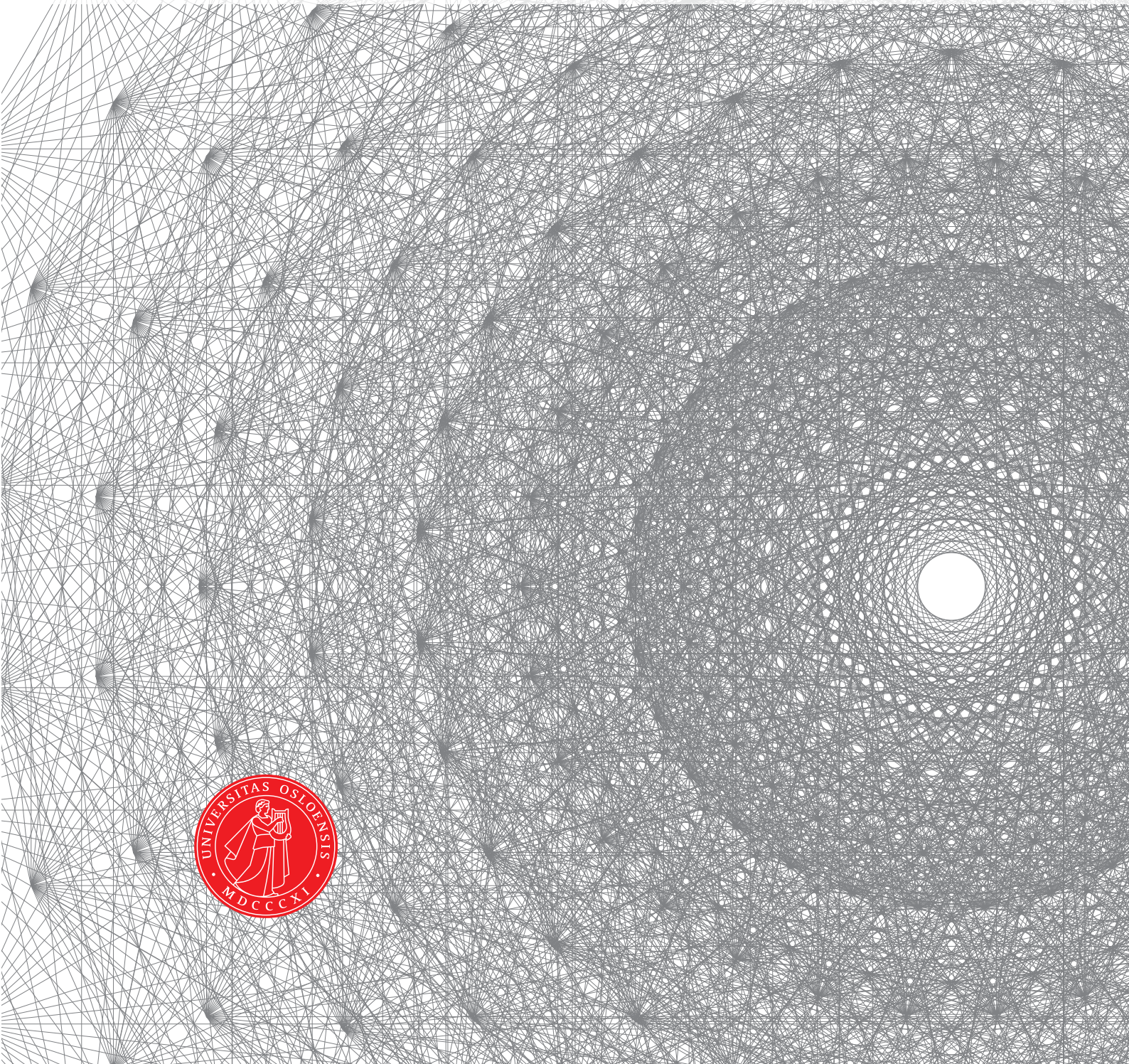
UiO : Department of Mathematics
University of Oslo

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

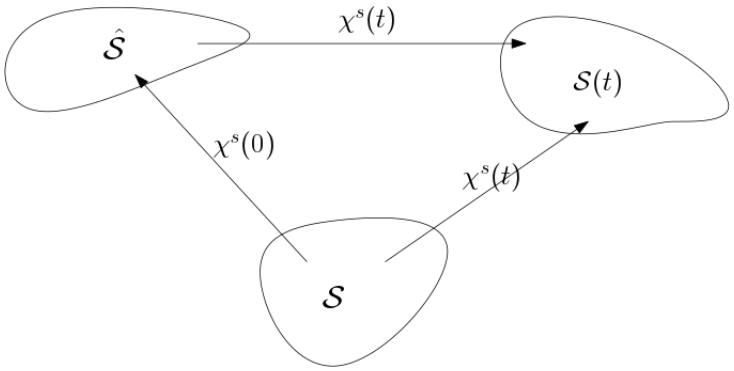
Continuum mechanics in different frames of reference

All materials are made up of atoms, and between atoms there is space. The laws that govern these atoms are complex and are very difficult to model. However materials like solids and fluids can be modeled if we assume them to exist as a continuum. This means that we assume that there exist no space inside the materials and they fill completely up the space they occupy. We can use mathematics with basic physical laws to model fluids and solids when they are assumed continuous. These laws are generally expressed in two frames of reference, Lagrangian and Eulerian. To exemplify these frameworks we can imagine a river running down a mountain. In the Eulerian framework we are the observer standing still besides the river looking at the flow. We are not interested in each fluid particle but only how the fluid acts as a whole flowing down the river. This approach fits the fluid problem as we can imagine the fluid continuously deforming along the river side.

In the Lagrangian description we have to imagine ourselves on a leaf going down the river with the flow. Looking out as the mountain moves and we stand still compared to the fluid particles. This description fits a solid problem nicely since we are generally interested in where the solid particles are in relation to each other. Modeling for instance a beam attached to a wall at one end and a weight at the other end. We can imagine the beam bending and to model this bending we need to where all the particles are compared to each other. The more the particles move in relation to each other the more stress there is in the beam.

In this chapter I will introduce both of these frameworks and the equations which are needed to model Fluid and Structure separately. I start with the Lagrangian description, by providing a short introduction to Lagrangian physics for the sake of completeness. Then introduce the solid and fluid equations. For a more detailed look on Lagrangian physics and the solid equation see [4].

