Contents

Chapter 1

A motivation for studying fluid-structure interaction

Fluid-structure interaction(FSI) is an interdiciplinary field, appearing in many applications. In nature, FSI forms the basis of many physical phenomia. A fish swimming utpstream, generating thrust from the surrounding fluid by wave-like movements of its fin and body. Or a tree, bedning back and fourth due to strong winds of a storm passing by. Both examples are understandable, but points out two main instances of how FSI occur. When the fish swims, it deformes the fluid, altering the nearby flowfield. For the tree however, the swinging and bending is induced by the pressure of passing wind acting on the tree trunk and branches. Ultimatly, fluid-structure interaction occurs due to both initial effect of either fluid, structure or a combination.

Computational fluid-structure interaction (CFSI) has grown wast within engingeering in the recent years, and proved to be essential for design development and performance optimalizastion of many applications. Applications are, but not limited to biomedical computations such as heart valves and aneurisms([?], [?]) inflation of parachutes [?] ,Underwater explosions [?] and wind turbines [?].

Within aeronautics, CFSI have proven to be crucial for advances within flight characteristics and fuel economy. Due to a wide range of wing materials and flow profiles to be studied, CFSI have made testing of proposed models possible, while saving expences regarding small and full-scale experiments.

Winglet, a near vertical tip replacement for a conventional wingtip of an aircraft, have reduced drag induced by wingtip vortices during flight. As a result, the overall fuel consumption of long-distance flights have been reduced by $\sim 5\%$, which is why winglets can be observed within many airliners today. Another consequence of installing wingelts is the recution of wingtip vortices, which in turn reduces trailing turbulence behind the aircraft. The trailing turbulence can intervene with flight controls of aircraft passing through it, making wingles an important safety feature for flight traffic.

Figure 1.1: A comparison of shedding vortices from conventional wingtip, versious a winglet.

Given the multidiciplinary nature of FSI, significant developments within the field have occured within recent years. Traditionally, fluid and structure mechanics have been considered separate scientific fields, hower the complex interaction There are several causes for , coupling of equations and nonlinearity

Even though FSI play an important role within many scientific apllications,

Computational stuff, why now, refer to computational power etc

Få med beregningsorienterte, to interdiciplinary.

Chapter 2

Governing equations

Computational fluid-structure interaction (CFSI) is a multi-physics field of science, combining two separate fields of computational mechanics, computational fluid dynamics (CFD), and computational structure dynamics (CSM). While CFD and CSM traditionally have been considered as two distinct fields of science, the goal of CFSI is to combine the separate fluid and structure problems ,and their interaction or *coupling* to one another. Therefore, the study CFSI demands understanding of each separate field. This chapter presents the governing equations of the individual fluid and structure problem. Balance laws for each separate problem, together with auxiliary kinematic, dynamic and material relations will be described briefly.

2.1 Continuum Mechanics

To interpret nature, mathematical models are needed to describe how object around us reacts to external and/or internal stimuli. The mathematical models forms a basis for establishing elementary conservation laws and partial differential equations (PDE's), making scientist and engineers not only able to understand physical phenomena, but also predict them.
Fluid and solids are both materials built up by a sequence of atoms, meaning on a microscopic level, an observer will locate discontinuties and space within the material. Evaluating each atom, or *material point*, is not impossible from a mathematical point of view. However, for mathematical moddeling and applications, the evaluation of each material point remains unpractical. In *Continuum mechanics*, first formulated by Augustin-Louis Cauchy [**?**], the microscopic structure of materials are ignored, assuming the material of interest is *continuously distributed* in space.

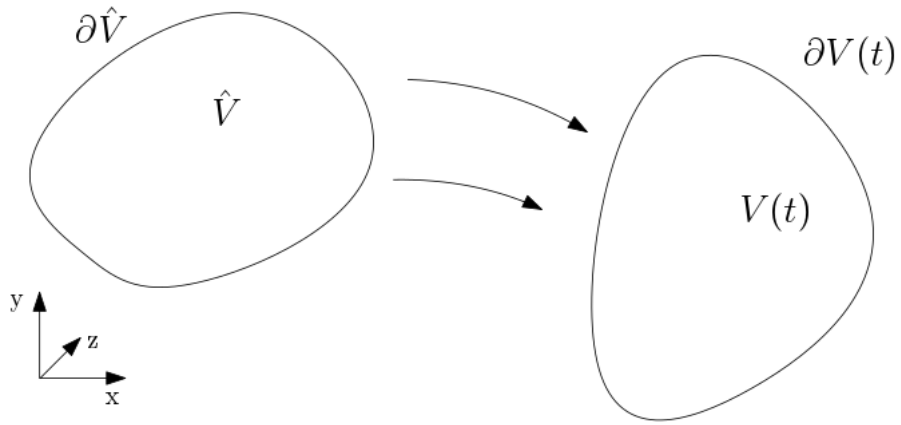

Figure 2.1: Unitpotato

A *continuum* is defined as a volume $V(t) \subset \mathbb{R}^d$ $d \in (2, 3)$, continuously distributed throughout its own volume. The initial shape of the continuum, the *reference configuration* $V(t = t_0) = \hat{V}$, is assumed to be stress free. When the *continuum* undergoes deformation due to applied internal or external forces , the new configuration $V(t)$ for $t \geq t_0$, deviates from its *reference configuration*. The new configuration $V(t) \neq \hat{V}$, is defined as the *current configuration*. If the continuum undergoes no deformation, the *reference* and *current* configuration simply coincide.

2.2 The Lagrangian and Eulerian description of motion

A fundamental difference between fluid and structure mechanics is the description of motion, the *Lagrangian* and *Eulerian* description. Within structure mechanics, deformation of a continuum due to internal/external forces are of primary interest. When the continuum undergoes deformation, it will deviate from is *reference configuration* $\hat{V}$. To measure the deformation, one must know the relative position of some particle $x(t) \in V(t)$, from its original configuration $\hat{\mathbf{x}} \in \hat{V}$.

Let $\hat{\mathrm{x}}$ be a particle in the reference $\hat{\mathrm{x}} \in \hat{\mathrm{V}}$. Further let $\mathrm{x}(\hat{\mathrm{x}}, \mathrm{t})$ be the new location of a particle $\hat{\mathrm{x}}$ for some time t such that $x \in V(t)$. Assume no two particles $\hat{\mathrm{x}}_a, \hat{\mathrm{x}}_b \in \hat{\mathrm{V}}$ occupy the same location for some time $V(t)$. Then the transformation $\hat{\mathrm{T}}(\hat{\mathrm{x}}, t) = x(\hat{\mathrm{x}}, t)$ maps a particle $\hat{\mathrm{x}}$ from the *reference configuration* $\hat{\mathrm{V}}$, to the *current configuration* $V(t)$ Assuming that the path for some $\hat{\mathrm{x}}$ is continuous in time, the inverse mapping $\hat{\mathrm{T}}^{-1}(x, t) = \hat{\mathrm{x}}(x, t)$, maps $x(\hat{\mathrm{x}}, t)$ back to its initial location at time $t = t_0$.
These mappings lets us track each particle from some *reference configuration* to some deformed state at time t. Tracking each particle $\hat{\mathrm{x}} \in \hat{\mathrm{V}}$ is often denoted the *Lagrangian Framework* and is a natural choice of describing structure mechanics.

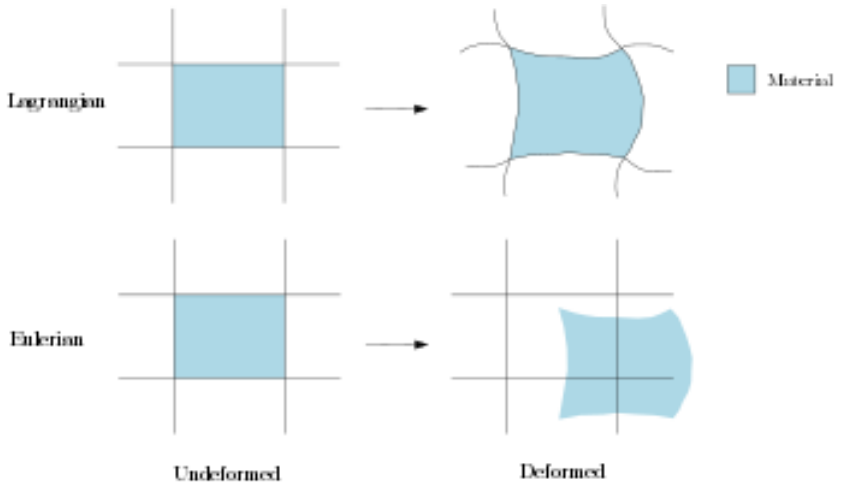

Figure 2.2: Comparison of the Lagrangian and Eulerian description of motion

By some particle $/batx \in \hat{\mathbf{V}}$, the *deformation* $\hat{\mathbf{u}}(\hat{\mathrm{x}}, t)$ is given by the relation

$$\hat{\mathbf{u}}(\hat{\mathrm{x}}, t) = x(\hat{\mathbf{x}}, t) - \hat{\mathbf{x}} = \hat{\mathrm{T}}(\hat{\mathbf{x}}, t) \tag{2.1}$$

and the *deformation velocity* is given by the time derivative of *deformation* such that

$$\hat{\mathbf{v}}(\hat{\mathrm{x}}, t) = d_t x(\hat{\mathbf{x}}, t) = d_t \hat{\mathbf{u}}(\hat{\mathbf{x}}, t) = \frac{\partial \hat{\mathrm{T}}(\hat{\mathrm{x}}, t)}{\partial t} \tag{2.2}$$

Considering a flow of fluid particles in a river, a *Lagrangian* description of the particles would be tedious as the number of particles entering and leaving the domain quickly rise to a immense number. Instead consider defining a view-point V fixed in time, and monitor every fluid particle passing the coordinate $x \in V(t)$ as time elapses. Such a description is defined as the *Eulerian framework*. Therefore the Eulerian formulation is natural for describing fluid dynamics.
We can describe the particles occupying the *current configuration* $V(t)$ for some time $t \geq t_0$

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

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

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

and its velocity

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

It is important to mention that the we are not interested in which particle is occupying a certain point in our domain, but only its properties.

2.3 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*. and is the primary property that causes and describe *stress*.

Strain is purely an observation, and it is not dependent on the material of interest. However one expects that a material undergoing strain, will give forces within a continuum due to neighboring material particles interacting with one another. Therefore one derive material specific models to describe how a certain material will react to a certain amount of strain. These strain measures are used to define models for *stress*, which is responsible for the deformation in materials [**?**]. 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 undeformend configuration. The choice of refering our equations to the current or reference configuration is indifferent from a theoretical point of view. In practice however this choice can have a severe impact on our strategy of solution methods and physical of modelling. [22]. Regardless of configuration, the *deformation gradient* and *determinant of the deformation gradient* are essential measurement in structure mechanics. By [15], both configurations are considered.

Reference configuration

Definition 2.1. Let $\hat{u}$ be a differential deformation field in the *reference* configuration, I be the Identity matrix and the gradient $\hat{\nabla} = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$. Then the *deformation gradient* is given by,

$$\hat{\mathbf{F}} = I + \hat{\nabla}\hat{\mathbf{u}} \tag{2.3}$$

expressing the local change of relative position under deformation.

Definition 2.2. Let $\hat{u}$ be a differential deformation field in the *reference* configuration, I be the Identity matrix and the gradient $\hat{\nabla} = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$. Then the *determinant of the deformation gradient* is given by,

$$J = det(\hat{\mathbf{F}}) = det(I + \hat{\nabla}\hat{\mathbf{u}}) \tag{2.4}$$

expressing the local change of volume the configuration.

From the assumption of linear operator $\mathbf{F}$, and no two particles $\hat{\mathbf{x}}_a, \hat{\mathbf{x}}_b \in \hat{\mathrm{V}}$ occupy the same location for some time $V(t)$, J to be greater than 0 [22].

Current configuration

Definition 2.3. 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,

$$\mathbf{F} = I - \nabla \mathbf{u} \tag{2.5}$$

expressing the local change of relative position under deformation.

Definition 2.4. 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,

$$J = det(\mathbf{F}) = det(I - \nabla \mathbf{u}) \tag{2.6}$$

expressing the local change of volume the configuration.

