to the reference values from [26]. The frequencies almost match

with a variance of 1.3 %. The deflection in the y-direction is about 6 % bigger in my simulation than in the reference case. My linear elastic beam hardly deforms in the x-direction, whereas the beam with St. Venant-Kirchhoff material shortens significantly under the perpendicular gravitation.

### 5.2.3 FSI3 Results

#### Stability

The simulation converges for  $\alpha_m = \alpha_f = 0$  for both timesteps  $\Delta t$ . However, it only converges for the larger timestep  $\Delta t = 0.005s$  for  $\alpha_m = 0.2$  and  $\alpha_f = 0.4$ . For  $\Delta t = 0.001s$ , the displacement of  $A$  shows high frequency oscillations as seen in Figure 5.10 before OpenFOAM crashes at  $t \approx 1.1s$ .

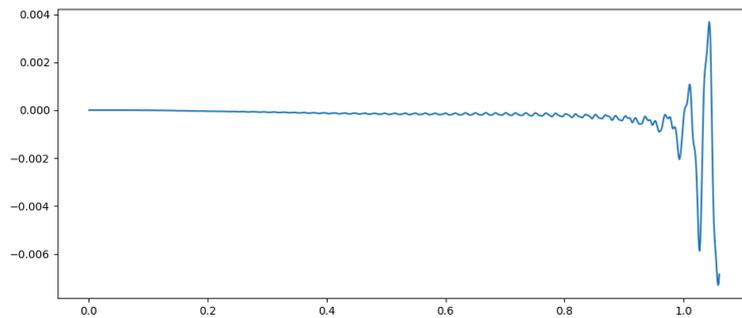


Figure 5.10: For a timestep  $\Delta t = 0.001s$ ,  $\alpha_m = 0.2$  and  $\alpha_f = 0.4$  the simulation crashes after 1.1 s. Before the crash, the flap oscillates with a high frequency.

#### Converging Simulations

For the other parameter choices, the elastic flap oscillates as seen in Figures 5.11 and 5.12. The strains are small during the ramp up of the velocity from 0 to 2 s. At 2.5 s, the deformations start to be visible and they peak between 4 and 4.5 s. After that point, the tip shows a periodic displacement in the y-direction. Table 5.5 gives characteristic values on the deformation of the structure.

	$\Delta t = 0.001,$ $\alpha_f = 0, \alpha_m = 0$	$\Delta t = 0.005$ $\alpha_f = 0, \alpha_m = 0$	$\Delta t = 0.005$ $\alpha_f = 0.4, \alpha_m = 0.2$
Frequency [Hz]	4.317	4.308	4.255
periodic $\Delta y(A)$ [ $10^{-3}m$ ]	$1.07 \pm 41.68$	$1.12 \pm 41.38$	$1.25 \pm 46.75$
$\Delta y_{max}$ [ $10^{-3}m$ ]	46.1	45.8	54.5
$t$ at $\Delta y_{max}$ [s]	4.08	4.31	4.31
periodic $\Delta x(A)$ [ $10^{-6}m$ ]	$50 \pm 36$	$48 \pm 38$	$65 \pm 65$

Table 5.5: Characteristic values of the three converging FSI3 simulations

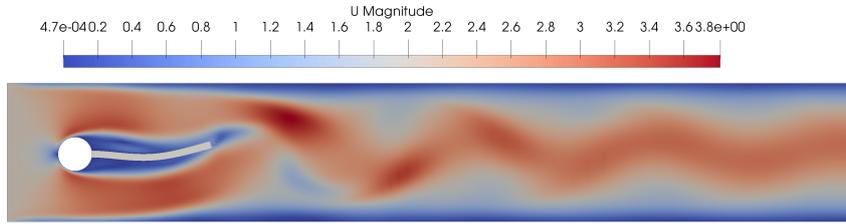


Figure 5.11: Exemplary velocity field of the FSI3 Case.

