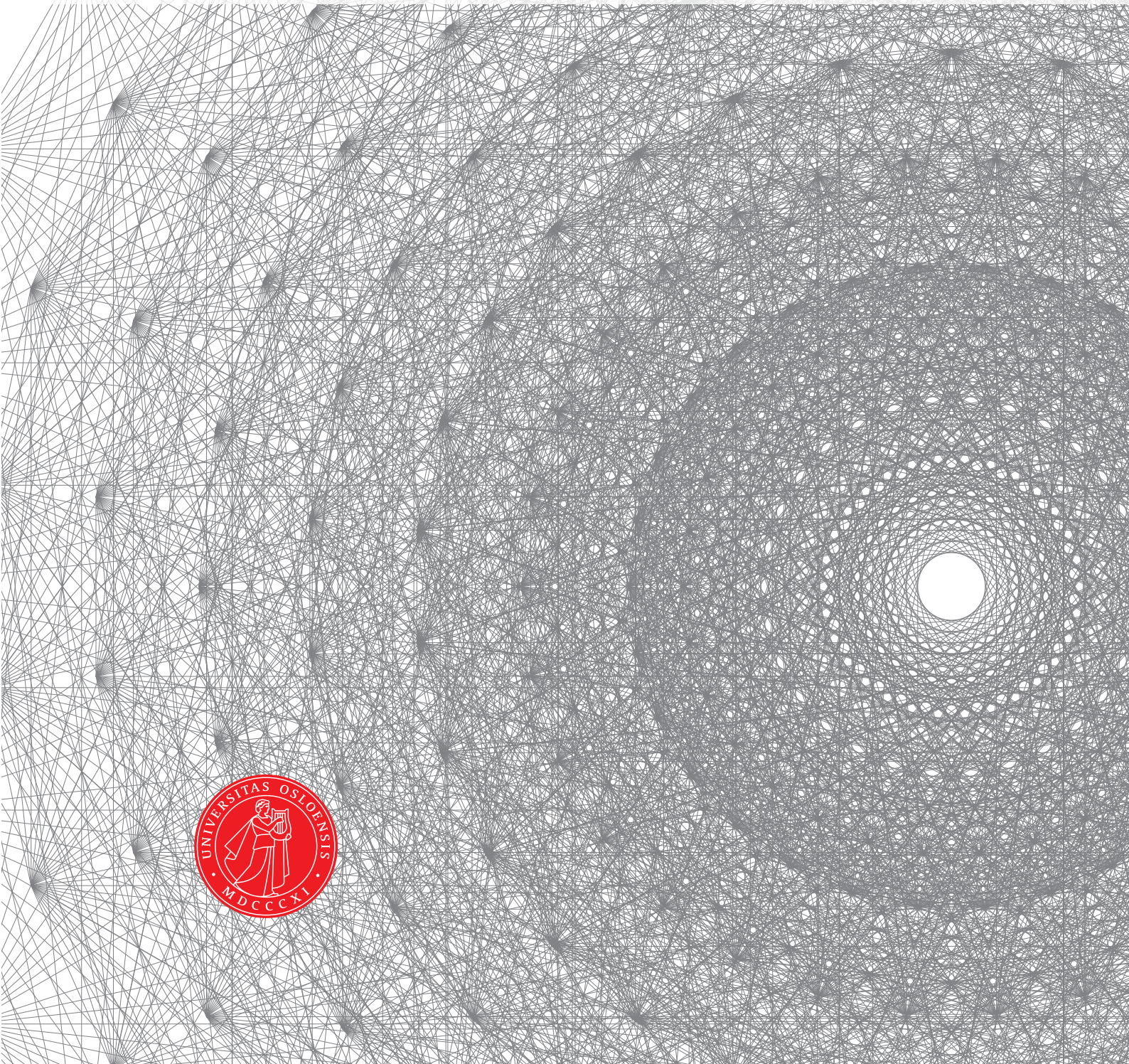
UiO : Department of Mathematics
University of Oslo

Monolithic solver for Computational Fluid-Structure Interaction

Sebastian Gjertsen

Master's Thesis, Spring 2017



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

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

Contents

Chapter 1

Verification and Validation of the Fluid Structure Interaction Implementation.

The general approach when solving a real world problem with numerical computing, starts by defining the mathematics, implementing the equations numerically and solve the equations on a computer. To gain insight into a real world problem the produced solutions are used to extract data of interest.

A question then immediately arises, is the solution and the extracted data trustworthy? To answer this question we need to adress another question, are the equations implemented correct? if so, is the problem defined correct mathematically? Without answering these questions, being confident that your solutions are correct is difficult [20]. This process of generating evidence that computed solutions meets certain requirements to fulfill an intended purpose, in the context of scientific computing, is known as Verification and Validation. The goal of this section will hence be to verify and validate the different numerical schemes outlined in the two previous chapters.

The chapter starts with the process of Verification where the fluid and structure parts of the code will be verified. This will be followed Validation of the code where I implement a well known benchmark testing the fluid and structure parts individually and as a full FSI problem.

1.1 Verification

Verification, in the context of scientific computing, is the process of determining wether or not the implementation of numerical algorithms in computer code, is correct. [14]. When mapping a mathematical model to a computational model there is always a risk of introducing errors to the computational model. Verification helps us identify, quantify and reduce said errors. Verification does not address wether or not the computed solutions are in alignment with physics in the real world. It only tells us that our model is computed correctly or not.

In Verification there are multiple classes of test that can be performed, one of which is *order of convergence tests*. Order of convergence are based on the behavior of the error between a known exact solution and a computed solution [17]. The most rigorous of the order of convergence test is the *Method of Manufactured Solution*(MMS) [14]. When performing MMS, rather than looking for an exact solution, we manufacture one. The idea is to make a solution *a priori*, and use this solution to generate an analytical source term for the governing PDEs and then run the PDE with the source term to produce a solution. The manufactured solution does not need to have a physically realistic relation, since the solution deals only with the mathematics.

When manufacturing a solution in MMS tests there are a number of criteria that needs to met for a solution to be sufficient. The manufactured solutions should be chosen to be non-trivial and analytic [14, 17]. The solutions should be manufactured so that no eventual derivatives vanish. For this reason trigonometric and exponential functions can be a smart choice, since they are smooth and infinitely differentiable. In short, a good manufactured solution is one that is complex enough so that it rigorously tests each part of the equations.