2.4 The solid problem

The governing equations for the solid mechanics are given by the balance law,

Equation 2.4.1. *Solid momentum*

$$\rho_s \frac{\partial \hat{\mathbf{v}}_s}{\partial t} = \nabla \cdot \hat{\mathbf{S}} + \rho_s \mathbf{f}_s \quad \text{in } \hat{\Omega}_s \tag{2.7}$$

$$\frac{\partial \hat{\mathbf{v}}_s}{\partial t} = \hat{\mathbf{u}_s} \quad \text{in } \hat{\Omega}_s \tag{2.8}$$

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, $\hat{\mathbf{S}}$ is the second Piola-Kirchhoff stress tensor, related to the Cauchy-stress tensor by,

$$\hat{\mathbf{S}}_s = \hat{J}\hat{\mathbf{F}}^{-1}\sigma_s\hat{\mathbf{F}}^{-T}$$

The elasticity of the material is expressed by the *Poisson ratio* ν_s, *Young modulus* E, or Lamè coefficients λ_s and μ_s. Their relation is given by,

$$E_y = \frac{\mu_s(\lambda_s + 2\mu_s)}{(\lambda_s + \mu_s)} \quad \nu_s = \frac{\lambda_s}{2(\lambda_s + \mu_s)}$$
$$\lambda_s = \frac{\nu E_y}{(1+\nu_s)(1-2\nu_s)} \quad \mu_s = \frac{E_y}{2(1+\nu_s)}$$

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. For small-deformations, *Hooke's law* assumes a linear relation between strain and stress,

Definition 2.5. 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,

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

Hooke's law is however limited to a small-deformation regime, and is not applicable for larger 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 2.6. 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. Vernant-Kirchhoff model* is then given by the relation,

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

where $\hat{\mathbf{C}}$ *is the right Cauchy-Green strain tensor and* $\hat{\mathbf{E}}$ *is the Green Lagrangian strain tensor* [1]

Though STVK can handle large deformations, it is not valid for large strain [**?**]. However since the strain considered in this thesis are small, it will remain our primary choice of strain-stress relation. STVK describes materials of compressible nature, but is should be mentioned that for large deformation models describing incompressible materials can be considered. Specially the Incompressible Neo-Hooke (INH) model is considered in several publications (see [**?**], [14]), sharing the same hyperelastic properties as the STVK model. While both models are valid for large deformations, the INH is superior compared to STVK in the sense that it holds for large strains aswell [**?**].

[1] See appendix A for definition

As for the fluid problem we define Dirichlet and Neumann boundary conditions on the form

$$\begin{aligned} \mathbf{v}_s &= \mathbf{v}_s^D \quad \text{on } \Gamma_s^D \subset \partial\Omega_s \\ \sigma_s \cdot \mathbf{n} &= \mathbf{g} \quad \text{on } \Gamma_s^N \subset \partial\Omega_s \end{aligned}$$

2.5 The Fluid problem

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

Equation 2.5.1. *Navier-Stokes equation*

$$\rho\frac{\partial \mathbf{v}_f}{\partial t} + \rho\mathbf{v}_f \cdot \nabla\mathbf{v}_f = \nabla \cdot \sigma + \rho\mathbf{f}_f \quad \text{in } \Omega_f \tag{2.9}$$

$$\nabla \cdot \mathbf{v}_f = 0 \quad \text{in } \Omega_f \tag{2.10}$$

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

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

Additional appropriate boundary conditions are supplemented to the equation for a given problem. The first type of of boundary conditions are Dirichlet boundary conditions,

$$\mathbf{v}_f = \mathbf{v}_f^D \quad \text{on } \Gamma_f^D \subset \partial\Omega_f \tag{2.11}$$

The second type of boundary condition are Neumann boundary conditions

$$\sigma_f \cdot \mathbf{n} = \mathbf{g} \quad \text{on } \Gamma_f^N \subset \partial\Omega_f \tag{2.12}$$

Chapter 3

Computational 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. CFSI adds another layer of complexity to the solution process, (1) the *coupling* of the fluid and solid equations , (2) 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.

$$\mathbf{v}_f = \mathbf{v}_s \tag{3.1}$$

$$\sigma_f \cdot \mathbf{n} = \sigma_s \cdot \mathbf{n} \tag{3.2}$$

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 aslo 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 simuntainiously and the strong coupling contributes to a stronger nonlinear behaviour of the whole system [20]. 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. Sovling 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 separatly was used in the initial efforts in CFSI, due to existing solvers for one or both problems [7]. Therefore, computational efficiency and code resue is one of the main reasons for choosing the partitioned approach. A major drawback is the methods ability to enfore 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 compuational time.

Regardless of framework, CFSI has to cope with a numerical artifact called the "added-mass effect" [5], [4], [6]. 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 [13]. Instead, the term is used to describe the numerical instabilities occurring for weakly coupled schemes, in conjunction with *in compressible fluids* and slender structures [6], 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 [7]. 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.[**?**]. In the *Interface-tracking* method, the mesh moves to accommodate for the movement of the structure as it deformes 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 yeilds 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 iteself. This approach is in general not limitied in terms of deformations, but suffers from reduced accuracy at the interface. [**?**].

Among the multiple approaches within CFSI, the arbitary Lagrangia-Eulerian methos is chosen for this thesis.

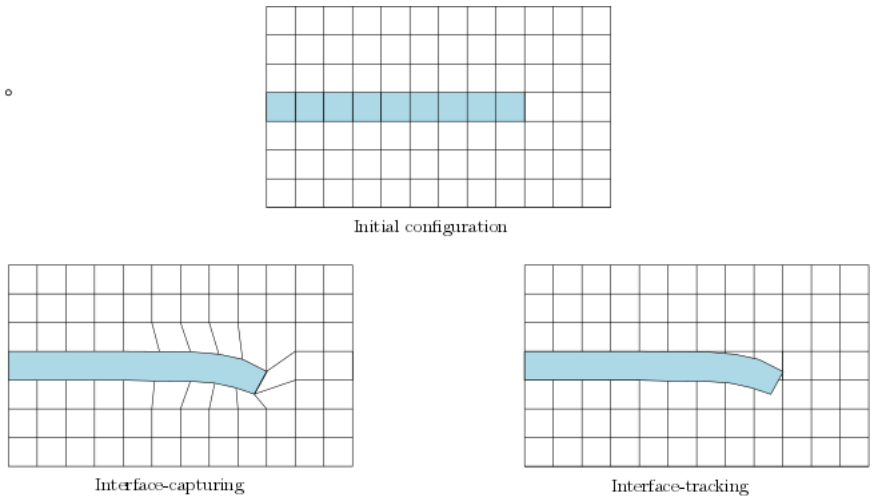

Figure 3.1: Comparison of interface-tracking and interface-capturing for an elastic beam undergoing deformation

3.1 Arbitary Lagrangian Eulerian formulation

The *arbitary Lagrangian-Eulerian* formulation is the most popular approach within *Interface-tracking* [**?**], [**?**]. 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 underformed *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-conctact with other structure parts [[22], [15]]. Since no natural displacement occur in the fluid domain, the transformation $\hat{\mathbf{T}}_f(t)$ has no directly physical meaning [**?**], [2]. Therefore, the construction of the transformation $\hat{\mathbf{T}}_f(t)$ is a purely numerical exercise.

3.1.1 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 underformed *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 [15], with help of a new arbitrary fixed reference system $\hat{\mathrm{W}}$

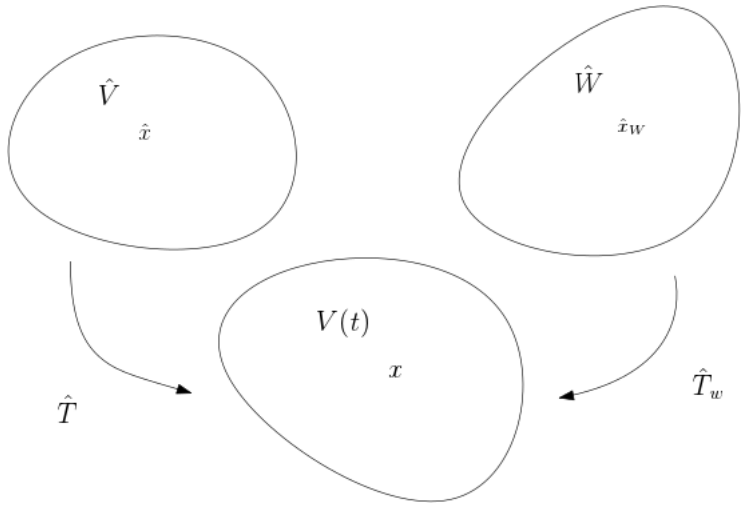

Figure 3.2: CFD-3, flow visualization of velocity time t = 9s

. Let $\hat{\mathrm{T}}_w : \hat{\mathrm{W}} \rightarrow V(t)$ be an invertable 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 $V(t)$,

Lemma 3.1. Local change of volume
Let V(t) be the reference configuration $V(t) \rightarrow \mathbb{R}^d$, *and* $\hat{W} \rightarrow \mathbb{R}^d$ *be the arbitary reference configuration. By the determinant of the deformation gradient* $\hat{J}_w$, *the the following relations holds,*

$$|V(t)| = \int_{\hat{W}} \hat{J}_w d\hat{x} \tag{3.3}$$

Lemma 3.2. Transformation of scalar spatial derivatives
Let f be a scalar function such that $f : V(t) \rightarrow \mathbb{R}$, *and* ∇f *be its gradient. Then its counterpart* $\nabla \hat{f}$, *by the scalar function* $\hat{f}\hat{W} \rightarrow \mathbb{R}$ *is given by the relation.*

$$\nabla f = \hat{F}_W^{-T} \hat{\nabla} \hat{\mathrm{f}} \tag{3.4}$$

Lemma 3.3. *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.*

$$\nabla\mathbf{w} = \hat{\nabla}\hat{\mathbf{w}}\hat{F}_W^{-1} \tag{3.5}$$

Lemma 3.4. 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,*

$$\frac{\partial f}{\partial t} = \frac{\partial \hat{\mathrm{f}}}{\partial t} - (\hat{F}_W^{-1}\frac{\partial \hat{\mathrm{T}}_W}{\partial t} \cdot \hat{\nabla})\hat{\mathrm{f}} \tag{3.6}$$

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,

$$\frac{d\mathbf{v}}{\partial t} = \frac{\partial \hat{\mathbf{v}}}{\partial t} - (\hat{F}_W^{-1}\frac{\partial \hat{\mathrm{T}}_W}{\partial t} \cdot \hat{\nabla})\hat{\mathbf{v}} \tag{3.7}$$

$$\mathbf{v} \cdot \nabla\mathbf{v} = \nabla\mathbf{v}\mathbf{v} = \hat{\nabla}\hat{\mathbf{v}}\hat{F}_W^{-1}\hat{\mathbf{v}} = (\hat{F}_W^{-1}\hat{\mathbf{v}} \cdot \hat{\nabla})\hat{\mathbf{v}} \tag{3.8}$$

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla\mathbf{v} = \frac{\partial \hat{\mathbf{v}}}{\partial t}(x,t) - (\hat{F}_W^{-1}\frac{\partial \hat{\mathrm{T}}_W}{\partial t} \cdot \hat{\nabla})\hat{\mathbf{v}} + (\hat{F}_W^{-1}\hat{\mathbf{v}} \cdot \hat{\nabla})\hat{\mathbf{v}} \tag{3.9}$$

$$= \frac{\partial \hat{\mathbf{v}}}{\partial t} + (\hat{F}_W^{-1}(\hat{\mathbf{v}} - \frac{\partial \hat{\mathrm{T}}_W}{\partial t}) \cdot \hat{\nabla})\hat{\mathbf{v}} \tag{3.10}$$

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 [15] we have the following relation,

$$\nabla \cdot \sigma = \nabla \cdot (\hat{\mathrm{J}}_W\hat{\sigma}\hat{F}_W^{-T}) \tag{3.11}$$

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* [15]such that

$$\nabla \cdot \mathbf{v} = \nabla \cdot (\hat{\mathrm{J}} \hat{F}_W^{-1} \hat{\mathbf{v}}) \tag{3.12}$$

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 invertable 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 3.1.1. ALE fluid problem
Let $\hat{\mathbf{v}}_f$ be the fluid velocity, ρ_f the fluid density, and ν_f the fluid viscosity.

$$\hat{\mathrm{J}}_f \frac{\partial \hat{\mathbf{v}}}{\partial t} + \hat{\mathrm{J}}_f (\hat{F}_f^{-1} (\hat{\mathbf{v}} - \frac{\partial \hat{\mathrm{T}}_W}{\partial t}) \cdot \hat{\nabla}) \hat{\mathbf{v}} = \nabla \cdot (\hat{\mathrm{J}}_W \hat{\sigma} \hat{F}_W^{-T}) + \rho_f \hat{\mathrm{J}} \mathbf{f}_f \quad \text{in } \hat{\Omega}_f \tag{3.13}$$

$$\nabla \cdot (\hat{\mathrm{J}} \hat{F}_W^{-1} \hat{v}) = 0 \quad \text{in } \hat{\Omega}_f \tag{3.14}$$

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

Due to the arbitary 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 [15]. The Lagrangian particle tracking $x \in \hat{\Omega}_f$, the Eulerian tracking $x \in \Omega_f$, and the arbitary tracking of the reference domain $x \in \hat{W}$ [15]. This concept can be further clairified by the introduction of *material* and *spatial* points.