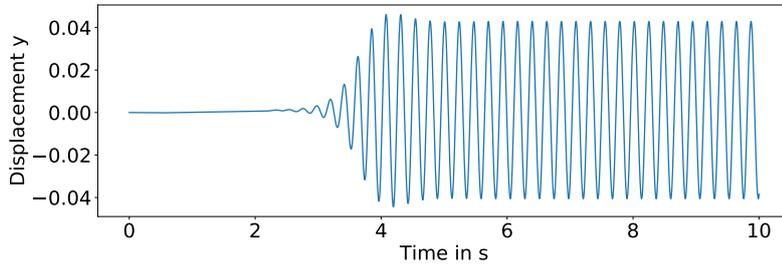


Figure 5.12: Tip displacement of point  $A$  in  $y$ -direction from 0 to 10 s for  $\Delta t = 0.001s$ ,  $\alpha_f = 0$  and  $\alpha_m = 0$ . For the other simulations the image looks very similar and is therefore not included here.

### Comparison to Reference Results

In Table 5.6, I compare characteristic values of the periodic oscillations of my FSI3 simulation with the results from [26] and two reference simulations with CalculiX instead of my structure solver. [26] has a timestep of  $\Delta t = 0.0005s$  and a parabolic

velocity profile on the inlet, whereas the other simulations use  $\Delta t = 0.001s$  and a constant velocity profile. Plus, the mesh in [26] is much finer with 253952 elements than the other simulations with only 6306 cells in the fluid domain. The setup with OpenFOAM, preCICE and CalculiX was validated in [23]. I used the same fluid case for the reference simulations as I used to couple my FEniCS structure solver. For CalculiX, I use the nonlinear case from the preCICE-tutorial on GitHub<sup>3</sup> and a linearized version. Figure 5.13 visually compares my simulation with the linear and nonlinear CalculiX simulations. The difference between my case and the linearized CalculiX is much smaller than between the two CalculiX simulations with less than 5% for frequency and y-displacement.

Quantity	FEniCS-OF	CCX-OF-lin	CCX-OF-nl	T & H
Frequency [Hz]	4.317	4.380	4.543	5.55
periodic $\Delta y(A)$ [ $10^{-3}m$ ]	$1.07 \pm 41.68$	$1.12 \pm 39.82$	$0.95 \pm 33.45$	$1.48 \pm 34.38$
periodic $\Delta x(A)$ [ $10^{-6}m$ ]	$50 \pm 36$	$50 \pm 25$	$-2075 \pm 2025$	$-2690 \pm 2530$

Table 5.6: Characteristic values of my FSI3 case compared to a linearized version of the FSI3 preCICE tutorial with CalculiX, the original tutorial and Turek & Hron [26].

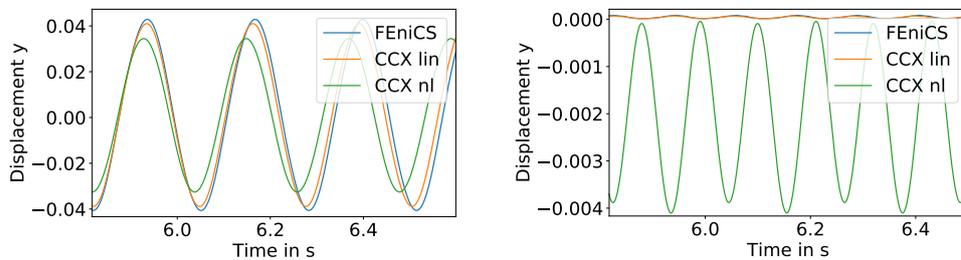


Figure 5.13: Displacement of the point A in  $m$  in the FSI3 test in y-direction (left) and x-direction (right). The results of the simulations using CalculiX are shifted such that the phases match on the left edge of the figures.

<sup>3</sup><https://github.com/precice/tutorials/tree/master/FSI/cylinderFlap/OpenFOAM-CalculiX>

## 6 Conclusions

In this thesis, I explored possibilities to simulate fluid structure interaction with FEniCS, preCICE and OpenFOAM. I developed a 2D structure solver in FEniCS and an adapter, such that the solver can be coupled with a pseudo-3D OpenFOAM simulation via preCICE. The setup was tested with two different testcases and for different parameters in the timestepping. The generalized  $\alpha$ -method becomes unstable in certain cases, when the forces are not evaluated at the correct time for small timesteps, as seen for the FSI3 case. The setups give qualitatively reasonable results for  $\alpha_m = \alpha_f = 0$ . In comparison with a validated linear structure solver, the difference is less than 5 % for frequency and displacement. This indicates that the preCICE-FEniCS adapter and the implemented structure solver work correctly. The CSM3 already shows the limitations of a linear elastic model: Large rotations cannot be represented correctly and deformations are overestimated. This is also the case for the FSI3 simulation, where the displacements in the y-direction are overestimated and the displacements in the x-direction are wrong, since they are the result of a rotation that cannot be displayed in a linear model.