1.1 Lagrangian physics



Let $\hat{\mathcal{S}}$, $\mathcal{S}$, $\mathcal{S}(t)$ be the initial stress free configuration of a given body, the reference and as the current configuration respectively. I define a smooth mapping from the reference configuration to the current configuration:

$$\chi^s(\mathbf{X}, t) : \hat{\mathcal{S}} \to \mathcal{S}(t) \tag{1.1}$$

Where $\mathbf{X}$ denotes a material point in the reference domain and χ^s denotes the mapping from the reference configuration to the current configuration. Let $d^s(\mathbf{X}, t)$ denote the displacement field which describes deformation on a body. The mapping χ^s can then be specified from its current position plus the displacement from that position:

$$\chi^s(\mathbf{X}, t) = \mathbf{X} + d^s(\mathbf{X}, t) \tag{1.2}$$

which can be written in terms of the displacement field:

$$d^s(\mathbf{X}, t) = \chi^s(\mathbf{X}, t) - \mathbf{X} \tag{1.3}$$

Let w($\mathbf{X}$,t) be the domain velocity which is the partial time derivative of the displacement:

$$w(\mathbf{X}, t) = \frac{\partial \chi^s(\mathbf{X}, t)}{\partial t} \tag{1.4}$$

1.1.1 Deformation gradient

To describe the rate at which a body undergoes deformation I will need to define a deformation gradient. Let $d(\mathbf{X}, t)$ be a differentiable deformation field in a given body, the deformation gradient is then:

$$F = \frac{\partial \chi^s(\mathbf{x}, t)}{\partial \mathbf{X}} = \frac{\partial \mathbf{X} + d^s(\mathbf{X}, t)}{\partial \mathbf{X}} = I + \nabla d(\mathbf{X}, t) \tag{1.5}$$

which denotes relative change of position under deformation in a Lagrangian frame of reference. We can observe that when there is no deformation. The deformation gradient F is simply the identity matrix.

We also need a way to change between volumes, from the reference ($\int dv$) to current ($\int dV$) configuration. This is defined with the Jacobian determinant, which is the determinant of the of the deformation gradient F:

$$J = \det(F) \tag{1.6}$$

The Jacobian determinant is used to change between volumes, assuming infinitesimal line and area elements in the current ds, dx and reference dV, dX configurations. The Jacobian determinant is therefore known as a volume ratio.

1.1.2 Strain

The relative change of location between two particles is called strain. Strain, strain rate and deformation is used to describe the relative motion of particles in a continuum. This is the fundamental quality that causes stress [9].

If we observe two neighboring points $\mathbf{X}$ and $\mathbf{Y}$. I can describe $\mathbf{Y}$ with:

$$\mathbf{Y} = \mathbf{Y} + \mathbf{X} - \mathbf{X} = \mathbf{X} + |\mathbf{Y} - \mathbf{X}| \frac{\mathbf{Y} - \mathbf{X}}{|\mathbf{Y} - \mathbf{X}|} = \mathbf{X} + d\mathbf{X} \tag{1.7}$$

Let $d\mathbf{X}$ be denoted by:

$$d\mathbf{X} = d\epsilon \mathbf{a}_0 \tag{1.8}$$

$$d\epsilon = |\mathbf{Y} - \mathbf{X}| \tag{1.9}$$

$$\mathbf{a}_0 = \frac{\mathbf{Y} - \mathbf{X}}{|\mathbf{Y} - \mathbf{X}|} \tag{1.10}$$

where $d\epsilon$ is the distance between the two points and $\mathbf{a}_0$ is a unit vector

We see now that $d\mathbf{X}$ is the distance between the two points times the unit vector or direction from $\mathbf{X}$ to $\mathbf{Y}$.

A certain motion transform the points $\mathbf{Y}$ and $\mathbf{X}$ into the displaced positions $\mathbf{x} = \chi^s(\mathbf{X}, t)$ and $\mathbf{y} = \chi^s(\mathbf{Y}, t)$. Using Taylor?s expansion $\mathbf{y}$ can be expressed in terms of the deformation gradient:

$$\begin{aligned} \mathbf{y} =& \chi^s(\mathbf{Y}, t) = \chi^s(\mathbf{X} + d\epsilon \mathbf{a}_0, t) && (1.11) \\ =& \chi^s(\mathbf{X}, t) + d\epsilon F \mathbf{a}_0 + \mathcal{O}(\mathbf{Y} - \mathbf{X}) && (1.12) \end{aligned}$$

where $\mathcal{O}(\mathbf{Y} - \mathbf{X})$ refers to the small error that tends to zero faster than $(\mathbf{X} - \mathbf{Y}) \to \mathcal{O}$.

If I set $\mathbf{x} = \chi^s(\mathbf{X}, t)$ It follows that:

$$\begin{aligned} \mathbf{y} - \mathbf{x} =& d\epsilon F \mathbf{a}_0 + \mathcal{O}(\mathbf{Y} - \mathbf{X}) && (1.13) \\ =& F(\mathbf{Y} - \mathbf{X}) + \mathcal{O}(\mathbf{Y} - \mathbf{X}) && (1.14) \end{aligned}$$

Let the **stretch vector** be $\lambda_{\mathbf{a}0}$, which goes in the direction of $\mathbf{a}_0$:

$$\lambda_{\mathbf{a}0}(\mathbf{X}, t) = F(\mathbf{X}, t)\mathbf{a}_0 \tag{1.15}$$

If we look at the square of λ:

$$\begin{aligned} \lambda^2 = \lambda_{\mathbf{a}0}\lambda_{\mathbf{a}0} &= F(\mathbf{X}, t)\mathbf{a}_0 F(\mathbf{X}, t)\mathbf{a}_0 && (1.16) \\ &= \mathbf{a}_0 F^T F \mathbf{a}_0 = \mathbf{a}_0 C \mathbf{a}_0 && (1.17) \end{aligned}$$

We have not introduced the important right Cauchy-Green tensor:

$$C = F^T F \tag{1.18}$$

Since $\mathbf{a}_0$ is just a unit vector, we see that this measures the squared length of change under deformation. We see that in order to determine the stretch one needs only the direction of $\mathbf{a}_0$ and the tensor C. C is also symmetric and positive definite $C = C^T$. I also introduce the Green-Lagrangian strain tensor E:

$$E = \frac{1}{2}(F^T F - I) \tag{1.19}$$

which is also symmetric since C and I are symmetric.