3.1.2 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 3.1.2. *ALE solid problem*

$$\rho_s \frac{\partial \hat{\mathbf{v}}_s}{\partial t} = \nabla \cdot \mathbf{FS} + \rho_s \mathbf{f}_s \quad \text{in } \Omega_s \tag{3.15}$$

$$\tag{3.16}$$

3.1.3 Fluid mesh movement

Let the total domain deformation $\hat{\mathrm{T}}(\hat{\mathrm{x}}, t)$ be divided into the solid $\hat{T}_s : \hat{\Omega}_s \to \Omega_s$, and fluid deformation $\hat{T}_f : \hat{\Omega}_f \to \Omega_f$. 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 arbitary, however the interface shared by both the fluid and solid domain, require an accuracte transformation of the interface points by $\hat{\mathrm{T}}_f$ [15],

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

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 [14]. The fluid domain deformation $\hat{\mathbf{u}}_f$ must therefore fulfill the boundary conditions

$$\hat{\mathbf{u}}_f(\hat{x}) = \hat{\mathbf{u}}_s \quad \hat{x} \in \hat{\Omega}_f \cup \hat{\Omega}_s \tag{3.17}$$

$$\hat{\mathbf{u}}_f(\hat{x}) \cdot \hat{\mathbf{n}} = 0 \quad \hat{x} \in \partial\hat{\Omega}_f \neq \hat{\Omega}_f \cup \hat{\Omega}_s \tag{3.18}$$

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 motion model*.

3.1.4 Mesh motion models

In the ALE framwork 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 nesecarry to avoid mesh entaglement [18]. Consequently, the choice of mesh moving technique is essential to generate a smooth evolution of the fluid mesh.Several mesh models have been proposed throughout the litterature, and for an overview the reader is refered to [12], 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 [21]. 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 ([9], [3]).
A limitation of the 2nd order mesh models is that by Dirichlet and Neumann boundary conditions, only mesh position or normal spacing spacing can be specified respectfully, but not both [8]. This limitation is overcome by 4th order biharmonic mesh model, since two boundary conditions can be specified at each boundary of

the fluid domain [8]. 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 steps needed for convergence at each time-step [21], 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 model

Equation 3.1.3. 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,

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

The

Most favourable, the largest mesh deformation occuring should be confined to the interal part of the mesh as it causes the least distortion [10]. Therefore the introduced diffusion parameter α, often raised to some power q, is introduced to manipulate this behaviour. The form of this paramter is often problem specific, as selective treatment of the elements may vary from different mesh deformation problems. A jacobian based method was introduced in [16]. In [10], the authors reviewed several distance based options, where α was some function of the distance to the closest moving boundary. This method was adopted in this thesis on the form

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

However as pointed out by [9], one of the main disadvantages of using the linear Laplace equation is that the equation solves the mesh deformation components independently of one another. Say one have deformation only in the x-coordinate direction, the interior mesh points will only be moved along this deformation. Such a behavior restricts the use to the Laplace equation of mesh extrapolation purposes.

Mesh motion by a Linear elastic model

Considering a linear elastic model for mesh moving was first introduced in [17]. Both [3]

$$\begin{aligned} \nabla \cdot \sigma &= 0 \\ \sigma &= \lambda Tr(\epsilon(u))I + 2\mu\epsilon(u) \\ \epsilon(u) &= \frac{1}{2}(\nabla u + \nabla u^T) \end{aligned}$$

Where Lamé constants λ and ν are given as

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

One of the main motivations for introducing such a model is the manipulation of Young's modulus E, and the poisson´s ration ν. Recall that Young's modulus is the measurement of the a materials stiffness, while the poission's ratio describe the materials stretching in the transverse direction under extension in the axial direction. Manipulating these parameters one can influence the mesh deformation, however the choice of these parameters have proven not to be consistent, and to be dependent of the given problem.

In [19] the author proposed a negative possion ratio, which makes the model mimic an auxetic material. Such materials becomes thinner in the perpendicular direction when they are submitted to compression, and this property is feasible for mesh under deformation.

One of the most common approach is to set ν as a constant in the range $\nu \in [0, 0.5)$ and let E be the inverse of the distance of an interior node to the nearest boundary surface [12]. The authors of [1] used this property and also argued that the Young's modulus also could be chosen as the inversely proportional to the cell volume. They also pointed out that both approaches would give the desired result that the small cells around the solid surface would modeled rigid, moving with the surface of the solid as it undergoes deformation. On the other hand cells further away will deform to counter the effects close to the solid surface.

Mesh motion by a Biharmonic model

Using a biharmonic mesh deformation model provides further freedom in terms of boundary conditions, and the reader is encoured to consult [8] for a deeper review. We will in combination with [19] present two main approaches the biharmonic model is defined as

$$\hat{\nabla}^2 \hat{\mathrm{u}} = 0 \quad \text{on } \hat{\Omega}_f$$

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

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

This model is defined in a mixed formulation, and as such the prize for quality and control of mesh deformation comes with the cost of more computational demanding problem.

For the boundary conditions two types has been proposed in [19]. 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

$$\textbf{Type 1} \quad \hat{\mathrm{u}}_f^{(k)} = \frac{\partial \hat{\mathrm{u}}_f^{(k)}}{\partial n} = 0 \quad \partial\hat{\Omega}_f/\Gamma \text{ for } k = 1, 2$$

$$\textbf{Type 2} \quad \hat{\mathrm{u}}_f^{(1)} = \frac{\partial \hat{\mathrm{u}}_f^{(1)}}{\partial n} = 0 \text{ and } \hat{\mathrm{w}}_f^{(1)} = \frac{\partial \hat{\mathrm{w}}_f^{(1)}}{\partial n} = 0 \quad \text{on } \hat{\Omega}_f^{in} \cup \hat{\Omega}_f^{out}$$

$$\hat{\mathrm{u}}_f^{(2)} = \frac{\partial \hat{\mathrm{u}}_f^{(2)}}{\partial n} = 0 \text{ and } \hat{\mathrm{w}}_f^{(2)} = \frac{\partial \hat{\mathrm{w}}_f^{(2)}}{\partial n} = 0 \quad \text{on } \hat{\Omega}_f^{wall}$$

With the first type of boundary condition the model can interpreted as the bending of a thin plate, clamped along its boundaries. 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 [11].

The main motivation for second type of boundary condition is for a rectangular domain where the coordinate axes match the Cartesian coordinate system [19]. In such a configuration, the mesh movement is only constrained in the perpendicular direction of the fluid boundary, leading to mesh movement in the tangential direction. This special case reduces the effect of distortion of the cells.

3.2 Discretization of the FSI problem

Say something general of FSI discretization.. FEM, FVM, ... In this thesis, the finite element method will be used to discretize the coupled fluid-structure interaction problem. It is beyound 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 nesecarry in order to define a foundation for problem solving will be introduced.

3.2.1 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-dimentional 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 unless specified. The outer boundary is set by $\partial\hat{\Omega}$, with $\partial\hat{\Omega}^D$ and $\partial\hat{\Omega}^N$ as the Dirichelt 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 appropiate 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. This reduces the differential equation of interest down to a system of linear equations. (skriv bedre..)
The velocity variable is continous through the solid and fluid domain

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

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

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

For simplification of notation the inner product is defined as

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

3.2.2 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 laplacian and linear elastic model will be shorted such that

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

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 3.1. *Coupled fluid structure interaction problem for laplace and elastic mesh moving model. Find* $\hat{\mathbf{u}}_s, \hat{\mathbf{u}}_f, \hat{\mathbf{v}}_s, \hat{\mathbf{v}}_f, \hat{p}_f$ *such that*

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

Problem 3.2. *Coupled fluid structure interaction problem for biharmonic mesh moving model. Find* $\hat{\mathbf{u}}_s, \hat{\mathbf{u}}_f, \hat{\mathbf{v}}_s, \hat{\mathbf{v}}_f, \hat{p}_f$ *such that*

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

for the first type of boundary conditions introduced.

Both problems introduced must handle the *kinematic* and *dynamic* boundary conditions in a consistent way. By a continuous velocity field on the whole domain,

the *kinematic* condition is strongly enforces on the interface $\hat{\Gamma}_i$ The continuity of normal stresses on the interface are defined as

$$\left(\hat{\mathrm{J}}_W \hat{\sigma} \hat{F}_W^{-T} \hat{\mathbf{n}}_f,\ \hat{\boldsymbol{\psi}}^u\right)_{\hat{\Omega}_f} = \left(\hat{\mathbf{F}}\hat{\mathbf{S}}\hat{\mathbf{n}}_s,\ \hat{\boldsymbol{\psi}}^u\right)_{\hat{\Omega}_s}$$

This condition is weakly imposed by omitting the boundary integral from the variational formulation [20], and it becomes an implicit condition for the system.

Chapter 4

Verification and Validation

Computer simulations are in many engineering applications a cost-efficient way for conducting design and performance optimalization of physical problems. However, thrusting blindly numbers generated from a computer code can prove to be naive. It doesn't take a lot of coding experience before one realizes the many things that can brake down and produce unwanted or unexpected results. Therefore, *credability* of computational results are essential, meaning the simulation is worthy of belief or confidence [**?**]. *Verification and validation* (VV) is the main approach for assessing and the reliability of computational simulations [**?**]. The terminology of (VV) have proven unconsistent across differnt engineering disciplines due to the variety of views regarding the fundaments of the method. A thorough review considering the development of (VV) concepts and terminology during the last century are studied in [**?**], where several attempts of definitions of *verification* and *validation* by different scientific communities are considered. In this thesis, the definitions provided by the *American Society of Mechanical Engineers guide for Verification and Validation in Computational Solid Mechanics* [**?**] are followed.

Definition 4.1. Verification: The process of determining that a computational model accurately represents the underlying mathematical model and its solution.

Definition 4.2. 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. [**?**]

To test a computational code for all possible parameters, conditions and applications are simply too time consuming. Verification and validation are therefore ongoing processes, with no clear boundary of completeness unless additional requirements are specified [**?**].

The goal of this chapter is to verify our implementations using the method of manufactured solution (MMS).

4.1 Verification of Code

Verification can be devided into *verification of code* and *verification of calculation* [[?] [?]

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 (PDE´s). A computer program therefore must evaluate mathematical identities such a differential operators and functions in order to produce accurate solutions of the governing PDE´s. Through verification of code, the ultimate goal is to ensure a computer program truly represents the mathematical model, forming a basis for the given problem.
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. Different acceptance criteria with increasing rigor are found in [?].

- Simple tests
- Code-to-code comparisons
- Discretization error quantification
- Convergence test
- Order-of-accuracy tests

The two first criteria have the advantage of complying in the absence of exact solutions, however not as rigorous as the three final criteria. *Simple tests* are applied directly on numerical solution, for example evaluating if the code preservers physical properties such as conservation laws. *Code-to-code comparisons* involves comparing the numerical solution of the code to another "reference code". However, the method proves useful only if the same mathematical models are used and the reference code have undergone rigorous verification [?].
The final three criteria *Discretization error quantification*, *Convergence test*, and *Order-of-accuracy tests* are all related to the discretization error E, defined as,

$$E = u_e - u_h$$

where u_e is the exact solution and u_h is the numerical solution at a given mesh and time step. Hence, an exact solution for the given problem is necessarily in order to evaluate the accuracy of the discretization of the mathematical model.

Discretization error quantification evaluates the quantitative error between the numerical solution and the exact solution, for a certain mesh resolution Δx and time step Δt. Even though error quantification surpass the previous criteria presented, the approache pose two major difficulties. First, what degree of spatial and temporal resolution are needed in order to reduce the discretization error E. Second, is

the subjective assessment of how small the error should be in order to give the code credability [**?**]. Since the previous criteria is, to some degree, bounded by subjective considerations, it lacks independence from the person conducting verification.

