UiO : Department of Mathematics
University of Oslo

FSI solver

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.

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 them on a computer. We then use the solutions to extract data that will answer the questions we set out to solve. A problem then immediately arises, is this 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 in accordance with the governing laws and equations? Without answering these questions, being confident that your solutions are correct is difficult [10]. The goal of this section will hence be to verify and validate the different numerical schemes.

We start with Verification, which is the process of assessing numerical correctness and accuracy of a computed solution. Then comes Validation, which is assessing physical accuracy of the numerical model, a process which is done by comparing numerical simulation with experimental data. In simple terms we check that we are solving the equations right and then that we are solving the right equations. The process of Verification has to always come before Validation. Because there is no need in checking if we are using the right equations if the equations are not solved right.

Verification

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. This does not address wether or not the mathematical model is in alignment with the real world only that our model is computed correctly. 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 tackling these problems there are multiple classes of test that can be performed, and the most rigorous is the *Method of manufactured solution* [8]. 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 [8]:

- 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 of order of convergence test is based on the behavior of the error between the manufactured exact solution and the computed solution. 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. Its the rate of this error that lets us now wether the solution is converging correctly. 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:

$$E = ||u - u_e|| \tag{1.1}$$

If we assume that the number of spatial points are equal in all directions the error is expressed as

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

where $k = m + 1$ and m is the polynomial degree of the spatial elements. If we for instance reduce Δt significantly than Δx will dominate, and Δt is negligible . If we then look at the errors in two timesteps, using (1.2):

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

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

We can use this to find the observed order of convergence and match with the theoretical for given

1.1 Structure MMS

We also want to test the coupled solid solver. We make a sourceterm f_s:

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

Solid variational formulation:

$$\big(\rho_s \frac{\partial u}{\partial t}, \phi\big)_{\hat{S}} + \big(P, \nabla \phi\big)_{\hat{S}} = f_s \tag{1.5}$$

$$\big(u - \frac{\partial d}{\partial t}, \psi\big)_{\hat{S}} = 0 \tag{1.6}$$

We should make another sourceterm f for the second equation but the solutions will be made so this line becomes zero. Using the solutions

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

and doing the order of convergence test we get, first in space:

Time:

1.2 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

N	Δt	m	E_u	k_u	E_d	k_d
4	$1x10^{-6}$	1	0.00688260782038		3.7854343057e-08	
8	$1x10^{-6}$	1	0.00172039793734	2.00021299904	9.46218870811e-09	2.00021299271
16	$1x10^{-6}$	1	0.000430084236596	2.00005114684	2.36546335807e-09	2.00005112024
32	$1x10^{-6}$	1	0.000107520101545	2.00001284898	5.91360617339e-10	2.00001274007
64	$1x10^{-6}$	1	2.68799509236e-05	2.00000399654	1.47839789951e-10	2.00000355583
4	$1x10^{-6}$	2	6.60233871884e-05	-	3.63128629891e-10	-
8	$1x10^{-6}$	2	8.28397328621e-06	2.9945823485	4.55618532822e-11	2.99458234332
16	$1x10^{-6}$	2	1.03646090432e-06	2.99865720273	5.70053508884e-12	2.99865718019
32	$1x10^{-6}$	2	1.2958771861e-07	2.99966479692	7.12732513531e-13	2.99966470221
64	$1x10^{-6}$	2	1.61994158345e-08	2.99991530218	8.90968200767e-14	2.99991489187

Table 1.1: Structure MMS

N	Δt	E_u	k_u	E_u	k_d
64	0.0008	2.40113737032e-06	-	1.76531251763e-08	-
64	0.0004	1.20432501777e-06	0.995493150627	8.68459764556e-09	1.02339269394
64	0.0002	5.91307436945e-07	1.02624446535	4.14332593927e-09	1.06766969495
64	0.0001	2.93267031994e-07	1.01169352445	2.02357953875e-09	1.03387975932
64	0.00005	1.46821750169e-07	0.998149192911	1.00209817454e-09	1.01388570182

Table 1.2: Structure MMS Time

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 1.3: 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 [10]. A model is made for a specific purpose, its only valid in respect to that purpose [7]. 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 [10]: 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 [10] 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.

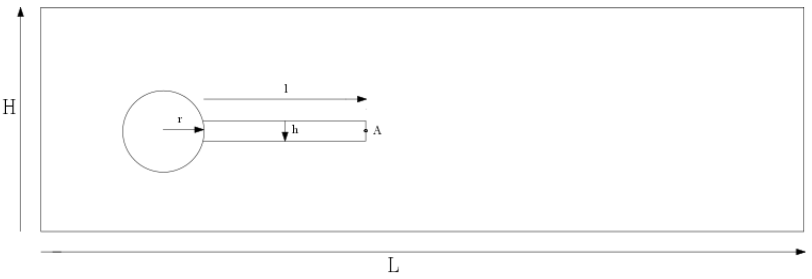
[10]
In the following we will look at tests for the fluid solvers both alone, testing laminar to turbulent flow, and with solid. We will test the solid solver, and lastly the entire coupled FSI problem.

1.3 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, and 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.

1.3.1 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:

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

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:

$$
\begin{aligned}
mean =& \frac{1}{2}(max + min) \quad (1.7)\\
amplitude =& \frac{1}{2}(max - min) \quad (1.8)\\
frequency =& \frac{1}{T} \quad (1.9)
\end{aligned}
$$

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

1.3.2 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.

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 1.4: 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 1.5: 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 1.6: 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.