1.1.3 Stress

While strain, deformation and strain rate only describe the relative motion of particles in a given volume. Stress give us the internal forces between neighboring particles. Stress is responsible for deformation and is therefore crucial in continuum mechanics. The unit of stress is force per area.

I introduce the Cauchy stress tensor σ_s by the constitutive law of St. Venant-Kirchhoff hyperelastic material model:

$$\sigma_s = \frac{1}{J} F(\lambda_s(trE)I + 2\mu_s E)F^T \tag{1.20}$$

If we use this tensor on an area that is taking the stress tensor times the normal vector $\sigma_s \mathbf{n}$ we get the forces acting on that area.

Using the deformation gradient and the Jacobian determinant., I get the first Piola-Kirchhoff stress tensor P:

$$P = J\sigma F^{-T} \tag{1.21}$$

This is known as the *Piola Transformation* and maps the tensor into a Lagrangian formulation which will be used when stating the solid equation.

I also introduce the second Piola-Kirchhoff stress tensor S:

$$S = JF^{-1}\sigma F^{-T} = F^{-1}P = S^T \tag{1.22}$$

from this relation the first Piola-Kirchhoff tensor can be expressed by the second:

$$P = FS \tag{1.23}$$

1.2 Solid equation

The solid equation describes the motion of a solid. It is derived from the principles of conservation of mass and momentum. Stated in the Lagrangian reference system [9]:

$$\rho_s \frac{\partial d^2}{\partial t^2} = \nabla \cdot (P) + \rho_s f \tag{1.24}$$

written in terms of the deformation d, where I used the first Piola-Kirchhoff stress tensor. f is a force acting on the solid body. Considering the material law from the previous section. The stresses depend on the strain which again depends of the displacements d. The solid equation will be finalized by stating the different boundary conditions needed to solve a solid problem.

1.2.1 Solid Boundary Conditions

The solid moves within the boundary of $\partial\mathcal{S}$. On the Dirichlet boundary $\partial\mathcal{S}_D$ we impose a given value. These can be initial conditions or set to a value set to a designated spot on the domain. These initial conditions are defined for d and w:

$$d = d_0 \text{ on } \partial\mathcal{S}_D \tag{1.25}$$

$$w(\mathbf{X}, t)_0 = \frac{\partial d(t=0)}{\partial t} \text{ on } \partial\mathcal{S}_D \tag{1.26}$$

The forces on the boundaries need to equal an eventual external force $\mathbf{f}$. These are are enforced on the Neumann boundaries:

$$P \cdot \mathbf{n} = f \text{ on } \partial\mathcal{S}_N$$

1.3 Fluid equations

The fluid equation will be stated in an Eulerian framework. In this framework the domain has fixed points where the fluid passes through. The Navier-Stokes(N-S) equations are, like the solid equation, derived using principles of mass and momentum conservation. N-S describes the velocity and pressure in a given fluid continuum. They are here written in the fluid time domain $\mathcal{F}$ as an incompressible fluid:

$$\rho_f\big(\frac{\partial u}{\partial t} + u \cdot \nabla u\big) = \nabla \cdot \sigma_f + f \tag{1.27}$$

$$\nabla \cdot u = 0 \tag{1.28}$$

where u is the fluid velocity, p is the fluid pressure, ρ stands for density which, will be kept constant. f is body force and σ_f is the Cauchy stress tensor, $\sigma_f = \mu_f(\nabla u + \nabla u^T) - pI$, where I is the Identity matrix.

There does not yet exist an analytical solutions to the N-S equations, only simplified problems can be solved [12]. Actually there is a prize set out by the Clay Mathematics Institute of 1 million dollars to whomever can show the existence and smoothness of Navier-Stokes solutions [3], as apart of their millennium problems. Nonetheless this does not stop us from discretizing and solving N-S numerically. The field of modeling fluids numerically is known

as Computational Fluid Dynamics(CFD). And CFD is extensively used in for instance weather forecasts, construction of aircraft, and biomedical engineering.

One difficulty in the N-S equations is the nonlinearity appearing in the convection term on the left hand side. Non-linearity is most often handled using Newtons method or Picard iteration.

Before these equations can be solved we need to impose boundary conditions.

1.3.1 Fluid Boundary conditions

Lastly I need to impose boundary conditions. The fluid flows within the boundary noted as $\partial\mathcal{F}$. On the Dirichlet boundary $\partial\mathcal{F}_D$ we impose a given value. This can be initial conditions or set to zero as on walls with "no slip" condition. These conditions are defined for u p and d:

$$u = u_0 \text{ on } \partial\mathcal{F}_D$$

$$p = p_0 \text{ on } \partial\mathcal{F}_D$$

The forces on the boundaries need to equal an eventual external force $\mathbf{f}$. These are are enforced on the Neumann boundaries:

$$\sigma \cdot \mathbf{n} = f \text{ on } \partial\mathcal{F}_N$$

Chapter 2

Verification and Validation in Computational Fluid Structure Interaction.

When we set out to solve a real world problem with numerical computing, we start by defining the mathematics, we implement the equations numerically and solve the equations on a computer. We use the solutions to extract data that will answer the questions we set out to answer. A problem then immediately arises, is the solution correct? To answer this we need to answer another question, are the equations solved correct numerically, if so is the problem defined correct mathematically, that is in accordance with the governing laws and equations? Without answering these questions, being confident that your solutions are correct is difficult [11]. The goal of this section will hence be to verify and validate the different numerical schemes.

I start with the process of Verification where the fluid and structure parts of the code will be verified. Following will be Validation of the code where I look at a well known benchmark testing the fluid and structure parts individually and as a full FSI problem. After that I investigate the impact of using different time order schemes, and different mesh motion techniques.