Convergence tests assumes the discretization of a PDE to be consistent, if the spatial ∇x and/or temporal ∇x refinement decreases to zero, etc $\nabla x, \nabla t \rightarrow 0.$, so does the discretization error $E \rightarrow 0$ [**?**]. The method is more consistent in comparison with discretization error quantification critera, as the choice of discretization parameters $\nabla x, \nabla t$ are not limited to one particular resolution. Even though the choice of temporal and spatial refinements are subjective, a reduction of the discretization error is expected for increasing temporal and spatial refinement. The method is considered the minimum criterion for rigorous code verification [**?**].
The final critera *Order-of-accuracy* is regarded as the most rigorous acceptance criterion for verification [**?**], [**?**], [**?**], which is employed in this thesis. In addition to error estimation and convergence of the numerical solution, the method ensure the discretization error E is reduced in coordinance 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 our code. The order of convergence is calculated xAssuming a PDE of space and time, order-of-accuracy tests are conducted separatly of

By monitoring the dicretization error E by spatial and temporal refinements, one assumes the asymptotic behavior,

$$E = E_x + E_t = u_e - u_h = C\ hx^p$$

where C is a constant, h represents the spatial or temporal resolution, and p is the convergence rate of the numerical sche me. The error E is assumed to be a sum of the discretization error of space and time.
Refinement is the act of solving the mathematical model for a set of discretization parameters h, such that $h_1 < h_2 < h_3..h_n \rightarrow 0$. Assuming a PDE discretized in space and time, the paramters takes the form (Δx, Δt). When conducting *order-of-accuracy* tests, the convergence rate p is evaluated for temporal and spatial refinements separatly. Therefore, if one consideres spatial refinement study $\Delta x \rightarrow 0$, one must choose Δt small such that the temporal discretization error can be exluded from the total error E. For convergence tests, the code is assumed verified and consistent if the discretization error is proportional to h^p

4.1.1 Method of manufactured solution

For conducting verification of code based on convergence tests, an exact solution of the mathematical model or PDE is needed in order to evaluate the discretization error E. However accurate solutions of PDE´s are limited, and often simplifications of the original problem are needed in order to produce analytically solutions for comparison. *The method of manufactured solutions* provides a simple yet robust

way of making analytic solutions for PDE´s. The problem is not attacked in a traditional way, by finding an analytic solution by retaining the overall physics defined by the model.

Let a 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.

In the method of manufactured solution one first manufactures a solution $\mathbf{u}$ for the given problem. In general, the choice of $\mathbf{u}$ will not satisfy the governing equations, producing a sourceterm $\mathbf{f}$ after differentiation by $\mathbf{L}$. The produced source term will cancel any imbalance formed by the manufactured solution $\mathbf{u}$ of the original problem. Therefore, the manufactured solution can be constructed without any physical reasoning, proving code verificaion as a purely a mathematical exercise were we are only interested if we are solving our equation right [**?**]. Even though the manufactured solution $\mathbf{u}$ can be chosen independently, some guidelines have been proposed for rigirous verification([**?**], [**?**], [**?**]).

- The manufactured solution (MS), should be composed of smooth analytic functions such as exponential, trigonometric or polynomials.
- The MS should exercise all terms and derivatives of the PDE´s.
- The MS should have sufficient number of derivatives

The guidelines presented are not limitation for choosing a MS, but rather improvements for ensuring the representation of the mathematical model is thoroughly tested.

Order-of-accuracy test or *order-of-convergence test* is the most rigorous code I wrote this to check if this works

To deeply verify the robustness of the method of manufactured solution, a report regarding code verification by MMS for CFD was published by Salari and Knupp [**?**]. This thorough work applied the method for both compressible and incompressible time-dependent Navier-Stokes equation. To prove its robustness the authors deliberate implemented code errors in a verified Navier-Stokes solver by MMS presented in the report. In total 21 blind testcases where implemented, where different approaches of verification frameworks were tested. Of these, 10 coding mistakes that reduces the observed order-of-accuracy was implemented. Here the method of manufactured solution captured all of them.

For construction of the sourceterm $\mathbf{f}$ the Unified Form Language (UFL) [**?**] provided in FEniCS Project will be used. UFL provides a simple yet powerfull method of declaration for finite element forms. An example will be provided in the Fluid Problem section.

4.1.2 Verification of the fluid-structure interaction solver by MMS

In the last section, the choice MMS for verification of code is stressed to not be limited by the overall physics of the given problem of interest, which is true considering the fluid and solid problems separate. The freedom of choosing a MMS for a monolithic fluid-structure interaction problem is however limited to some extent, due coupling coditions at the interface. Recall from chapter 4,

1. Kinematic boundary condition $\hat{\mathbf{u}}_s = \hat{\mathbf{u}}_f]$, enforced strongly by a continious velocity field in the fluid and solid domain.

2. Dynamic boundary condition $\sigma_s \cdot \mathbf{n} = \sigma_f \cdot \mathbf{n}$, enforced weakly by omitting the boundary integrals from the weak formulation in problem ?.

The construction of a MMS is therefore not obvious, as it must fulfill condition 1 and 2, in addition to the divergence-free condition in the fluid. The struggle is reflected of the abscence of research, regarding MMS for coupled FSI solvers in the litterature. The challenge are often disregarded, such as cite METHOLOGY for COMPARING.., presenting a thorough methodology for comparing fluid-structure interaction solvers based on the verification and validation framework. The verification process is conducted on the fluid and structure solver separatly, taking no consideration to the coupling. Instead, the correctness of the coupling is evaluated by the code validation.

The approach clearly ease the process, assuming verification of each codeblock is "sufficient" to declare the code verified. It must be stressed that solving each problem individually is not true verification, in reference to a monolithic approach where the problems are solved at the same time.
The construction of a MMS for a monolithic FSI problem is therefore out of the scope of this thesis. Conducting verification on the fluid and structure separatly is not , but considered "good enough" to show the mathematical model is discretized accuratly.

4.2 Validation

Through *verification*, one can assure that a scientific code implements a mathematical model correctly. However, c orrectness is unnecessarly, if the model fails to serve as an accurate representation of the physical problem of interest. By definition 1.2, *Validation* is the act of demonstrating that a mathematical model is applicable for its intended use with a certain degree of accuracy. This demonstration is not intended to portray the model as an absolute truth, nor the best model available [**?**]. The acceptence criteria of validation is based on the numerical solution produced, by comparison with existing experiment data. The dependency of thrusting experiments, makes *validation* assess a wide range of issuses [**?**]

- The relevance of mathematical model compared to the experiment.

- Assurence of that the experiments was conducted correctly, in accordinance with prescribed parameters, initial and boundary conditions e.t
- Uncertainty of experimental measurements

Comparing numerical results with existing experiments,raise some issues in the validation process. First, reproducing of experimental resuts...

Chapter 5

Implementation of a Fluid-Structure Interaction solver

This chapter will focus on the implementation of a fluid-structure interaction solver, with emphasis on general implentation issues.

5.1 Black-box solvers versus a self-made implementation

A black box is any device whose workings are not understood by or accessible to its user. In terms of computational science, the term is often associated with commercially available code where the user is limited to the control of input parameters, with no intervention of the code itself. Trusting commercial software blindly is risky. Even though the software has been rigorous tested, the lack of full transparancy of the implementation ...(full understanding og math). In addition a full understanding of the software also takes time to learn.

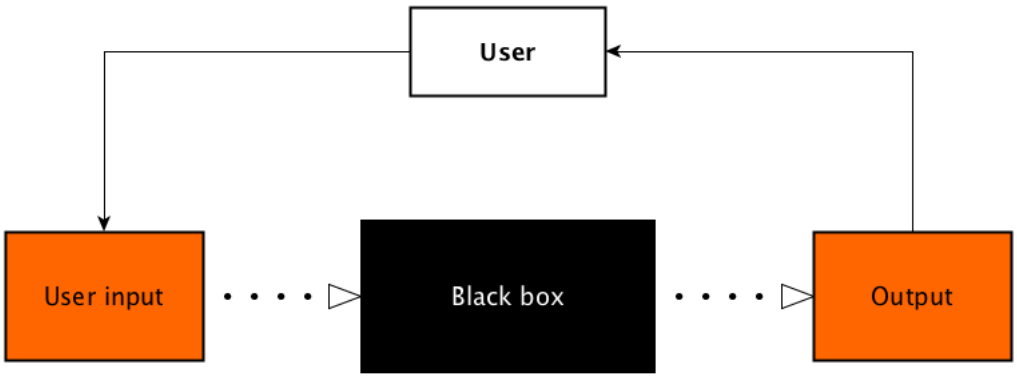

Figure 5.1: Computational domain of the validation benchmark

Several commercial and available solvers for FSI exists, both for monolithic and partitioned approaches. The commerical software ADINA makes solvers based on

both approaches available for the user [**?**]. The open-source OpenFOAM also incoporate both solvers, howevere enabling more flexibillity in combining different solution strategies ...

The disiplines of CFD and computational structural dynamics (CSD) have tradidionally been treated separatly. Hence, highly optimized and specialized software has been developed individually, with little or no regard to multiphysics such as fluid-structure interaction. Using a partitioned approach, the MpCCI interface is considered the "de-facto" software, for coupling separate fluid and solid solvers [[**?**], [**?**]].

In this thesis a monolithic FSI-solver is developed without the use of commercial available software. The main motivation for this choice is the full understanding of the implementation, and the ability for code intervention along the solution process. The process of implementing a self-made solver for fluid-structure interaction is not stright forward. Due to the overall size of the total system, and the assumption of a non-linear material and large deformations makes the process..

The implementation of the FSI-solver relies soley on the open-source project FEniCS, a computing platform for solving partial differential equation. The next section will present the general consepts for this..

5.2 FEniCS

The FEniCS project is an open-source finite element environment for solving partial differential equations (https://fenicsproject.org/). Using a combination of high-level Python and C++ interfaces, mathematical models can be implemented compactly and efficiently. FEniCS consists of several sub-modules and we will give a brief overview of the most central components used during implementation and computation.

5.2.1 DOLFIN

DOLFIN is the computational C++ backend of the FEniCS project, and the main user interface. It unifies several FEniCs components for implementing of computational mesh, function spaces, functions and finite element assembly.

- UFL (The Unified Form Language) is a domain specific language, used for the discretization of mathematical abstractions of partial differential equations on a finite element form. Its implementation on top of Python, makes it excellent to define problems close to their mathematical notation without the use of more complex features. One uses the term *form* to define any representation of some mathematical problem defined by UFL.

- FFC (The form compiler) compiles the finite elements variation forms given by UFL, generating low-level efficient C++ code

- FIAT the finite element backend, covering a wide range of finite element basis functions used in the discretization of of the the finite-element forms. It covers a wide range of finite element basis functions for lines, triangles and tetrahedras.

DOLFIN also incorporate the necessary interfaces to external linear algebra solvers and data structures. Within FEniCS terminology these are called linear algebra backends. PETSc is the default setting in FEniCS, a powerful linear algebra library with a wide range of parallel linear and nonlinear solvers and efficient as matrix and vector operations for applications written in C, C++, Fortran and Python.
A comprehensive introduction FEniCS is out of the scope for this thesis, and for further insight the reader is directed to the introductional demos presented on https://fenicsproject.org/.

5.2.2 Poission Equation, hello world

When learning programming, it is common to present a "Hello world!" program, presenting the fundamental concepts of the software of interest. For solving PDE's, one of the most basic challenges is solving the Possion's equation. Let Ω be the computational domain and let $\partial\Omega$, be the boundary of Ω. The Poission equation takes the form,

$$\begin{aligned} -\nabla^2 u &= f \quad \in \Omega \\ u &= u_d \quad \in \partial\Omega \end{aligned}$$

where u is the unknown of interest and f is some known function. ∇^2 is the Laplace operator and $u = u_d$ is a prescribed Dirichlet condition on the boundary. Assuming the reader has some knowledge of the finite-element method, the unknown function u is approximated by a *trial function*. The problem is then multiplied with a *test function* v, and then we use integration by-parts over the domain Ω.

$$\begin{aligned} -\int_\Omega \nabla^2 u v dx &= \int_\Omega f dx \\ -\int_{\partial\Omega} \frac{\partial u}{\partial n} v ds + \int_\Omega \nabla u \nabla v dx &= \int_\Omega f dx \end{aligned}$$

For simplicity, the *test function* $v = 0$ on the boundary $\partial\Omega$ due to the prescribed Dirichlet condition which leaves us with the system

$$\int_\Omega \nabla u \nabla v dx = \int_\Omega f dx$$

With the primilaries set, we focus on the implementation of the probelem in FEniCS. The poission problem can then be expressed,

```
from fenics import *

frame#frame frameCreateframe framemeshframe frameandframe framedefineframe framea
    frame frame1.frameorderframe framelagrangianframe framefunctionframe framespace
mesh = UnitSquareMesh(10, 10)
V = FunctionSpace(mesh, 'CG', 1)

frame#frame frameDefineframe framedirichletframe frameboundaryframe framecondition
u_bnd = Expression('1 + x[0]*x[0] + 2*x[1]*x[1]', degree=2)

bcs = DirichletBC(V, u_bnd, "on_boundary")

Define variational problem
u = TrialFunction(V)
v = TestFunction(V)
f = Constant(-10.0)
a = dot(grad(u), grad(v))*dx
L = f*v*dx

frame#frame frameSolveframe frameproblemframe frameandframe framecomputeframe
    framesolution
u = Function(V)
solve(a == L, u, bc)
```

Algorithm 5.1: posssion.py

The possion problem expressed with FEniCS points out two important properties. Frist, the overall problem is implemented compactly while each code segement for solving the problem remains clear. Second, the abstract notation remains close to the orignal problem of interest.