The procedure of MMS is as follows [14]:

- We define a mathematical model on the form $L(\mathbf{u}) = 0$ where $L(\mathbf{u})$ is a differential operator and u is a dependent variable.
- Define the analytical form of the manufactured solution $\hat{\mathbf{u}}$
- Use the model $L(u)$ with $\hat{\mathbf{u}}$ inserted to obtain an analytical source term $f = L(\hat{\mathbf{u}})$

- Initial and boundary conditions are enforced from $\hat{\mathbf{u}}$
- Then use this source term to calculate the solution $\mathbf{u}$, $L(\mathbf{u}) = f$

If we let $\mathbf{u}$ be the numerical solution and $\hat{\mathbf{u}}$ be the exact solution, $||.||$ be the L^2 norm, we define the error as:

$$E = ||\mathbf{u} - \hat{\mathbf{u}}|| \tag{1.1}$$

When we decrease the node spacing (Δx) or decrease time step size(Δt), we expect the solution to convergence towards a given solution and hence the error to get smaller. If we assume uniform node spacing in all spatial directions:

$$E = C_1 \Delta x^k + C_2 \Delta t^l \tag{1.2}$$

where $k = m + 1$ and m is the polynomial degree of the spatial elements. The error is hence dependent on the node spacing and the time step. If for instance Δt is small enough, Δx will dominate, and Δt will be negligible. Let E_{n+1} and E_n be the computed errors of a solution with fine and coarse node spacing respectively. If we divide the errors using (1.2):

$$\frac{E_{n+1}}{E_n} = \Big(\frac{\Delta x_{n+1}}{\Delta x_n}\Big)^k \tag{1.3}$$

$$k = \frac{log(\frac{E_{n+1}}{E_n})}{log(\frac{\Delta x_{n+1}}{\Delta x_n})} \tag{1.4}$$

After refining the mesh while keeping Δt fixed and sufficiently small, k can be found and compared to the theoretical order of convergence for each given problem. If the k that we have found matches the theoretical order of convergence, with small margin of error, we can be satisfied that the implementation of the code is correct.

1.1.1 Method of Manufactured Solution on the implementation of the Solid equation

The MMS test is constructed to verify the implementation of the solid equation (**??**), with the restriction $\mathbf{u} = \frac{\partial d}{\partial t}$.

Solutions $\hat{\mathbf{d}}$ and $\hat{\mathbf{u}}$ are manufactured with sine and cosine such that the derivatives does not become zero and we have temporal and spatial dependencies.

The solutions are also manufactured to uphold the restriction $\mathbf{u} = \frac{\partial d}{\partial t}$.

$$\hat{\mathbf{d}_\mathbf{e}} = (cos(y)sin(t), cos(x)sin(t))$$
$$\hat{\mathbf{u}_\mathbf{e}} = (cos(y)cos(t), cos(x)cos(t))$$

The manufactured solutions are used to produce a sourceterm f_s :

$$\rho_s \frac{\partial \hat{\mathbf{u}_\mathbf{e}}}{\partial t} - \nabla \cdot (P(\hat{\mathbf{d}_\mathbf{e}})) = f_s \tag{1.5}$$

The equations are solved for **d** and **u** on a unit square domain. The number N denotes the number of spatial points in x and y direction. The functions u and d will be computed to match the source term f_s. The computations were simulated for 10 time steps and the error was calculated for each time step and then the mean of all the errors were used as a measure of the error.

N	Δt	**m**	E_u	$\mathbf{k_u}$	E_d	$\mathbf{k_d}$
4	1×10^{-7}	**1**	0.0068828	**-**	3.7855×10^{-9}	**-**
8	1×10^{-7}	**1**	0.0017204	**2.0002577**	9.4622×10^{-10}	**2.0002577**
16	1×10^{-7}	**1**	0.0004300	**2.0000622**	2.3654×10^{-10}	**2.0000622**
32	1×10^{-7}	**1**	0.0001075	**2.0000154**	5.9136×10^{-11}	**2.0000154**
64	1×10^{-7}	**1**	0.0000268	**2.0000038**	1.4783×10^{-11}	**2.0000038**

Table 1.1: Method of Manufactured Solution on the implementation of the Solid equation in space with m = 1

In table 1.1 we set m = 1, and vary the number of spatial points from 4 to 64 keeping $\Delta t = 10^{-7}$. The error E_u and E_d gets smaller for increasing values of N.The order of convergence k_u and k_d converges toward the expected value of 2.

N	$\mathbf{\Delta t}$	$E_u[\times 10^{-6}]$	$\mathbf{k_u}$	$E_u[\times 10^{-8}]$	$\mathbf{k_d}$
64	**0.1**	0.027663		0.0034221	
64	**0.05**	0.013390	**1.0467**	0.0018093	**0.9194**
64	**0.025**	0.007016	**0.9324**	0.0009246	**0.9685**
64	**0.0125**	0.003645	**0.9444**	0.0004688	**0.9798**
64	**0.00625**	0.001828	**0.9957**	0.0002414	**0.9571**

Table 1.2: Method of Manufactured Solution on the implementation of the Solid equation in time

In table 1.2 we check the temporal convergence. The number of spatial points has been fixed to 64, with varying Δt from 0.1 halving each step to 0.0065.

The error E_u and E_d gets smaller for decreasing values of Δt. The scheme tested is temporal first order accurate, by setting a value $\theta = 1$ expecting a order of convergence in temporal direction of 1. In table 1.2 convergence k_u and k_d tends towards 1.

1.1.2 MMS of Fluid equations with prescribed motion

In this section we verify the fluid equations (**??**) in the ALE framework computed from a reference domain, with a prescribed motion.

The functions $\hat{\mathbf{u}}$, $\hat{\mathbf{d}}$ and $\hat{p}$ are manufactured to uphold the restriction (**??**) and incompressible fluid (**??**), and are made with sine and cosine function to uphold the criteria of MMS. The fluid and domain velocity are set to be equal: $\hat{\mathbf{u}} = \mathbf{w}$.

