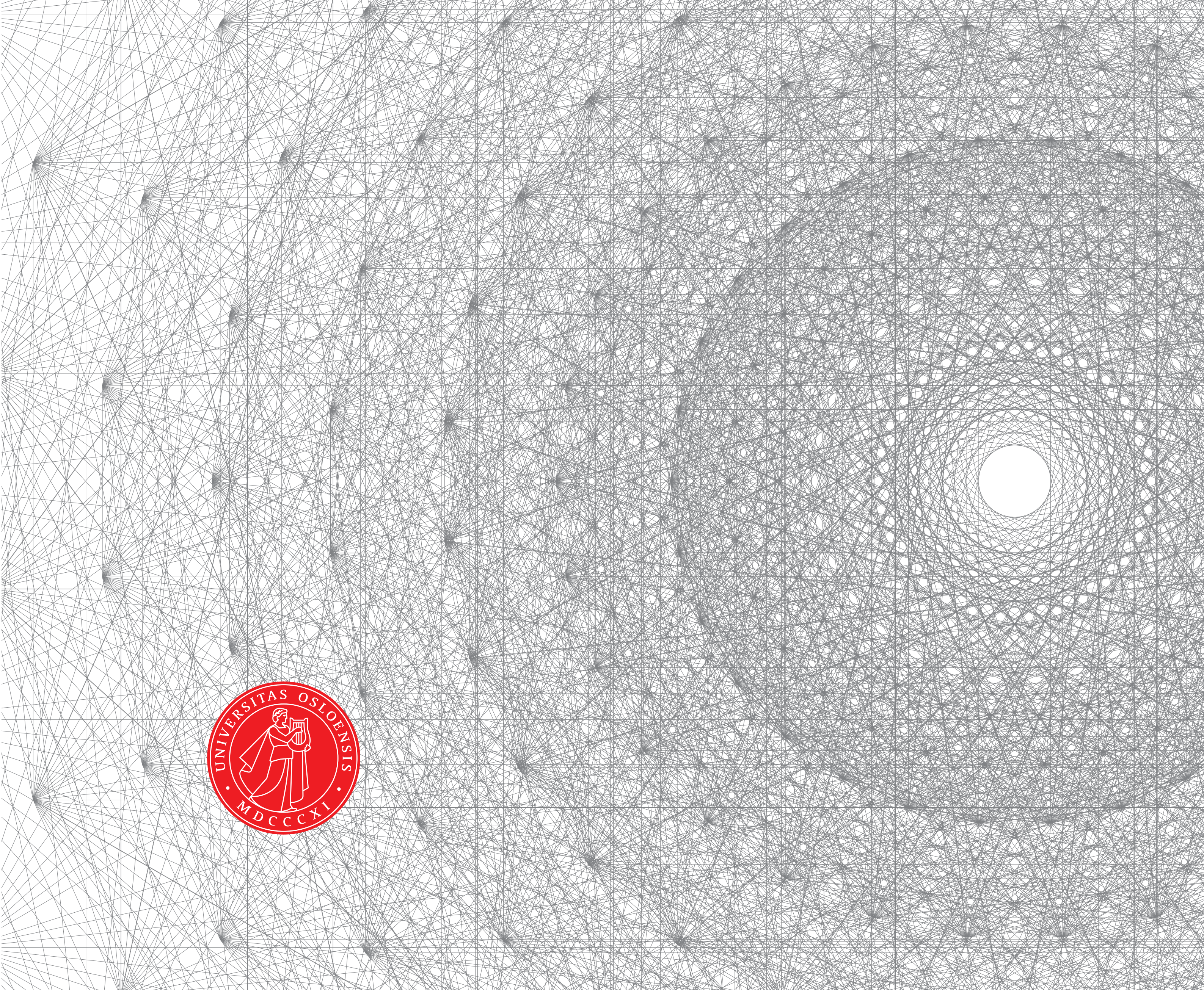
UiO : Department of Mathematics
University of Oslo

Verification and Validation of a Monolithic Fluid-Structure Interaction Solver in FEniCS

A comparison of mesh lifting operators

Andreas Slyngstad
Master's Thesis, Autumn 2017



This master's thesis is submitted under the master's programme *Computational Science and Engineering*, with programme option *Mechanics*, at the Department of Mathematics, University of Oslo. The scope of the thesis is 60 credits.

The front page depicts a section of the root system of the exceptional Lie group E_8, projected into the plane. Lie groups were invented by the Norwegian mathematician Sophus Lie (1842–1899) to express symmetries in differential equations and today they play a central role in various parts of mathematics.

Acknowledgements

First off, I would like to thank my supervisors Dr. Kristian Valen-Sendstad and Aslak Wigdahl Bergersen. Kristian, your guidance and dedication for mechanics have kept me highly encouraged throughout this thesis. Aslak, your insight and problem solving methodology have been invaluable to overcome the many problems I encountered in my numerical simulations. Thank you both for spending wast amounts of time, supervising me during the last two years. I would also like to thank Professor Mikael Mortensen and Miroslav Kuchta at the Department of Mathematics at the University of Oslo. Your open door policy and technical understanding of the FEniCS project have been important for completing this thesis.

I would like to express my deepest gratitude to my parents, Sveinung and Åse, for your unlimited support and daily phone calls. Finally I would like to thank my beloved partner Charlotte and daughter Linde Olivia. Your love and understanding have kept me highly motivated and well fed. This thesis is dedicated to you.

Contents

Fluid-structure interaction; what and why?

The interaction between fluid and solids can be observed all around us in nature and has shown crucial in engineering. Examples in nature include swimming fish, flying birds, or trees bending in the wind. Man has learned from nature and has traditionally relied upon laboratory experiments to design windmills, aircrafts, and bridges. The importance of understanding fluid-structure (or solids) interaction (FSI) cannot be overstated, as the lack of such has demonstrated to be disastrous in the design of everything from bridges to airplanes. Let alone to emphasize our incapability to replicate the performance of nature; we're far away from designing a drone capable of flying like a hummingbird. One can study FSI experimentally, however laboratory experiments are inherently noisy, expensive, and results can be difficult to reproduce. A much cheaper and indeed smarter approach to studying FSI is using computers, or more specifically numerical simulations to gain fundamental insight to the interaction between fluids and solids. The latter has on the other hand shown to be difficult to realize, for a number of reasons related to both mathematical and computational reasons. Therefore, the goal of this thesis is to develop an open-source framework using standard techniques for solving FSI problems that can be used as a point of reference for future benchmarking of FEniCS-based FSI solvers.

The main goal of this thesis is to create a verified and validated monolithic fluid-structure interaction solver in FEniCS, which can handle large deformations. To achieve this, I have defined four subgoals:

- Formulate a weak variation for a monolithic arbitrary Lagrangian Eulerian fluid-structure interaction problem.
- Construct a finite element solver for the fluid-structure interaction problem.
- Verify and validate a finite element solver for the fluid-structure interaction problem.
- Compare the impact of discretization and mesh lifting operators on the final solution.
- Improve computational efficiency of the implementation.

Each of the following subgoals will be addressed in separate chapters organized as follows: In chapter 1, balance of linear momentum for both solids and fluids are first introduced together with conservation of mass. In chapter 2, the Eulerian, Lagrangian, and the arbitrary Lagrangian-Eulerian (ALE) frames of reference are briefly introduced to express the governing equations, before the equations describing FSI are derived. Chapter 3 investigates the numerical implementation by verification, using the most rigorous convergence tests, before validation is performed against state-of-the-art benchmarks. Finally, computational speed-up is addressed in chapter 4, together with long-term numerical stability of the coupled problem, and methods to overcome these challenges.

Governing equations of solids and fluids

Fluid-structure interaction (FSI) combines two classical fields of mechanics, computational fluid mechanics (CFD), and computational structural mechanics (CSM). To complete FSI there is also the coupling, or interaction between these two. A separate understanding of the fluid and structure is therefore necessary to understand the full problem. This chapter presents the governing equations of the individual fluid and structure problem, together with auxiliary kinematic, dynamic, and material relations.

1.1 Continuum Mechanics

In our effort to understand and describe physical phenomenon in nature, we describe our observations and theories by creating mathematical models. The mathematical models makes scientist and engineers not only able to understand physical phenomena, but also predict them. All matter is built up by a sequence of atoms, meaning on a microscopic level, an observer will locate discontinuities and space within the material. Evaluating each atom, or *material point*, is not impossible from a mathematical point of view. However, for mathematical modeling and applications, the evaluation of each material point remains unpractical. In *continuum mechanics*, first formulated by Augustin-Louis Cauchy [18], the microscopic structure of materials are ignored, assuming the material of interest is *continuously distributed* in space, referred to as a continuum.

In context of this thesis I define a *continuum* as a continuous body $V(t) \subset \mathbb{R}^d$ $d \in (1, 2, 3)$, continuously distributed throughout its own extension. The continuum is assumed to be infinitely divisible, meaning one can divide some region of the continuum a indefinitely number of times. A continuum is also assumed to be locally homogeneous, meaning if a continuum is subdivided into infinitesimal regions, they would all have the same properties such as mass density. These two properties forms the baseline for deriving conservation laws and constitute equations, which are essential for formulating mathematical models for both CFD and CSM. However, a continuum remains a mathematical idealization, and may not be a reasonable model for certain applications. In general, continuum mechanics have proven to be applicable provided that $\frac{\delta}{l} << 1$ where δ is a characteristic length scale of the material micro-structure, and l is a length scale of the problem of interest [15].

1.2 The Lagrangian and Eulerian description of motion

In continuum mechanics, one makes use of two classical description of motion, the *Lagrangian* and *Eulerian* description. Both concepts are related to an observers view of motion, visually explained by the concepts of *material* and *spatial* points. A material points represents a particle within the material, moving with the material as it move and deform. A spatial point, refers to some reference at which the path of the material points are measured from.

Lagrangian

In the Lagrangian description of motion, the material and spatial points coincide, meaning the reference point of which motion is measured, follows the material as it diverts from its initial position. The initial position of all material points in a continuum extend a region, called the *reference configuration* $\hat{V}$. From now on, all identities in the *reference configuration* will be denoted with the notation "$\wedge$". If a continuum deviates from its reference configuration, a material point $\hat{\mathbf{x}}(x, y, z, t)$ may no longer be at its initial position, but moved to a new position $\mathbf{x}(x, y, z, t)$ at time t. The new positions of all material points extend a new region, called the *current configuration* $V(t)$.

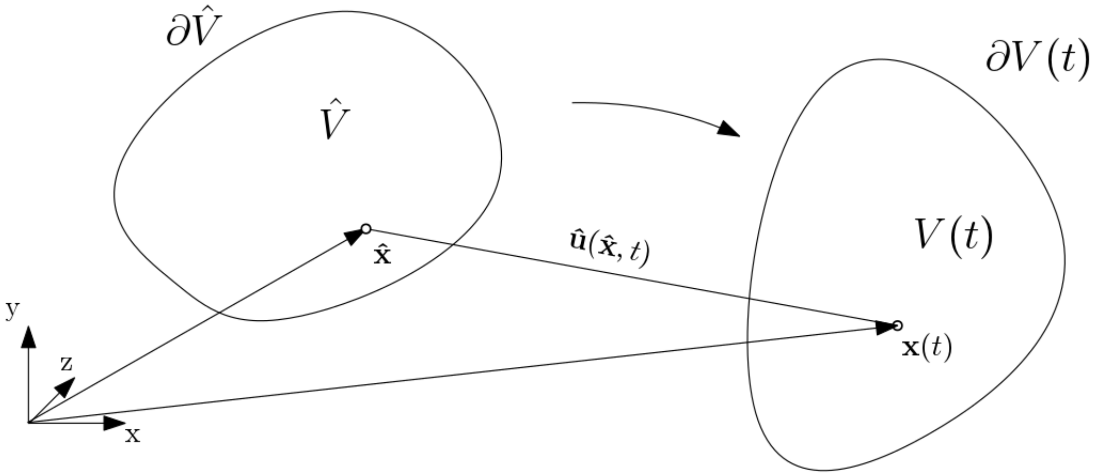


Figure 1.1: A visual representation of the Lagrangian description of motion.

To measure the displacement of a material point $\mathbf{x} \in V(t)$ for time t, from its initial point $\hat{\mathbf{x}} \in \hat{V}$, one defines a *deformation vector field*

$$\hat{\mathbf{u}}(\hat{\mathrm{x}}, t) = x(\hat{\mathbf{x}}, t) - \hat{\mathbf{x}} = \hat{\mathrm{T}}(\hat{\mathbf{x}}, t) \tag{1.1}$$

Mathematically, deformation is a 1:1 mapping $\hat{\mathrm{T}}(\hat{\mathbf{x}}, t)$, transforming material points from the *reference configuration* $\hat{V}$, to the *current configuration* $V(t)$. Visually, the deformation resembles the shape of continuum for some time t. To describe the continuums motion, one defines the *velocity vector field* given by the time derivative of the deformation field,

$$\hat{\mathbf{v}}(\hat{\mathrm{x}}, t) = d_t x(\hat{\mathbf{x}}, t) = d_t \hat{\mathbf{u}}(\hat{\mathbf{x}}, t) = \frac{\partial \hat{\mathrm{T}}(\hat{\mathrm{x}}, t)}{\partial t} \tag{1.2}$$

The Lagrangian description of motion is the natural choice when tracking particles and surfaces are of main interest. Therefore, it is mainly used within structure mechanics.

Eulerian

In the Eulerian description of motion, the material and spatial points are separated. Instead of tracking material points $\hat{x}(t) \in V(t)$, the attention brought to a fixed view-point V. In contrary with the Lagrangian description, the *current configuration* is chosen as the *reference configuration*, not the initial position of all material particles. The location or velocity of any material particle is not of interest, but rather the properties of a material particle happening to be at $\mathbf{x}(t)$ for some t.

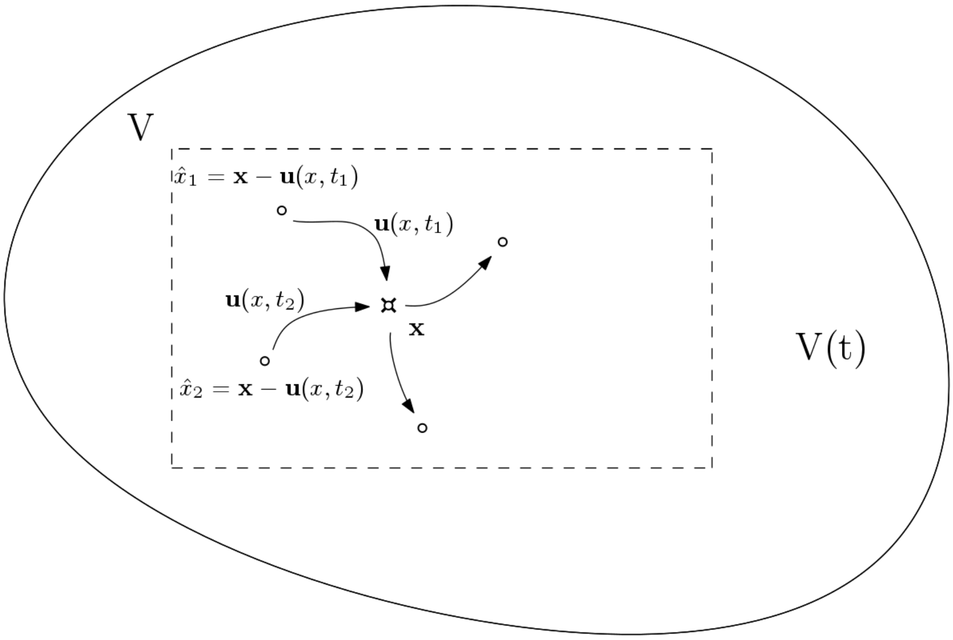


Figure 1.2: A visual representation of the Eulerian description of motion. For a view-point V fixed in time, a spatial coordinate $\mathbf{x}$ measures properties of a material particle $\hat{x}$ from the moving continuum $V(t)$.

We can describe the particles occupying the *current configuration* $V(t)$ for some time $t \geq t_0$

$$\mathbf{x} = \hat{\mathbf{x}} + \hat{\mathbf{u}}(\hat{\mathrm{x}}, t)$$

Since our domain is fixed we can define the deformation for a particle occupying position $x = x(\hat{\mathrm{x}}, t)$ as

$$\mathbf{u}(x, t) = \hat{\mathbf{u}}(\hat{\mathbf{x}}, t) = x - \hat{\mathbf{x}}$$

and its velocity

$$\mathbf{v}(\hat{\mathbf{x}}, t) = \partial_t \mathbf{u}(\hat{\mathbf{x}}, t) = \partial_t \hat{\mathbf{u}}(\hat{\mathbf{x}}, t) = \hat{\mathbf{v}}(\hat{\mathbf{x}}, t)$$

The Eulerian description falls naturally for describing fluid flow, due to local kinematic properties are of higher interest rather than the shape of fluid domain. Using a Lagrangian description for fluid flow would also be tedious, due to the large number of material particles appearing for longer simulations of fluid flow. A comparison of the two previous mentioned description is shown of In Figure 1.3.

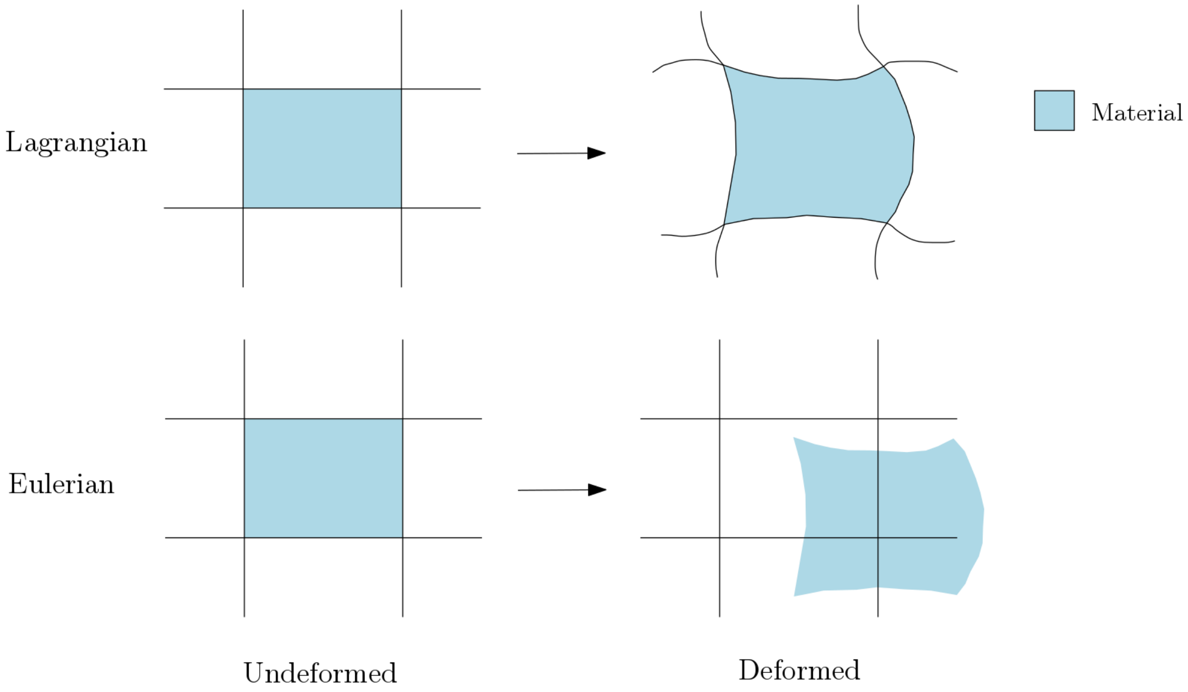


Figure 1.3: Comparison of the Lagrangian and Eulerian description of motion.

1.3 The Solid equations

The solid governing equations is given by,

Equation 1.3.1. *Solid equations*

$$\rho_s \frac{\partial \hat{\mathbf{v}}_s}{\partial t} = \nabla \cdot (\hat{J} \sigma_s \hat{\mathbf{F}}^{-T}) + \rho_s \mathbf{f}_s \quad \text{in } \hat{\Omega}_s \tag{1.3}$$

$$\frac{\partial \hat{\mathbf{v}}_s}{\partial t} = \hat{\mathbf{u}}_\mathbf{s} \quad \text{in } \hat{\Omega}_s \tag{1.4}$$

defined in a Lagrangian coordinate system, with respect to an initial reference configuration $\hat{\Omega}_s$. The structure configuration is given by the displacement $\hat{\mathbf{u}}_s$, with the relation $\frac{\partial \hat{\mathbf{v}}}{\partial t} = \hat{\mathbf{u}}_s$ to the solid velocity. The density of the structure is given by ρ_s, and $\hat{\mathbf{f}}_s$ express any exterior body forces acting. Finally, $\hat{\mathbf{F}} = I + \nabla \hat{\mathbf{u}}_s$ is the deformation gradient, and $\hat{J}$ is the determinant of $\hat{\mathbf{F}}$ [1].