5.3 implemented code

The overall codestructure of the implementation is based on dividing the full code into fragments of Python modules. The main idea is to keep key code segments functionality while maintaining modularity to each code segment.

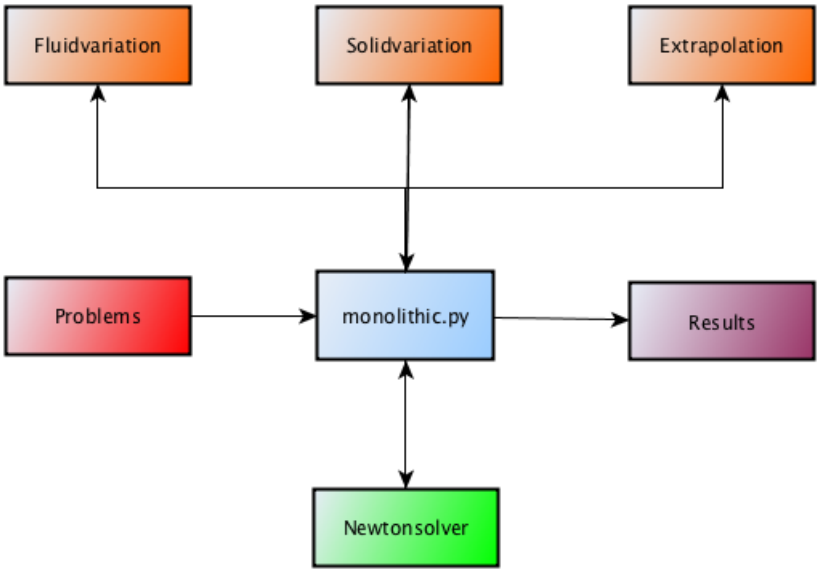

Figure 5.2: Computational domain of the validation benchmark

Chapter 6

Time discretization and optimalization

The aim of this chapter is to present some of the main challenges regarding discretization of a general monolithic fluid-structure interaction(FSI) problem, using the ALE-framework. Even separately, the discretization of fluid and structure problems impose rather difficult issues due to their non-linear nature. However, their long-time existence within research community makes them well known problems and a vast number of rigorous approaches and commercial software exist to solve them individually. When solving the fluid and structure simultaneously however, the overall problem gets more complex due to the overall dependency of the two sub-problems and their interaction to one another.

One of the main challenges is the additional non-linearty intruduced by the domain-velocity term in the fluid problem.

Problem 6.1. *ALE term*

$$\hat{\mathrm{J}}(\hat{F}_W^{-1}(\hat{\mathbf{v}} - \frac{\partial \hat{\Gamma}_W}{\partial t}) \cdot \hat{\nabla})\hat{\mathbf{v}}$$

Closer inspection of the convection term reviels spatial and temporal differential operators depending non-linearly on one another. Within computational science, these operators often appear separated. Therefore the discretization of a general time-stepping scheme is not directly intuitive, and often based on the experience of similar equations such as the Navier-Stokes equations. In this thesis, time-stepping schemes of second order will be considered. It has been reported in [] [], that the stability of first and second-order time stepping schemes are affected by the ALE-convection term, but to what extent remains unclear.
Though only the fluid problem will be discussed, it must be emphasized that the discretization of the solid problem is of great importance. Several studies exists for the individual solid problem, but a deeper analysis considering a fluid-structure interaction setting is abvient from the FSI litterature [**?**].

6.1 Implementation of a one-step θ scheme

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

Problem 6.2.

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

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 [**?**], a general time stepping algorithm for the coupled Navier-Stokes equation can be written as

Problem 6.3.

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

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 unessecary. A general mid-point rule by $\alpha = \vartheta = \frac{1}{2}$, a second order scheme in time would easily be acchieved. However, in [**?**] 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 litterature of fluid-structure interaction. One reason for this may be that the ALE fluid problem will remain non-linear due to the ALE-mapping.

For the structure problem, the Crank-Nicholson is of main interest due to energy preservation properties and second order convergence.

In light of By letting $\alpha = \vartheta \;\; \alpha, \vartheta \in [0,1]$ for the fluid problem, and generalising the consepts in an ALE context, we derive the one-stepl θ scheme found in [19].

Problem 6.4. *One-step θ-scheme for laplace and elastic mesh moving model. Find* $\hat{\mathbf{u}}_s, \hat{\mathbf{u}}_f, \hat{\mathbf{v}}_s, \hat{\mathbf{v}}_f, \hat{p}_f$ *such that*

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

Deeper analysis in [19], specify to important properties of the one-step *theta* scheme. Firstly, it is unconditionally stable regardless of time step for the interval $\theta = [\frac{1}{2}, 1]$.

Chapter 7

Numerical Results

In this chapter the numerical results for this thesis will be presented. The first and section chapters conserns the verification and validation of the *One-step* θ scheme respectivly. In the third and final chapter, different speed-up strategies are presented and compared

7.1 Verification

7.1.1 Fluid Problem

One question which arises during the construction of the manufactured solution is, which formulation of the Navier-Stokes equation do we want to calculate the sourceterm. From a numerical point of view constructing the sourceterm from the Eulerian formulation and then map the equation would be feasable. Such an apporach limits the evaluation of computational demanding routines such as the generation of the deformation gradient $\hat{F}$ and its Jacobian $\hat{\mathrm{J}}$. Even though refinement studies of spatial and temporal discretizations are often computed on small problems, such speed-ups are important when running larger simulations. Recall from Chapter ??? the ALE formulation of the Navier Stokes equation.

$$\rho_f \hat{\mathrm{J}} \frac{\partial \hat{u}}{\partial t} + \hat{\mathrm{J}}\hat{F}^{-1}(\hat{u} - \hat{w}) \cdot \nabla \hat{u} - \nabla \cdot \hat{\mathrm{J}}\sigma\hat{F}^{-T} = f$$
$$\hat{\mathrm{d}}\mathrm{iv}(\hat{\mathrm{J}}\hat{F}^{-1}\hat{\mathrm{u}}) = 0$$

Algorithm 7.1: Descriptive Caption Text

```
u_x = "cos(x[0])*sin(x[1])*cos(t_)"
u_y = "-sin(x[0])*cos(x[1])*cos(t_)"
p_c = "sin(x[0])*cos(x[1])*cos(t_)"

f = rho*diff(u_vec, t_) + rho*dot(grad(u_vec), (u_vec - w_vec)) -
div(sigma_f(p_c, u_vec, mu))
```

We will on the basis of the presented guidelines define the manufactured solution.

$$u = sin(x + y + t)^2$$
$$v = cos(x + y + t)^2$$
$$p = cos(x + y + t)$$

7.2 Validation of a One-step θ scheme

The numerical benchmark presented in [**?**] has been chosen for validation of the *One-step* θ scheme presented in chapter. The benchmark has been widely accepted throughout the fluid-structure interaction community as a rigid validation benchmark. This is mainly due to the diversity of tests included, challenging all the main components of a fluid-structure interaction scheme.

The computational domain is based on the *von Kármán vortex street* se (cite), where a cylinder is intentionally placed off center in a pipe. This configuration initiates a periodic shedding of vortices, as some fluid moves past the cylinder. In [**?**], an elastic flag is placed behind the cylinder.

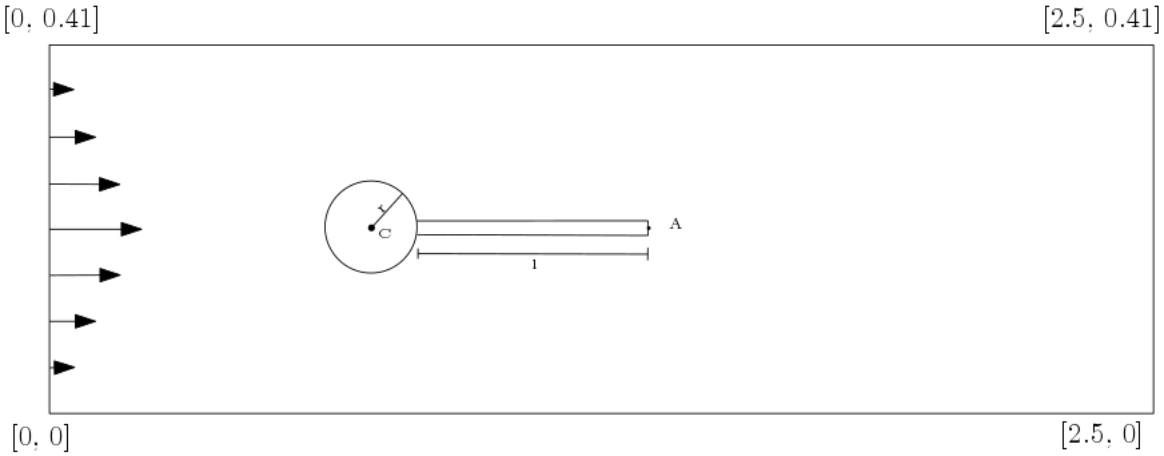

Figure 7.1: Computational domain of the validation benchmark

The benchmark is divided into three main test environments. In the first environment the fluid solver is tested for a series of different flow profiles.
The second environment regards the structure implementation, regarding bending of the elastic flag. The third environment the full fluid-structure interaction problem. The test environments are further divided into three different problems with increasing difficulty, posing different challenges to the implementation.

Several quantites for comparion are presented in [**?**] for validation purposes. The evaluation of these quantities are considered for fully developed flow,

- 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.

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

The following environments and their sub-problems presents both steady state and periodic solutions. For the steady state solutions, the quantity of interest will be calculated for the last timestep. For the periodic solutions, the amplitude and mean values for the time dependent quantity are calculated from the last period of oscillations.The mean value and amplitude is given by,

$$\text{mean} = \frac{1}{2}\text{max} + \text{min}$$
$$\text{amplitude} = \frac{1}{2}\text{max} \ . \ \text{min}$$

from the maximum and minimum value of the quantity of interest from the last period.

In [**?**], all steady state solutions seems to be calulated by solving a steady state equation such as the stokes equation for the fluid problem. The assumtion is based on simulation paramters regarding time-step are only reported for the periodic solutions. In this thesis all problems in [**?**],will be calculated by time integration. The main motivation is based upon that any given numerical errors regarding time integration will be intercepted at an earlier stage for a simpler problem. Therefore, the choice of timestep is chosen such that reasonable accuracy of the reference solution is attained.

In [**?**], details such as finite-element spaces and newton iteration critera are not reported. Therefore, the following numerical results have been a process of trial and error. In the following section, an overview of each environment togheter with the numerical results will be presented. A formal discussion of the results are given at the end of each simulation environment. For each table, the error of the finest spatial and temporal refinement compared to the reference solution is reported .

Since the first two simulation environmens are presented mainly in support of the third and final environment, they where not reported in OTHER CITE. Therefore results in the first two subsections will be compared with [**?**], while the third will consider both [**?**] and OTHER CITE.