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

Whilst testing the implementations of the fluid equations, the opportunity arises to also test the mappings between current and reference configurations. The source term f_f is produced without mappings:

$$\rho_f \frac{\partial \hat{\mathbf{u}}}{\partial t} + \nabla \hat{\mathbf{u}}(\hat{\mathbf{u}} - \frac{\partial \hat{\mathbf{d}}}{\partial t}) - \nabla \cdot \sigma(\hat{\mathbf{u}}, \hat{p})_f = f_f$$

To be specific, we use f_f from the current configuration and map it to the reference:

$$\rho_f J \frac{\partial \mathbf{u}}{\partial t} + (\nabla \mathbf{u}) F^{-1}(\mathbf{u} - \frac{\partial \mathbf{d}}{\partial t}) + \nabla \cdot (J \hat{\sigma_f} F^{-T}) = J f_f$$

The computations are performed on a unit square domain and the computations were simulated with 10 timesteps and the error was calculated for each time step and then the mean of all the errors was taken and used as a measure of the error.

N	Δt	E_u	$\mathbf{k_u}$	E_p	$\mathbf{k_p}$
64	0.1	5.1548×10^{-5}	**-**	0.008724	**-**
64	0.05	2.5369×10^{-5}	**1.0228**	0.004290	**1.0240**
64	0.025	1.2200×10^{-5}	**1.0561**	0.002058	**1.0596**
64	0.0125	0.56344×10^{-5}	**1.1145**	0.0009556	**1.1068**

Table 1.3: Results for Method of Manufacture Solutions test for fluid equations

In table 1.3 we check the temporal convergence keeping the spatial points constant with $N = 64$. Decreasing Δt by half, from 0.1 to 0.0125. The errors for fluid velocity and pressure E_u and E_p decrease with decreasing time steps. The compute convergence k_u and k_p tends toward a value of 1.

N	Δt	**m**	E_u	k_u	E_p	k_p
2	1×10^{-6}	**2**	8.6955×10^{-4}	**-**	0.01943	**-**
4	1×10^{-6}	**2**	1.0844×10^{-4}	**3.0032**	0.00481	**2.0140**
8	1×10^{-6}	**2**	0.1354×10^{-4}	**3.0007**	0.00119	**2.0120**
16	1×10^{-6}	**2**	0.0169×10^{-4}	**3.0001**	0.00029	**2.0074**

Table 1.4: Results of MMS ALE FSI u=w

In table 1.4 we check the spatial convergence keeping the time step constant as $\Delta t = 10^{-6}$. Increasing spatial points N from 2 to 16. The error $E_u and E_p$ decreases with increasing spatial points. Computed convergence k_u tends toward 3 and k_p tends towards 2.

Discussion of the MMS tests

The MMS test of the solid equation has a clear trend toward 2 in spatial direction, and 1 in temporal direction. The temporal convergence rate k is not exactly 1, and could be because of the number of spatial points $N = 64$ is not high enough. With this in mind the trends shows convergence towards the theoretical convergence, which concludes that the solid equation has been implemented correctly.

The MMS test of the fluid equations computed from the reference domain shows trends in the spatial convergence toward 3 in the fluid velocity and 2 in pressure, which is expected. For the temporal convergence of the fluid equation the trend is towards 1 but is not exactly 1. The reason could be

that the number of spatial points are not high enough, and also that the fluid equations have been computed on a reference domain. Nonetheless the convergence rates are sufficient and giving the conclusion that the fluid equations are implemented correctly.

It should be noted that a more rigorous MMS test of the FSI problem would be to test the entire FSI problem, and not splitting the test into parts. To do a full MMS of the entire FSI problem, one needs to take into account the condition of continuity of velocity on the interface [4], the stresses need to equal on the interface and the flow needs to be divergence free. Manufacturing such a solution is very difficult [3]. The author has yet to find a paper that manufactures a solution fulfilling all the condition in a rigorous manner. For this reason the MMS was split into parts, and for the intended use the author finds the results from MMS tests sufficient .

1.2 Validation

After the code has been verified, we move on to Validation which is the process of determining if a model gives an accurate representation of real world physics within the bounds of the intended use [20]. A model trying to simulate a real world problem is only useful if we are certain that the solutions reflect the real world. The process of validation gives the evidence that the model indeed reflects the real world.

A computational model is made for a specific purpose, its only valid with respect to that purpose [12]. If the purpose of the computational model is complex and trying to answer multiple questions, then the validity needs to be determined for each question. The idea is to validate the solver, *brick by brick*, starting with simple testing of each part of the model and building more complexity and eventually testing the entire model.

Three aspects have been identified in the process of validating a computational model [20]. These are: quantifying the accuracy of the model by comparing responses with experimental responses, interpolation of the model to conditions corresponding to the intended use and determining the accuracy of the model for the conditions under which its meant to be used.

The process of Validation is also, as I have experienced, a way to figure out at

what size timestep and number of spatial points the model can handle. As we will see in the chapter all the benchmarks are run with different timesteps and number of cells to see how the model reacts. The problem with using benchmarks with known data for comparison is that we do not test the model blindly. It is easier to mold the model to the data we already have. As Oberkampf and Trucano in [20] puts it "Knowing the "correct" answer beforehand is extremely seductive, even to a saint". Knowing the limitations of our tests will therefore strengthen our confidence in the model. The process of verifying and validating, if one does not clearly know the bounds of sufficient accuracy, is an endless process. [20]

This chapter comprises of the implementation of a well known benchmark test case, used to validate the computational FSI model.

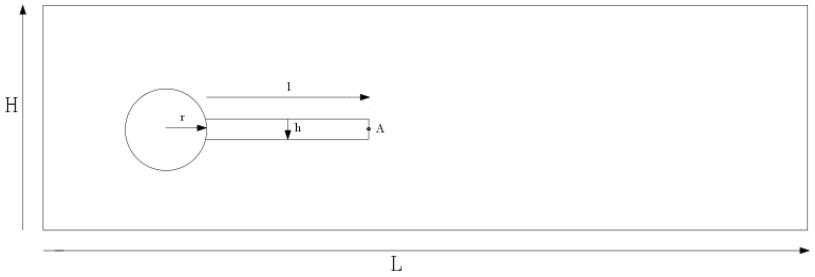
1.2.1 Fluid-Structure Interaction between an elastic object and laminar incompressible flow

This benchmark is based on the older CFD benchmark "Benchmark Computations of Laminar Flow Around a Cylinder" [18]. The benchmark is called "Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow", and provides a proposal for a FSI benchmark [9]. The authors provide a problem setup and results from their implementation, and opens for others to also contribute data.