Material models express the dependency between strain tensors and stress. The validity of material models is often limited by their ability to handle deformation and strain to some extent, before it breaks down or yields nonphysical observations of the material. In this thesis, a linear relation between stress and strain is assumed, where the elasticity of the material is expressed by the *Poisson ratio* ν_s, *Young modulus* E, or Lamés coefficients λ_s and μ_s. Their relation is given by,

$$\begin{aligned} E_y &= \frac{\mu_s(\lambda_s + 2\mu_s)}{(\lambda_s + \mu_s)} & \nu_s &= \frac{\lambda_s}{2(\lambda_s + \mu_s)} \\ \lambda_s &= \frac{\nu E_y}{(1+\nu_s)(1-2\nu_s)} & \mu_s &= \frac{E_y}{2(1+\nu_s)} \end{aligned}$$

Hooke's law is a linear relation applicable for small-scale deformations,

Definition 1.1. Let $\hat{u}$ be a differential deformation field in the *reference* configuration, I be the Identity matrix, and the gradient $\hat{\nabla} = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$. *Hooke's law* is then given by,

$$\begin{aligned} \sigma_s &= \frac{1}{\hat{J}} \hat{\mathbf{F}} (\lambda_s (Tr(\epsilon) I + 2\mu\epsilon) \hat{\mathbf{F}} \\ \hat{\mathbf{S}}_s &= \lambda_s (Tr(\epsilon) I + 2\mu\epsilon \\ \epsilon &= \frac{1}{2} (\hat{\nabla} \hat{\mathbf{u}} + (\hat{\nabla} \hat{\mathbf{u}})^T) \end{aligned}$$

However, as Hooke's law is limited to a small-deformation, it is not valid for large deformations encountered in this thesis. A valid model for larger deformations is the hyper-elastic *St. Vernant-Kirchhoff model*(STVK), extending Hooke's law into a non-linear regime.

Definition 1.2. Let $\hat{u}$ be a differential deformation field in the *reference* configuration, I be the Identity matrix and the gradient $\hat{\nabla} = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$. The *St.*

---

[1] See Appendix A for further detail

Vernant-Kirchhoff model is then given by the relation,

$$\begin{aligned}
&\sigma_s = \frac{1}{\hat{J}}\hat{\mathbf{F}}(\lambda_s(Tr(\hat{\mathbf{E}})I + 2\mu\hat{\mathbf{E}})\hat{\mathbf{F}}^{-T} \\
&\hat{\mathbf{S}}_s = \lambda_s(Tr(\hat{\mathbf{E}})I + 2\mu\hat{\mathbf{E}} \\
&\hat{\mathbf{E}} = \frac{1}{2}(\hat{\mathbf{C}} - I) \quad \hat{\mathbf{C}} = \hat{\mathbf{F}}\hat{\mathbf{F}}^{-T}
\end{aligned}$$

where $\hat{\mathbf{C}}$ is the right Cauchy-Green strain tensor and $\hat{\mathbf{E}}$ is the Green Lagrangian strain tensor.

Though STVK can handle large deformations, it is not valid for large strain [22]. However since the strain considered in this thesis are small, it will remain our primary choice of strain-stress relation. In addition, initial condition and boundary condition is supplemented for the problem to be well posed. The first type of of boundary conditions are Dirichlet boundary conditions,

$$\mathbf{v}_s = \mathbf{v}_s^D \quad \text{on } \Gamma_s^D \subset \partial\Omega_s \tag{1.5}$$

$$\mathbf{d}_s = \mathbf{d}_s^D \quad \text{on } \Gamma_s^D \subset \partial\Omega_s \tag{1.6}$$

$$\tag{1.7}$$

The second type of boundary condition are Neumann boundary conditions

$$\sigma_s \cdot \mathbf{n} = \mathbf{g} \quad \text{on } \Gamma_s^N \subset \partial\Omega_s \tag{1.8}$$

1.4 The Fluid equations

The fluid is assumed to be express by the incompressible Navier-Stokes equations,

Equation 1.4.1. *Navier-Stokes equation*

$$\rho \frac{\partial \mathbf{v}_f}{\partial t} + \rho \mathbf{v}_f \cdot \nabla \mathbf{v}_f = \nabla \cdot \sigma + \rho \mathbf{f}_f \quad \text{in } \Omega_f \tag{1.9}$$

$$\nabla \cdot \mathbf{v}_f = 0 \quad \text{in } \Omega_f \tag{1.10}$$

defined in an Eulerian description of motion. The fluid density as ρ_f and fluid viscosity ν_f are assumed to be constant in time, and $\mathbf{f}_s$ represents any body force. The fluid is assumed Newtonian, where *Cauchy stress sensor* follows Hooke's law

$$\sigma = -p_f I + \mu_f(\nabla \mathbf{v}_f + (\nabla \mathbf{v}_f)^T$$

As for the solid equations, boundary conditions are supplemented considering Dirichlet boundary conditions,

$$\mathbf{v}_f = \mathbf{v}_f^D \quad \text{on } \Gamma_v^D \subset \partial\Omega_f \tag{1.11}$$

$$p_f = p_f^D \quad \text{on } \Gamma_p^D \subset \partial\Omega_f \tag{1.12}$$

The second type of boundary condition are Neumann boundary conditions

$$\sigma_f \cdot \mathbf{n} = \mathbf{g} \quad \text{on } \Gamma_f^N \subset \partial\Omega_f \tag{1.13}$$

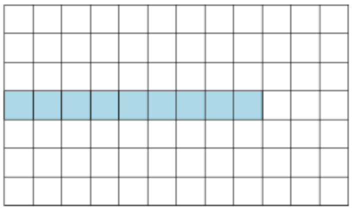
Fluid Structure Interaction

The multi-disciplinary nature of computational fluid-structure interaction, involves addressing issues regarding computational fluid dynamics and computational structure dynamics. In general, CFD and CSM are individually well-studied in terms of numerical solution strategies. FSI adds another layer of complexity to the solution process by the *coupling* of the fluid and solid equations, and the tracking of *interface* separating the fluid and solid domains. The coupling pose two new conditions at the interface absent from the original fluid and solid conditions, which is *continuity of velocity* and *continuity of stress* at the interface.

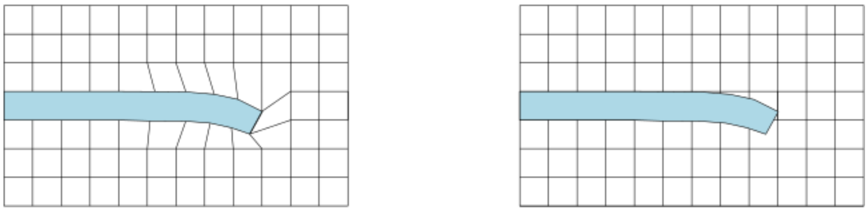
$$\mathbf{v}_f = \mathbf{v}_s \tag{2.1}$$

$$\sigma_f \cdot \mathbf{n} = \sigma_s \cdot \mathbf{n} \tag{2.2}$$

The tracking of the interface is a issue, due to the different description of motion used in the fluid and solid problem. If the natural coordinate system are used for the fluid problem and solid problem, namely the Eulerian and Lagrangian description of motion, the domains doesn't match and the interface. Tracking the interface is also essential for fulfilling the interface boundary conditions. As such only one of the domains can be described in its natural coordinate system, while the other domain needs to be defined in some transformed coordinate system. Fluid-structure interaction problems are formally divided into the *monolithic* and *partitioned* frameworks. In the monolithic framework, the fluid and solid equations together with interface conditions are solved simultaneously. The monolithic approach is *strongly coupled*, meaning the *kinematic* (1.1) and *dynamic*(1.2) interface conditions are met with high accuracy. However, the complexity of solving all the equations simultaneously and the strong coupling contributes to a stronger nonlinear behavior of the whole system [42]. The complexity also makes monolithic implementations *ad hoc* and less modular, and the nonlinearity makes solution time slow. In the *partitioned* framework one solves the equations of fluid and structure subsequently. Solving the fluid and solid problems individually is beneficial, in terms of the wide range of optimized solvers and solution strategies developed for each sub-problem. In fact, solving the fluid and solid separately was used in the initial efforts in FSI, due to existing solvers for one or both problems [10]. Therefore, computational efficiency and code reuse is one of the main reasons for choosing the partitioned approach. A major drawback is the methods ability to enforce the *kinematic* (1.1) and *dynamic*(1.2) conditions at each timestep. Therefore partitioned solution strategies are defined as *weakly* coupled. However, by sub-iterations between each sub-problem at each timestep, (1.1) and (1.2) can be enforced with high accuracy, at the cost of increased computational time. Regardless of framework, FSI has to cope with a numerical artifact called the "added-mass effect" [5], [4], [8]. The term is not to be confused with added mass found in fluid mechanics, were virtual mass is added to a system due to an accelerating or de-accelerating body moving through a surrounding fluid [20]. Instead, the term is used to describe the numerical instabilities occurring for weakly coupled schemes, in conjunction with *in compressible fluids* and slender structures [8], or where the density of the incompressible fluid is close to the structure. For partitioned solvers, sub-iterations are needed when the "added-mass effect" is strong, but for incompressible flow the restrictions can lead to unconditional instabilities [10]. The *strong coupled* monolithic schemes have proven overcome "added-mass effect" preserving energy balance, at the prize of a highly non-linear system to be solved at each time step [5]. Capturing the interface is matter of its own, regardless of the the monolithic and partitioned frameworks. The scope of interface methods are divided into *interface-tracking* and *interface-capturing* methods, visualized in figure 2.1.



Initial configuration



Interface-capturing

Interface-tracking

Figure 2.1: Comparison of interface-tracking and interface-capturing for an elastic beam undergoing deformation

In the *Interface-tracking* method, the mesh moves to accommodate for the movement of the structure as it deforms the spatial domain occupied by the fluid. As such, the mesh itself "tracks" the fluid-structure interface as the domain undergoes deformation. Interface-capturing yields better control of mesh resolution near the interface, which in turn yields better control of this critical area in terms of enforcing the interface conditions. However, moving the mesh-nodes pose potential problems for mesh-entanglements, restricting the possible extent of deformations. In *interface-capturing* methods one distinguish the fluid and solid domains by some phase variable over a fixed mesh, not resolved by the mesh itself. This approach is in general not limited in terms of deformations, but suffers from reduced accuracy at the interface. Among the multiple approaches within FSI, the arbitrary Lagrangian-Eulerian method is chosen for this thesis.

2.1 Arbitrary Lagrangian Eulerian formulation

The *arbitrary Lagrangian-Eulerian* formulation is the most popular approach within *Interface-tracking* [24, 9]. In this approach the structure is given in its natural *Lagrangian coordinate system*, while the fluid problem is formulated in an artificial coordinate system similar to the *Lagrangian coordinate system*, by an artificial fluid domain map from the undeformed *reference configuration* $\hat{\mathbf{T}}_f(t) : \hat{V}_f(t) \rightarrow V_f(t)$. The methods consistency is to a large extent dependent on the regularity of the artificial fluid domain map. Loss of regularity can occur for certain domain motions, were the structure makes contact with domain boundaries or self-contact with other structure parts [23]. Since no natural displacement occur in the fluid domain, the transformation $\hat{\mathbf{T}}_f(t)$ has no directly physical meaning [24, 3]. Therefore, the construction of the transformation $\hat{\mathbf{T}}_f(t)$ is a purely numerical exercise.

2.1.1 ALE formulation of the fluid problem

The original fluid problem, defined by the incompressible Navier-Stokes equations (Equation 1.5.1). are defined in an *Eulerian description of motion* $V_f(t)$. By changing the computational domain to an undeformed *reference configuration* $V_f(t) \rightarrow \hat{V}_f(t)$, the original problem no longer comply with the change of coordinate system. Therefore, the original Navier-Stokes equations needs to be transformed onto the *reference configuration* $\hat{V}_f$. Introducing the basic properties needed for mapping between the sub-system $\hat{V}_f(t)$ and $V_f(t)$, we will present the ALE time and space derivative transformations found in [25], with help of a new arbitrary fixed reference system $\hat{\mathrm{W}}$. Let $\hat{\mathrm{T}}_w : \hat{\mathrm{W}} \rightarrow V(t)$ be an invertible mapping, with the scalar $\hat{\mathrm{f}}(\hat{\mathrm{x}}_W, t) = f(x, t)$ and vector $\hat{\mathbf{w}}(\hat{\mathrm{x}}_W, t) = \mathbf{w}(x, t)$ counterparts. Further let the deformation gradient $\hat{F}_w$ and its determinant $\hat{\mathrm{J}}_w$, be defined in accordance wit definition 1.1 and 1.2 in Chapter 1. Then the following relations between temporal and spatial derivatives apply, between the two domains $\hat{W}(t)$ and $V(t)$,

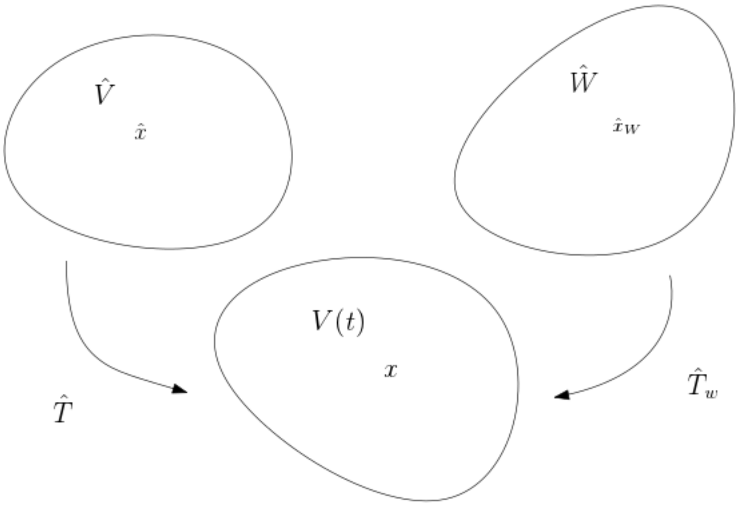


Figure 2.2: CFD-3, flow visualization of velocity time t = 9s

Lemma 2.1. Local change of volume
Let V(t) be the reference configuration $V(t) \to \mathbb{R}^d$, and $\hat{W} \to \mathbb{R}^d$ be the arbitrary reference configuration. By the determinant of the deformation gradient $\hat{J}_w$, the the following relations holds,

$$|V(t)| = \int_{\hat{W}} \hat{J}_w d\hat{x} \tag{2.3}$$

Lemma 2.2. Transformation of scalar spatial derivatives
Let f be a scalar function such that $f : V(t) \to \mathbb{R}$, and ∇f be its gradient. Then its counterpart $\nabla \hat{f}$, by the scalar function $\hat{f}\hat{W} \to \mathbb{R}$ is given by the relation.

$$\nabla f = \hat{F}_W^{-T} \hat{\nabla} \hat{\mathrm{f}} \tag{2.4}$$

Lemma 2.3. *Let $\boldsymbol{w}$ be a vector field such that $\mathbf{w} : V(t) \to \mathbb{R}^d$, and $\nabla \mathbf{w}$ be its gradient. Then its counterpart $\hat{\nabla}\hat{\mathbf{w}}$, by the vector field $\hat{\mathbf{w}} : \hat{W} \to \mathbb{R}^d$ is given by the relation.*

$$\nabla \mathbf{w} = \hat{\nabla} \hat{\mathbf{w}} \hat{F}_W^{-1} \tag{2.5}$$

Lemma 2.4. Transformation of scalar temporal derivatives
Let f be a scalar function such that $f : V(t) \to \mathbb{R}$, and $\frac{\partial f}{\partial t}$ be its time derivative. *Then its counterpart $\frac{\partial \hat{f}}{\partial t}$, by the scalar function $\hat{f}\hat{W} \to \mathbb{R}$ is given by the relation,*

$$\frac{\partial f}{\partial t} = \frac{\partial \hat{\mathrm{f}}}{\partial t} - (\hat{F}_W^{-1} \frac{\partial \hat{\mathrm{T}}_W}{\partial t} \cdot \hat{\nabla}) \hat{\mathrm{f}} \tag{2.6}$$

where $\frac{\partial \hat{\mathrm{T}}_W}{\partial t}$ the domain velocity of $\hat{W}$

With the necessary preliminaries set, the original fluid problem (Equation 1.1) can be derived with respect to $\hat{W}$. By *Lemma 2.2, 2.3* the *material derivative* $\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}$ is transformed by,

$$\frac{d\mathbf{v}}{\partial t} = \frac{\partial \hat{\mathbf{v}}}{\partial t} - (\hat{F}_W^{-1} \frac{\partial \hat{\mathrm{T}}_W}{\partial t} \cdot \hat{\nabla}) \hat{\mathbf{v}} \tag{2.7}$$

$$\mathbf{v} \cdot \nabla \mathbf{v} = \nabla \mathbf{v} \mathbf{v} = \hat{\nabla} \hat{\mathbf{v}} \hat{F}_W^{-1} \hat{\mathbf{v}} = (\hat{F}_W^{-1} \hat{\mathbf{v}} \cdot \hat{\nabla}) \hat{\mathbf{v}} \tag{2.8}$$

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = \frac{\partial \hat{\mathbf{v}}}{\partial t}(x,t) - (\hat{F}_W^{-1} \frac{\partial \hat{\mathrm{T}}_W}{\partial t} \cdot \hat{\nabla}) \hat{\mathbf{v}} + (\hat{F}_W^{-1} \hat{\mathbf{v}} \cdot \hat{\nabla}) \hat{\mathbf{v}} \tag{2.9}$$

$$= \frac{\partial \hat{\mathbf{v}}}{\partial t} + (\hat{F}_W^{-1} (\hat{\mathbf{v}} - \frac{\partial \hat{\mathrm{T}}_W}{\partial t}) \cdot \hat{\nabla}) \hat{\mathbf{v}} \tag{2.10}$$

The transformation of temporal derivatives, introduces an additional convection term $(\hat{F}_W^{-1} \frac{\partial \hat{\mathrm{T}}_W}{\partial t} \cdot \hat{\nabla})\hat{\mathrm{f}}$, which is accounts for the movement of the domain $\hat{W}$. Moving on to the right hand side of Equation 1.1, we will consider the transformation of the divergence of stress onto the reference domain $\hat{W}$. By [25] we have the following relation,

$$\nabla \cdot \sigma = \nabla \cdot (\hat{\mathrm{J}}_W \hat{\sigma} \hat{F}_W^{-T}) \tag{2.11}$$

Were $\hat{\mathrm{J}}_W \hat{\sigma} \hat{F}_W^{-T}$ is the *first Piola Kirchhoff* stress tensor, relating forces from a Eulerian description of motion to the reference domain $\hat{W}$. Assuming a Newtonian fluid, the *Cauchy stress tensor* takes the form $\sigma = -pI + \mu_f(\nabla \mathbf{v} + (\nabla \mathbf{v})^T$. Since $\sigma \neq \hat{\sigma}$ in $\hat{W}$, the spatial derivatives must be transformed, by using *Lemma 2.2*

$$\sigma = -pI + \mu_f(\nabla \mathbf{v} + (\nabla \mathbf{v})^T$$
$$\hat{\sigma} = -\hat{\mathrm{p}}I + \mu_f(\hat{\nabla}\hat{\mathbf{v}}\hat{F}_W^{-1} + \hat{F}_W^{-T}\hat{\nabla}\hat{\mathbf{v}}^T)$$

For the conservation of continuum we apply the *Piola Transformation* [25], such that

$$\nabla \cdot \mathbf{v} = \nabla \cdot (\hat{\mathrm{J}}\hat{F}_W^{-1}\hat{\mathbf{v}}) \tag{2.12}$$

As the central concepts for transforming the fluid problem on an arbitrary reference domain are introduced, the notation $\hat{W}$ will no longer be used, instead replaced with the fluid domain $\hat{\Omega}_f$, inheriting all previous concepts presented in reference with $\hat{W}$. Let $\hat{\mathrm{T}}_f : \hat{\Omega}_f \to \Omega_f(t)$ be an invertible mapping, with the scalar $\hat{\mathrm{f}}(\hat{\mathrm{x}}_f, t) = f(x, t)$ and $\hat{\mathbf{v}}_{\mathbf{f}}(\hat{\mathrm{x}}_f, t) = \mathbf{v}_f(x, t)$ counterparts. Further let $\hat{F}_f$ be the deformation gradient and $\hat{\mathrm{J}}_w$ its determinant.