7.2.1 Validation of fluid solver

The first test environment conserns the fluid dynamics part of the total FSI problem, to ensure the solver can handle flows in low Reynold-numbers regime. Two approaches for the validation are given in [**?**]. The first approach consideres setup as a fluid-structure interaction problem, by setting the elastic flag close to rigid by manipulation of the structure paramters. Second, the flag can be set fully rigid and considered a purly flow problem. Hence, the fluid variation formulation can be reduced to

Find $\hat{\mathbf{v}}_f, \hat{\mathrm{p}}_f$ such that

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

The latter approach is chosen for this thesis, as only the variational formulation for the fluid is tested and removes any influence of the strucutre and mesh extrapolation discretization. Since $\hat{\Omega}_f = \Omega_f(t)\ \ t \in T$, the mesh velocity of the fluid $\frac{\partial \hat{\Gamma}_W}{\partial t} = 0$ and no deformation of the fluid domain is present.

The validation of the fluid solver is divided into the three sub-cases CFD1, CFD2 and CFD3.While CFD1 and CFD2 yields steady state solutions, CFD3 is a periodic solution.

Table 7.1: Benchmark environment

Fluid parameters			
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

A parabolic velocity profile on the form,

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

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 flow profiles.

At the right 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 is based on the evaluation of drag and lift forces on the inner geometry for each sub-case. with comparison to [**?**]. Each sub-case will be conducted on four different mesh, with increasing refinement. The following tables presents the numerical results for each sub-case.

Table 7.2: CFD 1 Results

$\Delta t = 0.1 \;\; \theta = 1.0$			
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 7.3: CFD-2

$\Delta t = 0.01 \;\; \theta = 1.0$			
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 %

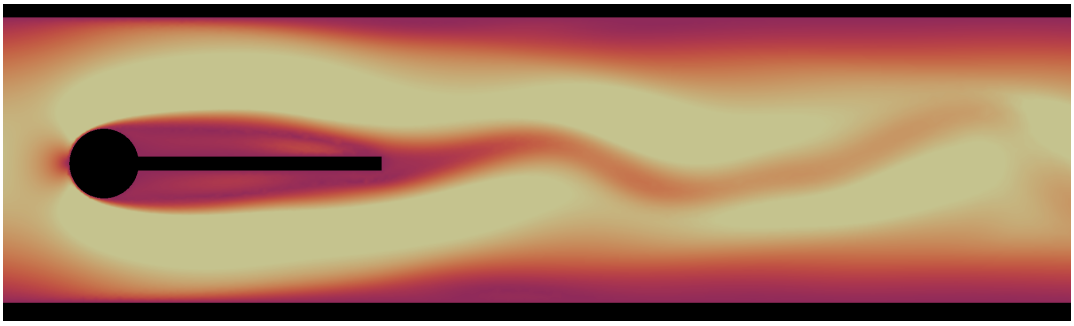

Figure 7.2: CFD-3, flow visualization of velocity time t = 9s

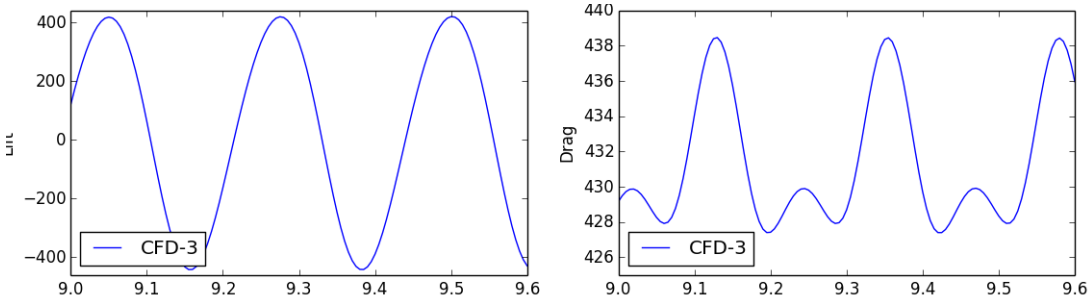

Figure 7.3: CFD-3, lift and drag forces at time t = [9, 9.6]

Table 7.4: CFD-3

$\Delta t = 0.01$ $\theta = 0.5$			
nel	ndof	Drag	Lift
1438	6881 (P2-P1)	417.23 +/- 0.0217	-249.21 +/- 0.32
	16474 (P3-P2)	414.86 ± 5.6282	-7.458 ± 444.07
2899	13648 (P2-P1)	408.50 ± 4.3029	-19.731 ± 373.45
	32853 (P3-P2)	432.86 ± 5.5025	-9.686 ± 431.28
7501	34657 (P2-P1)	431.57 ± 5.2627	-12.497 ± 429.76
	83955 (P3-P2)	438.20 ± 5.5994	-11.595 ± 438.00
19365	88520 (P2-P1)	435.43 ± 5.4133	-11.545 ± 438.89
	215219 (P3-P2)	438.80 ± 5.6290	-11.158 ± 439.23
Reference		439.95 ± 5.6183	-11.893 ± 437.81
Error		0.002 % ± 0.001 %	0.061 % ± 0.003%
$\Delta t = 0.005$ $\theta = 0.5$			
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%

7.2.2 Discussion of results

The numerical results for CFD1, CFD2 and CFD3 are all within reasonable range of the reference solutions presented in [**?**]. For CFD1 and CFD2, the choice of P2-P1 elements together with a fully implicit schene $\theta = 1$ gained sufficient accuracy in comparison with the reference solution. The second order cranc-nicholoson scheme $\theta = 0.5$ was investigated for CFD1 and CFD2, 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 unsuficcient for capturing the expected periodic solution. Only a steady-state flow profile was observed.. By cranc-nicolson time-stepping scheme $\theta = 0.5$, the periodic solution was attained. Since the choice of finite-elemt pair is not reported in the original work, both P3-P2 and P2-P1 element pairs for fluid and pressure repsectivly was compared in combination with spatial mesh refinement. From Table 1.4, the choice P3-P2 element pair is eminent to achieve reasonable results for the first and second

mesh regardless of timestep. However, the third and fourth mesh shows close resemblance with the reference solution.
On this basis, the choice of P2-P1 element pair is sufficient for the evaluation of drag and lift on the inner geometry with increasing mesh resolution.

7.2.3 Validation of solid solver

Table 7.5: CSM validation environment

Fluid parameters			
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

The validation of the solid solver are conducted on three refined mesh, where the number of elements are chosen in close resemblance with the original work in [**?**]. 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 by the motivation of shorter simulation time while retaining 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 FSI-solver.ß

Table 7.6: CSM 1 Results

$\Delta t = 0.1 \;\; \theta = 1.0$			
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 %

7.2.4 Discussion of results

The results for sub-problems CSM-1 and CSM-2 each coincide with the reference solution. The study of lower-grade elements proved successful for both problems, justifying accurate results can be achieved for polynomials grade 1 and 2 for all mesh refinements. This obeservation is furhter justified in the CSM-3 reults. In table 1.4, the displacement components of P3-P3 and P2-P2 elements can hardly be distinguished.

Table 7.7: CSM 2 Results

$\Delta t = 0.05$ $\theta = 1.0$			
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 7.8: CSM 3 Results

$\Delta t = 0.02$ $\theta = 0.5$			
nel	ndof	ux of A [x 10^3]	uy of A [x 10^3]
319	832 P1-P1	-10.790 +/- 10.797	-55.184 +/- 56.682
	2936 P2-P2	-14.380 +/- 14.387	-63.198 +/- 65.147
	6316 P3-P3	-14.409 +/- 14.417	-63.288 +/- 65.225
1365	3140 P1-P1	-13.032 +/- 13.041	-60.446 +/- 62.075
	11736 P2-P2	-14.407 +/- 14.416	-63.283 +/- 65.220
	25792 P3-P3	-14.412 +/- 14.421	-63.310 +/- 65.246
5143	11084 P1-P1	-14.059 +/- 14.071	-62.591 +/- 64.473
	42736 P2-P2	-14.412 +/- 14.421	-63.313 +/- 65.249
	94960 P3-P3	-14.416 +/- 14.425	-63.328 +/- 65.263
Reference		-14.305 +- -14.305	-63.607 +- 65.160
Error		%	%

$\Delta t = 0.01$ $\theta = 0.5$			
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		%	%

$\Delta t = 0.005$ $\theta = 0.5$			
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		%	%

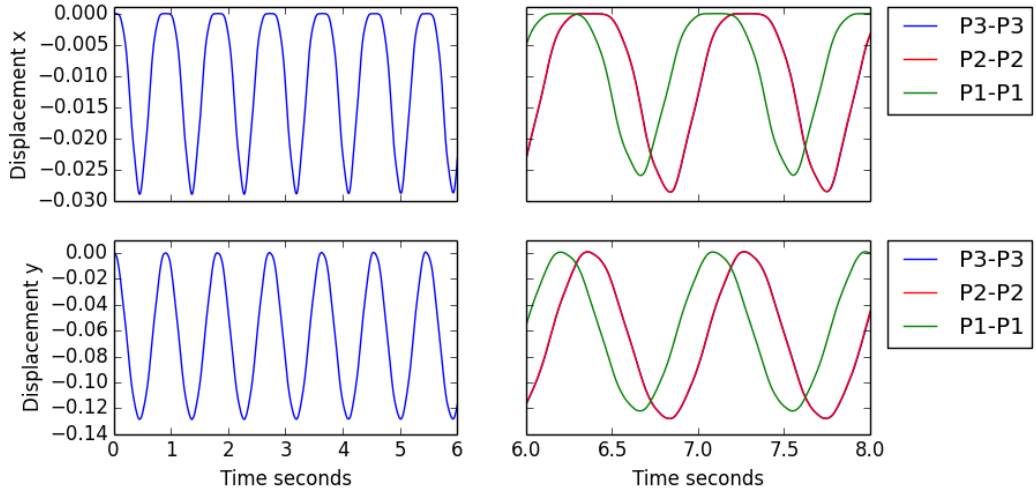

Figure 7.4: CSM-3, deformation components of A(t) for two different time intervals. Time interval $t \in [0, 6]$ shows the P3-P3 element pair, while$t \in [6, 8]$ compares all finite elemet pair chosen for the experiment

7.2.5 Validation of fluid structure interaction solver

The validation of the FSI solver constist of three sub-cases which will be referred to FSI-1, FSI-2 and FSI-3. The FSI-1 environment yields a steady state solution for the system, inducing small deformations to the elastic flag. This environment is exelent to ensure the overall coupling of the FSI-problem is exectuted properly. The FSI-2 and FSI-3 environment results in a periodic solution, where the elastic flag oscilates behind the sylinder.

For all sub-cases a parabolic velocity profile on the form,

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

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 flow profiles. At the right 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 circle of the inner geometry. The structure deformation and velocity is set to zero on the left side of the flag, where the flag is ancored to the circle. On the fluid-structure interface Γ, we enfore the kinematic and dynamic boundary condition

$$\mathbf{v}_f = \mathbf{v}_s \tag{7.1}$$

$$\sigma_f \cdot \mathbf{n} = \sigma_s \cdot \mathbf{n} \tag{7.2}$$

From chapter ?, (1.1) is enforced strongly due to the continious velocity field, while (1.2) is enforced weakly by omtitting form the weak formulation by.

Apart from the accuracy of the reported values, the main purpose of the validation of the fluid solver is twofold. Firstly, it is of great importance to ensure that the overall coupling of the fluid-structure interaction problem are executed correctly. Second, a good choice of mesh extrapolation model is essential to ensure that mesh entanglement is not present. Based on the experience with the previous sub-problems, the finite element group of P2-P2-P1 is chosen for deformation, velocity and pressure respectivly.