The difference from the older benchmark is that now there has been added an elastic "flag" behind the cylinder in the flow direction. The benchmark provides a computational domain, and boundary conditions. It provides 9 subproblems in 3 parts split up into: a solid mechanical part named CSM(1,2 and 3), fluid mechanical part named CFD(1,2 and 3) and a full FSI part named FSI(1,2 and 3). The chapter starts by defining the computing domain, the boundary conditions and quantities for comparison. We then split up into the three parts, CSM, CFD and FSI, listing how the subtests are computed and providing results.

Problem Defintion

Domain



The computational domain consists of a circle with an elastic bar behind the circle. The circle is positioned at (0.2, 0.2) making it 0.05 of center from bottom to top. This shift in the domain is done to induce oscillations to an otherwise steady flow.

L	H	l	h	A
2.5	0.41	0.35	0.02	(0.2, 0.6)

Table 1.5: Domain parameters

Boundary conditions

A parabolic profile has been prescribed to the inlet velocity that increases from $t = [0, 2]$ and is kept constant after $t = 2$. The fluid velocity on upper and lower walls are set to zero, normally called a "no slip" condition.

$$u(0, y) = 1.5u_0 \frac{y(H - y)}{(\frac{H}{2})^2}$$
$$u(0, y, t) = u(0, y) \frac{1 - cos(\frac{\pi}{2}t)}{2} \text{ for } t < 2.0$$
$$u(0, y, t) = u(0, y) \text{ for } t \leq 2.0$$

Quantities for comparison

When the fluid moves around the circle and bar it exerts a frictional force. These forces are split into drag and lift and calculated as follows:

$$(F_d, F_L) = \int_S \sigma_f n dS$$

where S is the part of the circle and bar in contact with the fluid.
We set a point $A = (0.2, 0.6)$ on the right side of the bar. Where this point is in the spatial direction gives a value for how much the bar has deformed. For some given inflow conditions, unsteady solutions appear. For the unsteady solutions the values, meaning drag and lift, and displacement in x and y directions, are represented by the mean and amplitude values:

$$mean = \frac{1}{2}(max + min) \tag{1.6}$$

$$amplitude = \frac{1}{2}(max - min) \tag{1.7}$$

$$\tag{1.8}$$

In each test the values denoted as **ref** are the values taken from the original benchmark proposal paper [9]. Elements are the computational elements in the mesh and dofs are the *degrees of freedom* often simply called the number of unknowns.

CFD test cases

The CFD tests can be simulated in two ways. The first way assumes the bar to be rigid object, meaning that the computational domain is just the fluid domain, and a no slip condition has been set on the interface. The other way, which is implemented in this thesis, is by computing the problem as a full FSI problem by setting $\rho_s = 10^6$ and $\mu_s = 10^{12}$, such that the bar is almost completely rigid, only giving rise to very small deformation (in the $10^{-9} - 10^{-10}$ range).

The CFD tests cases, CFD1 and CFD2 are simulated with Reynolds numbers 20 and 100 converging to steady solutions. The CFD3 test case has a Reynolds number 200 which will induce oscillations behind the circle and bar, giving fluctuations in the fluid velocity, hence giving unsteady solutions.

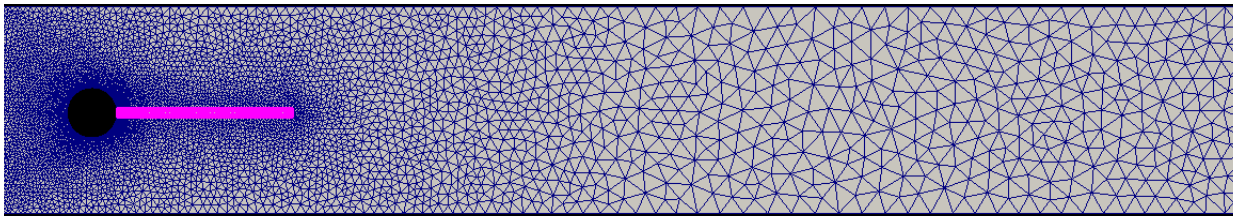


Figure 1.1: Picture of entire FSI computing domain with 11556 cells

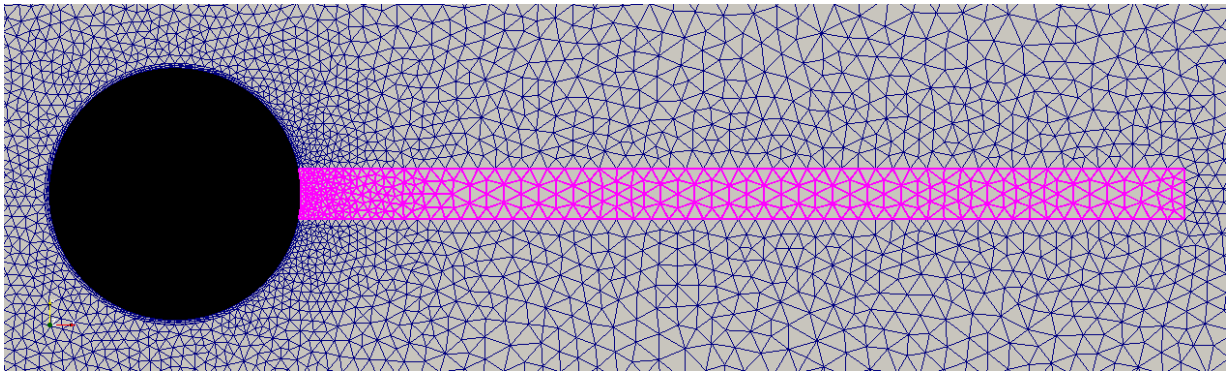


Figure 1.2: Picture of FSI computing domain with 11556 cells, zoomed in with the solid domain marked in pink. Around the circle we can see a small boundary layer with the width of 2 cells

Parameters	CFD1	CFD2	CFD3
$\rho_f[10^3 \frac{kg}{m^3}]$	1	1	1
$\nu_f[10^{-3} \frac{m^2}{s}]$	1	1	1
$U[\frac{m}{s}]$	0.2	1	2
$\mathrm{Re} = \frac{Ud}{\nu_f}$	20	100	200

Table 1.6: Summary of all the parameters in CFD tests

elements	**dofs**	**Drag**	**Lift**
2474	21749	14.059	1.100
7307	63365	14.110	1.080
11556	99810	14.200	1.1093
ref		**14.29**	**1.119**

Table 1.7: Results of CFD1 case run as full FSI, with almost rigid bar

Table 1.7 shows the results for the CFD1 testcase, showing convergence towards the referential values in Drag and Lift for increasing elements and dofs.

elements	**dofs**	**Drag**	**Lift**
2474	21749	134.9	10.38
7307	63365	135.4	10.0
11556	99810	136.1	10.41
ref		**136.7**	**10.53**

Table 1.8: Results of CFD2 case run as full FSI with almost rigid bar

Table 1.8 shows the results for CFD2 tending towards the referential values in Drag and Lift for increasing elements and dofs.