Equation 2.1.1. ALE fluid problem
Let $\hat{\mathbf{v}}_f$ *be the fluid velocity,* ρ_f *the fluid density, and* ν_f *the fluid viscosity.*

$$\hat{\mathrm{J}}_f \frac{\partial \hat{\mathbf{v}}}{\partial t} + \hat{\mathrm{J}}_f(\hat{F}_f^{-1}(\hat{\mathbf{v}} - \frac{\partial \hat{\mathrm{T}}_W}{\partial t}) \cdot \hat{\nabla})\hat{\mathbf{v}} = \nabla \cdot (\hat{\mathrm{J}}_W \hat{\sigma} \hat{F}_W^{-T}) + \rho_f \hat{\mathrm{J}} \mathbf{f}_f \quad \text{in } \hat{\Omega}_f \tag{2.13}$$
$$\nabla \cdot (\hat{\mathrm{J}}\hat{F}_W^{-1}\hat{v}) = 0 \quad \text{in } \hat{\Omega}_f \tag{2.14}$$

where $\mathbf{f}_s$ represents any exterior body force.

Due to the arbitrary nature of the reference system $\hat{W}$, the physical velocity $\hat{\mathbf{v}}$ and the velocity of arbitrary domain $\frac{\partial \hat{\mathrm{W}}_w}{\partial t}$ doesn't necessary coincide, as it deals with three different reference domains [25]. The Lagrangian particle tracking $x \in \hat{\Omega}_f$, the Eulerian tracking $x \in \Omega_f$, and the arbitrary tracking of the reference domain $x \in \hat{W}$ [25]. This concept can be further clarified by the introduction of *material* and *spatial* points.

2.1.2 ALE formulation of the solid problem

With the introduced mapping identities we have the necessary tools to derive a full fluid-structure interaction problem defined of a fixed domain. Since the structure already is defined in its natural Lagrangian coordinate system, no further derivations are needed for defining the total problem.

Equation 2.1.2. *ALE solid problem*

$$\rho_s \frac{\partial \hat{\mathbf{v}}_s}{\partial t} = \nabla \cdot \mathbf{F}\mathbf{S} + \rho_s \mathbf{f}_s \quad \text{in } \Omega_s \tag{2.15}$$
$$\tag{2.16}$$

2.1.3 Fluid mesh movement

Let the total domain deformation $\hat{\mathrm{T}}(\hat{\mathrm{x}}, t)$ be divided into the solid $\hat{T}_s : \hat{\Omega}_s \rightarrow \Omega_s$, and fluid deformation $\hat{T}_f : \hat{\Omega}_f \rightarrow \Omega_f$. The physical motivated solid domain deformation, defined as $\hat{T}_s : \hat{x}_s + \hat{\mathbf{u}}_s$ were $\hat{\mathbf{u}}_s$ is the structure deformation, is a consistent mapping from the *reference configuration* to the *current configuration* of the solid domain. As pointed out in section 2.2.2, the deformation of the fluid domain doesn't inherit any physical relation between the two configurations. Despite this fact, one still introduce a fluid deformation variable $\hat{\mathbf{u}}_f$, letting the fluid domain transformation be given by

$$\hat{\mathrm{T}}_f(\hat{\mathrm{x}}, t) = \hat{\mathbf{x}} + \hat{\mathbf{u}}_f(\hat{x}, t)$$

The construction of $\hat{\mathrm{T}}_f(\hat{\mathrm{x}}, t)$ remains arbitrary, however the interface shared by both the fluid and solid domain, require an accurate transformation of the interface points by $\hat{\mathrm{T}}_f$ [25],

$$\hat{\mathrm{T}}_f(\hat{\mathrm{x}}, t) = \hat{\mathrm{T}}_s(\hat{\mathrm{x}}, t) \quad \leftrightarrow \quad \hat{\mathbf{x}} + \hat{\mathbf{u}}_f(\hat{x}, t) = \hat{\mathbf{x}} + \hat{\mathbf{u}}_s(\hat{x}, t)$$

Therefore the fluid deformation $\hat{\mathbf{u}}_f$ must have a continuous relation to the structure deformation $\hat{\mathbf{u}}_s$, enforced by $\hat{\mathbf{u}}_f = \hat{\mathbf{u}}_s$ on the interface.For the non-moving boundaries in the fluid domain, tangential deformation are allowed, however normal deformations in relation the the boundaries are not allowed [24]. The fluid domain deformation $\hat{\mathbf{u}}_f$ must therefore fulfill the boundary conditions

$$\hat{\mathbf{u}}_f(\hat{x}) = \hat{\mathbf{u}}_s \quad \hat{x} \in \hat{\Omega}_f \cup \hat{\Omega}_s \tag{2.17}$$

$$\hat{\mathbf{u}}_f(\hat{x}) \cdot \hat{\mathbf{n}} = 0 \quad \hat{x} \in \partial\hat{\Omega}_f \neq \hat{\Omega}_f \cup \hat{\Omega}_s \tag{2.18}$$

In accordance with conditions 2.17, 2.18, the fluid transformation $\hat{\mathrm{T}}_f(\hat{\mathrm{x}}, t)$ is constructed such that $\hat{\mathbf{u}}_f$ is an extension of the solid deformation $\hat{\mathbf{u}}_s$ into the fluid domain. The extension is constructed by a partial differential equation, called a *mesh lifting operator*.

2.1.4 Mesh lifting operators

In the ALE framework one of the most limiting factors is the degeneration of the mesh due to large deformations. Even the most advanced mesh motion model reaches a limit when only re-meshing is necessary to avoid mesh entanglement [38]. Consequently, the choice of mesh lifting operator is essential to generate a smooth evolution of the fluid mesh. Several mesh models have been proposed throughout the literature, and for an overview the reader is referred to [19], and the reference therein. In this thesis, the 2nd order *Laplacian* and *pseudo-elasticity* mesh model, together with the 4th order biharmonic mesh model will be considered. The 2nd order *Laplacian* and *pseudo-elasticity* mesh model are beneficial in terms of simplicity and computational efficiency, at the cost of the regularity of the fluid cells [44]. Hence, the 2nd order models are only capable of handling moderate fluid mesh deformations. Using geometrical or mesh position dependent parameters, the models can be improved to handle a wider range of deformations, by increasing the stiffness of the cell close to the interface [14].

A limitation of the 2nd order mesh models is that by Dirichlet and Neumann boundary conditions, only mesh position or normal mesh spacing can be specified respectfully, but not both [11]. This limitation is overcome by 4th order biharmonic mesh model, since two boundary conditions can be specified at each boundary of the fluid domain [11]. The 4th order biharmonic mesh model is superior for handling large fluid mesh deformations, as the model generates a better evolution of the fluid cells. A better regularity of the fluid cells also have the potential of less Newton iterations needed for convergence at each time-step [44], discussed in section 5.5. The model is however much more computational expensive compared to the 2nd order mesh models.

Mesh motion by a Laplacian lifting operator

Equation 2.1.3. The Laplace equation model
Let $\hat{\mathbf{u}}_f$ *be the fluid domain deformation,* $\hat{\mathbf{u}}_s$ *be the structure domain deformation, and let* α *be diffusion parameter raised to the power of some constant* q*. The Laplacian mesh model is given by,*

$$\begin{aligned} &-\hat{\nabla}\cdot(\alpha^q\hat{\nabla}\hat{\mathbf{u}}) = 0 \;\; \hat{\Omega}_f \\ &\hat{\mathbf{u}}_f = \hat{\mathbf{u}}_s \;\; \text{on} \;\; \Gamma \\ &\hat{\mathbf{u}}_f = 0 \;\; \text{on} \;\; \partial\hat{\Omega}_f/\Gamma \end{aligned}$$

The choice of diffusion parameter is often problem specific, as selective treatment of the fluid cells may vary from different mesh deformation problems. For small deformations, the diffusion-parameter α can be set to a small constant [45, 24]. To accommodate for larger deformations, a diffusion-parameter dependent of mesh parameters, such as fluid cell volume [2] or the Jacobian of the deformation gradient [35] have proven beneficial. In [16], the authors reviewed several options based on the distance to the closest moving boundary. This approach will be used in this thesis, using a diffusion-parameter inversely proportional to the magnitude of the distance x, to the closest moving boundary,

$$\alpha(x) = \frac{1}{x^q} \quad q = -1$$

Mesh motion by a Linear elastic lifting operator

Equation 2.1.4. The linear elastic model
Let $\hat{\mathbf{u}}_f$ *be the fluid domain deformation,* $\hat{\mathbf{u}}_s$ *be the structure domain deformation, and let* σ *be the Cauchy stress tensor. The linear elastic mesh model is given by,*

$$\begin{aligned}
&\nabla \cdot \sigma = 0 \;\; \hat{\Omega}_f \\
&\hat{\mathbf{u}}_f = \hat{\mathbf{u}}_s \;\text{ on }\; \Gamma \\
&\hat{\mathbf{u}}_f = 0 \;\text{ on }\; \partial\hat{\Omega}_f/\Gamma \\
&\sigma = \lambda Tr(\epsilon(\hat{\mathbf{u}}_f))I + 2\mu\epsilon(\hat{\mathbf{u}}_f) \qquad\qquad \epsilon(u) = \frac{1}{2}(\nabla u + \nabla u^T)
\end{aligned}$$

Where λ, μ are Lamés constants given by Young's modulus E, and Poisson's ratio ν.

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \;\; \mu = \frac{E}{2(1+\nu)}$$

The fluid mesh deformation characteristics are in direct relation which the choice of the material specific parameters, Young's modulus E and Poisson's ratio μ. Young's modulus E describes the stiffness of the material, while the Poisson's ratio relates how a materials shrinks in the transverse direction, while under extension in the axial direction. However the choice of these parameters have proven not to be consistent, and to be dependent of the given problem. In [43] the author proposed a negative Poisson ratio, which makes the model mimic an auxetic material, which becomes thinner in the perpendicular direction when submitted to compression. Another approach is to set $\nu \in [0, 0.5)$ and let E be inversely proportional to the cell volume [1], or inverse of the distance of an interior node to the nearest deforming boundary surface [19]. In this thesis, the latter is chosen merely for the purpose of code reuse from the Laplace mesh model, defined as,

$$\nu = 0.1 \quad E(x) = \frac{1}{x^q} \quad q = -1$$

Mesh motion by a Biharmonic lifting operator

Equation 2.1.5. The biharmonic mesh model
Let $\hat{\mathbf{u}}_f$ *be the fluid domain deformation,* $\hat{\mathbf{u}}_s$ *be the structure domain deformation. The biharmonic mesh model is given by,*

$$\hat{\nabla}^4\hat{\mathbf{u}}_f = 0 \quad \text{on } \hat{\Omega}_f$$

By introducing a second variable on the form $\hat{\mathrm{w}} = -\hat{\nabla}\hat{\mathrm{u}}$, we get the following system defined by

$$\begin{aligned} \hat{w} &= -\hat{\nabla}^2\hat{u} \\ -\hat{\nabla}\hat{w} &= 0 \end{aligned}$$

In combination with [43], two types of boundary conditions are proposed. Let $\hat{u}_f$ be decomposed by the components $\hat{u}_f = (\hat{\mathrm{u}}_f^{(1)}.\hat{\mathrm{u}}_f^{(2)})$. Then we have

$$\begin{aligned} &\textbf{Type 1} \quad \hat{\mathrm{u}}_f^{(k)} = \frac{\partial \hat{\mathrm{u}}_f^{(k)}}{\partial n} = 0 \quad \partial\hat{\Omega}_f/\Gamma \text{ for } k = 1, 2 \\ &\textbf{Type 2} \quad \hat{\mathrm{u}}_f^{(1)} = \frac{\partial \hat{\mathrm{u}}_f^{(1)}}{\partial n} = 0 \text{ and } \hat{\mathrm{w}}_f^{(1)} = \frac{\partial \hat{\mathrm{w}}_f^{(1)}}{\partial n} = 0 \quad \text{on } \hat{\Omega}_f^{in} \cup \hat{\Omega}_f^{out} \\ &\qquad\qquad \hat{\mathrm{u}}_f^{(2)} = \frac{\partial \hat{\mathrm{u}}_f^{(2)}}{\partial n} = 0 \text{ and } \hat{\mathrm{w}}_f^{(2)} = \frac{\partial \hat{\mathrm{w}}_f^{(2)}}{\partial n} = 0 \quad \text{on } \hat{\Omega}_f^{wall} \end{aligned}$$

The first type of boundary condition the model can interpreted as the bending of a thin plate, clamped along its boundaries. In addition to prescribed mesh position as the Laplacian and linear-elastic model, an additional constraint to the mesh spacing is prescribed at the fluid domain boundary. The form of this problem has been known since 1811, and its derivation has been connected with names like French scientists Lagrange, Sophie Germain, Navier and Poisson [17]. The second type of boundary condition is advantageous when the *reference domain* $\hat{\Omega}_f$ is rectangular, constraining mesh motion only in the perpendicular direction of the fluid boundary. This constrain allows mesh movement in the tangential direction of the domain boundary, further reducing distortion of the fluid cells [43].

2.2 Discretization of the FSI problem

In this thesis, the finite element method will be used to discretize the coupled fluid-structure interaction problem. It is beyond of scope of this thesis, to thorough dive into the analysis of the finite element method regarding fluid-structure interaction problems. Only the basics of the method, which is necessary in order to define a foundation for problem solving will be introduced.

2.2.1 Finite Element method

Let the domain $\Omega(t) \subset \mathbb{R}^d$ $(d = 1, 2, 3)$ be a time dependent domain discretized a by finite number of d-dimensional simplexes. Each simplex is denoted as a finite element, and the union of these elements forms a mesh. Further, let the domain be divided by two time dependent subdomains Ω_f and Ω_s, with the interface $\Gamma = \partial\Omega_f \cap \partial\Omega_s$. The initial configuration $\Omega(t), t = 0$ is defined as $\hat{\Omega}$, defined in the same manner as the time-dependent domain. $\hat{\Omega}$ is known as the *reference configuration*, and hat symbol will refer any property or variable to this domain. The outer boundary is set by $\partial\hat{\Omega}$, with $\partial\hat{\Omega}^D$ and $\partial\hat{\Omega}^N$ as the Dirichlet and Neumann boundaries respectively.

The family of Lagrangian finite elements are chosen, with the function space notation,

$$\hat{V}_{\Omega} := H^1(\Omega) \quad \hat{V}_{\Omega}^0 := H_0^1(\Omega)$$

where H^n is the Hilbert space of degree n.
Let Problem 2.1 denote the strong formulation. By the introduction of appropriate trial and test spaces of our variables of interest, the weak formulation can be deduced by multiplying the strong form with a test function and taking integration by parts over the domain. The velocity variable is continuous through the solid and fluid domain

$$\hat{V}_{\Omega,\hat{v}} := \hat{\boldsymbol{v}} \in H_0^1(\Omega), \;\; \hat{\boldsymbol{v}}_f = \hat{\boldsymbol{v}}_s \text{ on } \hat{\Gamma}_i$$
$$\hat{V}_{\Omega,\hat{\psi}} := \hat{\boldsymbol{\psi}}^u \in H_0^1(\Omega), \;\; \hat{\boldsymbol{v}}_f = \hat{\boldsymbol{v}}_s \text{ on } \hat{\Gamma}_i$$

For the deformation, and the artificial deformation in the fluid domain let

$$\hat{V}_{\Omega,\hat{v}} := \hat{\boldsymbol{u}} \in H_0^1(\Omega), \;\; \hat{\boldsymbol{u}}_f = \hat{\boldsymbol{u}}_s \text{ on } \hat{\Gamma}_i$$
$$\hat{V}_{\Omega,\hat{\psi}} := \hat{\boldsymbol{\psi}}^v \in H_0^1(\Omega), \;\; \hat{\boldsymbol{\psi}}_f^v = \hat{\boldsymbol{\psi}}_s^v \text{ on } \hat{\Gamma}_i$$

For simplification of notation the inner product is defined as

$$\int_{\Omega} \hat{\boldsymbol{v}} \; \hat{\boldsymbol{\psi}} \; dx = (\hat{\boldsymbol{v}}, \; \hat{\boldsymbol{\psi}})_{\Omega}$$

2.2.2 Variational Formulation

With the primaries set, we can finally define the discretization of the monolithic coupled fluid-structure interaction problem. For full transparency, variation formulation of all previous suggested mesh motion models will be shown. For brevity, the Laplace and linear elastic model will be shorted such that

$$
\begin{aligned}
\hat{\sigma}_{\text{mesh}} &= \alpha \nabla \hat{\mathbf{u}}_f && \text{Laplace} \\
\hat{\sigma}_{\text{mesh}} &= \lambda Tr(\epsilon(\hat{\mathbf{u}}_f))I + 2\mu\epsilon(\hat{\mathbf{u}}_f) && \text{Linear Elasticity}
\end{aligned}
$$

Further, only the biharmonic model for the first type of boundary condition will be introduced as the second boundary condition is on a similar form. By the concepts of the finite element method, the weak variation problem yields.

Problem 2.1. *Coupled fluid structure interaction problem for Laplace and elastic mesh lifting operator. Find* $\hat{\mathbf{u}}_s, \hat{\mathbf{u}}_f, \hat{\mathbf{v}}_s, \hat{\mathbf{v}}_f, \hat{p}_f$ *such that*

$$
\begin{aligned}
\big(\hat{\mathrm{J}}_f \frac{\partial \hat{\mathbf{v}}_f}{\partial t},\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} + \big(\hat{\mathrm{J}}_f (\hat{F}_f^{-1}(\hat{\mathbf{v}}_f - \frac{\partial \hat{\mathrm{T}}_f}{\partial t}) \cdot \hat{\nabla})\hat{\mathbf{v}}_f,\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} &+ \big(\hat{\mathrm{J}}_f \hat{\sigma} \hat{F}_f^{-T} \hat{\mathbf{n}}_f,\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Gamma}_i} \\
-\big(\hat{\mathrm{J}}_f \hat{\sigma} \hat{F}_f^{-T},\ \hat{\nabla}\hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} - \big(\rho_f \hat{\mathrm{J}}_f \mathbf{f}_f,\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} &= 0 \\
\big(\rho_s \frac{\partial \hat{\mathbf{v}}_s}{\partial t},\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_s} + \big(\hat{\mathbf{F}}\hat{\mathbf{S}}\hat{\mathbf{n}}_f,\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Gamma}_i} - \big(\hat{\mathbf{F}}\hat{\mathbf{S}},\ \nabla\hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_s} - \big(\rho_s \hat{\mathbf{f}}_s,\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_s} &= 0 \\
\big(\frac{\partial \hat{\mathbf{v}}_s - \hat{\mathbf{u}}_s}{\partial t},\ \hat{\boldsymbol{\psi}}^v\big)_{\hat{\Omega}_s} &= 0 \\
\big(\nabla \cdot (\hat{\mathrm{J}}_f \hat{F}_f^{-1} \hat{\mathbf{v}}_f),\ \hat{\boldsymbol{\psi}}^p\big)_{\hat{\Omega}_f} &= 0 \\
\big(\hat{\sigma}_{\text{mesh}},\ \hat{\nabla}\hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} &= 0
\end{aligned}
$$

Problem 2.2. *Coupled fluid structure interaction problem for biharmonic mesh lifting operator. Find* $\hat{\mathbf{u}}_s, \hat{\mathbf{u}}_f, \hat{\mathbf{v}}_s, \hat{\mathbf{v}}_f, \hat{p}_f$ *such that*

$$
\begin{aligned}
\big(\hat{\mathrm{J}}_f \frac{\partial \hat{\mathbf{v}}_f}{\partial t},\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} + \big(\hat{\mathrm{J}}_f (\hat{F}_f^{-1}(\hat{\mathbf{v}}_f - \frac{\partial \hat{\mathrm{T}}_f}{\partial t}) \cdot \hat{\nabla})\hat{\mathbf{v}}_f,\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} &+ \big(\hat{\mathrm{J}}_f \hat{\sigma} \hat{F}_f^{-T} \hat{\mathbf{n}}_f,\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Gamma}_i} \\
-\big(\hat{\mathrm{J}}_f \hat{\sigma} \hat{F}_f^{-T},\ \hat{\nabla}\hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} - \big(\rho_f \hat{\mathrm{J}}_f \mathbf{f}_f,\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} &= 0 \\
\big(\rho_s \frac{\partial \hat{\mathbf{v}}_s}{\partial t},\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_s} + \big(\hat{\mathbf{F}}\hat{\mathbf{S}}\hat{\mathbf{n}}_f,\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Gamma}_i} - \big(\hat{\mathbf{F}}\hat{\mathbf{S}},\ \nabla\hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_s} - \big(\rho_s \hat{\mathbf{f}}_s,\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_s} &= 0 \\
\big(\frac{\partial \hat{\mathbf{v}}_s - \hat{\mathbf{u}}_s}{\partial t},\ \hat{\boldsymbol{\psi}}^v\big)_{\hat{\Omega}_s} &= 0 \\
\big(\nabla \cdot (\hat{\mathrm{J}}_f \hat{F}_f^{-1} \hat{\mathbf{v}}_f),\ \hat{\boldsymbol{\psi}}^p\big)_{\hat{\Omega}_f} &= 0 \\
\big(\hat{\nabla}\hat{\mathbf{u}},\ \hat{\nabla}\hat{\boldsymbol{\psi}}^\eta\big)_{\hat{\Omega}_f} - \big(\hat{\mathbf{w}},\ \hat{\nabla}\hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} &= 0 \\
\big(\hat{\nabla}\hat{\mathbf{w}},\ \hat{\nabla}\hat{\boldsymbol{\psi}}^v\big)_{\hat{\Omega}_f} &= 0
\end{aligned}
$$

for the first type of boundary conditions introduced.