2.1 Verification

Verification is the process of determining wether or not the implementation of numerical algorithms in computer code, is done correctly. [7]. In verification we get evidence that the numerical model derived from mathematics is solved correctly by the computer. The strategy will be to identify, quantify and reduce errors cause by mapping a mathematical model to a computational model. Verification does not address wether or not the computed solutions are in alignment with the real world. It only tells us that our model is computed correctly or not. To verify that we are computing correctly we can compare our computed solution to an exact solution. But the problem is that there are no known exact solution to for instance the Navier-Stokes equations, other than for very simplified problems.

In Verification there are multiple classes of test that can be performed, and the most rigorous is the *Method of manufactured solution*(MMS) [7]. 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 than run the PDE with the source term to get a solution hopefully matching the manufactured one. The manufactured solution does not need to have a physically realistic relation, since the solution deals only with the mathematics. The procedure is as follows [7]:

- We define a mathematical model on the form $L(u) = 0$ where $L(u)$ is a differential operator and u is a dependent variable.
- Define the analytical form of the manufactured solution $\hat{u}$
- Use the model $L(u)$ with $\hat{u}$ inserted to obtain an analytical source term $f = L(\hat{u})$
- Initial and boundary conditions are enforced from $\hat{u}$
- Then use this source term to calculate the solution u, $L(u) = f$

After the solution has been computed we perform systematic convergence tests [**?**]. The idea behind order of convergence tests is based on the behavior of the error between the manufactured exact solution and the computed solution. If we let u be the numerical solution and u_e be the exact solution, $||.||$ be the L^2 norm, we define the error as:

$$E = ||u - u_e|| \tag{2.1}$$

When we increase the number of spatial points ($\Delta x, \Delta y$ or Δz) or decrease timestep(Δt), we expect the error to get smaller. It is the rate of this error that lets us know wether the solution is converging correctly. If we assume that the number of spatial points are equal in all directions the error is expressed as

$$E = C_1 \Delta x^k + C_2 \Delta t^l \tag{2.2}$$

where $k = m+1$ and m is the polynomial degree of the spatial elements. The error is hence dependent on the number of spatial points and the timestep. If we for instance reduce Δt significantly, Δx will dominate, and Δt will be negligible. If we then look at two error where E_{n+1} has finer mesh than E_n, using (2.2):

$$\frac{E_{n+1}}{E_n} = \big(\frac{\Delta x_{n+1}}{\Delta x_n}\big)^k \tag{2.3}$$

$$k = \frac{log(\frac{E_{n+1}}{E_n})}{log(\frac{\Delta x_{n+1}}{\Delta x_n})} \tag{2.4}$$

We use k to find the observed order of convergence and match with the theoretical order of convergence for each given problem.

The manufactured solutions should be chosen to be non-trivial and analytic [10] [7]. The solutions should be so that no 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 equation.

We will generate a manufactured solution (pressure, viscosity and velocity field distributions) which satisfies the continuity equation on the deformed fluid domain, continuity of tractions and displacements at the fluid-solid interface

I begin with verifying the solid part of the code. Then the fluid part of the code will be tested with a given displacement, also testing the mappings between configurations. A full MMS of the entire FSI problem with all the conditions is very difficult [1]. One needs to take into account the condition of continuity of velocity on the interface [2]. The stresses need to equal on the interface and the flow needs to be divergence free. MMS of full FSI is therefore out the scope of this thesis.

2.2 Structure MMS

To do MMS of the solid I use the solid equation and make a sourceterm f_s:

$$\rho_s \frac{\partial u}{\partial t} - \nabla \cdot (P) = f_s$$

The solid variational formulation is written as:

$$\left(\rho_s \frac{\partial u}{\partial t}, \phi\right)_{\hat{\mathcal{S}}} + \left(P, \nabla\phi\right)_{\hat{\mathcal{S}}} = f_s \tag{2.5}$$

$$\left(u - \frac{\partial d}{\partial t}, \psi\right)_{\hat{\mathcal{S}}} = 0 \tag{2.6}$$

These equations are solved together and we solve for d and u. The functions u and d will be computed to match the source term. In the tables below we investigate convergence in space and time.

Even though we have two equations we do not make a source for the second. This is because the solutions are made to uphold criteria of $u = \frac{\partial d}{\partial t}$:

$$d = (cos(y)sin(t), cos(x)sin(t))$$
$$u = (cos(y)cos(t), cos(x)cos(t))$$

To meet the requirements of MMS such as smoothness and complexity, i chose functions with sine and cosine. The derivatives does not become zero and we have time and space dependencies.

I start with checking order of convergence in space. Setting m = 1, the expected order of convergence will 2.

N	Δt	**m**	$E_u \times 10^{-3}$	$\mathbf{k_u}$	$E_d \times 10^{-8}$	$\mathbf{k_d}$
4	1×10^{-6}	**1**	6.88		3.78	
8	1×10^{-6}	**1**	1.72	**2.000212**	0.94	**2.000212**
16	1×10^{-6}	**1**	0.43	**2.000051**	0.23	**2.000051**
32	1×10^{-6}	**1**	0.10	**2.000012**	0.05	**2.000012**
64	1×10^{-6}	**1**	0.026	**2.000003**	0.0014	**2.000003**

Table 2.1: Structure Method of Manufactured of Solution in space in m = 1

Up next I set m=2 changing the expected order of convergence to 2:

N	Δt	**m**	$E_u[\times 10^-5]$	$\mathbf{k_u}$	$E_d[\times 10^{-10}]$	$\mathbf{k_d}$
4	1×10^{-6}	**2**	6.60	**-**	3.63	**-**
8	1×10^{-6}	**2**	0.82	**2.99458**	0.45	**2.99458**
16	1×10^{-6}	**2**	0.10	**2.99865**	0.057	**2.99865**
32	1×10^{-6}	**2**	0.012	**2.99966**	0.0071	**2.99966**
64	1×10^{-6}	**2**	0.00161	**2.99991**	0.00089	**2.99991**

Table 2.2: Structure Method of Manufactured of Solution in space in m = 2

Lastly I check convergence in time. Here i set N = 64 and the timestep is halved for each computation.