Table 7.9: Benchmark environment

Solid parameters			
parameter	FSI1	FSI2	FSI3
$\rho^s[10^3 \frac{kg}{m^3}]$	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	FSI1	FSI2	FSI3
Re	20	100	200

FSI1

Table 7.10: FSI 1 Results

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
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
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
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

Table 7.11: FSI 1 - No extrapolation

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
REF	REF	0.0227	0.8209	14.295	0.7638

FSI2

FSI2

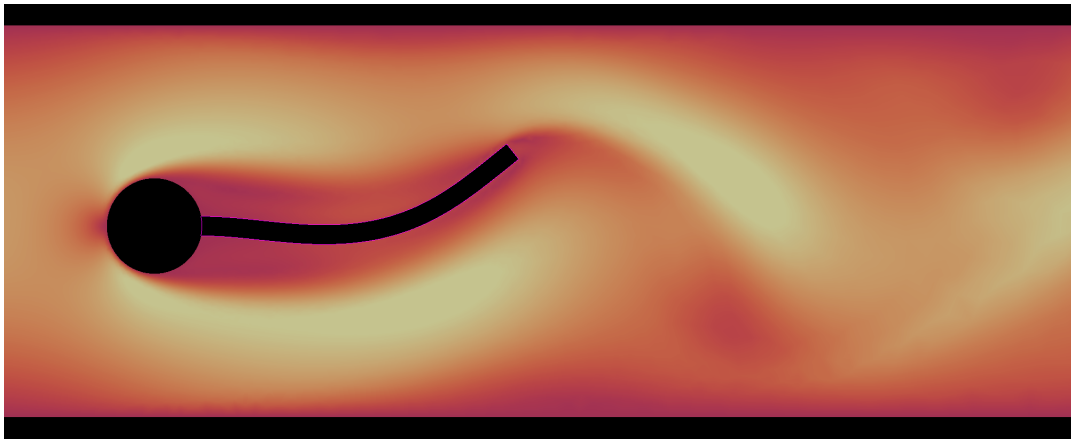

Figure 7.5: FSI-2, visualization of fully developted flow with structure deformation at time t = 9s

FSI3

Table 7.12: 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 ± 3.22	449.40 ± 14.70	0.55 ± 155.80
7307	63365	-2.32 ± 2.30	1.34 ± 3.17	451.78 ± 16.08	1.13 ± 151.22
11556	99810	-2.34 ± 2.34	1.57 ± 3.19	455.92 ± 17.32	-0.10 ± 151.03
$\Delta t = 0.001 \theta = 0.501$					
nel	ndof	ux of A [x 10^3]	uy of A [x 10^3]	Drag	Lift
1216	5797	–2.17± 2.08	3.32 ± 29.07	439.98 ± 14.08	1.91 ± 151.71
2295	10730	-3.04 ± 2.88	1.51 ± 35.88	452.04 ± 22.41	3.30 ± 160.11
5963	27486	-3.03± 2.85	1.23 ± 35.97	459.45 ± 23.80	1.53 ± 160.14
Reference		136.7	10.53		
Error		0.007 %	0.001 %		

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	7.96 ± 8.10	-3.84 ± 1.02	450.16 ± 15.11	-20.09 ± 148.17
7307	63365	3.10 ± 3.06	-1.90 ± 4.21	457.37 ± 15.24	-51.77 ± 127.28
11556	99810	-2.18 ± 9.65	1.31 ± 4.93	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
1216	5797	-2.18 ± 2.10	3.56 ± 2.90	435.19 ± 9.77	-1.57 ± 151.43
7307	63365	-1.42 ± 4.70	7.77 ± 2.85	454.38 ± 19.75	17.97 ± 155.08
11556	99810	-2.23 ± 6.16	1.72 ± 4.48	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		0.007 %	0.001 %		

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
1216	5797	-1.74 ± 1.76	3.56 ± 26.01	439.41± 12.21	-1.35 ± 138.74
2295	10730	-2.39 ± 2.40	1.76 ± 32.27	449.71± 18.16	3.71 ± 149.97
$\Delta t = 0.001 \theta = 0.501$					
nel	ndof	ux of A [x 10^3]	uy of A [x 10^3]	Drag	Lift
1216	5797	-3.39 ± 3.38	1.23 ± 36.61	413.26 ± 51.82	57.19 ± 222.65
2295	10730	-4.70 ± 4.71	1.49 ± 44.62	427.91± 93.17	44.38 ± 268.05
Reference		-2.69 ± 2.56	1.48 ± 34.38	457.3 ± 22.66	2.22 ±- 149.78
Error		0.007 %	0.001 %		

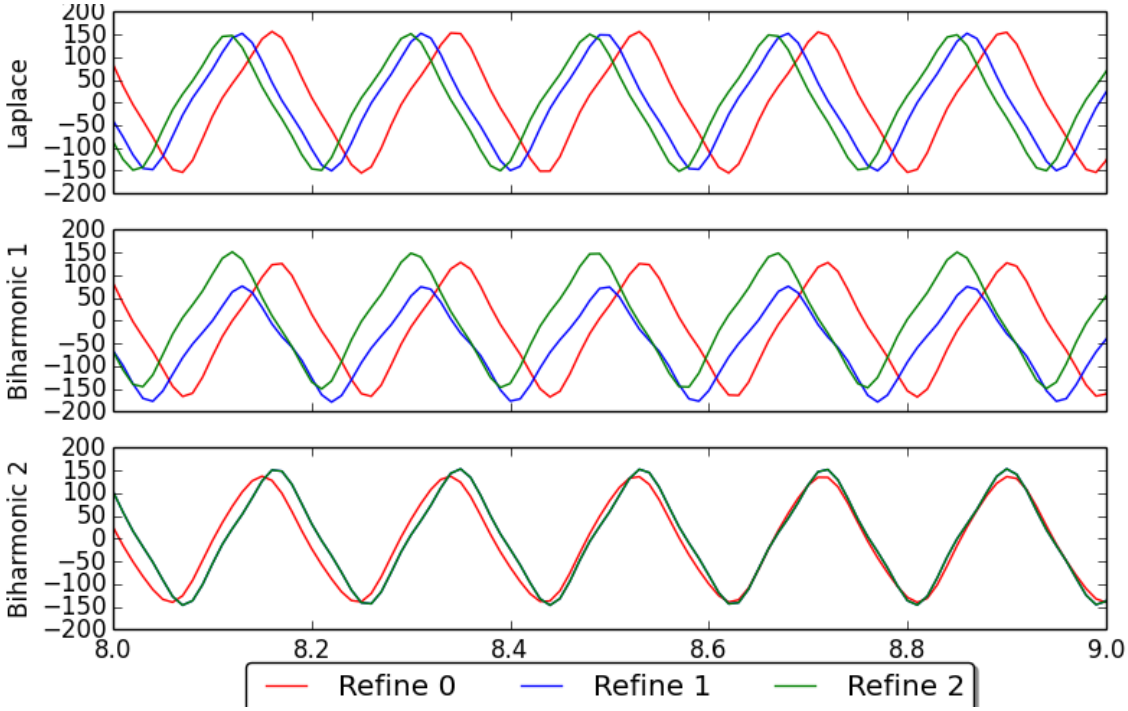

Figure 7.6: Comparing mesh extrapolation models

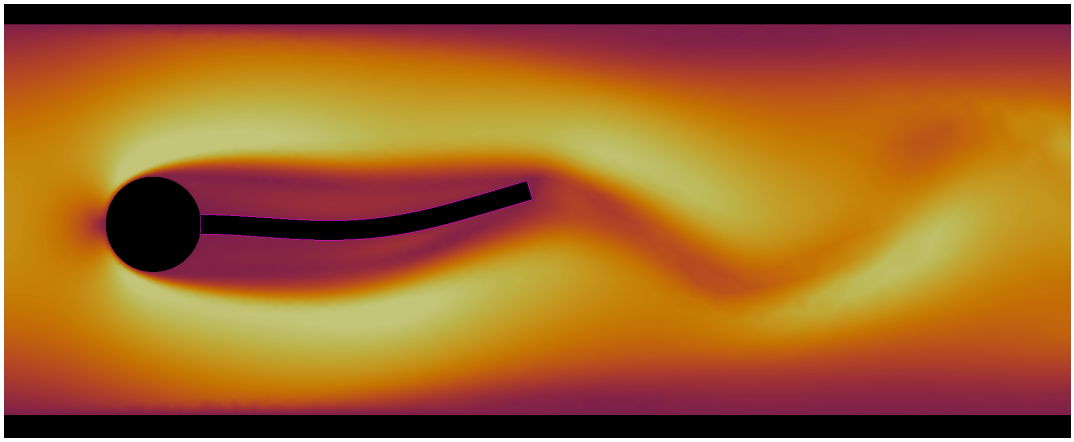

Figure 7.7: FSI-3, visualization of fully developted flow with structure deformation at time t = 5.1s

7.2.6 Discussion of results

Considering FSI1, all mesh extraplation models are of high accuracy compared to the reference solution. However, due to the small deformations of order 10^{-6}, FSI1 doesn't provide a rigorous test of the chosen mesh extrapolation model. By omitting mesh extrapolation from the variatonal formulation, reasonable results are still obtained. This proves that the FSI-1 validation case can be misguiding, in terms of validating the chosen mesh extrapolation model.

The FSI2 case proved to be one of the most demanding tests, due to the large deformation of the elastic flag. leading to the risk of entangled mesh cells. Therefore

a high quality extrapolation of the solid deformation into the fluid is needed. All mesh extrapolation models proved to

The FSI3 environment does not induce deformation to the extent of the FSI2. However a critical phase in the transition to the periodic solution was discovered, where the pressure oscillation induces a large deformation to the system.

7.3 Investigation of temporal stability

Preliminary work regarding discretization and numerical analysis of Crank-Nicholson time stepping schemes for fluid structure interaction can be found in cite WIck papers. Two main properties of interest of higher-order methods have proven to be the stability of long-time simulation, and obtaining the expected physics for the problem of interest.

It is known that the Crank-Nicolson scheme can suffer from temporal stability, for long-term simulations [**?**]. Therefore, the authors of [**?**], investigated temporal stability of the Crank-Nicolson scheme for the validation benchmark found in [**?**]. The critera for the numerical experiements was to obatin a stable solution in the time interval [0, 10] minutes, by temporal and spatial refinement studies. The fully monolithic FSI problem discretized with second-order Crank-Nicolson, proved to give general stability problems for long-term simulation for certain time-steps k.

Following the ideas of [**?**],, a second order scheme based on the Cranck-Nicholson yields two possibilities.

Discretization 7.1. *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 7.2. *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 1.1 and 1.2 , and significant differences of temporal accuracy was not found.
Two options to coupe with the presented unstabilities are the *shifted Crank-Nicolson* [**?**], [19], [**?**], and the *frac-step method*. Both of these methods are defined as A-stable time-stepping schemes meaning.. In this thesis the shifted Crank-Nocolson scheme will be considered.
The shifted Crank-Nicolson scheme introduce further stability to the overall system, by shifting the θ paramter slightly to the implicit side. 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 [**?**].

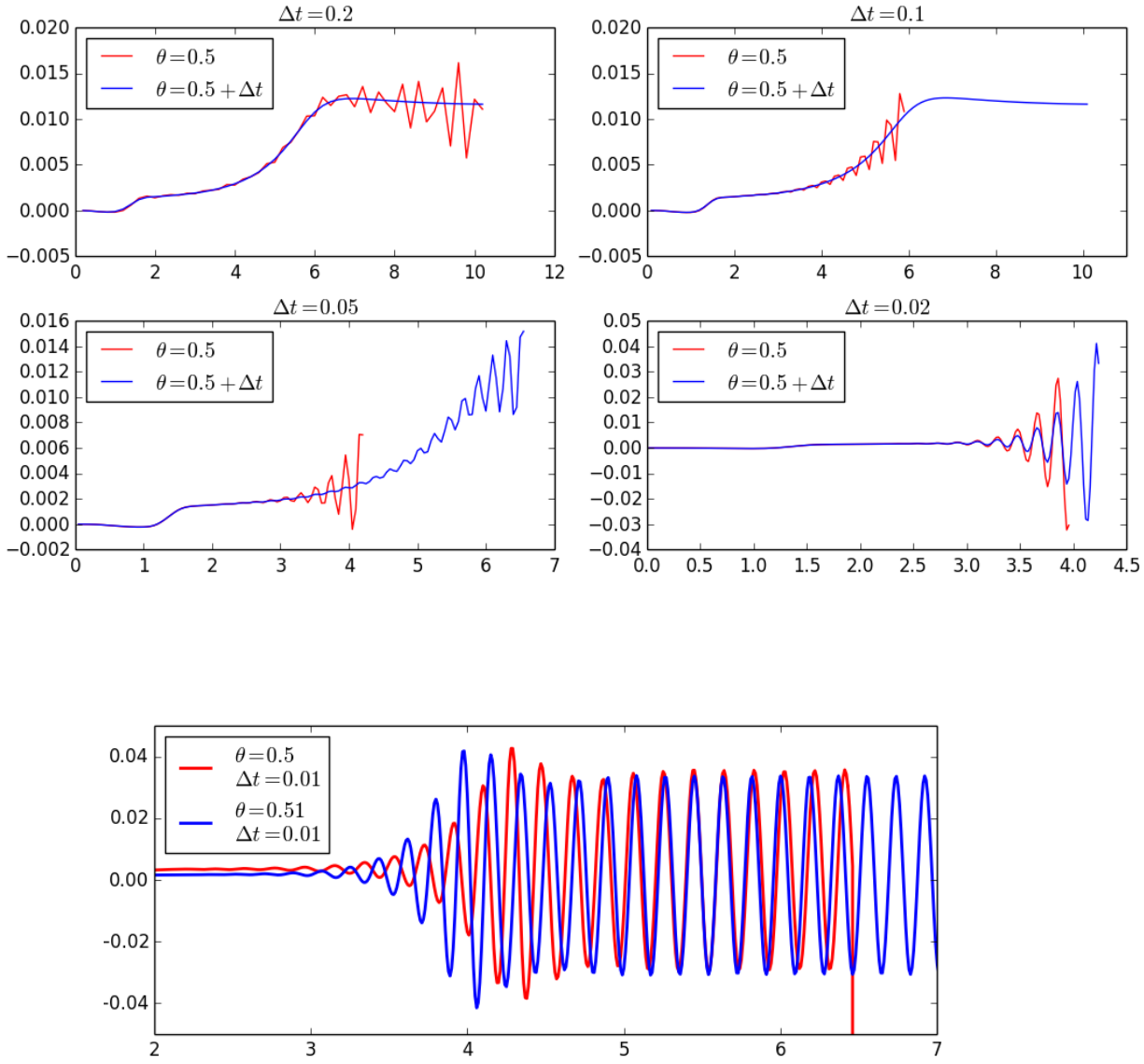

Figure 7.8: Investigation of temporal stability for the FSI3 benchbark in the time interval $t \in [0, 10]$, comparing the shifted crank nicolson to the original cranc nicolson scheme.