Coupling conditions

Equation 2.2.1. Interface coupling conditions

$$\begin{aligned} &\mathbf{v}_f = \mathbf{v}_s && \textit{kinematic boundary condition} \\ &\big(\hat{\mathrm{J}}_W \hat{\sigma} \hat{F}_W^{-T} \hat{\mathbf{n}}_f,\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} = \big(\hat{\mathbf{F}}\hat{\mathbf{S}}\hat{\mathbf{n}}_s,\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_s} && \textit{dynamic boundary condition} \end{aligned}$$

By a continuous velocity field on the whole domain, the *kinematic* condition is strongly enforced on the interface $\hat{\Gamma}_i$. The *dynamic* boundary condition is weakly imposed by omitting the boundary integral from the variational formulation, becoming an implicit condition for the system [42].

2.3 One-step θ scheme

For both the fluid problem and the structure problem, we will base our implementation on a θ-scheme. A θ-scheme is favorable, making implementation of classical time-stepping schemes simple. For the structure problem, θ-scheme takes the form

$$\begin{aligned} \rho_s \frac{\partial \hat{\mathbf{v}}_s}{\partial t} - \theta \nabla \cdot \hat{\mathbf{F}}\hat{\mathbf{S}} - (1-\theta)\nabla \cdot \hat{\mathbf{F}}\hat{\mathbf{S}} - \theta \rho_s \hat{\mathbf{f}}_s - (1-\theta)\rho_s \hat{\mathbf{f}}_s &= 0 \\ \frac{\partial \hat{\mathbf{v}}_s}{\partial t} - \theta \hat{\mathbf{u}}_s - (1-\theta)\hat{\mathbf{u}}_s &= 0 \end{aligned}$$

For $\theta \in [0,1]$ classical time-stepping schemes are obtained such as the first-order forward-Euler scheme $\theta = 0$, backward-Euler scheme $\theta = 1$, and the second-order Crank-Nicholson scheme $\theta = \frac{1}{2}$. Studying the fluid problem, it is initially simpler to consider the Navier-Stokes equation in an Eulerian formulation rather the ALE-formulation Following [33], a general time stepping algorithm for the coupled Navier-Stokes equation can be written as

$$\begin{aligned} \frac{1}{\Delta}(\mathbf{u}^{n+1} - \mathbf{u}^n) + B(\mathbf{u}^*)\mathbf{u}^{n+\alpha} - \nu \nabla^2 \mathbf{u}^{n+\alpha} &= -\nabla p + \mathbf{u}^{n+\alpha} \\ \nabla \cdot \mathbf{u}^{n+\alpha} &= 0 \end{aligned}$$

Here $\mathbf{u}^{n+\alpha}$ is an "intermediate" velocity defined by,

$$\mathbf{u}^{n+\alpha} = \alpha \mathbf{u}^{n+1} + (1-\alpha)\mathbf{u}^n \quad \alpha \in [0,1]$$

while $\mathbf{u}^*$ is on the form

$$\mathbf{u}^* = \mathbf{u}^{n+\vartheta} = \begin{cases} \vartheta \mathbf{u}^{n+1} + (1-\vartheta)\mathbf{u}^n & \vartheta \geq 0 \\ \vartheta \mathbf{u}^{n-1} + (1-\vartheta)\mathbf{u}^n & \vartheta \leq 0 \end{cases}$$

At first glance, defining an additional parameter ϑ for the fluid problem seems unnecessary. A general mid-point rule by $\alpha = \vartheta = \frac{1}{2}$, a second order scheme in time would easily be achieved. However, in [33] an additional second order

scheme is obtained by choosing e $\alpha = \frac{1}{2}$, $\vartheta = -1$, where $\mathbf{u}^*$ is approximated with an Adam-Bashforth linear method. Making the initial fluid problem linear while maintaining second order convergence is an important result, which have not yet been investigated thorough in literature of fluid-structure interaction. However, in the monolithic ALE method presented in this thesis, the fluid problem will still remain non-linear due to the ALE-mapping of the convective term, but making the overall problem "more linear" in contrary with a second order Crank-Nicolson scheme. The idea was initially pursued in this thesis but left aside, as discretization of the fluid convective term was not intuitive.
By letting $\alpha = \vartheta \;\; \alpha, \vartheta \in [0, 1]$ for the fluid problem, and generalizing the concepts in an ALE context, we derive the one-step θ scheme found in [43].

Problem 2.3. The one-step θ scheme *Find* $\hat{\mathbf{u}}_s, \hat{\mathbf{u}}_f, \hat{\mathbf{v}}_s, \hat{\mathbf{v}}_f, \hat{p}_f$ *such that*

$$\begin{aligned}
\big(\hat{\mathrm{J}}^{n,\theta} \frac{\partial \hat{\mathbf{v}}}{\partial t},\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} + \\
\theta\big(\hat{\mathrm{J}}\hat{F}_W^{-1}(\hat{\mathbf{v}} \cdot \hat{\nabla})\hat{\mathbf{v}},\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} + (1-\theta)\big(\hat{\mathrm{J}}\hat{F}_W^{-1}(\hat{\mathbf{v}} \cdot \hat{\nabla})\hat{\mathbf{v}},\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} \\
-\big(\hat{\mathrm{J}}\frac{\partial \hat{\mathrm{T}}_W}{\partial t} \cdot \hat{\nabla})\hat{\mathbf{v}},\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} - \theta\big(\hat{\mathrm{J}}_W \hat{\sigma} \hat{F}_W^{-T},\ \hat{\nabla}\hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} - -(1-\theta)\big(\hat{\mathrm{J}}_W \hat{\sigma} \hat{F}_W^{-T},\ \hat{\nabla}\hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} \\
-\theta\big(\rho_f \hat{\mathrm{J}}\mathbf{f}_f,\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} - (1-\theta)\big(\rho_f \hat{\mathrm{J}}\mathbf{f}_f,\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} = 0 \\
\big(\rho_s \frac{\partial \hat{\mathbf{v}}_s}{\partial t},\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_s} + -\theta\big(\hat{\mathbf{F}}\hat{\mathbf{S}},\ \nabla\hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_s} + -(1-\theta)\big(\hat{\mathbf{F}}\hat{\mathbf{S}},\ \nabla\hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_s} \\
-\theta\big(\rho_s \hat{\mathbf{f}}_s,\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_s} - (1-\theta)\big(\rho_s \hat{\mathbf{f}}_s,\ \hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_s} = 0 \\
\big(\frac{\partial \hat{\mathbf{v}}_s}{\partial t} - \theta\hat{\mathbf{u}}_s - (1-\theta)\hat{\mathbf{u}}_s,\ \hat{\boldsymbol{\psi}}^v\big)_{\hat{\Omega}_s} = 0 \\
\big(\nabla \cdot (\hat{\mathrm{J}}\hat{F}_W^{-1}\hat{\mathbf{v}}),\ \hat{\boldsymbol{\psi}}^p\big)_{\hat{\Omega}_f} = 0 \\
\big(\hat{\sigma}_{\text{mesh}},\ \hat{\nabla}\hat{\boldsymbol{\psi}}^u\big)_{\hat{\Omega}_f} = 0
\end{aligned}$$

Verification and Validation

Computer simulations are in many engineering applications a cost-efficient method of conducting design and optimize performance. However, blindly trusting results generated from a computer simulations can prove to be naive. It doesn't take a lot of coding experience before one realizes many things that can brake down and produce unwanted or unexpected results. Therefore, *credibility* of computational results are essential, meaning the simulation is worthy of belief or confidence [21]. For rigid evaluation of numerical models we use *verification and validation* (*V&V*) [34]. For a in-depth discussion of all aspects surrounding *V&V* the reader is referred to [21]. In this thesis, we follow the definitions provided by the *American Society of Mechanical Engineers guide for Verification and Validation in Computational Solid Mechanics* [31]:

Definition 3.1. Verification: The process of determining that a computational model accurately represents the underlying mathematical model and its solution.

Definition 3.2. Validation: The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model.

Simplified, *verification* considers if one solves the equations right, while *validation* is checking if one solves the right equations for the given problem [28]. Verification and validation is per definition an ongoing processes, with no clear boundary of completeness unless additional requirements are specified [28]. The goal of this chapter is to verify the implementations using the method of manufactured solution (MMS), and addressing validation in a subsequent chapter.

3.1 Verification of Code

Within scientific computing a mathematical model is often the baseline for simulations of a particular problem of interest. For scientists exploring physical phenomena, the mathematical model is often on the form of systems of partial differential equations (PDEs). Through verification of code, the ultimate goal is to ensure that the computer program correctly represents the mathematical model. To accumulate sufficient evidence that a mathematical model is solved correctly by a computer code, it must excel within predefined criteria. If the acceptance criterion is not satisfied, a coding mistake is suspected. Should the code pass the preset criteria, the code is considered verified. Of the different classes of test found in [28], *order-of-accuracy* is regarded as the most rigorous [30, 28, 36]. The method tests if the discretization error E is reduced in accordance with the *formal order of accuracy* expected from the numerical scheme. The formal order of accuracy is defined to be the theoretical rate at which the truncation error of a numerical scheme is expected to reduce. The *observed order of accuracy* is the actual rate produced by the numerical solution. For order of convergence tests, the code is assumed to be verified if we recover the theoretical convergence from the discretization error. Monitoring the discretization error E by spatial and temporal refinements, one assumes the error E can be expressed as,

$$E = C\Delta t^p + D\Delta x^l$$

where C and D are constants, Δt and Δx represents the spatial and temporal resolution, while p and l is the observed order of accuracy of the numerical scheme. In order to calculate the convergence in space l, the spatial discretization error must be negligible compared to the temporal discretization error $C\Delta t^p$. The total error can then by expressed as $E = D\Delta x^l$, and we calculate the convergence rate for subsequent spatial mesh refinement by,

$$\frac{E_2}{E_1} = (\frac{\Delta x_2}{\Delta x_1})^l \tag{3.1}$$

$$l = \frac{log(\frac{E_2}{E_1})}{log(\frac{\Delta x_2}{\Delta x_1})} \tag{3.2}$$

where E_2 is computed on a finer mesh compared to E_1 on a courser mesh. For spatial convergence tests, the same procedure applies by choosing a high resolution temporal discretization, and calculating the error E_1,E_2 by subsequent smaller time steps. In order to calculate order of convergence we need to find an exact solution of the problem. Creating an exact solution is often non-trivial. However, the method of manufactured solution provides an efficient way of generating exact solutions.

3.1.1 Method of manufactured solution

Solutions to Navier-Stokes is limited and simplifications of the original problem are often necessary to produce analytical solutions. *The method of manufactured solutions* provides a simple yet robust way of creating analytic solutions. Let partial differential equation of interest be on the form

$$\mathbf{L}(\mathbf{u}) = \mathbf{f}$$

Here $\mathbf{L}$ is a differential operator, $\mathbf{u}$ is variable the of interest, and $\mathbf{f}$ is some sourceterm. Normally, one would find u by solving the system. However, in MMS one first chooses a suitable $\mathbf{u}$, and insert it into equation 3.3, which produces a source term $\mathbf{f}$. Thus, when solving the system with the obtained $\mathbf{f}$, we know the exact solution. Another appealing feature of MMS is that the chosen $\mathbf{u}$ does not have to take into account the physical properties of the problem [27].

If the MMS is not chosen properly, the test will not work. Therefore, some guidelines for rigorous verification have been proposed in [36, 30, 27]:

- The manufactured solution should be composed of smooth analytic functions such as exponential, trigonometric, or polynomials.
- The manufactured solution should should have sufficient number of derivatives, exercising all terms and derivatives of the PDEs.

To properly verify the robustness of the method of manufactured solution, a report regarding code verification through the method manufactured solution for the time-dependent Navier-Stokes equation was published by Salari and Knupp [30]. To prove its robustness, the authors deliberate implemented code errors in a verified Navier-Stokes solver. In total 21 blind test-cases where implemented, where different approaches of verification frameworks were tested. Of these, 10 coding mistakes that reduced the observed order-of-accuracy was implemented. The MMS captured all coding mistakes, except one. This mistake would, accordingly to Roach [30], been captured if his guidelines for exact initial conditions had been followed.
In general, computing the source term $\mathbf{f}$ can be quite challenging and error prone. Therefore, symbolic computation of the sourceterm is advantageous to overcome mistakes which can easily occur when calculating by hand. For construction of the sourceterm $\mathbf{f}$, the Unified Form Language (UFL) provided in FEniCS Project will be used.

Comment on verification of the fluid-structure interaction solver by MMS

Although the MMS does not need to take any physics into account, there are often mathematical constrictions from the problem it self. From section Section 2.2 we have:
Let $\hat{\mathbf{v}}_s, \hat{\mathbf{v}}_f$ be the structure and fluid velocity, and let σ_s, σ_f be the Cauchy stress tensor for the structure and fluid respectively. Let $\mathbf{n}_i$ be the normal vector pointing out of the domain i. We then have the following interface boundary conditions.:

1. Kinematic boundary condition $\hat{\mathbf{v}}_s = \hat{\mathbf{v}}_f$, enforced strongly by a continuous velocity field in the fluid and solid domain.
2. Dynamic boundary condition $\sigma_s \cdot \mathbf{n}_s = \sigma_f \cdot \mathbf{n}_f$, enforced weakly by omitting the boundary integrals from the weak formulation in problem.

The choice of a MMS is therefore not trivial, as it must fulfill condition 1 and 2, in addition to the divergence-free condition in the fluid, and avoiding cancellation of the ALE-convective term $\frac{\partial \hat{T}_f}{\partial t}$. The construction of a MMS for a monolithic FSI problem is therefore out of the scope of this thesis. The struggle is reflected of the absence of research, regarding MMS for coupled FSI solvers in the literature. The problem is often circumvented, such as [32], where the verification process is conducted by the fluid and structure solver separately. Instead, the accuracy of the coupling is evaluated by the code validation. The approach clearly ease the process,

assuming verification of each code-block is "sufficient" to declare the code verified. In this thesis, the approach found in [32] was followed, but it must be stressed that solving each problem individually is not true verification.
Following the previous mentioned guidelines, we choose a 2D manufactured solution on a unit square. Trigonometric functions are chosen to accommodate smoothness and sufficient number of derivatives of the manufactured solution. The error is computed by comparing the numerical solution to the exact solution with higher order elements, in order to avoid super convergence. For temporal refinement we simulate ten time steps, and use the final time step solution to compute the error. Temporal refinement is computed on a uniform unit square mesh with increasing equal number of cells in the x,y direction. Fluid and solid material parameters are equally set to 1, and we choose a backward-Euler ($\theta = 1$) for the one-step theta scheme (section 2.3) initially. For the 2D incompressible Naiver-Stokes equation we choose the manufactured solution:

$$\begin{aligned}
\mathbf{u} &= [cos(x)sin(y)cos(t), -sin(x)cos(y)cos(t)] \\
p &= cos(y)sin(x)sin(t)
\end{aligned}$$

For the 2D structure equation the manufactured solution is chosen as:

$$\begin{aligned}
\mathbf{d} &= [cos(y)sin(x)sin(t), cos(x)sin(y)sin(t)] \\
\mathbf{u} &= [cos(y)sin(x)cos(t), cos(x)sin(y)cos(t)]
\end{aligned}$$

Figure 3.1 and 3.2 shows order-of-accuracy test for the fluid and solid solver. The spatial refinement study is in agreement with the expected theoretical convergence rate for both solvers, while temporal refinement fails to achieve satisfying results. Both solvers went trough rigorous testing and debugging but no error was found which could explain the poor temporal convergence. An investigation of other manufactured solutions also proved unsuccessful in order to achieve expected convergence rates. In terms of the acceptance criteria, the code is not verified. As the expected order of convergence was not achieved for a backward-Euler scheme, the second order Crank-Nicolson was not pursued. On this basis, moving on to validation deserves skepticism from the reader, as the previous results indicate that the code misrepresents the mathematical model. However, the numerical results in the upcoming chapter with comparison to several sources within the FSI literature, imply that the mathematical code is implemented correctly.

Δt	N	m	q	E_u	r_u	E_p	r_p
$1 \cdot 10^{-5}$	**8**	2	1	$2.17 \cdot 10^{-5}$	-	$1.68 \cdot 10^{-3}$	-
$1 \cdot 10^{-5}$	**16**	2	1	$2.72 \cdot 10^{-6}$	**2.99**	$4.15 \cdot 10^{-4}$	**2.02**
$1 \cdot 10^{-5}$	**32**	2	1	$3.41 \cdot 10^{-7}$	**2.99**	$1.03 \cdot 10^{-4}$	**2.01**
$1 \cdot 10^{-5}$	**64**	2	1	$4.26 \cdot 10^{-8}$	**2.99**	$2.57 \cdot 10^{-5}$	**2.00**
t	N	m	q	E_u	r_u	E_p	r_p
$\mathbf{4 \cdot 10^{-1}}$	**8**	2	1	$3.98 \cdot 10^{-3}$	-	$1.68 \cdot 10^{-3}$	-
$\mathbf{2 \cdot 10^{-1}}$	**16**	2	1	$3.95 \cdot 10^{-3}$	**0.001**	$4.44 \cdot 10^{-2}$	**0.001**
$\mathbf{1 \cdot 10^{-1}}$	**32**	2	1	$2.8 \cdot 10^{-3}$	**0.003**	$4.33 \cdot 10^{-2}$	**0.003**
$\mathbf{5 \cdot 10^{-2}}$	**64**	2	1	$2.72 \cdot 10^{-3}$	**0.003**	$4.23 \cdot 10^{-2}$	**0.004**

Table 3.1: Order of convergence study for the fluid solver. . Δt is the time step, while m and q are the polynomial degree of finite elements for velocity and pressure respectively. N is the number of elements at each side of a unit square, and r_u, r_p is the observed order of accuracy for velocity and pressure. Theoretical convergence is achieved for spatial refinement, while temporal refinements shows low convergence.

Δt	N	s	p	E_u	r_u	E_d	r_d
$1 \cdot 10^{-5}$	8	2	2	$2.16 \cdot 10^{-5}$	-	$1.18 \cdot 10^{-10}$	-
$1 \cdot 10^{-5}$	16	2	2	$2.72 \cdot 10^{-6}$	**2.99**	$1.49 \cdot 10^{-11}$	**2.02**
$1 \cdot 10^{-5}$	32	2	2	$3.40 \cdot 10^{-7}$	**2.99**	$1.87 \cdot 10^{-12}$	**2.01**
$1 \cdot 10^{-5}$	64	2	2	$4.27 \cdot 10^{-8}$	**2.99**	$2.37 \cdot 10^{-13}$	**2.00**
$\mathbf{4 \cdot 10^{-1}}$	64	1	1	2.54	-	$9.05 \cdot 10^{-1}$	-
$\mathbf{2 \cdot 10^{-1}}$	64	1	1	1.48	**0.77**	$0.28 \cdot 10^{-1}$	**0.46**
$\mathbf{1 \cdot 10^{-1}}$	64	1	1	1.08	**0.46**	$0.15 \cdot 10^{-1}$	**0.45**
$\mathbf{5 \cdot 10^{-2}}$	64	1	1	$7.83 \cdot 10^{-1}$	**0.45**	$0.11 \cdot 10^{-1}$	**0.45**

Table 3.2: Order of convergence study for the solid solver. Δt is the time step, while s and p are the polynomial degree of finite elements for velocity and deformation respectively. N is the number of elements at each side of a unit square, and r_u, r_p is the observed order of accuracy for velocity and deformation. Theoretical convergence is achieved for spatial refinement, while temporal refinements shows low convergence in comparison with theoretical convergence.

3.2 Validation

Through *verification*, one can assure that a scientific code evaluate mathematical model correctly. However, accuracy is unnecessary if the model fails to serve as an appropriate representation of the physical problem. By definition 3.2, *Validation* is the act of demonstrating that a mathematical model is applicable for its intended use with a certain degree of accuracy. That is, a mathematical model is validated if it meets some predefined criteria within a specific context. Validation is therefore not intended to portray the model as an absolute truth, nor the best model available [29].