N	**Δt**	$E_u[\times 10^{-6}]$	$\mathbf{k_u}$	$E_u[\times 10^{-8}]$	$\mathbf{k_d}$
64	**0.0008**	2.40	**-**	1.76	**-**
64	**0.0004**	1.20	**0.995**	0.86	**1.0233**
64	**0.0002**	0.59	**1.026**	0.41	**1.0676**
64	**0.0001**	0.29	**1.011**	0.20	**1.0338**
64	**0.00005**	0.14	**0.998**	0.10	**1.0138**

Table 2.3: Structure Method of Manufactured of Solution in time

2.3 MMS on FSI ALE

In this section we use the method of manufactured solutions to verify the FSI ALE monolithic solver. We start by prescribing a motion to d and w and give a solution to u and p. We set $u = w$ to start with:

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

We make the solutions to uphold the criterias : $\nabla \cdot u = 0$ and $\frac{\partial d}{\partial t} = w$

To test the mapping we make the source term f without mappings:

$$\rho_f \frac{\partial u}{\partial t} + \nabla u(u - \frac{\partial d}{\partial t}) - \nabla \cdot \sigma_f = f$$

Then we use this f and map it to the reference configuration and compute:

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

The computations are done on a unitsquare domain and the computations ran with 10 timesteps and the error was calculated for each time step and then the mean of all the errors was taken and used to calculate the convergence rates.

N	Δt	m	E_u	k_u	E_p	k_p
64	0.1	2	0.0140496662424	-	4.78779559903	-
64	0.05	2	0.00697215098985	1.01086014072	2.38002096658	1.00838727906
64	0.025	2	0.00341287458821	1.03061641184	1.18981484439	1.00023719999
64	0.0125	2	0.00164214907307	1.05540230133	0.595733372533	0.99799839775
2	$10x^{-6}$	2	0.000520027806571	-	0.0194221106771	-
4	$10x^{-6}$	2	6.60205272446e-05	2.97760220293	0.00480815191132	2.01414560945
8	$10x^{-6}$	2	8.28184559099e-06	2.99489045	0.00118568799584	2.0197580517
16	$10x^{-6}$	2	1.0417232845e-06	2.99098020306	0.000281586546806	2.0740741124

Table 2.4: MMS ALE FSI u=w

Validation

After the code has been verified to see that we are indeed computing in the right fashion. We move on to Validation which is the process of determining if the model gives an accurate representation of the real world within the bounds of the intended use [11]. A model is made for a specific purpose, its only valid in respect to that purpose [6]. If the purpose is complex and trying to answer multiple question then the validity need to be determined to each question. The idea is to validate the solver *brick by brick*. We start with simple testing of each part of the model and build more complexity and eventually testing the whole model.Three issues have been identified in this process [11]: 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. For example if our solver needs to model fluid which is turbulent we have to validate our model to catch these turbulences and as we shall see later the Taylor-Green benchmark is a good test. Well known benchmarks will be used as validation, we will see in this chapter that these tests supply us with a problem setup, initial and boundary conditions, and lastly results that we can compare with. 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 to run. As we will see in the chapter all the benchmarks are run with different timesteps and number of cells to see how it 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 [11] 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. It really can be an endless process of verifying and validating if one does not clearly now the bounds of sufficient accuracy.

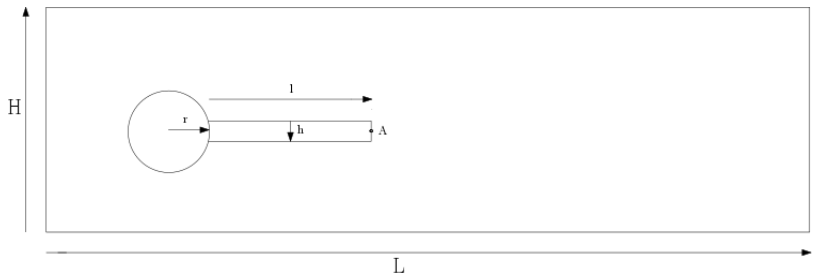
[11]
In the following we will look at tests for the fluid solvers both alone, testing laminar to turbulent flow, and with solid. We will test the solid solver, and lastly the entire coupled FSI problem.

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

The goal of this benchmark is to test the fluid and solid solver first separately and then together as a full FSI problem [5]. This benchmark is based on the older benchmark " flow around cylinder" with fluid considered incompressible and in the laminar regime where the structure deformations are significant. The problem is setup with the solid submerged in the fluid, so that oscillations in the fluid deform the structure. We will measure the drag and lift around the circle and bar, and measure structural displacement at a given point. This benchmark will be used to test and verify different numerical methods and code implementations. Testing robustness and efficiency.

2.4.1 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 is done to induce oscillations to an otherwise laminar flow. This gives a force to the elastic bar. The parameters of the domain are: L = 2.5, H = 0.41, l = 0.35, h = 0.02, A = (0.2, 0.6)

Boundary conditions

The fluid velocity has a parabolic profile on the inlet that changes over time:

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

We set no slip on the "floor" and "ceiling" so to speak.

$$u(x,y,t) = 0 \text{ on } \partial\mathcal{F}_{floorandceiling}$$

$$p(x,y,t) = 0 \text{ on } \partial\mathcal{F}_{outlet}$$

Quantities for comparison

When the fluid moves around the circle and bar it exerts a force. These 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 on the right side of the bar. This point is used to track the deformation in CSM and FSI tests.
In the unsteady time dependent problems the values are represented with a mean , amplitude and frequency:

$$mean = \frac{1}{2}(max + min) \tag{2.7}$$
$$amplitude = \frac{1}{2}(max - min) \tag{2.8}$$
$$frequency = \frac{1}{T} \tag{2.9}$$

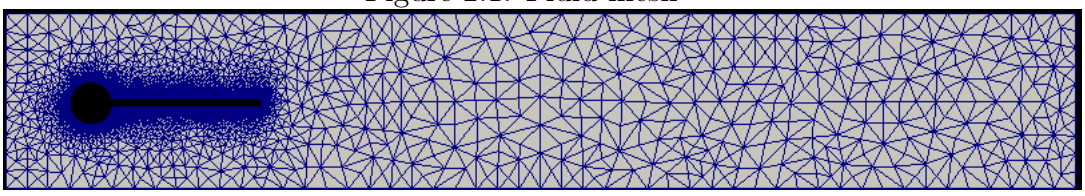
In each test the numbers with ref are the values taken from the benchmark paper [5]

2.4.2 Results

CFD test

The first two CFD tests are run with Reynolds number 20 and 100 giving steady drag and lift around the circle. CFD 3 has a Reynolds number 200 which will induce oscillations behind the circle, giving fluctuations in the drag and lift. The CFD tests were run using the the bar as rigid object, that is the domain calculated is just the fluid domain. It is possible to also calculate with the bar and setting ρ_s and μ_s to a large value.