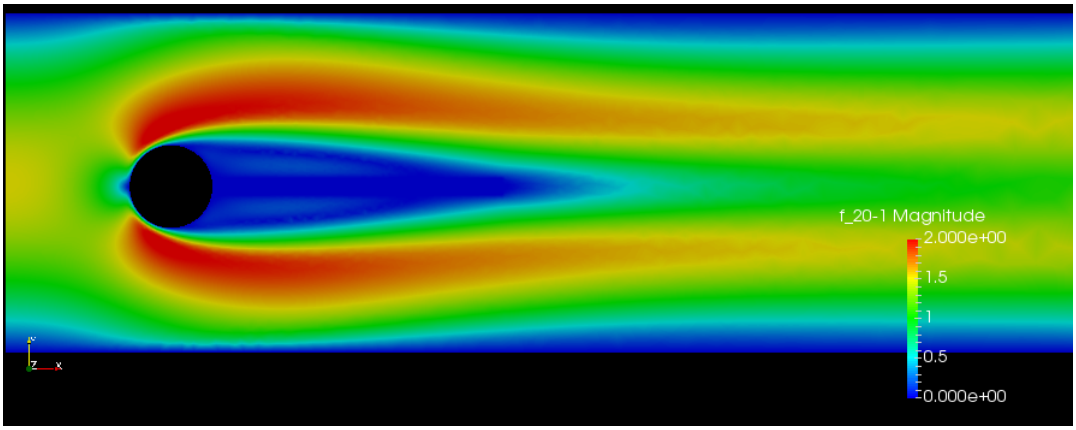


Figure 1.3: CFD2 steady state case fluid flow with 11556 cells

In the CFD2 case notice, in figure 1.2.1, that since the circle and bar are positioned non symmetric in the y direction there is more fluid flowing closer to the upper boundary. This non symmetry is what induces oscillations when the fluid inlet velocity is higher. The CFD2 case has an inlet velocity just below the point of inducing oscillations.

elements	**dofs**	**Drag**	**Lift**
2474	21749	434.42 ± 4.28	-15.63 ± 407.59
7307	63365	435.54 ± 5.06	-11.77 ± 425.73
11556	99810	438.13 ± 5.42	-10.01 ± 435.40
ref	ref	$\mathbf{439.45 \pm 5.61}$	$\mathbf{-11.893 \pm 437.81}$

Table 1.9: Results of unsteady state case CFD 3 with $\Delta t = 0.01$

Table 1.9 shows the results for the CFD3 testcase with $\Delta t = 0.001$, the results show clear convergence toward the **ref** value for increased number of cells and dofs.

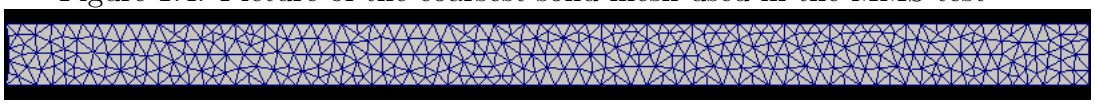
Discussion of CFD results

The CFD1 and CFD2 steady solution test cases gives satisfactory results compared to the referential values given in the benchmark paper.

CSM test cases

The CSM tests are calculated using only the bar as computing domain. A body force f_s is set a gravitational force g, which has been kept fixed throughout the CSM tests, changing only the material parameters of the solid. The tests CSM1 and CSM2 gives rise to steady state solutions. The difference between them is a more slender bar. The CSM 3 gives unsteady solutions and even more slender causing the bar, if modeled correctly to down and back up. Since there is no friction from the movement in the model it should,if energy is preserved, make the down to a point and up to its initial state, and repeat this motion.

Figure 1.4: Picture of the coarsest solid mesh used in the MMS test



Parameters	CSM1	CSM2	CSM3
$\rho_f[10^3 \frac{kg}{m^3}]$	1	1	1
$\nu_f[10^{-3} \frac{m^2}{s}]$	1	1	1
u_0	0	0	0
$\rho_s[10^3 \frac{kg}{m^3}]$	1	1	1
ν_s	0.4	0.4	0.4
$\mu_s[10^6 \frac{m^2}{s}]$	0.5	2.0	0.5
g	2	2	2

Table 1.10: Summary table of the parameters used in the CSM tests

elements	dofs	$d_x(A)[\times 10^{-3}]$	$d_y(A)[\times 10^{-3}]$
725	1756	-5.809	-59.47
2900	6408	-6.779	-64.21
11600	24412	-7.085	-65.63
46400	95220	-7.116	-65.74
ref	**ref**	**-7.187**	**-66.10**

Table 1.11: Results of the steady CSM1 case from coarse to fine mesh.

Elements	Dofs	$d_x(A)[\times 10^{-3}]$	$d_y(A)[\times 10^{-3}]$
725	1756	-0.375	-15.19
2900	6408	-0.441	-16.46
11600	24412	-0.462	-16.84
46400	95220	-0.464	-16.87
ref	**ref**	**-0.469**	**-16.97**

Table 1.12: Results of the steady CSM2 case from coarse to fine mesh.

elements	dofs	$d_x(A)[\times 10^{-3}]$	$d_y(A)[\times 10^{-3}]$
725	1756	-11.743 ± 11.744	-57.952 ± 58.940
2900	6408	-13.558 ± 13.559	-61.968 ± 63.440
11600	24412	-14.128 ± 14.127	-63.216 ± 64.744
46400	95220	-14.182 ± 14.181	-63.305 ± 64.843
ref		$\mathbf{-14.305 \pm 14.305}$	$\mathbf{-63.607 \pm 65.160}$

Table 1.13: Results of the unsteady CSM3 case with mesh from coarse to fine.

The tables 1.11 ,1.12 and 1.12 shows the results of the CSM1, CSM2 and CSM3 cases respectively. All three show a clear tendency towards the referential values when increasing the number of elements.