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 1.7: Parameters

elements	dofs	ux $[\times 10^{-3}]$	uy $[\times 10^{-3}]$
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 1.8: CSM 1

Elements	Dofs	ux $[\times 10^{-3}]$	ux $[\times 10^{-3}]$
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 1.9: CSM 2

elements	dofs	ux $[\times 10^3]$	uy $[\times 10^3]$
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		-14.305 ± 14.305	-63.607 ± 65.160

Table 1.10: CSM 3

Figure 1.1: Displacement of point A, CSM3

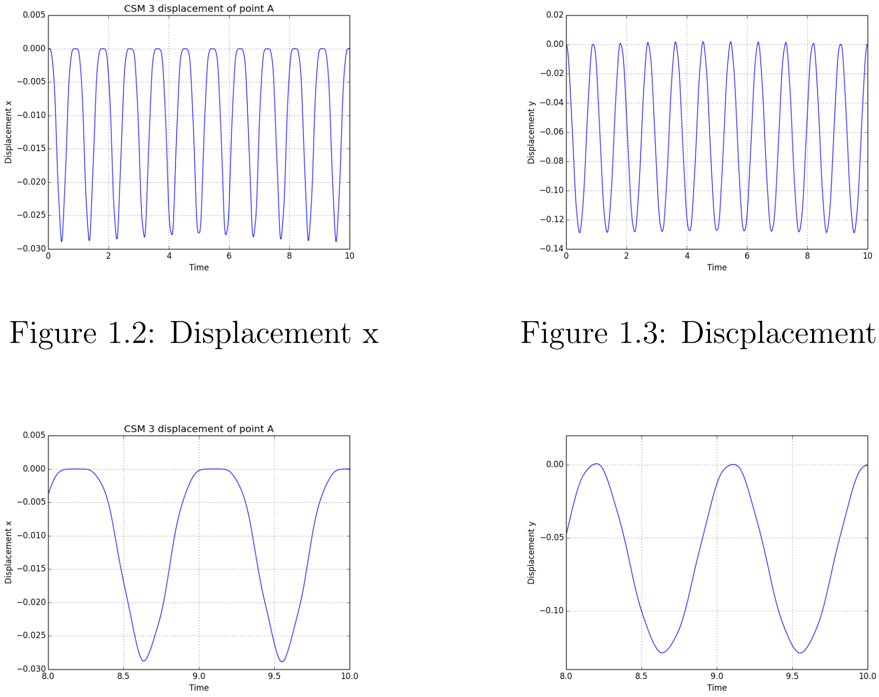


Figure 1.2: Displacement x

Figure 1.3: Discplacement y

Figure 1.4: Displacement x

Figure 1.5: Discplacement y

The plots 1.3.2 is a plot of the CSM3 test. This was run with Crank-Nicholson and as we can see the energy has been preserved.

FSI test

Results:

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 1.11: FSI Parameters

Cells	Dofs	ux of A $[x10^{-3}]$	uy of 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 1.12: FSI 1

The FSI-2 results were with $\Delta t = 0.01, \theta = 0.51, biharmonicbc1$

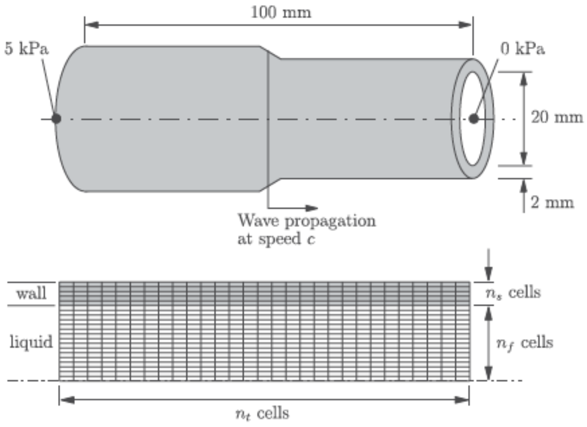
Cells	Dofs	ux of A [x10?3]	uy of A [x10?3],	Drag	Lift
2698	23563		-1.36 ± 78.92	160.61 ± 18.09	5.20 ± 266.02
ref	ref	-14.58 ± 12.44	1.23 ± 80.6	208.83 ± 73.75	0.88 ± 234.2

Table 1.13: FSI-2

1.4 Flexible tube

This benchmark details fluids in flexible cylindrical tubes [3], which is relevant to practical problems as pressure surge in pipelines and blood flow in arteries. Specifically this deals with wave propagation due to pressure drop. Pressure will be added to one side of the cylinder producing a wave in which theoretical wave and flow speed can be calculated and compared to numerical findings.

Problem definition



The properties and geometry were selected for its representation of blood flow in a large artery.

l [mm]	D [mm]	t [mm]	E [MPa]	$K_s [kPa]$	$K_f [GPa]$	ν_s	$\rho_s [\frac{kg}{m^3}]$	$\mu_s [kPa]$	$\rho_f []$
100	20	2	1	833	2.2	0.3	1000	385	1000

Table 1.14: My caption

Boundary conditions

Quanities for comparison

Results

1.5 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 1.3 . 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.

1.6 Temporal stability

Bibliography

[1] 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.

[2] 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.

[3] Chris J. Greenshields and H. G. Weller. A unified formulation for continuum mechanics applied to fluid-structure interaction in flexible tubes. *International Journal for Numerical Methods in Engineering*, 64(12):1575–1593, 2005.

[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] 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.

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

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

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

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

[11] 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.

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

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

[14] 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.

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