The preCICE-FEniCS adapter was designed such that other researchers can use it to couple their own FEniCS solvers for FSI and is provided on GitHub<sup>1</sup>. The whole structure setup can be used to extend CFD simulations to FSI due to the modular nature of simulations with preCICE as coupling tool. The components are well separated, such that also other CFD solvers than OpenFOAM may be used.

### 6.1 Future Work

There are still aspects in the structure solver and the adapter that can be improved. The structure solver could be extended for hyperelastic materials that it can express large deformations and rotations better. preCICE developers work on higher order coupling schemes, such that the forces can be evaluated at arbitrary points in time. This

---

<sup>1</sup><https://github.com/precice/fenics-adapter>

would allow the FEniCS solver to actually read the forces at  $t_{n+1-\alpha_f}$  and the generalized  $\alpha$ -method could be correctly applied for various choices of  $\alpha_f$ .

The computational performance was sufficient for the testcases in this thesis with a maximum runtime of 10 hours for the FSI3 on a virtual machine. preCICE and OpenFOAM are ready for intra- and inter-solver parallelism. For more sophisticated simulations, also the FEniCS-part could be parallelized, such that the solvers run on multiple cores and at the same time instead of in an alternating fashion on a single core.

# List of Figures

2.1	Nearest-neighbor vs RBF-mapping . . . . .	9
3.1	General scheme of a coupled FSI simulation using preCICE. . . . .	10
4.1	Overview over the structure solver . . . . .	14
4.2	Runtimes of the simulation of an elastic flap . . . . .	16
5.1	Setup of the channel with an elastic flap. . . . .	19
5.2	Shift of the fluid domain near the Dirichlet boundary of the structure . . . . .	20
5.3	PointSources on a Dirichlet boundary . . . . .	21
5.4	Velocity magnitude and normal stresses at 5 s in the elastic flap case. . . . .	21
5.5	PointSource vs workaround: Displacements and differences . . . . .	22
5.6	Elastic flap: FEniCS vs CalculiX . . . . .	23
5.7	Setup of the FSI3 Benchmark Scenario. . . . .	24
5.8	Mesh of and around the flap for the FSI3 simulation. . . . .	25
5.9	CSM3: Displacements . . . . .	26
5.10	Unstable generalized $\alpha$ -method . . . . .	27
5.11	Exemplary velocity field of the FSI3 Case. . . . .	28
5.12	FSI3: Tip displacement . . . . .	28
5.13	FSI3: FEniCS vs CalculiX . . . . .	29

## List of Tables

5.1	Parameters of the elastic flap testcase. . . . .	19
5.2	Geometry of the FSI3 Simulation. . . . .	24
5.3	Material parameters in the FSI3 Simulation. . . . .	25
5.4	Comparison of my partitioned CSM3 with FEniCS and OpenFOAM to reference simulations. . . . .	26
5.5	Characteristic values of the three converging FSI3 simulations . . . . .	28
5.6	FSI3: Comparison to other simulations . . . . .	29