In scientific computing, validation is traditionally conducted by comparing numerical results against existing experimental data, considered to be ground truth. The design of validation experiments vary by the motivation of their creators. Validated experiments for computational science can be divided into three groups [34]: (1) To improve fundamental understanding of a physical process, (2) Discovery or enhancement of mathematical models of well known physical processes, (3) To conclude the reliability and performance of systems. Comparing numerical results and experimental data, makes *validation* assess a wide range of issues [34]. Is the experiment relevant, and conducted correctly in accordance with prescribed parameters? What about the measurement uncertainty of reference experimental data? These issues must be addressed in order to raise sufficient confidence that the mathematical model is credible for its intended use.

3.2.1 Validation benchmark

The numerical benchmark presented in [13] has been chosen for validation of the *One-step θ* scheme from chapter 3. The benchmark has been widely accepted as a rigid validation benchmark throughout the literature [41, 42, 10]. This is mainly due to the diversity of tests included, challenging all the main components of a fluid-structure interaction scheme. The benchmark is based on the a CFD benchmark [39], where a cylinder is placed off-center in a 2D channel. In [13], an additional elastic flag is placed behind the cylinder, see Figure 4.1 The benchmark is divided into three problems, each further divided into three different sub-problems with increasing complexity. In the first problem, the fluid solver is tested for different inlet flow profiles. The second problem considers the structure solver, evaluating the bending of the elastic flag. And the final problem concerns validation of a full fluid-structure interaction problem with the fluid and the elastic flag. Several quantities for comparison are presented in [13] for validation purposes:

- The position (x,y) of point A(t) as the elastic flag undergoes deformation.
- Drag and lift forces exerted on of the whole interior geometry in contact with the fluid, consisting of the rigid circle and the elastic beam.

$$(F_D, F_L) = \int_\Gamma \sigma \cdot \mathbf{n} dS$$

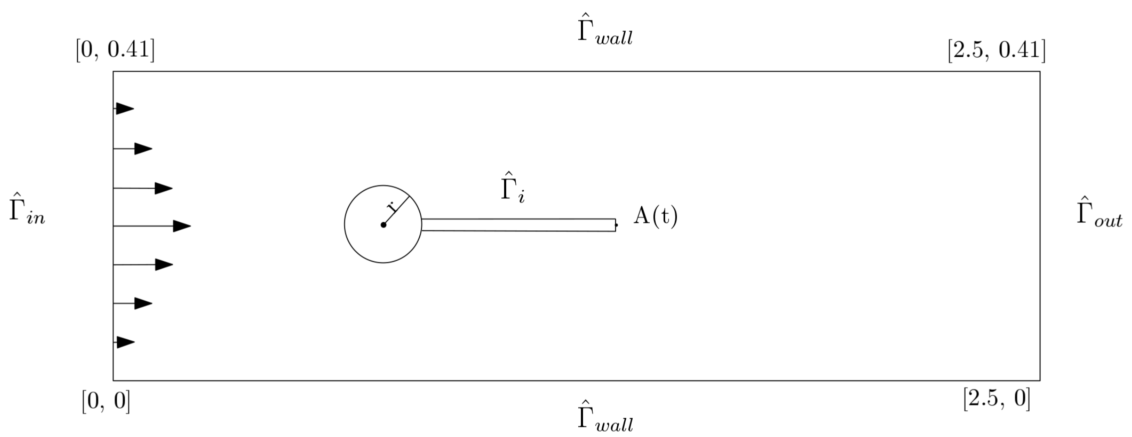


Figure 3.1: Computational domain of the validation benchmark.

All problems pose both steady state and periodic solutions. For the steady state solutions, the quantity of interest will be calculated based on a transient simulation, that has converged towards a steady state solution. For the periodic solutions, the amplitude and mean values for the time dependent quantity are calculated from the last period.

$$\text{mean} = \frac{1}{2}\text{max} + \text{min} \tag{3.3}$$

$$\text{amplitude} = \frac{1}{2}\text{max - min} \tag{3.4}$$

In [13], all steady state solutions seems to be calculated by solving a steady state equation since time-step are only reported for the periodic solutions. In this thesis, all problems in [13] are calculated using time integration. The main reason for solving the problem transiently rather than steady state, is that numerical errors associated with initial transients are negligible with a sufficiently low time step size, without laborious changes to the numerical implementation. In the following section, an overview of each problem together with numerical results will be presented. A discussion of the results are given at the end of each simulation problem. For each table, the relative error of the finest spatial and temporal refinement compared to the reference solution is reported in [13].

3.2.2 Validation of fluid solver

The validation test of the fluid solver addresses transient flow for a low Reynolds-number regime. We can take to different approaches to this problem [13]. The first one considers the setup as a fluid-structure interaction problem, setting the material properties to mimic a stiff rod. In the second approach, the flag is excluded from the computational domain. Thus, any influence from the structure is eliminated. In this thesis, I choose to use second approach.
Let $\mathbf{v}_f$*,* p_f *be the fluid velocity and pressure, and let* σ_f *be the Cauchy stress tensor, and* $\mathbf{f}_f$ *denote any sourceterm, Find* $\mathbf{v}_f$*,* p_f *such that :*

$$
\begin{aligned}
\big(\frac{\partial \mathbf{v}_f}{\partial t},\ \boldsymbol{\psi}^u\big)_{\hat{\Omega}_f} + \big((\mathbf{v}_f \cdot \nabla)\mathbf{v}_f,\ \boldsymbol{\psi}^u\big)_{\Omega_f} - \big(\hat{\sigma},\ \nabla\boldsymbol{\psi}^u\big)_{\Omega_f} - \big(\rho_f \mathbf{f}_f,\ \boldsymbol{\psi}^u\big)_{\Omega_f} &= 0 \\
\big(\nabla \cdot \mathbf{v}_f),\ \boldsymbol{\psi}^p\big)_{\Omega_f} &= 0
\end{aligned}
$$

The validation of the fluid solver is divided into three sub-problems; CFD-1, CFD-2, and CFD-3, each with different fluid parameters shown in Table 3.3. While CFD-1 and CFD-2 are steady state solutions, it is expected that the CFD-3 results is temporally varying with a von Karman street behind the flag. A parabolic velocity profile on the form,

parameter	CFD-1	CFD-2	CFD-3
$\rho^f [10^3 \frac{kg}{m^3}]$	1	1	1
$\nu^f [10^{-3} \frac{m^2}{s}]$	1	1	1
U	0.2	1	2
Re	20	100	200

Table 3.3: Parameters for the fluid validation set-up. Note that only the inlet velocity is changing.

$$
v_f(0, y) = 1.5U \frac{(H-y)y}{(\frac{H}{2})^2}
$$

is set on the left channel inflow. H is the height of the channel, while the parameter U is set differently to each problem to induce different inlet flow profiles. At the channel outflow, the pressure is set to $p = 0$. No-slip boundary conditions for the fluid are enforced on the channel walls, and on the inner geometry consisting of the circle and the elastic flag. The validation of the fluid solver is based on the evaluation of drag and lift forces on the inner geometry, compared against a reference solution. A spatial and temporal convergence study is conducted on all sub-problems.

Results

Table 3.4, 3.5, and 3.6 below shows the numerical solution of each sub-problem, CFD-1, CFD-2, and CFD-3. Each sub-problem is evaluated on four different mesh with increasing resolution. For the numerical solution of CFD-3 in Table 4.4, additional temporal and spatial refinement studies are conducted. Figure 4.1 shows the evaluation of lift and drag for the finest spatial and temporal resolution, while Figure 4.3 shows a visual representation of the fluid flow through the channel. The numerical solutions of CFD-1 in Figure 4.2 shows convergence against the reference solution. Choosing P2-P1 elements together with a fully implicit scheme $\theta = 1$, a relative error of 0.006% for lift, and 0% for drag is attained. For the numerical solution of CFD-2 presented in Figure 4.3, the same observations apply. The second order Crank-Nicolson scheme $\theta = 0.5$ was investigated for CFD-1 and CFD-2, however only improving the results of order 10^{-6} for both lift and drag. For the periodic problem CFD-3, the choice of P2-P1 elements with a fully implicit time-stepping scheme proved insufficient for capturing the expected periodic solution. Using Crank-Nicolson time-stepping scheme $\theta = 0.5$, the periodic solution was attained.

$\Delta t = 0.1 \;\; \theta = 1.0$			
nel	ndof	Drag	Lift
1438	6881	13.60	1.089
2899	13648	14.05	1.126
7501	34657	14.17	1.109
19365	88520	14.20	1.119
Reference		14.29	1.119
Error		0.006 %	0.00 %

Table 3.4: CFD-1 results, lift and drag evaluated at the inner geometry surface for increasing spatial refinement. The error is computed as the relative error from the highest mesh resolution against the reference solution.

$\Delta t = 0.01 \;\; \theta = 1.0$			
nel	ndof	Drag	Lift
1438	6881 (P2-P1)	126.0	8.62
2899	13648 (P2-P1)	131.8	10.89
7501	34657 (P2-P1)	135.1	10.48
19365	88520(P2-P1)	135.7	10.55
Reference		136.7	10.53
Error		0.007 %	0.001 %

Table 3.5: CFD-2 results, lift and drag evaluated at the inner geometry surface for increasing spatial refinement. The error is computed as the relative error from the highest mesh resolution against the reference solution.

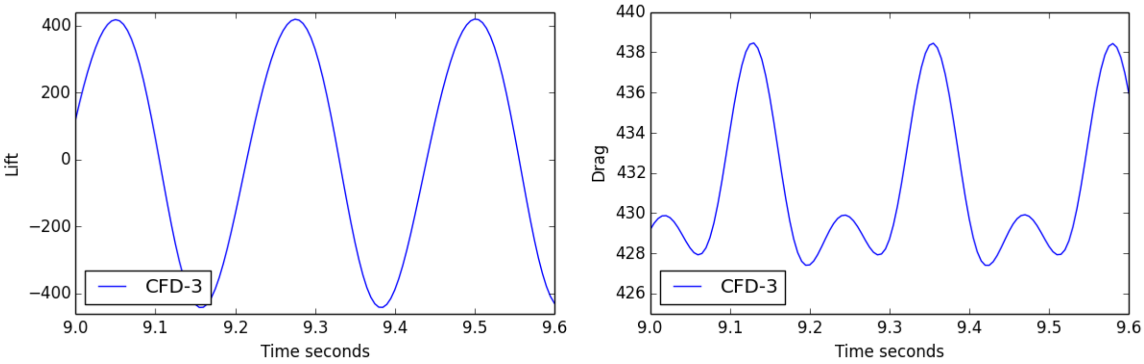


Figure 3.2: CFD-3, lift and drag forces at time t = [9, 9.6].

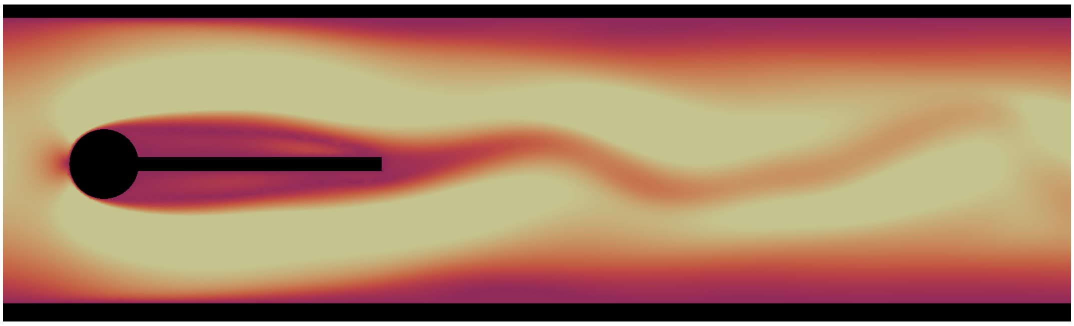


Figure 3.3: CFD-3, flow visualization of velocity time t = 9s.

$\Delta t = 0.01 \;\; \theta = 0.5$			
nel	ndof	Drag	Lift
1438	6881 (P2-P1) 16474 (P3-P2)	417.23 ± 0.0217 414.86 ± 5.6282	-249.21 ± 0.32 -7.458 ± 444.07
2899	13648 (P2-P1) 32853 (P3-P2)	408.50 ± 4.3029 432.86 ± 5.5025	-19.731 ± 373.45 -9.686 ± 431.28
7501	34657 (P2-P1) 83955 (P3-P2)	431.57 ± 5.2627 438.20 ± 5.5994	-12.497 ± 429.76 -11.595 ± 438.00
19365	88520 (P2-P1) 215219 (P3-P2)	435.43 ± 5.4133 438.80 ± 5.6290	-11.545 ± 438.89 -11.158 ± 439.23
Reference		439.95 ± 5.6183	-11.893 ± 437.81
Error		0.002 % ± 0.001 %	0.061 % ± 0.003%

Discussion

Since the choice of finite-element pair is not reported in the original work, both P3-P2 and P2-P1 element pairs for fluid and pressure respectively was compared in combination with spatial mesh refinement. From Table 4.3, a relative error $< 0.08\%$ of the mean and amplitude for lift and drag is attained. The choice P3-P2 element pair is eminent to achieve reasonable results for the first and second mesh regardless of time step. However, the third and fourth mesh resolution shows close resemblance with the reference solution, independent of finite-element pair. On

$\Delta t = 0.005$ $\theta = 0.5$			
nel	ndof	Drag	Lift
1438	6881 (P2-P1)	417.24 ± 0.0084	-249.386 ± 0.1345
1438	16474 (P3-P2)	414.90 ± 5.7319	-8.467 ± 443.45
1438	13648 (P2-P1)	408.27 ± 4.0192	-18.981 ± 363.84
2899	32853 (P3-P2)	432.90 ± 5.5333	-11.382 ± 430.60
1438	34657 (P2-P1)	431.59 ±5.2979	-13.644 ± 429.68
7501	83955 (P3-P2)	438.23 ± 5.6393	-12.917 ± 437.78
1438	88520 (P2-P1)	435.46 ± 5.4579	-13.190 ± 438.05
19365	215219 (P3-P2)	438.84 ± 5.6576	-12.786 ± 438.36
Reference		439.95 ± 5.6183	-11.893 ± 437.81
Error		0.002 % ± 0.006 %	0.075 % ± 0.001%

Table 3.6: CFD-3 results, lift and drag evaluated at the inner geometry surface. A spatial refinement study is conducted for increasing mesh resolution and two different finite element pairs. The relative error is computed from the solution of the highest mesh resolution, against the reference solution.

basis of the presented results, the fluid solver is validated in accordance with the proposed benchmark.

3.2.3 Validation of solid solver

The validation of the solid solver is conducted on a rectangular domain, representing the elastic structure in Figure 3.1. The structure is fixed to a fictional wall on the left side of the domain, pulled by a gravitational force $\mathbf{g} = (0, g)$. The validation of the solid solver is based on comparison of the deflection of point $A(t) = [A_x(t), A_y(t)]$, conducted on three refined mesh, where the number of finite elements are chosen in close resemblance with the original work in [13]. A simple investigation of different finite-element pairs, suggest that P3-P3 elements where used for making the reference solution. In this study, lower order finite-element pair was included, comparing shorter simulation time with solution accuracy. While computational time is not a major concern for the solid solver, the study is important for potentially reducing the computational time for the final validation problem.

.

parameter	CSM 1	CSM 2	CSM 3
$\rho^s[10^3 \frac{kg}{m^3}]$	1	1	1
ν^s	0.4	0.4	0.4
$\mu^s[10^6]$	0.5	2.0	0.5
$g\frac{m}{s^2}]$	2.0	2.0	2.0

Table 3.7: Parameters for the solid validation set-up.

Results

The numerical results for CSM-1, CSM-2, and CSM-3 are presented in table 3.8, 3.9, 3.10, and 3.11. For the steady state sub-problems CSM-1 and CSM-2, a spatial convergence study is conducted through mesh refinement with three different finite-element pairs. For the periodic CSM-3 problem, an additional temporal study was conducted for two different time steps. In Figure 3.4, a visualization of CSM-3 is provided for three different time steps. Finally, Figure 3.5 shows the displacement vector components, comparing all finite-element pairs for the finest mesh resolution. For CSM-1, the relative error of deformation found in Table 4.6, is 1.41% and 0.8% for the x and y coordinate respectively. In Table 4.7, a relative error of 1.49% and 0.88% for the x,y components can be found for CSM-2, proving both steady state problems coincide with the reference solution. In Table 4.8, the numerical solutions CSM-3 for time steps $\Delta t = 0.01$ and $\Delta t = 0.005$, are in close resemblance with the reference solution. The study of lower-order elements proved successful for all problems, justifying accurate results can be achieved using P2-P2 elements for deformation and velocity, even for coarse mesh resolution.

$\Delta t = 0.1$ $\theta = 1.0$			
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]
319	832 P1-P1	-5.278	-56.6
	2936 P2-P2	-7.056	-65.4
	6316 P3-P3	-7.064	-65.5
1365	3140 P1-P1	-6.385	-62.2
	11736 P2-P2	-7.075	-65.5
	25792 P3-P3	-7.083	-65.5
5143	11084 P1-P1	-6.905	-64.7
	42736 P2-P2	-7.083	-65.4
	94960 P3-P3	-7.085	-65.5
Reference		-7.187	-66.1
Error		1.41 %	0.8 %

Table 3.8: CSM-1, deformation components of A(t) for $\Delta t = 0.1$ and increasing spatial refinement. The error is computed as the relative error from the highest mesh resolution against the reference solution.

$\Delta t = 0.05 \;\; \theta = 1.0$			
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]
319	832 P1-P1	-0.3401	-14.43
	2936 P2-P2	-0.460	-16.78
	6316 P3-P3	-0.461	-16.79
1365	3140 P1-P1	-0.414	-15.93
	11736 P2-P2	-0.461	-16.81
	25792 P3-P3	-0.461	-16.82
5143	11084 P1-P1	-0.449	-16.60
	42736 P2-P2	-0.461	-16.82
	94960 P3-P3	-0.462	-16.82
Reference		-0.469	-16.97
Error		1.49%	0.88 %

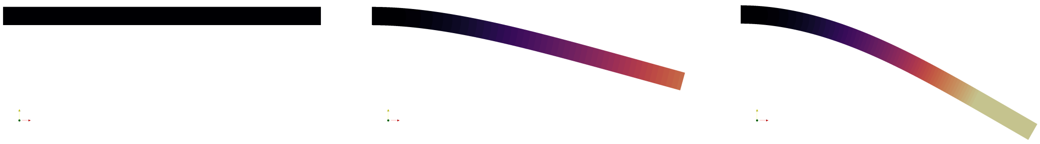
Table 3.9: CSM-2, deformation components of A(t) for $\Delta t = 0.05$ and increasing spatial refinement. The error is computed as the relative error from the highest mesh resolution against the reference solution.

$\Delta t = 0.01 \;\; \theta = 0.5$			
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]
319	832 P1-P1	-10.835 ± 10.836	-55.197 ± 56.845
	2936 P2-P2	-14.390 ± 14.392	-63.303 ± 65.149
	6316 P3-P3	-14.432 ± 14.435	-63.397 ± 65.263
1365	3140 P1-P1	-13.053 ± 13.054	-60.367 ± 62.241
	11736 P2-P2	-14.428 ± 14.432	-63.388 ± 65.256
	25792 P3-P3	-14.444 ± 14.446	-63.432 ± 65.287
5143	11084 P1-P1	-14.082 ± 14.084	-62.656 ± 64.495
	42736 P2-P2	-14.444 ± 14.447	-63.435 ± 65.288
	94960 P3-P3	-14.449 ± 14.452	-63.449 ± 65.296
Reference		-14.305 +- -14.305	-63.607 +- 65.160
Error		1% ± 1%	0.24% ± 0.24%

Table 3.10: CSM-3, deformation components of A(t) for $\Delta t = 0.01$, with increasing temporal refinement. The error is computed as the relative error from the highest mesh resolution.

$\Delta t = 0.005$ $\theta = 0.5$			
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]
319	832 P1-P1	-10.846 ± 10.848	-56.049 ± 56.053
	2936 P2-P2	-14.390 ± 14.391	-63.738 ± 64.703
	6316 P3-P3	-14.429 ± 14.430	-63.833 ± 64.810
1365	3140 P1-P1	-13.057 ± 13.057	-60.813 ± 61.826
	11736 P2-P2	-14.426 ± 14.427	-63.827 ± 64.801
	25792 P3-P3	-14.440 ± 14.441	-63.854 ± 64.845
5143	11084 P1-P1	-14.091 ± 14.091	-63.195 ± 63.981
	42736 P2-P2	-14.441 ± 14.441	-63.856 ± 64.847
	94960 P3-P3	-14.446 ± 14.446	-63.865 ± 64.860
Reference		-14.305 +- -14.305	-63.607 +- 65.160
Error		1% ± 1%	0.4% ± 0.4%

Table 3.11: CSM-3, deformation components of A(t) for $\Delta t = 0.005$, with increasing temporal refinement. The error is computed as the relative error from the highest mesh resolution.