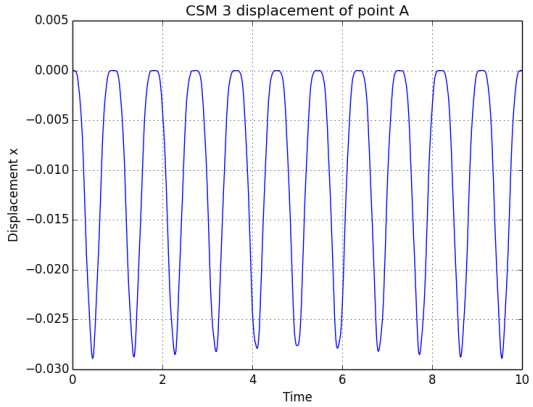


Figure 1.5: Displacement in x direction, timeinterval (0,10)

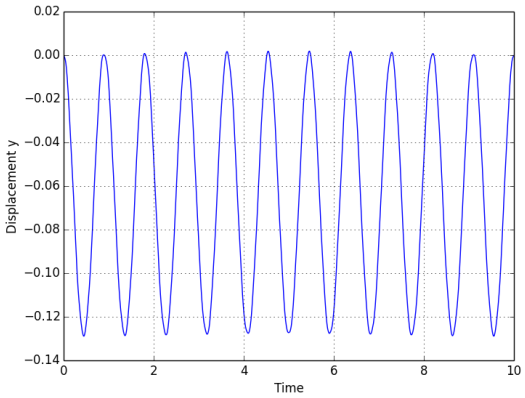


Figure 1.6: Displacement in y direction, timeinterval (0,10)

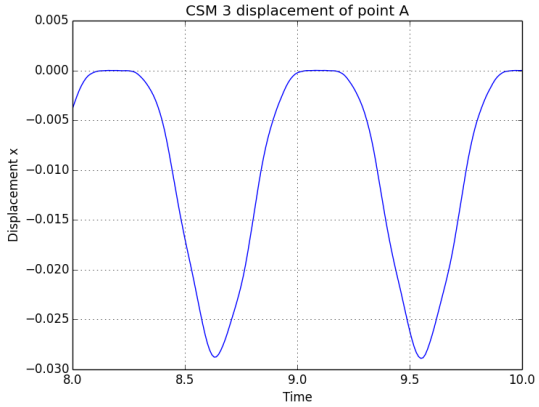


Figure 1.7: Displacement in x direction, timeinterval (8,10)

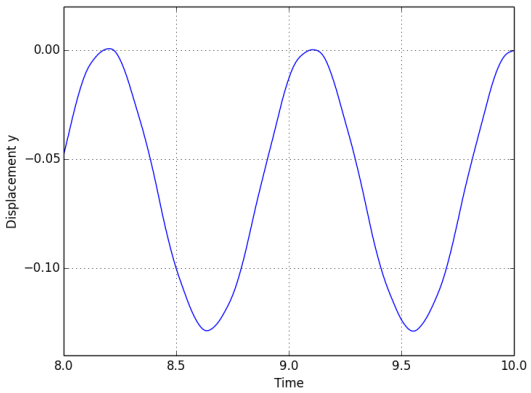


Figure 1.8: Displacement in y direction, timeinterval (8,10)

Figure 1.9: Plots of the results for CSM3 showing Displacement of point A

The figure 1.9 is of displacement at the point A in x and y direction of the CSM3 test. The CSM3 test was run with Crank-Nicholson, $\theta = 0.5$, and it can be seen that in the y displacement the bar returns to it initial state, that is with zero displacement. This results indicates that the energy in the system has been preserved.

Discussion of the CSM results

FSI tests

The FSI tests are run with 2 different inflows conditions. FSI1 gives a steady state solution while the others are unsteady. FSI-2 gives the largest deformation is therefore considered the most difficult of the three [15], giving deformations of 2.5 times greater than the flag height. Vinje 2016 [25] reported not being able to compute FSI-2 model due to shortcomings in the linear elasticity solid model. The FSI-3 test has the highest inflow speed giving medium deformations but more rapid oscillations.

Parameters	FSI1	FSI2	FSI3
$\rho_f[10^3 \frac{kg}{m^3}]$	1	1	1
$\nu_f[10^{-3} \frac{m^2}{s}]$	1	1	1
u_0	0.2	1	2
$\text{Re} = \frac{Ud}{\nu_f}$	20	100	200
$\rho_s[10^3 \frac{kg}{m^3}]$	1	10	1
ν_s	0.4	0.4	0.4
$\mu_s[10^6 \frac{m^2}{s}]$	0.5	0.5	2

Table 1.14: FSI Parameters

FSI1

Cells	Dofs	$d_x(A)[\times 10^{-3}]$	$d_y(A)[\times 10^{-3}]$	Drag	Lift
2474	21749	0.0229	0.8265	14.0581	0.7546
7307	63365	0.02309	0.7797	14.1077	0.7518
11556	99810	0.02295	0.8249	14.2046	0.7613
ref	**ref**	**0.0227**	**0.8209**	**14.295**	**0.7638**

Table 1.15: Results of FSI 1 test case

FSI2

Cells	Dofs	$d_x(A)[\times 10^{-3}]$	$d_y(A)[\times 10^{-3}]$	Drag	Lift
2474	21749	-15.26 ± 13.44	1.34 ± 82.38	157.02 ± 14.79	-1.426 ± 258.4
7307	63365	-14.96 ± 13.24	1.01 ± 81.67	159.01 ± 16.33	1.88 ± 254.2
11556	99810	-14.96 ± 13.23	1.29 ± 81.9	161.09 ± 17.66	0.06 ± 255.78
ref	**ref**	$\mathbf{-14.58 \pm 12.44}$	$\mathbf{1.23 \pm 80.6}$	$\mathbf{208.83 \pm 73.75}$	$\mathbf{0.88 \pm 234.2}$

Table 1.16: FSI2 test case results, $\Delta t = 0.01$, using the harmonic lifting operator

Cells	Dofs	$d_x(A)[x10^{-3}]$	$d_y(A)[x10^{-3}]$	Drag	Lift
2474	21749	-15.10 ± 13.32	1.16 ± 82.46	159.53 ± 17.44	0.68 ± 259.10
7307	63365	-14.85 ± 13.14	1.21 ± 81.72	160.72 ± 17.84	0.93 ± 255.14
11556	99810	-14.83 ± 13.11	1.24 ± 81.6	161.50 ± 18.17	0.62 ± 254.40
ref	**ref**	$\mathbf{-14.58 \pm 12.44}$	$\mathbf{1.23 \pm 80.6}$	$\mathbf{208.83 \pm 73.75}$	$\mathbf{0.88 \pm 234.2}$

Table 1.17: FSI-2 with $\Delta t = 0.001$, using harmonic lifting operator

The tables 1.16 and 1.17 shows results for the FSI2 test case. The displacements in both results for both of the $\Delta t = 0.01$ and $\Delta t = 0.001$ show convergence towards the **ref** values. The lift converges more slowly towards the ref value, while the drag values are off by about a value of 45 in both cases.