A numerical investigation of temporal stability in shown in Table 1.7, where the shifted crank-nicolson scheme $\theta = 0.5 + k$, is compared the original crank-nicolson $\theta = 0.5$. The shifted version clearly show stability properties surpassing the original crank-nicolson scheme, for all numerical experiments. However, for $\Delta t \in [0.2, 0.1, 0.05, 0.02]$ the scheme clearly lacks the ability to capture the overall physics of the validation problem. Long-time stability and expected physical beavior is obtained for $\Delta t = 0.01$. However, numerical experiments showed that for $\Delta t \leq 0.005$ numerical stability was achieved regardless of both methods. This result is important, reducing the overall computational time needed to achieve reasonable accuracy.

7.4 Optimization of Newtonsolver

A *bottleneck* express a phenomena where the total performance of a complete implementation is limited to small code fragments, accounting for the primary consumption of computer resources.

As for many other applications, within computational science one can often assume the consummation 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 computational demanding operations comes from 20% of the code. In our case, the bottleneck is the newtonsolver. The two main reasons for this is

- **Jacobian assembly**
  The construction of the Jacobian matrix for the total residue of the system, is the most time demanding operations within the whole computation.

- **Solver**.
  As iterative solvers are limited for the solving of fluid-structure interaction problems, direct solvers was implemented for this thesis. As such, the operation of solving a linear problem at each iteration is computational demanding, leading to less computational efficient operations. Mention order of iterations?

Facing these problems, several attempts was made to speed-up the implementation. The FEniCS project consist of several nonlinear solver backends, were fully user-customization option are available. However one main problem which we met was the fact that FEniCS assembles the matrix of the different variables over the whole mesh, even though the variable is only defined in one to the sub-domains of the system.In our case the pressure is only defined within the fluid domain, and therefore the matrix for the total residual consisted of several zero columns within the structure region. FEniCS provides a solution for such problems, but therefore we were forced to construct our own solver and not make use of the built-in nonlinear solvers.

The main effort of speed-up were explored around the Jacobian assembly. Of the speed-ups methods explored in this thesis, some are *consistent* while others are *nonconsistent*. Consistent methods are methods that always will work, involving smarter approaches regarding the linear system to be solved. The non-consistent method presented involves altering the equation to be solved by some simplification of the system. As these simplifications will alter the expected convergence of the solver, one must take account for additional Newton iterations against cheaper Jacobi assembly. Therefore one also risk breakdown of the solver as the Newton iterations may not converge.

7.5 Consistent methods

7.5.1 Jacobi buffering

By inspection of the Jacobi matrix, some terms of the total residue is linear terms, and remain constant within each time step. By assembling these terms only in the first Newton iteration will save some assembly time for the additional iterations needed each time step. As consequence the convergence of the Newton method should be unaffected as we do not alter the system.

7.6 Non-consisten methods

7.6.1 Reuse of Jacobian

As the assembly of the Jacobian at each iteration is costly, one approach of reusing the Jacobian for the linear system was proposed. In other words, the LU-factorization of the system is reused until the Jacobi is re-assembled. This method greatly reduced the computational time for each time step. By a user defined parameter, the number of iterations before a new assembly of the Jacobian matrix can be controlled.

7.6.2 Quadrature reduce

The assemble time of the Jacobian greatly depends on the degree of polynomials used in the discretisation of the total residual. Within FEniCS t he order of polynomials representing the Jacobian can be adjusted. The use of lower order polynomials reduces assemble time of the matrix at each newton-iteration, however it leads to an inexact Jacobian which may results to additional iterations.

Table 7.13: Comparison of speedup techniques

Laplace					
Implementation	Naive	Buffering	Reducequad.	Reusejacobi	Combined
Mean time/-timestep	104.5		125.5	48.3	6.8
Speedup %	1.0		-20%	54 %	94 %
Mean iteration	4.49		30.59	10.29	10.29
Biharmonic Type 1					
Implementation	Naive	Buffering	Reducequad.	Reusejacobi	Combined
Mean time/-timestep	243.3	307.6	51.6	76.7	24.8
Speedup %	1.0	-26%	78.7%	68.4 %	89.7%
Mean iteration	4.1	6.2	4.6	7.1	6.8

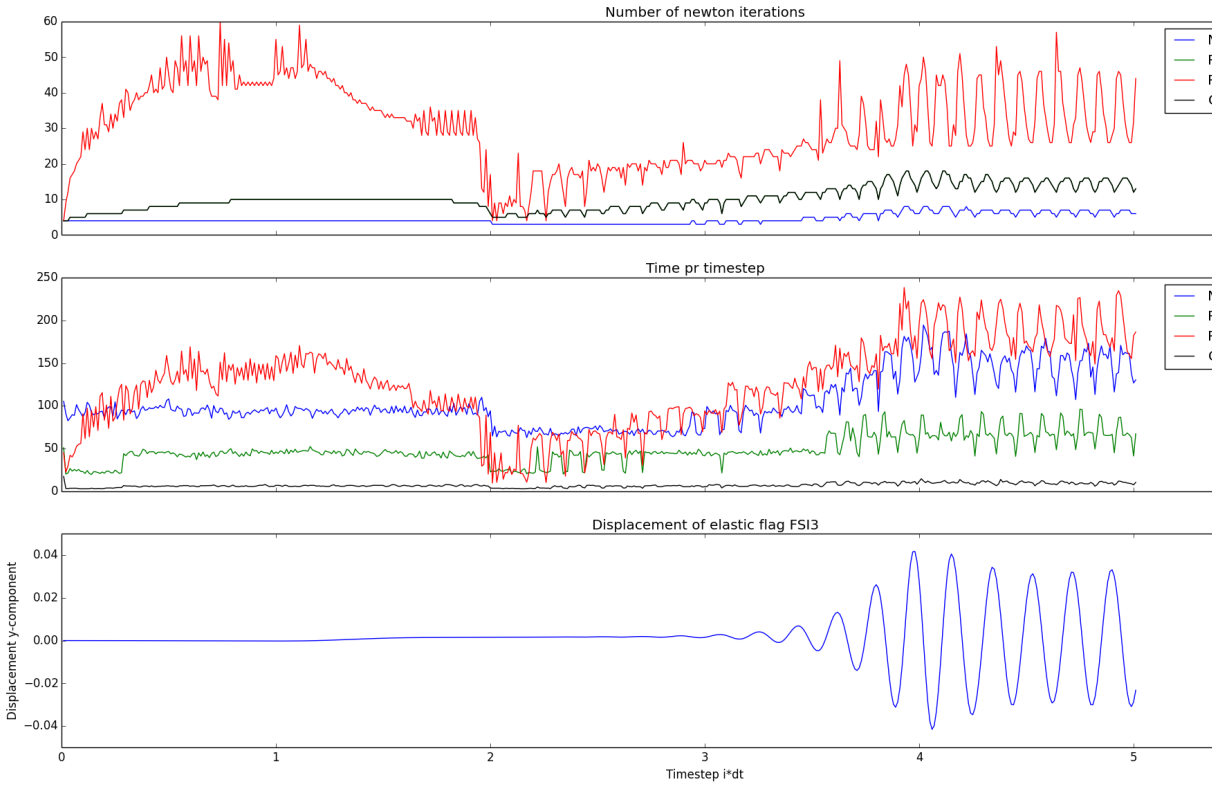

Figure 7.9: Comparison of speed-up techniques for the laplace mesh model

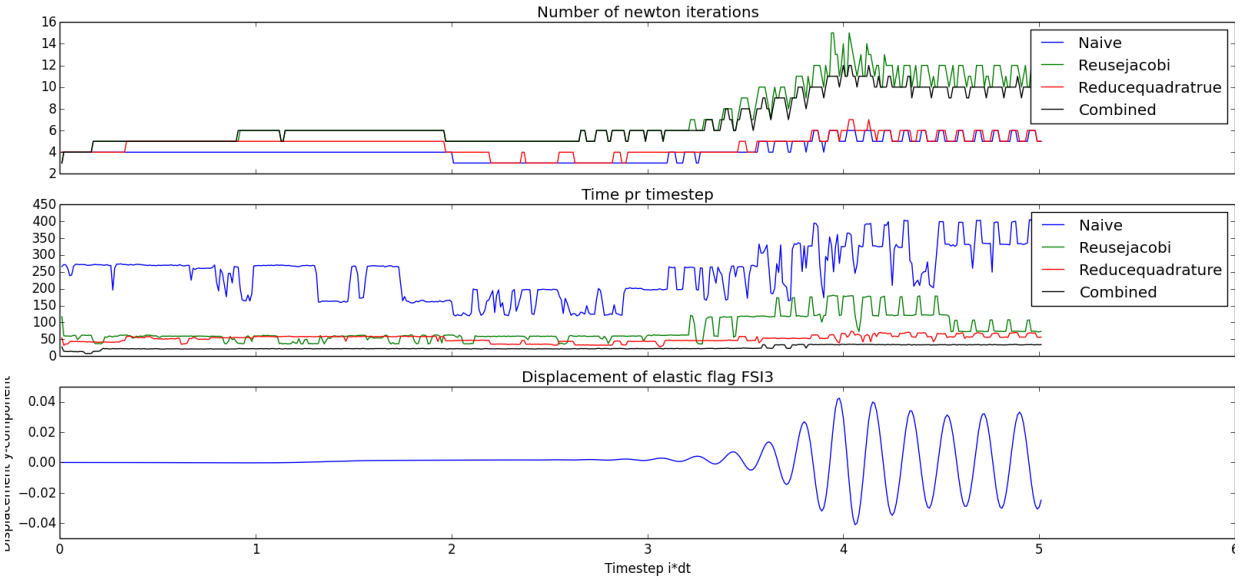

Figure 7.10: Comparison of speed-up techniques for the biharmonic type 1 mesh model

Bibliography

[1] Robert T Biedron and Elizabeth M Lee-Rausch. Rotor Airloads Prediction Using Unstructured Meshes and Loose CFD/CSD Coupling.

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

[3] Richard P Dwight. Robust Mesh Deformation using the Linear Elasticity Equations.

[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] 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.

[7] Bernhard Gatzhammer. Efficient and Flexible Partitioned Simulation of Fluid-Structure Interactions. page 261, 2014.

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

[9] Su-Yuen Hsu, Chau-Lyan Chang, and Jamshid Samareh. A Simplified Mesh Deformation Method Using Commercial Structural Analysis Software.

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

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

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

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

[14] 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.

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

[16] K Stein, T Tezduyar, and R Benney. Mesh Moving Techniques for Fluid-Structure Interactions With Large Displacements.

[17] T E Tezduyar, M Behr, S Mittal, and A A Johnson. COMPUTATION OF UNSTEADY INCOMPRESSIBLE FLOWS WITH THE STABILIZED FINITE ELEMENT METHODS: SPACE-TIME FORMULATIONS, ITERATIVE STRATEGIES AND MASSIVELY PARALLEL IMPLEMENTATIONSt. *New Methods in Transient Analysis ASME*, 246(143), 1992.

[18] 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.

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

[20] Thomas Wick. Solving Monolithic Fluid-Structure Interaction Problems in Arbitrary Lagrangian Eulerian Coordinates with the deal.II Library.

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

[22] P. Wriggers. *Computational contact mechanics, second ed., Springer*. 2006.