(a) $t = 0$ (b) $t = 0.22$ (c) $t = 0.45$

Figure 3.4: CSM-3, visualization of deformation of the elastic flag for three time steps: (a) initial configuration, (b) half way extension, (c) full extension

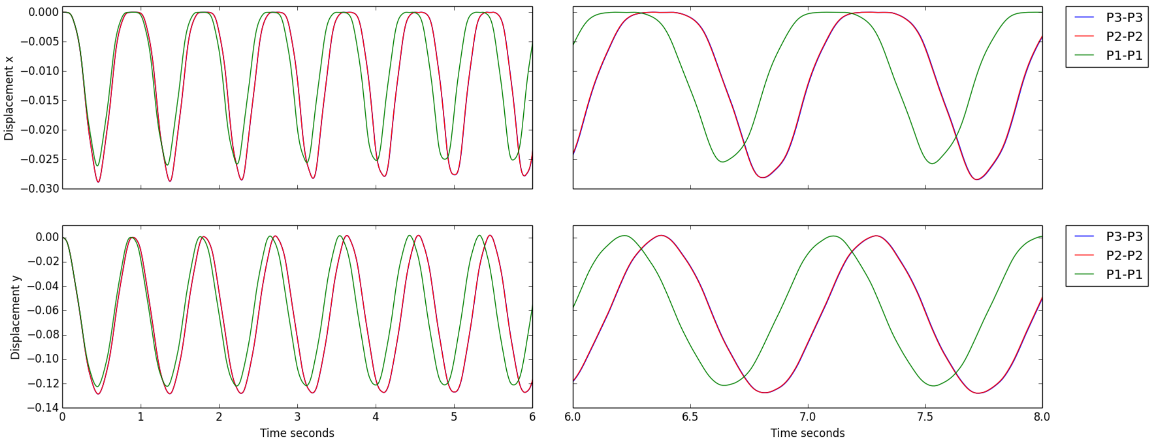


Figure 3.5: CSM-3 $\Delta t = 0.01$, deformation components of A(t) with finest mesh resolution, comparing all finite element pairs for time interval $t \in [0, 6]$ and $t \in [6, 8]$.

Discussion

Comparing all finite-element pairs for CSM-3, visualized in figure 4.5, shows P2-P2 and P3-P3 elements hardly can be distinguished from each other. In accordance with previous mentioned results and observations, the solid solver is validated in accordance with the validation benchmark.

3.2.4 Validation of fluid structure interaction solver

The validation of the FSI solver consist of three sub-problems which will be referred to FSI-1, FSI-2 and FSI-3. The FSI-1 problem yields a steady state solution for the system, inducing small deformations to the elastic flag. The FSI-2 and FSI-3 problems results in a periodic solution, where the elastic flag oscillates behind the cylinder. All sub-problems inherit the conditions from the previous validation branches, with the exception of no gravitational force on the elastic flag. On the fluid-structure interface Γ, we enforce the kinematic and dynamic boundary condition

$$\mathbf{v}_f = \mathbf{v}_s \tag{3.5}$$

$$\sigma_f \cdot \mathbf{n} = \sigma_s \cdot \mathbf{n} \tag{3.6}$$

Apart from the accuracy of the reported values, the main purpose of the validation of the solver is twofold. First, it is of great importance to ensure that the overall coupling of the fluid-structure interaction problem is executed correctly. Second, a good choice of mesh extrapolation model is essential to avoid divergence of the numerical solution, due to mesh entanglement. Based on experience in section, 4.2.1-2, the finite element pair P2-P1 for the fluid solver, and P2-P2 for the solid solver proved successful. Therefore the finite-elements P2-P2-P1 for deformation, velocity, and pressure are chosen for the numerical experiments. Higher order elements will not be examined, mainly due to long computational time, even for optimized solver approaches.

Solid parameters			
parameter	FSI-1	FSI-2	FSI-3
$\rho^s[10^3\frac{kg}{m^3}]$	1	10	1
ν^s	0.4	0.4	0.4
$\mu^s[10^6\frac{kg}{ms^2}]$	0.5	0.5	2.0
Fluid parameters			
$\rho^f[10^3\frac{kg}{m^3}]$	1	1	1
$\nu^f[10^{-3}\frac{m^2}{s}]$	1	1	1
U	0.2	1	2
parameter	FSI-1	FSI-2	FSI-3
Re	20	100	200

Table 3.12: Fluid-structure interaction sub-problem parameters

Results

The numerical results for FSI-1, FSI-2, and FSI-3 are shown in Table 4.10-12. For all sub-problems, a spatial convergence study has been conducted on three different meshes with increasing resolution, with the relative error of the finest spatial and temporal resolution. For FSI-1 in Table 4.10, an additional option is proposed, by omitting mesh lifting operator from the monolithic variational form from section 3.2.2. A comparison of the validation parameters lift, drag, and displacement with different mesh lifting operators can be found in Figure 4.2-3. Finally, Figure 4.7 and 4.9 visualize the flow field and deformation of the elastic flag for a given time.

FSI-1

Laplace					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	0.0226	0.8200	14.061	0.7542
7307	63365	0.0227	0.7760	14.111	0.7517
11556	99810	0.0226	0.8220	14.201	0.7609
Reference		0.0227	0.8209	14.295	0.7638
Error		$< 10^{-6}$ %	$< 10^{-6}$ %	0.66 %	0.38 %

Linear Elastic					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	0.0226	0.8198	14.061	0.7541
7307	63365	0.0227	0.7762	14.111	0.751
11556	99810	0.0226	0.8222	14.201	0.7609
Reference		0.0227	0.8209	14.295	0.7638
Error		$< 10^{-6}$ %	$< 10^{-6}$ %	0.66 %	0.38 %

Biharmonic bc1					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	0.0226	0.8200	14.061	0.7541
7307	63365	0.0227	0.7761	14.111	0.7517
11556	99810	0.0227	0.8017	14.205	0.9248
Reference		0.0227	0.8209	14.295	0.7638
Error		$< 10^{-6}$ %	$< 10^{-6}$ %	0.63 %	21.08 %

Biharmonic bc2					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	0.0226	0.8200	14.061	0.7543
7307	63365	0.0227	0.7761	14.111	0.7518
11556	99810	0.0227	0.8020	14.205	0.9249
Reference		0.0227	0.8209	14.295	0.7638
Error		$< 10^{-6}$ %	$< 10^{-6}$ %	0.63 %	21.09 %

No extrapolation					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	0.0224	0.9008	14.064	0.7713
7307	63365	0.0226	0.8221	14.117	0.7660
11556	99810	0.0225	0.8787	14.212	0.7837
Reference		0.0227	0.8209	14.295	0.7638
Error		$< 10^{-6}$ %	$< 10^{-5}$ %	0.58 %	2.61 %

Table 3.13: FSI 1 - Comparison of mesh extrapolation models for three spatial refinements

FSI-2

Laplace $\Delta t = 0.01$ $\theta = 0.51$					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	-15.27 ± 13.45	1.34 ± 82.4	157.00 ±14.85	-1.09 ±258.47
7307	63365	-14.23 ±13.37	1.31 ± 82.2	159,3 ± 15.43	0.92± 254.53
11556	99810	-14.96 ± 13.24	1.28 ± 81.9	161.07 ± 17.81	0.02 ± 256.04
$\Delta t = 0.001$ $\theta = 0.5$					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	-15.61± 13.21	1.34 ± 83.6	155.38 ± 13.98	-3.00 ± 289.06
7307	63365	-15.31 ± 13.07	1.02 ± 82.8	156.81 ± 14.95	-2.00 ± 276.24
11556	99810	-15.28 ± 13.04	1.28 ± 82.9	158.45 ± 16.09	-2.53 ± 276.13
Reference		-14.58 ± 12.44	1.23 ±80.6	208.83 ± 73.75	0.88 ± 234.2
Error		(4.8 ± 4.8)10^{-6} %	(4 ± 2.8) 10^{-6}%	24.1 % ± 78.1 %	387.5 % ± 17.9 %

Biharmonic 1 $\Delta t = 0.01$ $\theta = 0.51$					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	-15.44 ± 13.24	-1.38 ± 82.3	157.67 ± 15.02	-0.89± 258.87
7307	63365	-15.04 ± 12.96	0.99 ± 81.9	159.83± 16.83	0.98 ± 245.40
11556	99810	-15.29± 13.17	1.29 ± 82.5	161.69 ± 18.73	-1.86 ± 251.30
$\Delta t = 0.001$ $\theta = 0.5$					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	-15.36 ± 13.12	1.35 ± 83.1	155.38 ± 13.74	-2.55 ± 285.19
7307	63365	-15.23 ± 12.97	1.03± 82.4	157.14 ± 15.18	-8.62 ± 263.87
11556	99810	-15.27 ± 12.99	1.31 ± 82.7	157.72 ± 15.58	3.34 ± 258.76
Reference		-14.58 ± 12.44	1.23 ±80.6	208.83 ± 73.75	0.88 ± 234.2
Error		(4.7 ± 4.4)10^{-6} %	(6.5 ± 2.6)10^{-6} %	208.83 ± 73.75	0.88 ± 234.2

Biharmonic 2 $\Delta t = 0.01$ $\theta = 0.51$					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	-14.93 ± 13.22	1.35 ± 81.5	157.76 ± 15.04	-0.49 ± 254.13
7307	63365	-14.67± 13.05	1.00± 80.9	159.59 ± 16.77	2.22 ± 248.11
11556	99810	1.58 ± 12.86	1.23± 81.5	161.85 ±18.84	-1.64 ± 247.04
$\Delta t = 0.001$ $\theta = 0.5$					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	-15.63 ± 12.7	1.31 ± 82.9	155.55 ± 13.82	-2.45 ± 281.18
7307	63365	-14,99 ± 12.81	0.99± 82.14	156.86 ± 15.05	-1.65 ± 269.84
11556	99810	-15.26 ± 12.91	1.27 ± 81.8	156.86 ± 15.05	-1.65 ± 269.84
Reference		-14.58 ± 12.44	1.23 ±80.6	208.83 ± 73.75	0.88 ± 234.2
Error		(4.6 ± 3.7)10^{-6} %	(3.2 ± 1.4)10^{-6} %	24.8 % ± 79.5 %	287.5 % ± 15.2 %

Table 3.14: FSI 1 - Comparison of mesh extrapolation models for $\Delta t = [0,01,0,001]$, for three spatial refinements

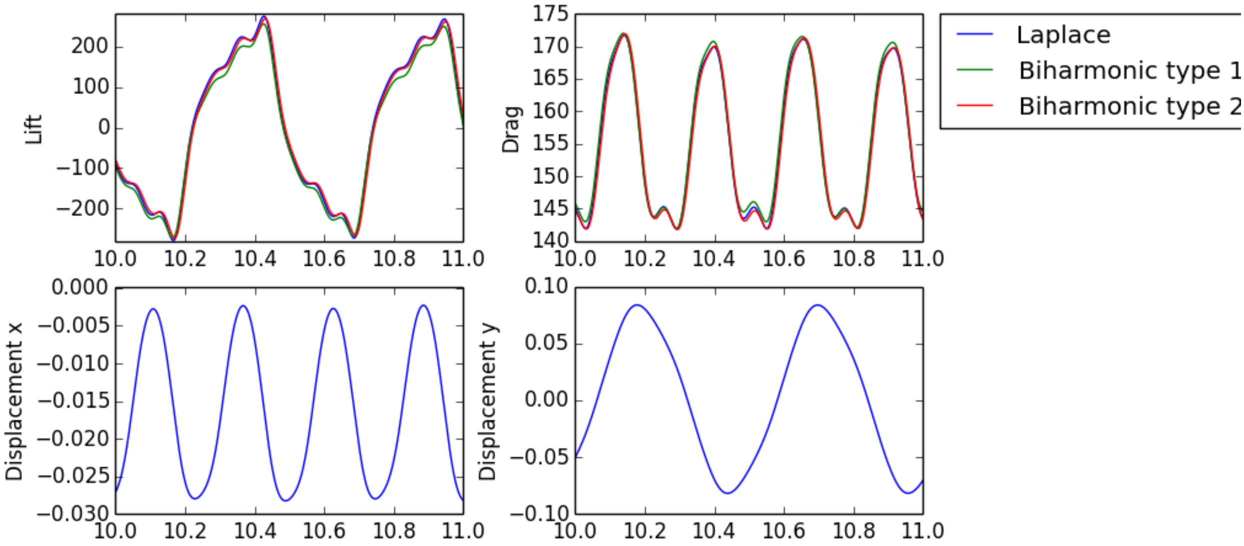


Figure 3.6: FSI-2, visualization of fully developed flow with structure deformation at time t = 9s.

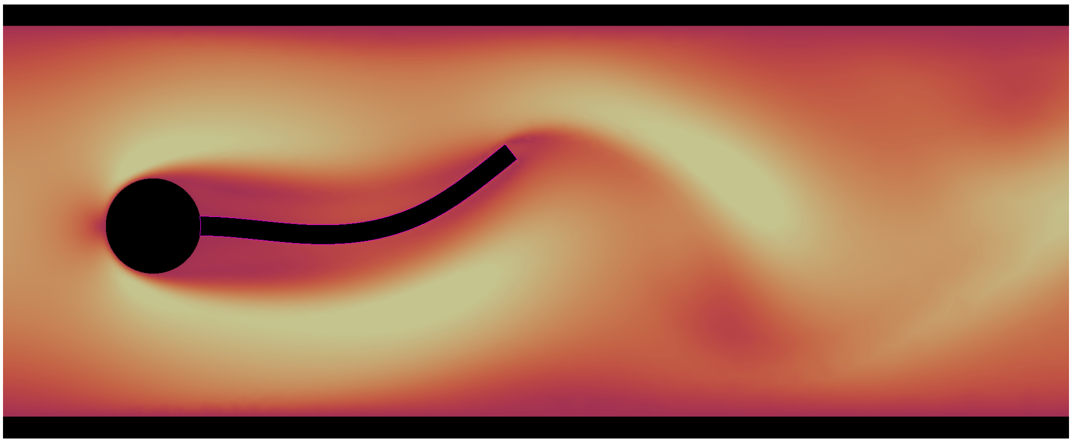


Figure 3.7: FSI-2, visualization of fully developed flow with structure deformation at time t = 9s.

FSI-3

Table 3.15: FSI 3 - Comparison of mesh extrapolation models

Laplace $\Delta t = 0.01\theta = 0.51$					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	-2.41 ± 2.41	1.49 ± 32.21	449.39 ± 14.72	0.55 ± 155.80
7307	63365	-2.32 ± 2.31	1.32 ± 31.80	451.76 ± 16.10	1.04 ± 151.51
11556	99810	-2.34 ± 2.34	1.59 ± 31.91	455.94 ± 17.34	-0.01 ± 151.36
$\Delta t = 0.001\theta = 0.5$					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	-2.91 ± 2.74	1.28 ± 35.01	450.90 ± 18.11	2.28 ±161.13
7307	63365	-2.82 ± 2.66	1.24 ± 34.69	453.56 ± 19.80	2.94 ± 158.67
11556	99810	-2.88 ± 2.72	1.49 ± 34.97	458.60 ± 22.12	2.23 ± 158.95
Reference		-2.69 ± 2.56	1.48 ± 34.38	457.3 ± 22.66	2.22 ± 149.78
Error		$(7.0 \pm 6.2)10^{-6}$ %	$(6.7 \pm 1.7)10^{-6}$ %	0.28 % ± 2.38 %	0.45 % ± 6.12 %

Biharmonic 1 $\Delta t = 0.01\theta = 0.51$					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	-2.40 ± 2.38	1.58 ± 32.07	450.16 ± 15.11	-20.09 ± 148.17
7307	63365	-2.26 ± 2.14	1.70 ± 31.3	457.37 ± 15.24	-51.77 ± 127.28
11556	99810	-2.33 ± 2.32	1.93 ± 31.5	456.40 ± 17.45	0.45 ± 149.68
$\Delta t = 0.001\theta = 0.5$					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	-2.18 ± 2.10	3.52 ± 2.90	435.19 ± 9.77	-1.59± 151.45
7307	63365	-2.80 ± 2.64	1.25 ± 3.45	454.38 ± 19.76	17.97 ± 155.08
11556	99810	-2.84 ± 2.68	1.50 ± 3.47	459.12 ± 22.97	-3.12 ± 171.22
Reference		-2.69 ± 2.56	1.48 ± 34.38	457.3 ± 22.66	2.22 ±- 149.78
Error		$(5.5 \pm 4.6)10^{-6}$ %	$(1.3 \pm 8.9)10^{-6}$ %	0.40 % ± 1.37 %	240.5 % ± 14.3 %

Biharmonic 2 $\Delta t = 0.01\theta = 0.51$					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	-2.33 ± 2.33	1.57 ± 31.6	449.44 ± 14.82	0.80 ±152.03
7307	63365	-2.25 ± 2.23	1.35 ± 31.3	452.63 ±16.29	17.11 ± 146.05
11556	99810	-2.25 ± 2.29	1.59 ± 31.4	457.89 ± 17.26	57.83 ± 141.69
$\Delta t = 0.001\theta = 0.5$					
nel	ndof	ux of A [x 10^{-3}]	uy of A [x 10^{-3}]	Drag	Lift
2474	21249	-2.83 ± 2.66	1.31 ± 34.5	450.24 ± 18.25	2.57 ± 175.42
7307	63365	-2.77 ± 2.61	0.98± 34.6	453.53 ± 20.01	2.60 ± 159.13
11556	99810	-2.80 ± 2.65	1.37 ± 34.7	458.41 ± 22.23	15.56 ± 157.78
Reference		-2.69 ± 2.56	1.48 ± 34.38	457.3 ± 22.66	2.22 ±- 149.78
Error		$(4.0 \pm 3.5)10^{-6}$ %	$(7.4 \pm 9.3)10^{-6}$ %	0.24 % ± 1.90 %	600.9 % ± 5.34 %

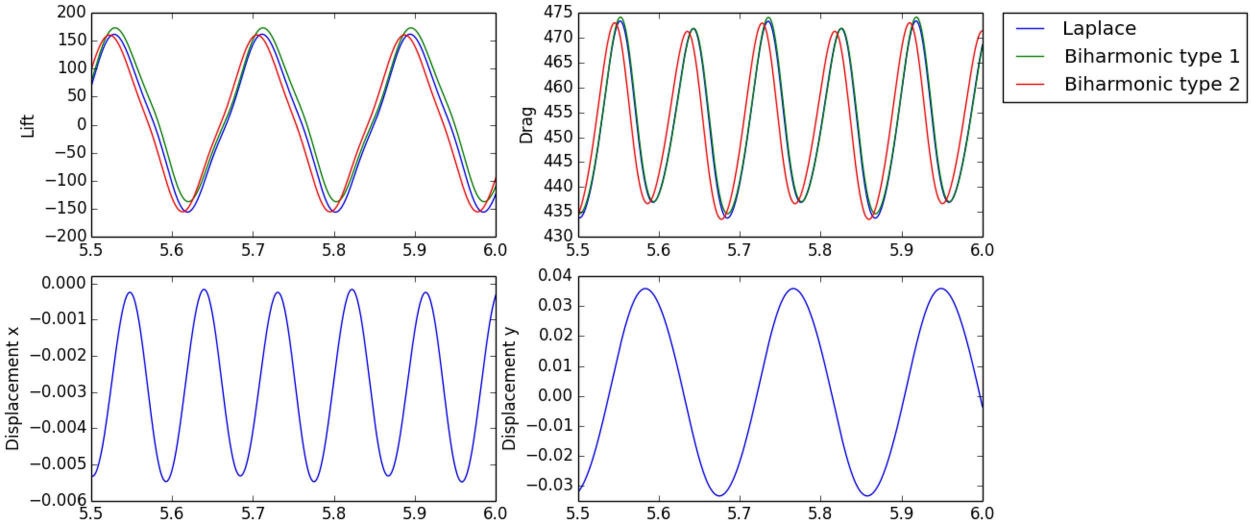


Figure 3.8: Comparison of mesh motion models for FSI-3, in time interval t $t \in [5.5, 6]$.