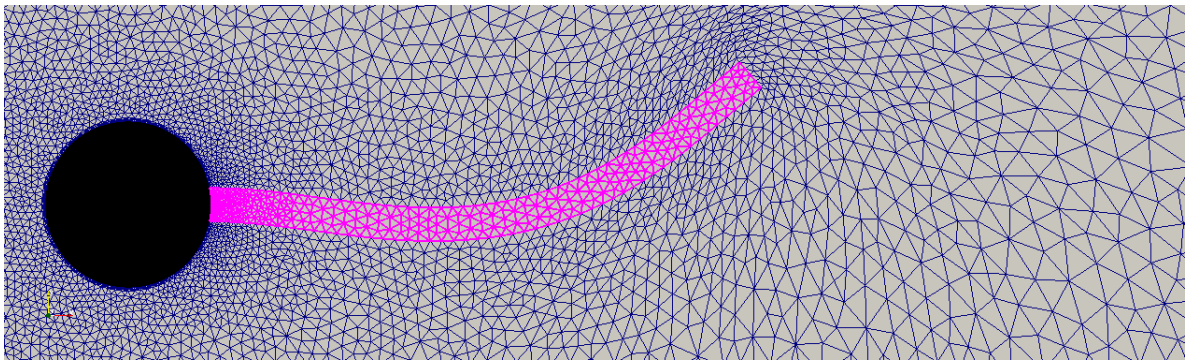


Figure 1.10: Deformation at $t = 9.70$sec. The bar is marked with pink colour and deformed using warp by vector in paraview.

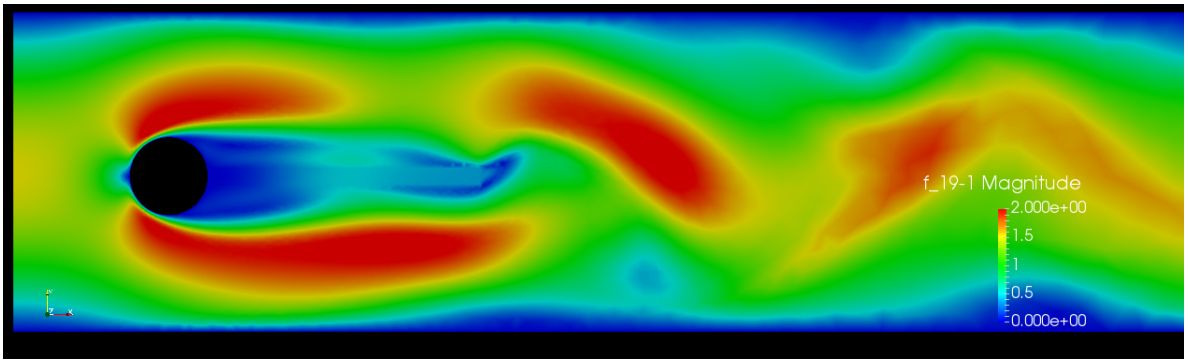


Figure 1.11: Fluid velocity at $t = 9.70$sec on the reference mesh

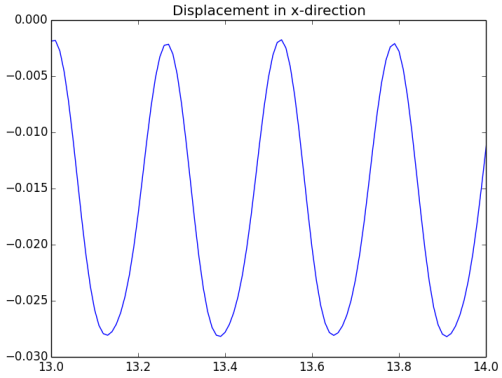


Figure 1.12: Displacement x

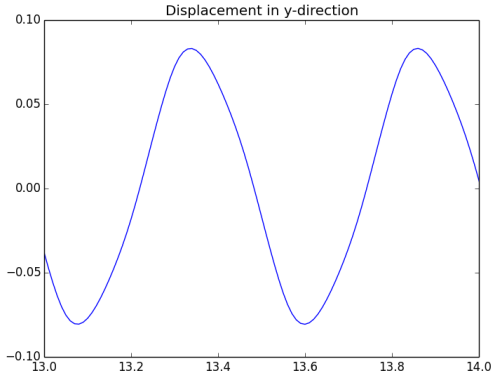


Figure 1.13: Displacement x

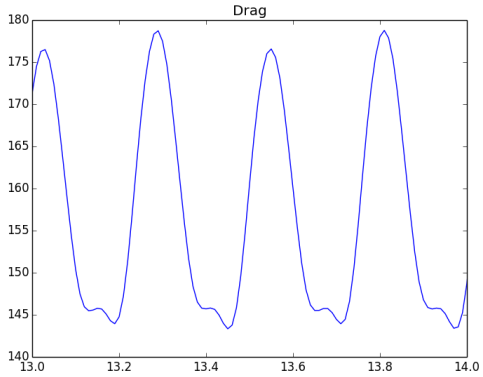


Figure 1.14: Drag

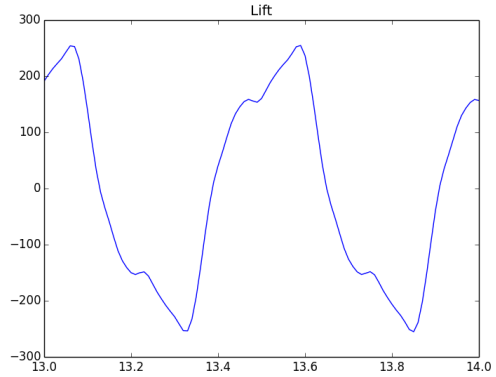


Figure 1.15: Lift

Figure 1.16: Plots of FSI2 result values for, $\Delta t = 0.001$, with 11556 elements

and figure **??**

FSI3 results

Cells	Dofs	$d_x(A)[\times 10?3]$	$d_y(A)[\times 10?3]$	Drag	Lift
2474	21249	-1.79 ± 1.80	3.29 ± 2.64	439.36 ± 12.04	1.96 ± 142.31
7307	63365	-2.48 ± 2.48	1.64 ± 3.28	449.77 ± 18.02	3.41 ± 153.47
11556	99810	-2.47 ± 2.45	1.27 ± 3.28	456.60 ± 18.73	1.55 ± 153.46
ref	ref	$\mathbf{-2.69 \pm 2.56}$	$\mathbf{1.48 \pm 34.38}$	$\mathbf{457.3 \pm 22.66}$	$\mathbf{2.22 \pm 149.78}$