# Bibliography

- [1] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. “The FEniCS Project Version 1.5.” In: *Archive of Numerical Software* 3.100 (2015). DOI: 10.11588/ans.2015.100.20553.
- [2] J. Bleyer. “Numerical Tours of Computational Mechanics with FEniCS.” In: (2018), 25ff. DOI: 10.5281/zenodo.1287832.
- [3] F. J. Blom. “A monolithical fluid-structure interaction algorithm applied to the piston problem.” In: *Computer Methods in Applied Mechanics and Engineering* 167.3-4 (Dec. 1998), pp. 369–391. DOI: 10.1016/S0045-7825(98)00151-0.
- [4] H.-J. Bungartz, F. Lindner, B. Gatzhammer, M. Mehl, K. Scheufele, A. Shukaev, and B. Uekermann. “preCICE – A fully parallel library for multi-physics surface coupling.” In: *Computers and Fluids* 141 (2016), pp. 250–258. DOI: <https://doi.org/10.1016/j.compfluid.2016.04.003>.
- [5] P. Causin, J. F. Gerbeau, and F. Nobile. “Added-mass effect in the design of partitioned algorithms for fluid–structure problems.” In: *Computer Methods in Applied Mechanics and Engineering* 194.42 (2005), pp. 4506–4527. DOI: <https://doi.org/10.1016/j.cma.2004.12.005>.
- [6] L. Cheung Yau. *Conjugate Heat Transfer with the Multiphysics Coupling Library preCICE, Master’s Thesis*. 2016. URL: [https://www5.in.tum.de/pub/Cheung2016\\_Thesis.pdf](https://www5.in.tum.de/pub/Cheung2016_Thesis.pdf).
- [7] G. Chourdakis. *A general OpenFOAM adapter for the coupling library preCICE, Master’s Thesis*. 2017. URL: [https://www5.in.tum.de/pub/Chourdakis2017\\_Thesis.pdf](https://www5.in.tum.de/pub/Chourdakis2017_Thesis.pdf).

- [8] J. Chung and G. M. Hulbert. "A Time Integration Algorithm for Structural Dynamics With Improved Numerical Dissipation: The Generalized- $\alpha$  Method." In: *Journal of Applied Mechanics* 60.2 (June 1993), pp. 371–375. DOI: <http://dx.doi.org/10.1115/1.2900803>.
- [9] J. Degroote, R. Haelterman, S. Annerel, P. Bruggeman, and J. Vierendeels. "Performance of partitioned procedures in fluid-structure interaction." In: *Computers and Structures* 88.7-8 (Apr. 2010), pp. 446–457. ISSN: 00457949. DOI: 10.1016/j.compstruc.2009.12.006.
- [10] B. S. M. Ebna Hai. *Numerical Simulation of Fluid Structure Interaction (FSI) Effect on an Aircraft Wing*. Sept. 2013.
- [11] C. Farhat, K. G. van der Zee, and P. Geuzaine. "Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity." In: *Computer Methods in Applied Mechanics and Engineering* 195.17-18 (Mar. 2006), pp. 1973–2001. DOI: 10.1016/J.CMA.2004.11.031.
- [12] B. Gatzhammer. "Efficient and Flexible Partitioned Simulation of Fluid-Structure Interactions." PhD thesis. München: Technische Universität München, 2014. URL: [https://www5.in.tum.de/pub/Gatzhammer2014\\_preCICE.pdf](https://www5.in.tum.de/pub/Gatzhammer2014_preCICE.pdf).
- [13] M. Heil. "An efficient solver for the fully coupled solution of large-displacement fluid–structure interaction problems." In: *Computer Methods in Applied Mechanics and Engineering* 193.1-2 (Jan. 2004), pp. 1–23. DOI: 10.1016/J.CMA.2003.09.006.
- [14] M. Ismail, V. Gravemeier, A. Comerford, and W. A. Wall. "A stable approach for coupling multidimensional cardiovascular and pulmonary networks based on a novel pressure-flowrate or pressure-only Neumann boundary condition formulation." en. In: *International Journal for Numerical Methods in Biomedical Engineering* 30.4 (2014), pp. 447–469. DOI: 10.1002/cnm.2611.
- [15] D. E. Keyes, L. C. McInnes, C. Woodward, W. Gropp, E. Myra, M. Pernice, ..., and B. Wohlmuth. "Multiphysics simulations: Challenges and opportunities." In: *The International Journal of High Performance Computing Applications* 27.1 (Feb. 2013), pp. 4–83. DOI: 10.1177/1094342012468181.
- [16] W. M. Lai, D. Rubin, and E. Krempl. "CHAPTER 3 - Kinematics of a Continuum." In: *Introduction to Continuum Mechanics (Fourth Edition)*. Ed. by W. M. Lai, D. Rubin, and E. Krempl. Fourth Edition. Boston: Butterworth-Heinemann, 2010,