Figure 2.1: Fluid mesh



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 2.5: CFD parameters

elements	**dofs**	**Drag**	**Lift**
6616	32472	14.2439	1.0869
26464	124488	14.2646	1.11085
105856	487152	14.2755	1.11795
ref		**14.29**	**1.119**

Table 2.6: CFD 1

elements	**dofs**	**Drag**	**Lift**
6616	32472	135.465	6.27158
26464	124488	136.566	9.82166
105856	487152	136.573	10.4441
ref		**136.7**	**10.53**

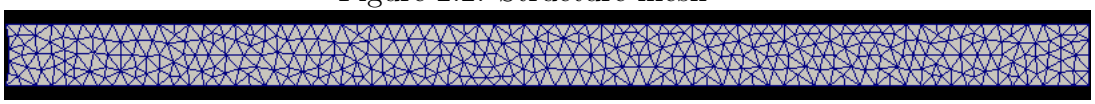
Table 2.7: CFD 2

CSM test

The CSM test are calculated using only the bar and adding a gravity term g with the same value but changing the parameters of solid. The tests CSM1 and CSM2 are steady state solutions. The difference is a more slender bar. The CSM 3 test is unsteady and even more slender causing the bar to move up and down. Since there is no resistance from any fluid the unsteady test should if energy is preserved make the bar move up and down infinitely.

Our quantity for comparing there will be the deformation at the point A.

Figure 2.2: Structure mesh



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 2.8: Parameters

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 2.9: CSM 1

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 2.10: CSM 2

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 2.11: CSM 3

Figure 2.3: Displacement of point A, CSM3

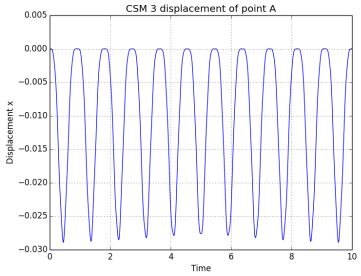


Figure 2.4: Displacement x

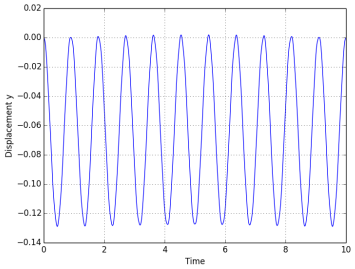


Figure 2.5: Discplacement y

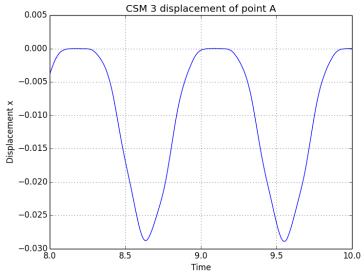


Figure 2.6: Displacement x

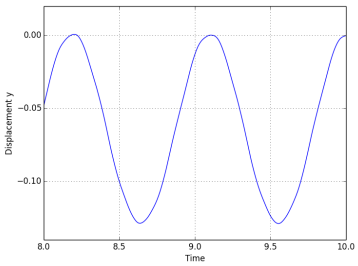


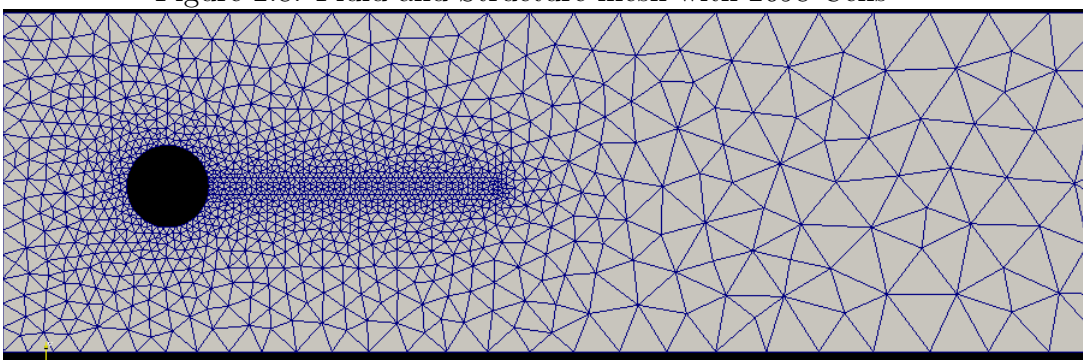
Figure 2.7: Discplacement y

The plots 2.4.2 is a plot of the CSM3 test. This was run with Crank-Nicholson, $\theta = 0.5$ and as we can see the energy has been preserved.

FSI test

Lastly we run the full FSI problem. Here we can see in 2.5 that now both fluid and structure has a mesh. The tests are run with 2 different inflows. 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 [8]. The FSI-3 test has the highest inflow speed giving more rapid oscillations.

Figure 2.8: Fluid and Structure mesh with 2698 Cells



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
$\mathrm{Re} = \frac{Ud}{\nu_f}$	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 2.12: FSI Parameters

FSI1

Cells	Dofs	$d_x(A)[x10^{-3}]$	$d_y(A)[x10^{-3}]$	Drag	Lift	Spaces
2698	23563	0.0227418	0.799314	14.1735	0.761849	P2-P2-P1
10792	92992	0.0227592	0.80795	14.1853	0.775063	P2-P2-P1
43168	369448	00.227566	0.813184	14.2269	0.771071	P2-P2-P1
ref	**ref**	**0.0227**	**0.8209**	**14.295**	**0.7638**	**ref**

Table 2.13: FSI 1

FSI2

The FSI-2 results were with $\Delta t = 0.01, \theta = 0.51$, biharmonic bc 1, $\alpha_u = 0.01$