Table 1.18: FSI3 unsteady test case results with $\Delta t = 0.01$, with harmonic lifting operator

Comment on FSI tests

A very important thing to notice about this paper [9] is that it is a proposal for a benchmark, as it is called "Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow". So the point of the paper is give a specific problem setup which others can contribute result-data. A paper was published in 2010 by J. Hron, Turek, et al, [22] that compared results of different discretizations and solution approaches. This paper [22] gives 7 different methods and results for two of the FSI test cases. They state in the numerical results that "However, also clear differences between the different approaches with regard to accuracy are visible. Particularly for the drag and lift values, which lead to differences of up to order 50%, and also for the displacement valueswhich are in the range of 10% errors.". With this in mind it is important to know that the referential values used are only those reported from the original paper, which only looked at one implementation. While the paper which compares results, only 2 of the 7 contributions were schemes of monolithic nature, which are the closest one should refer to in this thessis. In the Appendix **??** is a copy of the results from the paper comparing schemes, showing different results for different discretization, with different time steps and unknowns.

The FSI1 test gives a low fluid velocity and gives very low displacements. FSI1 is therefore not a rigid test for FSI. In fact I experienced in the beginning of making the FSI solver, that even with a wrong implementation i got good FSI1 results. However it is a good test for early checks, because if FSI1 is wrong the rest will definitely not work.

For the FSI2 test case we only have results from the initial Hron and Turek paper [9]. As previously stated the results for the FSI3 case differ by in some

cases 50% for Drag and Lift. With this in mind, in the FSI2 case I am off by less then 10% for displacement in x and y direction and for Lift. While Drag is off by about 50%. It is reasonable to assume that since there were such differences in the results for different implementations for the FSI3 results, we would expect similar behavior in the FSI2 results.

If we compare the results reported in the FSI3 case in the inital paper by Hron and Turek 2006, to their reported results in 2010 **??** implementation 3, we can see that they do not report the same results same scheme, leading one believe that that they have changed their implementation a bit.

Bibliography

[1] K. Yusuf Billah. Resonance, Tacoma Narrows bridge failure, and undergraduate physics textbooks. *American Journal of Physics*, 59(2):118, 1991.

[2] David J Charlesworth. Solution of the Incompressible Navier- Stokes Equations on Unstructured Meshes by. (August), 2003.

[3] S. Étienne, A. Garon, and D. Pelletier. Some manufactured solutions for verification of fluid-structure interaction codes. *Computers and Structures*, 106-107:56–67, 2012.

[4] Stéphane Étienne, D Tremblay, and Dominique Pelletier. Code Verification and the Method of Manufactured Solutions for Fluid-Structure Interaction Problems. *36th AIAA Fluid Dynamics Conference and Exhibit*, (June):1–11, 2006.

[5] Charles L. Fefferman. Existence and smoothness of the Navier-Stokes equation. *The millennium prize problems*, (1):1–5, 2000.

[6] Miguel A. Fernández, Jimmy Mullaert, and Marina Vidrascu. Explicit robin-neumann schemes for the coupling of incompressible fluids with thin-walled structures. *Computer Methods in Applied Mechanics and Engineering*, 267:566–593, 2013.

[7] Miguel A. Fernández, Jimmy Mullaert, and Marina Vidrascu. Generalized Robin-Neumann explicit coupling schemes for incompressible fluid-structure interaction: Stability analysis and numerics. *International Journal for Numerical Methods in Engineering*, 101(3):199–229, 2015.

[8] G Holzapfel. Nonlinear solid mechanics: A continuum approach for engineering, 2000.

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

[10] Jie Liu, Rajeev K. Jaiman, and Pardha S. Gurugubelli. A stable second-order scheme for fluid-structure interaction with strong added-mass effects. *Journal of Computational Physics*, 270:687–710, 2014.

[11] Anders Logg, Harish Narayanan, Marie Rognes, Johannes Ring, Kristian B. Ølgaard, and Garth N. Wells. FEniCS Project, 2011.

[12] Cm Macal. Proceedings of the 2005 Winter Simulation Conference ME Kuhl, NM Steiger, FB Armstrong, and JA Joines, eds. *Simulation*, pages 1643–1649, 2005.

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

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

[15] T Richter and T Wick. On time discretizations of Fluid-structure interactions. *Multiple Shooting and Time Domain Decomposition MEthods*, pages 377–400, 2013.

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

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

[18] M. Schäfer, S. Turek, F. Durst, E. Krause, and R. Rannacher. Benchmark Computations of Laminar Flow Around a Cylinder. pages 547–566, 1996.

[19] Natural Sciences. A Newton ' s Method Finite Element Algorithm for Fluid-Structure Interaction. (October), 2012.

[20] Noelle Selin. Verification and Validation. (February), 2014.

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

[22] S Turek, J Hron, M Razzaq, H Wobker, and M Sch. Fluid Structure Interaction II. 73, 2010.

[23] Boris Valkov, Chris H Rycroft, and Ken Kamrin. Eulerian method for fluid – structure interaction and submerged solid – solid contact problems.

[24] E. H. van Brummelen. Added Mass Effects of Compressible and Incompressible Flows in Fluid-Structure Interaction. *Journal of Applied Mechanics*, 76(2):021206, 2009.

[25] V Vinje. Simulating Cerebrospinal Fluid Flow and Spinal Cord Movement Associated with Syringomyelia. 2016.

[26] Frank M. White. Chapter 3 - Solutions of the Newtonian viscous-flow equa- tions. *Viscous Fluid Flow*, (5), 2006.

[27] Thomas Wick. Adaptive Finite Element Simulation of Fluid-Structure Interaction with Application to Heart-Valve Dynamics. *Institute of Applied Mathematics, University of Heidelber*, page 157, 2011.

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