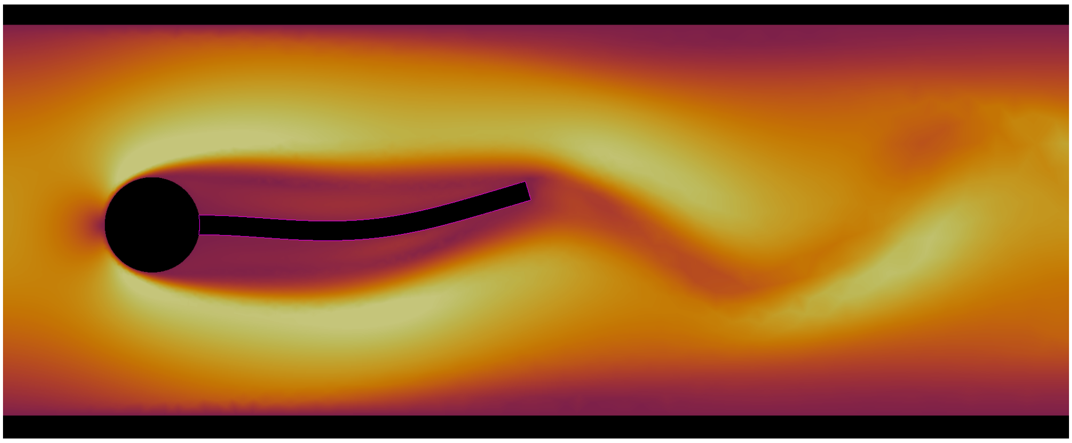


Figure 3.9: FSI-3, visualization of fully developed flow with structure deformation at time t = 5.1s.

Discussion

For FSI-1, all models excel well in comparison with the reference solution, even at coarse mesh resolution. Due to low reynolds number flow the induced deformation of the elastic flag is very small, FSI-1 proves to be excellent for initial validation of fluid-structure interaction solvers. However, due the small deformations of the elastic flag of order 10^{-5}, FSI-1 doesn't provide a rigorous test case for mesh extrapolation models. By omitting mesh extrapolation from the variational formulation in section 3.3.2, reasonable results are still obtained in Table 4.10. This fact proves FSI-1 to be misleading in terms of mesh extrapolation model, but remains excellent for initial validation of fluid-structure interaction solvers and the overall coupling of the fluid and solid equations.

The FSI-2 problem proved to be one of the most demanding tests, due to the large deformation of the elastic flag. For large deformations, the chance of fluid mesh entanglement was considerably high, stressing the mesh lifting operator extensively. The linear elastic model failed for both time sizes, but not due to mesh entanglement but early failure of the Newton-solver. This finding is comparable with the investigation conducted in [26], where early failure of the Newton-solver is in context with long-term simulation of the implicit Crank-Nicolson scheme. In their study, a shifted implicit shifted Crank-Nicolson scheme $\theta = 0.5 + \Delta t$ proved to further improve stability for the newton-solver, making the numerical scheme stable for coarse time-step. Further, numerical investigation in [26] showed that for both Crank-Nicolson and shifted Crank-Nicolson are stable for $\Delta t < 0.003$ for the same benchmark. The numerical results in this thesis proved both implicit schemes was applicable for all mesh lifting operators, except the linear elastic model.

In general, the numerical solution regarding deformation of the elastic flag proved accurate in accordance with the reference solution for all sub-problems. However, the evaluation of drag and lift proved challenging for the periodic FSI-2 and FSI-3 problems. For FSI-2, poor accuracy was observed for all mesh resolutions and time steps, while for FSI-3 the evaluation of drag remained accurate. The same observations was found in [37], a followup work of the original benchmark [13], where numerical solutions committed by different research communities was compared. The diversity of lift and drag values provided by different research communities was surprising, as differences of order 50% for drag and lift values, and 10% for displacement was observed. More surprisingly was that the authors of the original benchmark [13], who also committed their numerical results, didn't match their own reference solution with the same solver. Therefore, comparison of lift and drag forces with the reference solution alone can be misleading, and should not be the main acceptance criteria for code validation for this benchmark. Given the remarks in [37], the comparison of deformation is arguably a better main acceptance criteria. On this basis, the FSI code is validated in accordance with the original benchmark.

Numerical Experiments

4.1 Comparison of mesh lifting operators

Mesh lifting operators are essential for numerical stability of fluid-structure interaction. If the fluid mesh doesn't conform with the solid deformation, mesh entanglement increases with the possibility of numerical instabilities. In general, mesh models have shown to be either robust concerning mesh entanglements at the cost of computational time, or computational efficient with less robustness [19]. However, computational efficiency has proven not only to be dependent of the complexity of model, but also the regularity of the fluid mesh, reducing Newton iterations needed per time step [41]. In this section, a comparison of the mesh lifting operators from section 3.1.4. for the FSI-3 benchmark are presented. The linear elastic model was found not applicable in section 4.2.3. Therefore, only the Laplace and biharmonic lifting operators will be considered, comparing vertical displacement of the elastic flag, regularity of the fluid mesh, and number of Newton iterations per time step. To evaluate the regularity of the fluid mesh, the minimum value of the Jacobian of the deformation gradient have been considered in [41]. The Jacobian serves as a measure of mesh entanglement, meaning if $J_f \geq 0$, there are no crossing cells in the fluid mesh.

$$J_f = det(\hat{\mathbf{F}}_f) = det(I + \hat{\nabla}\hat{\mathbf{u}}_f)$$

where I is the identity matrix and $\hat{\mathbf{u}}_f$ is the fluid mesh deformation.

Results

Figure 4.1 shows a comparison the mesh lifting operators at the time interval $t = [0, 5]$, when a stable periodic solution is obtained. As shown in the middle figure all models shows a minima of mesh regularity at $3.8s < t < 4.2$, which is expected due to the largest deformation of the elastic flag. A larger number of Newton iterations for all models shows are needed at each time step, when the elastic flag starts oscillating for $t > 3s$. The biharmonic lifting operator is superior in terms of number of iterations need per time step, and mesh regularity in comparison with the Laplace model. Further, the biharmonic 2 model shows better mesh regularity than biharmonic 1, but shows equal behavior in terms of Newton iterations. Recall, the biharmonic 2 operator allows tangential mesh motion at the boundaries, while the biharmonic 1 model constraining mesh motion in both perpendicular and tangential direction. For all models, no distinct difference in deformation of the y-component is found.

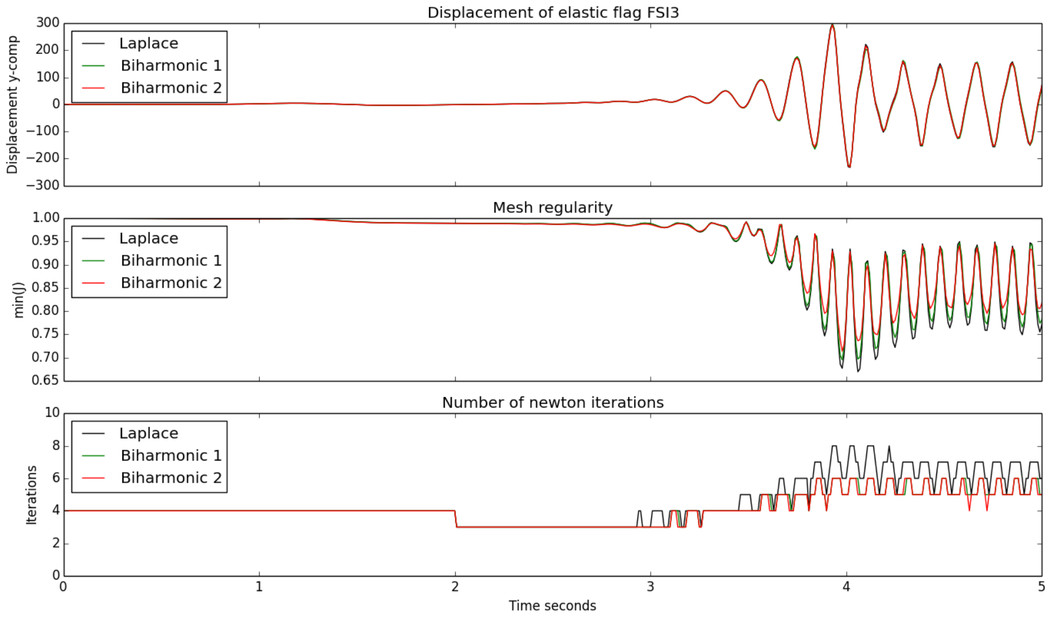


Figure 4.1: Investigation of mesh lifting operators for the FSI3 benchmark in the time interval $t \in [0, 5]$, comparing vertical displacement of elastic flag, mesh regularity, and number of Newton iterations.

Discussion

The numerical results confirms biharmonic models produce a better regularity of the fluid mesh cells, which in turn reduces the number of Newton iterations needed per time step. However, better evolution of mesh cells is by no means necessary for solving the FSI-3 problem. Therefore, the Laplace model remains a good choice, and its simplicity is preferable in terms of computational time (a topic to be discussed in section 4.2 4.3).

4.2 Investigation of long term temporal stability

One of the main challenges for constructing time-stepping schemes for ALE-methods, is the additional non-linearity introduced by the domain-velocity term in the fluid problem [7]:

$$\hat{\mathrm{J}}_f(\hat{F}_f^{-1}(\hat{\mathbf{v}}_f - \frac{\partial \hat{\mathrm{T}}_f}{\partial t}) \cdot \hat{\nabla})\hat{\mathbf{v}}_f \tag{4.1}$$

Closer inspection of the convective term reveal spatial and temporal differential operators depending non-linearly on one another. These differential operators often appear separated, making discretization of a general time-stepping scheme not directly intuitive. The domain velocity $\frac{\partial \hat{\mathrm{T}}_f}{\partial t}$ have proven to effect the stability of first and second-order accurate time stepping schemes on fixed domains, but to what extent remains unclear [7, 6]. The second order Crank-Nicolson used in this thesis, have also shown to suffer from temporal stability for long-term simulations of fluid problems. The unconditionally stable Crank-Nicolson scheme is restricted by the condition [40]:

$$k \leq ch^{\frac{2}{3}} \tag{4.2}$$

Were c is a constant, while k and h are the time-step and a mesh-size parameters. While for the stability of the time derivative of the ALE-mapping, no accurate restriction is obtained (although thoroughly explored in [7]). As a result, time step restriction is necessary to ensure that numerical stability [7]. The temporal stability for the implicit Crank-Nicolson scheme, for the validation benchmark chosen in this thesis, was studied in [26]. The criteria for the numerical experiments was to obtain a stable solution in the time interval of 10 seconds, by temporal and spatial refinement studies. Following the ideas outlined in [26], a second order scheme based on the Crank-Nicolson yields two possibilities:

Discretization 4.1. *Crank-Nicolson secant method*

$$\Big[\frac{\hat{\mathrm{J}}(\hat{\mathbf{u}}^n)\hat{\nabla}\hat{\mathbf{v}}^n\hat{\mathbf{F}}_W^{-1}}{2} + \frac{\hat{\mathrm{J}}(\hat{\mathbf{u}}^{n-1})\hat{\nabla}v^{n-1}\hat{\mathbf{F}}_W^{-1}}{2}\Big]\frac{\hat{\mathbf{u}}^n - \hat{\mathbf{u}}^{n-1}}{k}$$

Discretization 4.2. *Crank-Nicolson midpoint-tangent method*

$$\Big[\frac{\hat{\mathrm{J}}(\hat{\mathbf{u}}_{cn})\hat{\nabla}\hat{\mathbf{v}}_{cn}\hat{\mathbf{F}}_W^{-1}}{2}\Big]\frac{\hat{\mathbf{u}}^n - \hat{\mathbf{u}}^{n-1}}{k} \quad \hat{\mathbf{u}}_{cn} = \frac{\hat{\mathbf{u}}^n + \hat{\mathbf{u}}^{n-1}}{2} \;\; \hat{\mathbf{v}}_{cn} = \frac{\hat{\mathbf{v}}^n + \hat{\mathbf{v}}^{n-1}}{2}$$

The numerical experiments showed very similar performance for discretization 4.1, 4.2, and significant differences of long-term temporal stability was not found. However, spatial and temporal refinement showed the implicit Crank-Nicolson suffered from long-term stability problems for certain time steps. Choosing the time step $k = [0.005, 0.003]$, the FSI-3 problem (section 3.2.4) suffered from numerical instabilities. Interestingly, the instabilities occurred earlier in simulation time for increasing mesh refinement. A similar experiment in [43], showed reducing the time step $k = 0.001$ yield stable long-time simulation for both Discretization 4.1 and 4.2.
To overcomes the numerical instabilities two approaches have been suggested in the literature, the *shifted Crank-Nicolson* and the *frac-step method* [26, 43, 40]. In this thesis the shifted Crank-Nicolson scheme was considered, introducing stability to the overall system by shifting the centered scheme by $\theta = \frac{1}{2} + k$. If the shift is dependent of the time-step k such that $\frac{1}{2} \leq \theta \leq \frac{1}{2} + k$, the scheme will be of second order [26].

Results

A numerical investigation of long term numerical stability is shown in Figure 4.2, 4.3, where the shifted Crank-Nicolson scheme $\theta = \frac{1}{2} + \Delta t$, is compared the original Crank-Nicolson $\theta = \frac{1}{2}$. The shifted surpasses the original Crank-Nicolson scheme in terms of temporal stability, for all time steps. In Figure 4.3, the shifted Crank-Nicholson scheme retain long-time temporal stability for $\Delta t = 0.01$.. While for the ordinary Crank-Nicholson scheme, numerical experiments showed choosing $\Delta t = 0.001$ was necessarily to ensure stability, confirming the results found in [43].
Figure 4.2 shows choosing $\Delta t \in [0.2, 0.1]$ results in a steady-state solution, which can be explained by influence the solid problem. A centered Crank-Nicolson scheme $\theta = \frac{1}{2}$ is energy preserving, meaning no energy is dissipated from the system. Since the shifted Crank-Nicolson scheme is not centered, the amount of dissipated of energy is related to the time step. If the time step is sufficiently high such as $\Delta t \in [0.2, 0.1]$, the scheme will dissipate energy at a higher rate. Therefore, no periodic oscillation of the elastic flag is obtained. The validation of the solid solver shows the difference of energy dissipation for a centered and backward numerical scheme. Given the same solid parameters, a steady-state solution is obtained for CSM-1 ($\theta = 1.0$), while CSM-3 yields a periodic solution CSM-3 ($\theta = \frac{1}{2}$), shown in Figure 4.4 .
For $\Delta t \in [0.05, 0.02]$, the shifted Crank-Nicolson scheme is close to centered, meaning energy is nearly preserved in the structure. However, the Newton-solver diverges before full time interval of 10 seconds is finished. Surprisingly, a finer time step $\Delta t = 0.02$ fails at an earlier stage than $\Delta t = 0.05$. The numerical divergence is not due to mesh entanglement of the ALE-mapping, but divergence of Newton method itself [26]. It is believed the divergence of the Newton solver is linked to the influence of the domain velocity, by the research found in [7], but no clear time step restriction is obvious. Hence, several publications indicates choosing time step for a shifted Crank-Nicolson scheme is based on trial and error [43, 40].

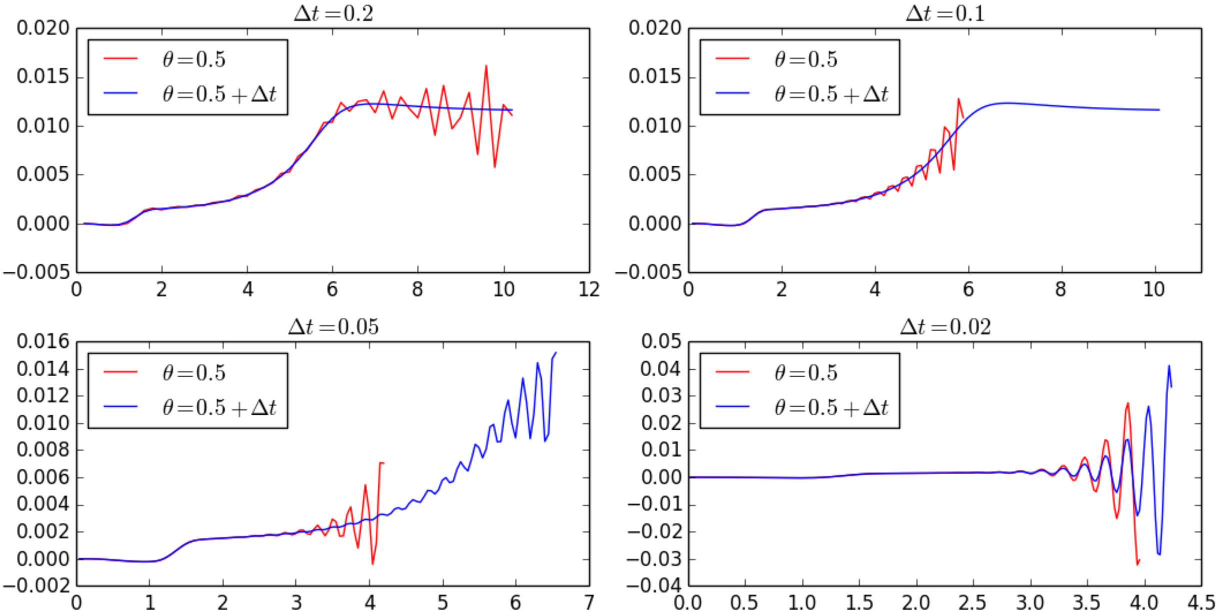


Figure 4.2: Investigation of long term numerical stability for the FSI3 benchmark in the time interval $t \in [0, 10]$, comparing the shifted and centered Crank-Nicolson scheme.

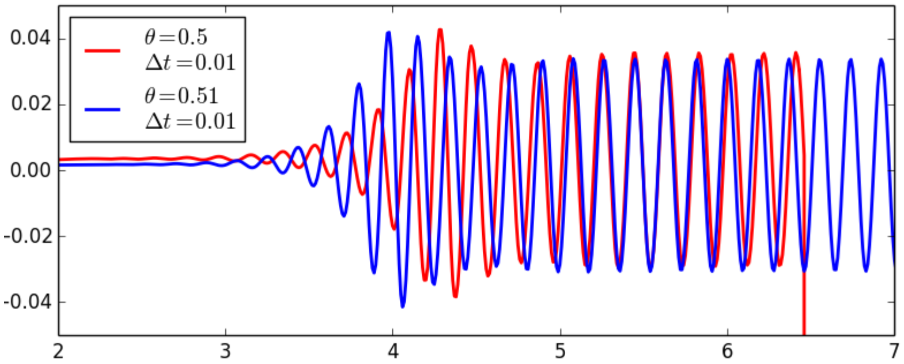


Figure 4.3: Investigation of long term numerical stability for the FSI3 benchmark in the time interval $t \in [0, 10]$, comparing the shifted and centered Crank-Nicolson scheme.

4.3 Optimization of the Newton solver

Software profiling is a dynamic program analysis, with the purpose of finding software sections which can be optimized. In scientific computing, software profiling often focuses on minimizing memory usage or executing speed, by collecting performance data to identify bottlenecks. A *bottleneck* is a term used when the total performance of a complete implementation is limited to small code fragments, accounting for the primary consumption of computer resources. For many applications, one can often assume the usage of resources follows the *The Pareto principle*. Meaning that for different types of events, roughly 80% of the effects come from 20% of the causes. An analogy to computational sciences it that 80% of the compu-

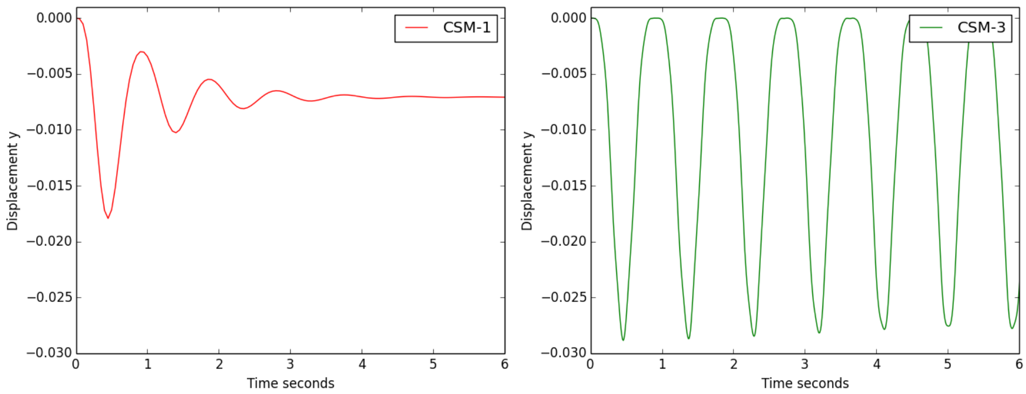


Figure 4.4: A comparison of different choices of θ, for the CSM-3 structure validation benchmark. Using the same structure parameters, a backward-Euler scheme $\theta = 1$ yields steady-state solution, while a periodic solution is observed for CSM-3 using a centered scheme $\theta = \frac{1}{2}$
To the left, centered Crank-Nicolson scheme and backward Euler scheme, using the structure validation benchmark for the time interval $t \in [0, 6]$.

tational demanding operations comes from 20% of the code. In this thesis, software profiling identified the bottleneck to be the Newton solver, using roughly $\sim$ 98% of the total computational time for each individual numerical experiment. For a full simulation of the FSI-3 problem considered in Chapter 3.2.4, on a course mesh with time step $k = 0.01$, the total computation time was initially 68 hours. With additional spatial and temporal refinement study of four mesh lifting operators, it became clear that the whole process would be too time consuming. Optimization of the Newton solver was therefore eminent to complete the initial goals set for this thesis.
Newtons method can be written as,

$$\nabla \mathbf{F}(\mathbf{x}_n)(\mathbf{x}_n - \mathbf{x}_0) = -\mathbf{F}(\mathbf{x}_n) \tag{4.3}$$

where $\mathbf{F}$ is the residue of the variational formulation, $\mathbf{x}_n$, $\mathbf{x}_0$ is vector, and $\nabla \mathbf{F}$ is the Jacobian of the residue. The Newton method can be divided into two main computational demanding operations.