Cells	Dofs	$d_x(A)[x10^{-3}]$	$d_y(A)[x10^{-3}]$	Drag	Lift	Extrapolation
2698	23563	-15.17 ± 13.35	1.13 ± 82.5	160.12 ± 17.88	0.87 ± 259.62	Biharmonic
2698	23563	-14.92 ± 13.17	1.13 ± 81.86	160.28 ± 17.94	0.65 ± 254.07	Biharmonic
2698	23563	-15.10 ± 13.32	1.16 ± 82.46	$159.53\pm, 17.44$	0.68 ± 259.10	Harmonic
10792	92992	-14.85 ± 13.14	1.21 ± 81.72	160 ± 17.84	0.89 ± 255.10	Harmonic
ref	**ref**	$\mathbf{-14.58 \pm 12.44}$	$\mathbf{1.23 \pm 80.6}$	$\mathbf{208.83 \pm 73.75}$	$\mathbf{0.88 \pm 234.2}$	**ref**

Table 2.14: FSI-2 with $\Delta t = 0.01$

Cells	Dofs	$d_x(A)[x10^{-3}]$	$d_y(A)[x10^{-3}]$	Drag	Lift	Extrapolation
2698	23563	15.42 ± 13.10	1.14 ± 83.39	157.01 ± 15.69	-0.77 ± 174.36	Harmonic
ref	**ref**	$\mathbf{-14.58 \pm 12.44}$	$\mathbf{1.23 \pm 80.6}$	$\mathbf{208.83 \pm 73.75}$	$\mathbf{0.88 \pm 234.2}$	**ref**

Table 2.15: FSI-2 with $\Delta t = 0.001$

Figure 2.9: Deformation at $t = 9.20$sec using warp by vector in paraview

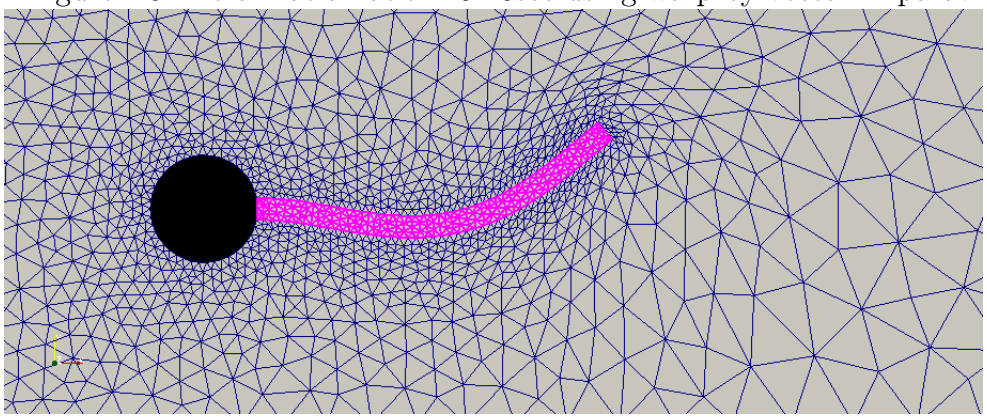
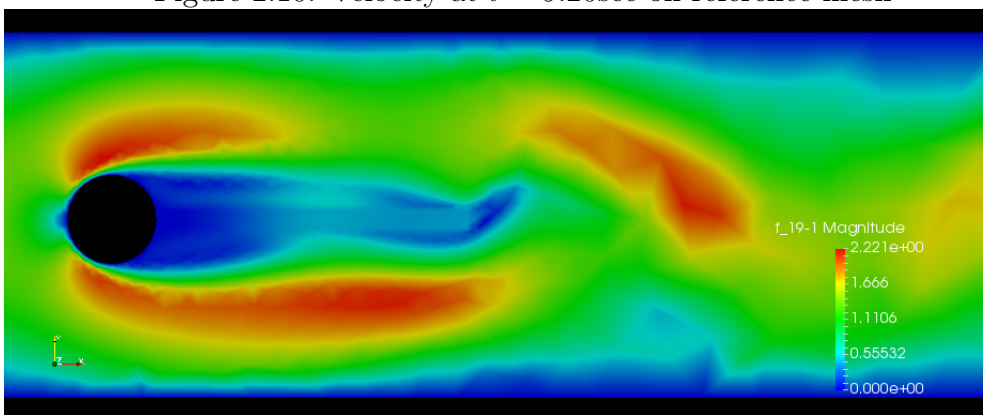


Figure 2.10: Velocity at $t = 9.20$sec on reference mesh



2.5 Mesh motion techniques

In this section we compare different mesh motion techniques from **??**. The test will be run using a version of the CSM test discussed earlier. The tests will compare the different techniques by looking at the how the deformation is lifted into the fluid domain. This is done by looking at a plot of the mesh after deformation to see how much cells distort and why. This is done using Paraview.
In these test cases we have the fluid initially at rest and with no inflow on the fluid. A gravitational force is applied to the structure much like the previous CSM test. The only difference is that we now use the full domain from the 2.4 . The tests are run as time-dependent with a the backward Euler scheme, leading to a steady state solution. In the first test case the parameters from CSM1 are used, and in the CSM4 the gravitational force has just been increased from 2 to 4.

Boundary conditions

The upper, lower and left boundary is set as no slip, that is no velocity in the fluid. On the left boundary there is a do nothing, and zero pressure.

Quantities for comparison

The different techniques will be plotted with the minimal value of the Jacobian. The Jacobian is if we remember the determinant of the deformation rate. If the jacobian is zero anywhere in the domain it means that the cells overlap and can cause singularity in the matrices during assembly.
We will also look at the a plot of the deformation in the domain. To visualize the how the different mesh motion techniques work. It is possible to see that if get thin triangles in the computational domain then the mesh motion operator is no good.