- pp. 69–153. ISBN: 978-0-7506-8560-3. DOI: <https://doi.org/10.1016/B978-0-7506-8560-3.00003-7>.
- [17] H. P. Langtangen and A. Logg. *Solving PDEs in Python*. Springer, 2017. ISBN: 978-3-319-52461-0. DOI: 10.1007/978-3-319-52462-7.
- [18] Y. Lee and C. Basaran. “A multiscale modeling technique for bridging molecular dynamics with finite element method.” In: *Journal of Computational Physics* 253 (Nov. 2013), pp. 64–85. DOI: 10.1016/J.JCP.2013.06.039.
- [19] F. Lindner, M. Mehl, and B. Uekermann. “Radial Basis Function Interpolation for Black-Box Multi-Physics Simulations.” In: *VII International Conference on Computational Methods for Coupled Problems in Science and Engineering*. May 2017.
- [20] H. G. Matthies and J. Steindorf. “Partitioned strong coupling algorithms for fluid–structure interaction.” In: *Computers & Structures* 81.8-11 (May 2003), pp. 805–812. ISSN: 0045-7949. DOI: 10.1016/S0045-7949(02)00409-1. URL: <https://www.sciencedirect.com/science/article/pii/S0045794902004091?via%3Dihub>.
- [21] N. M. Newmark. “A Method of Computation for Structural Dynamics.” In: *ASCE Journal of the Engineering Mechanics Division* 85 (1959), pp. 67–94.
- [22] preCICE Website. *preCICE :: Home, visited 2019-09-24*. URL: <https://www.precice.org>.
- [23] D. Risseeuw. *Fluid Structure Interaction Modelling of Flapping Wings Development and Validation of a General Open-source Fluid Structure Interaction Method with Analysis of Flexible Flapping Wing Aerodynamics*, MSc Thesis. 2019. URL: <http://resolver.tudelft.nl/uuid:70beddde-e870-4c62-9a2f-8758b4e49123>.
- [24] A. Rusch. *Extending SU 2 to fluid-structure interaction via preCICE*, Bachelor’s Thesis. 2016. URL: [https://www5.in.tum.de/pub/Rusch2016\\_BA.pdf](https://www5.in.tum.de/pub/Rusch2016_BA.pdf).
- [25] B. R uth. “Using FEniCS and OpenFOAM for the simulation of conjugate heat transfer in a partitioned fashion.” en. In: *90th GAMM Annual Meeting*. Vienna, Feb. 2019.
- [26] S. Turek and J. Hron. “Proposal for Numerical Benchmarking of Fluid-Structure Interaction between an Elastic Object and Laminar Incompressible Flow.” In: *Fluid-Structure Interaction*. Springer Berlin Heidelberg, June 2007, pp. 371–385. DOI: 10.1007/3-540-34596-5{\\_}15.

- [27] B. Uekermann. "Partitioned Fluid-Structure Interaction on Massively Parallel Systems." PhD thesis. Institut für Informatik, Technische Universität München, Oct. 2016. URL: <https://mediatum.ub.tum.de/doc/1320661/document.pdf>.
- [28] B. Uekermann, H.-J. Bungartz, L. Yau, G. Chourdakis, and A. Rusch. "Official preCICE Adapters for Standard Open-Source Solvers." In: 2017. URL: [https://www.researchgate.net/publication/321289778\\_Official\\_preCICE\\_Adapters\\_for\\_Standard\\_Open-Source\\_Solvers](https://www.researchgate.net/publication/321289778_Official_preCICE_Adapters_for_Standard_Open-Source_Solvers).
- [29] D. W. Olson, S. Wolf, and J. M. Hook. *The Tacoma Narrows Bridge collapse*. Vol. 68. Nov. 2015, pp. 64–65. DOI: 10.1063/PT.3.2991.
- [30] L. Wang, R. Quant, and A. Kolios. "Fluid structure interaction modelling of horizontal-axis wind turbine blades based on CFD and FEA." In: *Journal of Wind Engineering and Industrial Aerodynamics* 158 (2016), pp. 11–25.
- [31] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. "A tensorial approach to computational continuum mechanics using object-oriented techniques." In: *Computers in Physics* 12.6 (Nov. 1998), pp. 620–631. DOI: 10.1063/1.168744.
- [32] M. S. Zakaria, H. Abdullah, and K. A. Ahmad. "Fluid structure interaction simulation of large deformation and added-mass effect using OpenFOAM." In: *5th Mechanical Engineering Research Day (MERD'18), 3 May 2018, Kampus Teknologi UTeM, Melaka*. 2018, pp. 79–80. URL: <http://psasir.upm.edu.my/id/eprint/65093/>.