- **Jacobian assembly**
  The construction of the Jacobian matrix $\nabla \mathbf{F}(\mathbf{x}_n)$ of the total residue $\mathbf{F}(\mathbf{x}_n)$
- **Direct solver**
  LU factorization and solving the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$, where $\mathbf{A} = \nabla \mathbf{F}(\mathbf{x}_n)$ and $\mathbf{b} = -\mathbf{F}(\mathbf{x}_n)$

The speed-ups methods explored in this thesis are divided into *consistent* and *inconsistent* methods. Consistent methods speeds-up the solution process by efficient assembly methods of the linear system 4.3. Inconsistent method involves simplifications of the linear system 4.3, often at the cost of poor convergence of Newtons method. As a consequence, additional Newton iterations are often necessary for convergence at each time step.

4.3.1 Consistent methods

Jacobi buffering

The residue of the FSI problem consists of both linear and non-linear terms, $\mathbf{F} = \mathbf{F}_{lin} + \mathbf{F}_{nonlin}$. For each time step the linear part $\mathbf{F}_{lin}$ remains constant, which only need to be assembled on the first Newton iteration. The linear parts are then buffered, meaning saved in a separate matrix and added with the assembled non-linear $\mathbf{F}_{nonlin}$ matrix for each Newton iteration at each time step.

4.3.2 Inconsistent methods

Reuse of Jacobian

Reusing the Jacobian of the residue $\nabla\mathbf{F}(\mathbf{x}_n)$ have two beneficial consequences. First, the Jacobian of the residue $\nabla\mathbf{F}(\mathbf{x}_n)$ is only assembled once at each time step. Second, the LU factorization of the linear system 4.3 is only needed at the first iteration, as the Jacobian remains constant through the whole time step. The first Newton iteration is therefore solved without simplifications, while the remaining iterations uses an inexact Jacobian which reduces the convergence of the method.

Quadrature reduce

The assemble time of the Jacobian greatly depends on the degree of polynomials used in the discretization of the total residual. The use of lower order polynomials reduces assemble time of the Jacobian at each Newton iteration due decreased size of the Jacobian matrix. The decreased size of the matrix also reduces the time needed for LU-factorization of the linear system 4.3. However, the quadrature reduce method leads to an inexact Jacobian which may results to additional iterations.

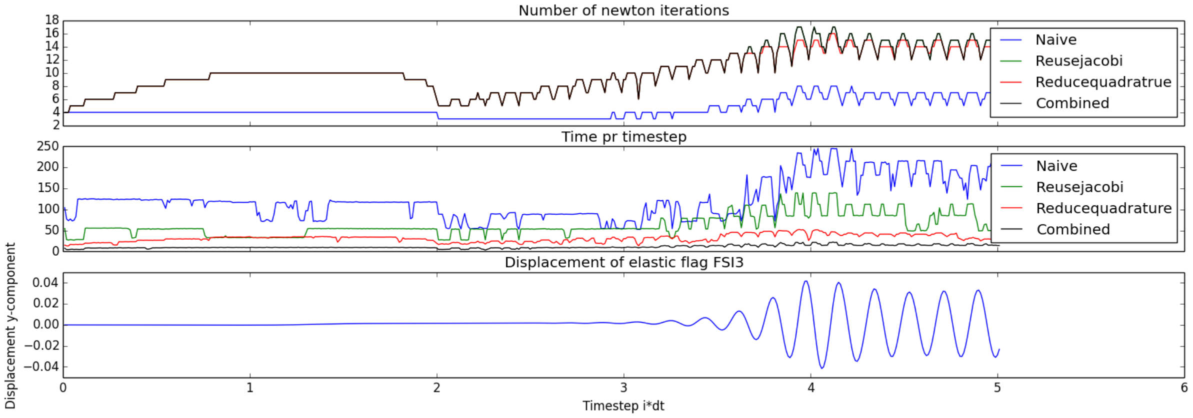
Combining quadrature reduce and Jacobian reuse

By using the previous inconsistent methods together, the first assembly of the Jacobian is done with lower order polynomials. This combination reduces assembly time of the Jacobian matrix at first iteration, and further speeds up LU-factorization of the linear system 4.3.

4.3.3 Comparison of speedup methods

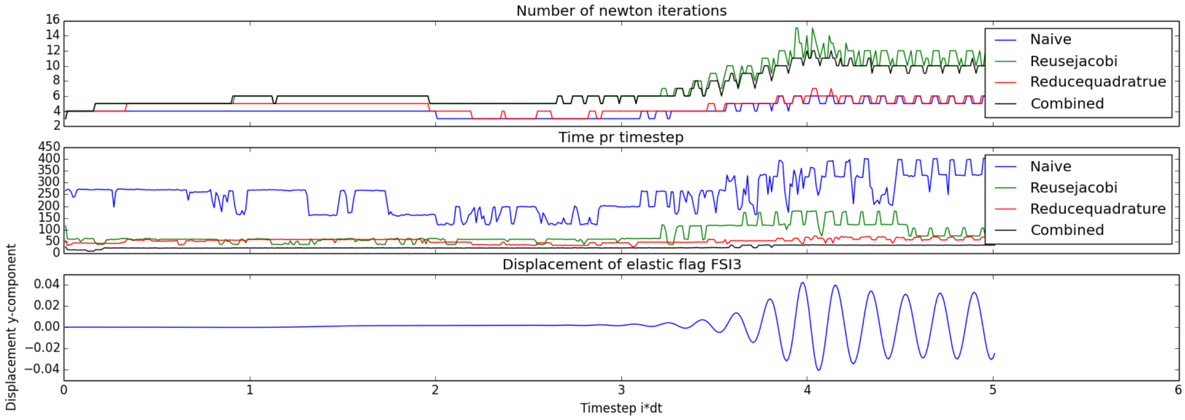
In Figure 4.5, all speed-up techniques are compared on the time interval $t = [0, 5]$, for the Laplacian model. Numerical simulations shows how all inconsistent methods increase the number of iterations needed for convergence at each time step, contrary to a consistent naive method. However, the inconsistent methods clearly dominate the time used for a full Newton iteration at each time step, in comparison with the naive approach. The fastest method proved to be the combined method, using 1.5 hours for solving the full time interval. The naive approach used 17.1 hours, meaning the combined method resulted in a 91% speedup.

The biharmonic mesh model results in Figure 4.6, shows similar observation compared to the Laplacian model. However, the benefit of a better evolution of mesh cells is reflected in the reduced number of iterations needed at each times step, for all methods. The quadrature reduce method even compares to the naive method in terms of iterations. The naive approach computed the whole time interval in 33.87 hours, while the combined method 3.46 hours, a speedup of 89%.



Laplace					
Implementation	Naive	Buffering	Reusejacobi	Reducequadrature	Combined
Mean time pr timestep	123.10	159.85	61.31	31.43	11.11
Mean iteration	4.50	7.79	10.22	10.08	10.22
Speedup ratio	-	0.12	0.50	0.74	0.91

Figure 4.5: Comparison of speed-up techniques for the Laplace mesh model.



Biharmonic Type 1					
Implementation	Naive	Buffering	Reusejacobi	Reducequad	Combined
Mean time pr timestep	243.39	307.67	76.77	51.64	24.87
Mean iteration	4.14	6.21	7.19	4.67	6.81
Speedup ration	-	-0.26	0.68	0.79	0.90

Figure 4.6: Comparison of speed-up techniques for the biharmonic type 1 mesh model.

Discussion

Of the speed-up methods considered in this thesis, the combined method proved to fastest for both mesh models. With a computational time difference of nearly ~ 2 hours, the Laplace model is superior to the biharmonic model in terms of efficiency. As long as mesh regularity isn't critical for the simulation, the Laplacian model is the preferable lifting operator. It is important to note that the introduced speed-up methods may not be applicable to other FSI problems, since inexact Jacobian methods are highly sensitive.

Conclusion and further research

In this thesis, a monolithic fluid-structure interaction solver in an arbitrary Lagrangian Eulerian description have been presented. This inline with the original goal of this thesis. However, verification of code was not achieved, however successful validation through the benchmark presented in [13] indicates the code still represents the mathematical model correctly enough. Thus it opens up the possibility that the verification of code test it self might have been erroneous.

An investigation of long term temporal stability of the FSI benchmark showed the implicit Crank-Nicolson was not applicable for time steps $\Delta t > 0.001$. To overcome the stability issues, a shifted Crank-Nicolson scheme was introduced, were long time temporal stability was obtained for $\Delta t \leq 0.01$. Software profiling motivated runtime optimizations of the Newton solver, where a combination of Jacobian reuse and lower-order polynomials to assemble the Jacobian matrix proved most beneficial in terms of computational efficiency.

In order to further explore all aspects of the FSI problem, I have also compared three different mesh lifting operators, focusing on mesh regularity and computational efficiency. The Laplace lifting operator proved to be the most efficient method, while the biharmonic operator was the most rigorous, however at the cost of computational cost. The linear elastic model failed for all tests expect the FSI-1 problem, meaning it was only valid for small deformations.

Future research

To pursue a successful verification of code is the primary future goal, which is necessary remove any doubt that the mathematical model is solved correctly. In addition, several extensions are possible. Several publications have showed the linear elastic lifiting operator applicable for a wide range of FSI problems. Therefore, further investigation is planned to investigate why the operator did not perform in my work.

Hoping to extend into 3D simulations, an implementation of iterative Krylov methods by block preconditioning are necessary, to overcome the CPU demanding nature of the monolithic FSI formulation. A substantial amount of time during the work of this thesis have been put into trying to implement the partitioned algorithm presented in [5]. In the end, I was not able to finalize this project, but future research will be spent on finding the last mistakes. A projection method would allow for a wider variety of fluid schemes, making the whole fluid equation linear. Thus, having the potential of further speeding up the solution process.

Appendices

The deformation gradient

Deformation is a major property of interest when a continuum is influenced by external and internal forces. The deformation results in relative change of position of material particles, called *strain*, is the primary property that causes and describe *stress*. Strain is purely an observation, not dependent on the material of interest. However one expects that a material undergoing strain, will apply forces within due to neighboring material particles interacting with one another. Therefore, material specific models are derived to describe how a certain material will react to a certain amount of strain. Strain measures are used to define models for *stress*, which is responsible for the deformation in materials [12]. Stress is defined as the internal forces that particles within a continuous material exert on each other, with dimension force per unit area.

The equations of continuum mechanics can be derived with respect to either a deformed or undeformed configuration. The choice of referring our equations to the current or reference configuration is indifferent from a theoretical point of view. Regardless of configuration, the *deformation gradient* and *determinant of the deformation gradient* are essential measurement in structure mechanics. By [25], both configurations are considered.

Reference configuration

Definition A.1. Let $\hat{\mathbf{u}}$ be a differential deformation field in the *reference* configuration, I be the Identity matrix and the gradient $\hat{\nabla} = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$. Then the *deformation gradient* is given by,

$$\hat{\mathbf{F}} = I + \hat{\nabla}\hat{\mathbf{u}} \tag{A.1}$$

expressing the local change of relative position under deformation.

Definition A.2. Let $\hat{\mathbf{u}}$ be a differential deformation field in the *reference* configuration, I be the Identity matrix and the gradient $\hat{\nabla} = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$. Then the *determinant of the deformation gradient* is given by,

$$J = det(\hat{\mathbf{F}}) = det(I + \hat{\nabla}\hat{\mathbf{u}}) \tag{A.2}$$

expressing the local change of volume of the configuration.

Current configuration

Definition A.3. Let $\mathbf{u}$ be a differential deformation field in the *reference* configuration, I be the Identity matrix and the gradient $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$. Then the *deformation gradient* is given by,

$$\mathbf{F} = I - \nabla \mathbf{u} \tag{A.3}$$

Definition A.4. Let $\mathbf{u}$ be a differential deformation field in the *reference* configuration, I be the Identity matrix and the gradient $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$. Then the *determinant of the deformation gradient* is given by,

$$J = det(\mathbf{F}) = det(I - \nabla \mathbf{u}) \tag{A.4}$$

Bibliography

[1] Robert Biedron and Elizabeth Lee-Rausch. Rotor Airloads Prediction Using Unstructured Meshes and Loose CFD/CSD Coupling. In *26th AIAA Applied Aerodynamics Conference*, pages 2–3, Reston, Virigina, aug 2008. American Institute of Aeronautics and Astronautics.

[2] P Crumpton and M Giles. Implicit time accurate solutions on unstructured dynamic grids. *Numerical methods in Fluids*, 25(11):1285–1300, jun 1997.

[3] J Donea, A Huerta, J.-Ph Ponthot, and A Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian methods. (1969):1–38, 2004.

[4] Miguel A Fernández and Jean-Frédéric Gerbeau. Algorithms for fluid-structure interaction problems. 2009.

[5] Miguel A. Fernández, Jean Frederic Gerbeau, and Ceremade Grandmont. A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. *International Journal for Numerical Methods in Engineering*, 69(4):794–821, 2007.

[6] L Formaggia and F Nobile. A stability analysis for the arbitrary Lagrangian Eulerian formulation with finite elements. *East-west journal of numerical mathematics*, Vol. 7(No. 2):105–131, 1991.

[7] Luca Formaggia and Fabio Nobile. Stability analysis of second-order time accurate schemes for ALE-FEM. *Computer Methods in Applied Mechanics and Engineering*, 193(39-41 SPEC. ISS.):4097–4116, 2004.

[8] Christiane Förster, Wolfgang A. Wall, and Ekkehard Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196(7):1278–1293, 2007.

[9] S. Frei, T. Richter, and T. Wick. Long-term simulation of large deformation, mechano-chemical fluid-structure interactions in ALE and fully Eulerian coordinates. *Journal of Computational Physics*, 321:874–891, 2016.

[10] Bernhard Gatzhammer. *Efficient and Flexible Partitioned Simulation of Fluid-Structure Interactions.* PhD thesis, Technische Universit¨at Munchen, 2014.

[11] Brian T. Helenbrook. Mesh deformation using the biharmonic operator. *International Journal for Numerical Methods in Engineering*, 2003.

[12] Gerhard A. Holzapfel. *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. John Wiley og Sons Ltd, 2000.

[13] Jaroslav Hron and Stefan Turek. Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow. *Fluid-Structure Interaction*, 53:371–385, 2006.

[14] Su-Yuen Hsu, Chau-Lyan Chang, and Jamshid Samareh. A Simplified Mesh Deformation Method Using Commercial Structural Analysis Software. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 2–5, Reston, Virigina, aug 2004. American Institute of Aeronautics and Astronautics.

[15] Jay D. Humphrey. *Cardiovascular Solid Mechanics*. Springer New York, New York, NY, 2002.

[16] Hrvoje Jasak and Željko Tuković. Automatic mesh motion for the unstructured Finite Volume Method. *Transactions of Famena*, 30(2):1–20, 2006.

[17] V V Meleshko. Bending of an Elastic Rectangular Clamped Plate: Exact Versus 'Engineering' Solutions. *Journal of Elasticity*, 48(1):1–50, 1997.

[18] José Merodio and Giuseppe Saccomandi. Continuum Mechanics - Volume I. In *Volume 1*, chapter 3, pages 82–84. EOLSS, 2011.

[19] Selim MM and Koomullil RP. Mesh Deformation Approaches – A Survey. *Journal of Physical Mathematics*, 7(2), 2016.

[20] J Newman. *Marine Hydrodynamics*. 1977.

[21] William L. Oberkampf and Christopher J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, Cambridge, 2010.

[22] M Razzaq, Stefan Turek, Jaroslav Hron, J F Acker, F Weichert, I Grunwald, C Roth, M Wagner, and B Romeike. Numerical simulation and benchmarking of fluid-structure interaction with application to Hemodynamics. *Fundamental Trends in Fluid-Structure Interaction*, 1:171–199, 2010.

[23] T Richter. Numerical Methods for Fluid-Structure Interaction Problems. *Course of lectures*, 2010(August 2010), 2010.

[24] T. Richter and T. Wick. Finite elements for fluid-structure interaction in ALE and fully Eulerian coordinates. *Computer Methods in Applied Mechanics and Engineering*, 199(41-44):2633–2642, 2010.

[25] Thomas Richter. Fluid Structure Interactions. 2016.

[26] Thomas Richter and Thomas Wick. On Time Discretizations of Fluid-Structure Interactions. pages 377–400. 2015.

[27] Patrick J. Roache. Code Verification by the Method of Manufactured Solutions. *Journal of Fluids Engineering*, 124(1):4, 2002.

[28] P.J. Roache. *Verification and Validation in Computational Science and Engineering.* Computing in Science Engineering, Hermosa Publishers, 1998, 8-9, 1998.

[29] Edward J. Rykiel. Testing ecological models: The meaning of validation. *Ecological Modelling*, 90(3):229–244, 1996.

[30] Kambiz Salari and Patrick Knupp. Code Verification by the Method of Manufactured Solution. Technical report, Sandia National Laboratories, 2000.

[31] LE Schwer. Guide for verification and validation in computational solid mechanics. *American Society of Mechanical Engineers*, PTC 60(V&V 10):1–15, 2006.

[32] Jason P Sheldon, Scott T Miller, and Jonathan S Pitt. Methodology for Comparing Coupling Algorithms for Fluid-Structure Interaction Problems. *World Journal of Mechanics*, 4(February):54–70, 2014.

[33] J.C. Simo and F. Armero. Unconditional stability and long-term behavior of transient algorithms for the incompressible Navier-Stokes and Euler equations. *Computer Methods in Applied Mechanics and Engineering*, 111(1-2):111–154, jan 1994.

[34] Ian Sommerville. Verification and Validation Benchmarks. Technical Report February, Sandia National Laboratories, 2006.

[35] K Stein, T Tezduyar, and R Benney. Mesh Moving Techniques for Fluid-Structure Interactions With Large Displacements. *Journal of applied mechanics*, 70(1):58–63, 2003.

[36] Dominique Tremblay, Stephane Etienne, and Dominique Pelletier. Code Verification and the Method of Manufactured Solutions for Fluid-Structure Interaction Problems. In *36th AIAA Fluid Dynamics Conference and Exhibit*, number June, pages 1–11, Reston, Virigina, jun 2006. American Institute of Aeronautics and Astronautics.

[37] Stefan Turek, Jaroslav Hron, Mudassar Razzaq, and Hilmar Wobker. Numerical Benchmarking of Fluid-Structure Interaction : A comparison of different discretization and solution approaches.

[38] Wolfgang A. Wall, Axel , Gerstenberger, Peter , Gamnitzer, Christiane , Förster, and Ekkehard , Ramm. Large Deformation Fluid-Structure Interaction – Advances in ALE Methods and New Fixed Grid Approaches. In *Fluid-Structure Interaction: Modelling, Simulation, Optimisation*, pages 195—-232. Springer Berlin Heidelberg, 2006.

[39] Frank White. *Viscous fluid flow.* McGraw-Hill, third edit edition, 2005.

[40] T. Wick. Stability Estimates and Numerical Comparison of Second Order Time-Stepping Schemes for Fluid-Structure Interactions. In *Numerical Mathematics and Advanced Applications 2011*, pages 625–632. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[41] T Wick and Thomas Wick. Variational-monolithic ALE fluid-structure interaction: Comparison of computational cost and mesh regularity using different mesh motion techniques. pages 2–12.

[42] Thomas Wick. Solving Monolithic Fluid-Structure Interaction Problems in Arbitrary Lagrangian Eulerian Coordinates with the deal.II Library. pages 25–30.

[43] Thomas Wick. *Adaptive Finite Element Simulation of Fluid-Structure Interaction with Application to Heart-Valve.* PhD thesis, Heidelberg, 2011.

[44] Thomas Wick. Fluid-structure interactions using different mesh motion techniques. *Computers and Structures*, 89(13-14):1456–1467, 2011.

[45] Thomas Wick. Fully Eulerian fluid-structure interaction for time-dependent problems. *Computer Methods in Applied Mechanics and Engineering*, 255:14–26, 2013.