CSM1

Figure 2.11: CSM1

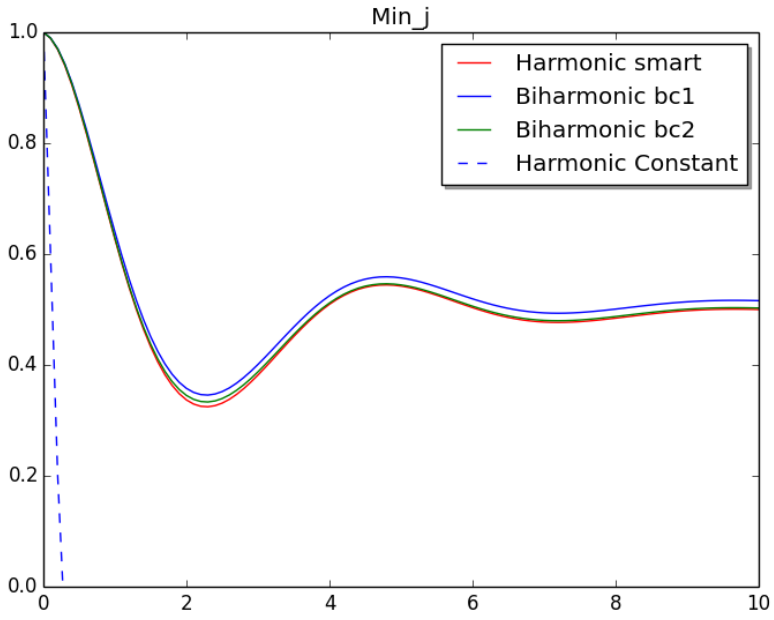


Figure 2.12: CSM1 with different techniques

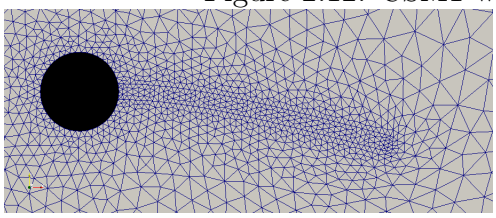


Figure 2.13: Harmonic smart

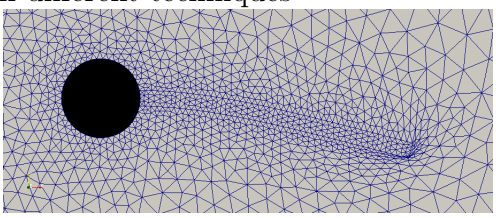


Figure 2.14: Harmonic constant

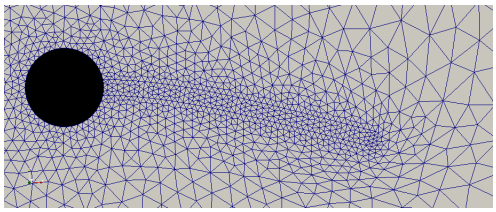


Figure 2.15: Biharmonic bc1

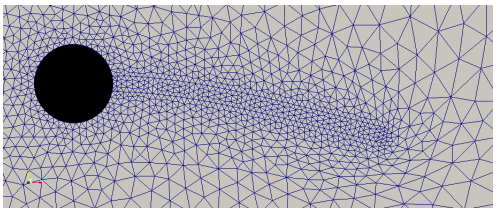


Figure 2.16: Biharmonic bc2

Technique	$d_y(A)[\times 10^{-3}]$	$d_x(A)[\times 10^{-3}]$
Harmonic	65.406	7.036
Constant	43.033	2.999
Bibc1	65.404	7.036
Bibc2	65.405	7.036

Table 2.16: Displacement of CSM1 test

2.6 Temporal stability

The following section will be looking at the results from choosing different θ values in our scheme. The benchmark test FSI-2 has been used to since it is known to blow up with certain values of θ and Δt. We only study the effects of Drag as the three other quantities shows the same behavior.

2.6 show the plots of Drag with $\Delta t = 0.01$. This will not produce very good drag values compared to the benchmark, but it shows the instability when choosing $\theta = 0.5$. The Crank-Nicholson scheme is stable until about 13 seconds where we see that it blows up and the solver diverges.

Figure 2.17: Drag for FSI2 with $\Delta t = 0.01$

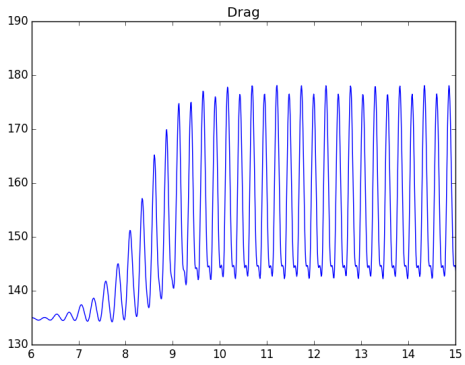


Figure 2.18: $\theta = 0.51$

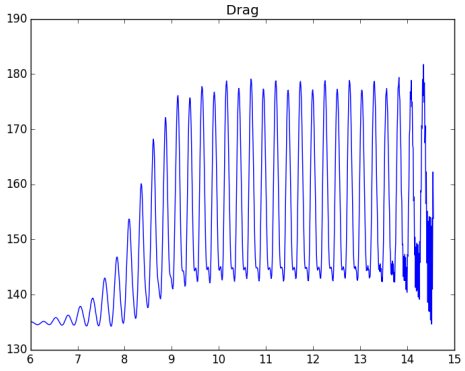


Figure 2.19: $\theta = 0.50$

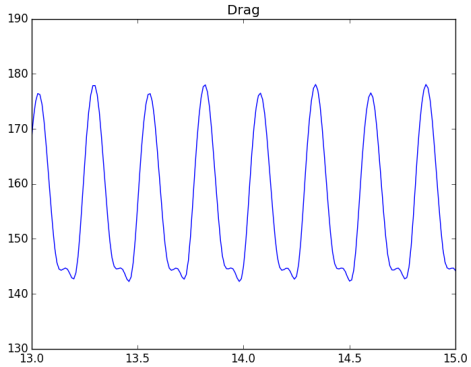


Figure 2.20: $\theta = 0.51$

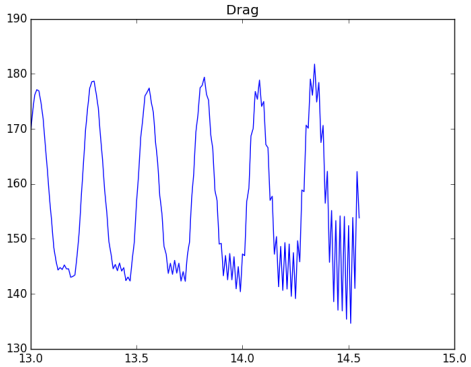


Figure 2.21: $\theta = 0.50$

Put in plot of $\theta = 0.5$ and $\Delta t = 0.001$ to see if its stable longer?

Bibliography

[1] 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.

[2] 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.

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

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

[5] 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.

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

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

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

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

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

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

[12] Frank M White. Viscous Fluid Flow Viscous. *New York*, Second:413, 2000.