

Thesis Title

Institution Name

Author Name

Day Month Year



# Kapittel 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 on a computer. We then use the solutions to extract data that will answer the questions of the problem we set out to solve. A problem then immediately arises, is this solution correct? To answer this we need to answer another question, is the problem defined correct mathematically, and if so are these equations solved correct numerically? Without answering these questions, being confident that your solutions are correct is difficult. [?] 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 or so called benchmark tests. 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. We will use a series of tests, in Verification we will use the Method of Manufactured solutions and in Validation we will use well known benchmarks.

### 1.1 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 caused by mapping a mathematical model to a computational model. This does not address whether or not the mathematical model is in alignment with the real world only that our model is computed correctly. In verifying the code, order of convergence tests will be the most rigorous. To do this test we will use the method of manufactured solutions (MMS) [?]. This method entails manufacturing an exact solution that is non trivial but analytical. The solution defines the boundary conditions and is passed through the equations giving a source term, named  $f$ . This source term is set to equalize the given equation, and then a solution is calculated. If the calculation is correct our calculated solution should equal the manufactured solution down to a given precision, computers are only precise to about  $10^{-16}$ . We can then increase for instance the number of cells in our computational domain, and see if the difference between the manufactured and computed solution (eg. error) gets smaller. The rate at which the error reduces can be checked with mathematical theory, we can then be more confident that our computation is correct. This will also be done in time, by reducing the time steps and investigating the error. The manufactured solution does not have to have any physical relation, and this fact does not implicate a less accurate verification. The solution needs only be non-trivial.

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 or decrease timestep, we expect the error to get smaller. It's the rate of this error that lets us know whether the

solution is converging correctly and hence that we are computing in the right fashion. If we assume that the number of spatial points are equal in all directions we know that the error behaves like

$$E = C_1 \Delta x^k + C_2 \Delta t^l$$

where  $k = m + 1$  and  $m$  is the polynomial degree of the spatial elements. This means that when we compute with Taylor-Hood elements (P2-P1) we should expect to get a convergence rate of 2 in space and 1 in time. The convergence rates are computed as:

$$\frac{E_{n+1}}{E_n} = \left( \frac{\Delta x_{n+1}}{\Delta x_n} \right)^k \quad (1.1)$$

$$k = \frac{\log\left(\frac{E_{n+1}}{E_n}\right)}{\log\left(\frac{\Delta x_{n+1}}{\Delta x_n}\right)} \quad (1.2)$$

### 1.1.1 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:

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

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

$$\rho_f J \frac{\partial u}{\partial t} + (\nabla u) F^{-1} \left( u - \frac{\partial d}{\partial t} \right) + \nabla \cdot (J \hat{\sigma}_f F^{-T}) = J f$$

The computations are done on a unit square 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.

Tabell 1.1: MMS ALE FSI  $u=w$

N	$\Delta t$	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

### 1.1.2 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:

$$\left(\rho_s \frac{\partial u}{\partial t}, \phi\right)_{\hat{\mathbf{S}}} + (P, \nabla \phi)_{\hat{\mathbf{S}}} = f_s \quad (1.3)$$

$$\left(u - \frac{\partial d}{\partial t}, \psi\right)_{\hat{\mathbf{S}}} = 0 \quad (1.4)$$

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:

Tabell 1.2: Structure MMS

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

Time:

Tabell 1.3: Structure MMS Time

N	$\Delta t$	$E_u$	$k_u$	$E_u$	$k_d$
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

## 1.2 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 [?]. A model is made for a specific purpose, its only valid in respect to that purpose [?]. 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 [?]: 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 [?] 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. [?]

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.

### 1.2.1 Taylor-Green vortex

The Taylor-Green vortex problem is used to examine if our N-S code has the ability to correctly simulate vortex decay and turbulence [?]. TGV is a non-trivial analytical solution to N-S. We will compute and compare the kinetic energy dissipation rate against a well known very good solver (smisk smisk). This will help us determine the solvers ability to handle turbulence.

#### Problem definition

Using a cube with sides  $2\pi$ .

We have an initial distribution of velocity  $\bar{u} = (u, v, w)$ :

$$u(x, y, z) = V_0 \sin(x) \cos(y) \cos(z) \quad (1.5)$$

$$v(x, y, z) = -V_0 \cos(x) \sin(y) \cos(z) \quad (1.6)$$

$$w(x, y, z) = 0 \quad (1.7)$$

The Reynolds number is defined as:  $Re = \frac{V_0 L}{\nu}$  where we set  $V_0 = 1$

#### Something about calculating data

The method used to evaluate the TGV solutions will be by investigating the rate of which the fluid dissipates the kinetic energi. The kinetic energi will be calculated as:

$$E_k = \frac{1}{\rho_0 \Omega} \int \rho \frac{uu}{2} = 0.5 \frac{u^2}{(2\pi)^3}$$

We use this to calculated the dissipation rate by differentiating  $E_k$  w.r.t time:

$$\epsilon(E_k) = -\frac{dE_k}{dt}$$

## Results

To help validate the Taylor-Green solutions we used the Oasis solver [?] which is known to handle TGV and other turbulent flows. We look at a plot of the dissipation rate over time and the kinetic energy compared with IPCS. These test were run with:  $N = 32$ ,  $\Delta t = 0.001$ ,  $\nu = 0.001$  giving  $Re = 1000$ :

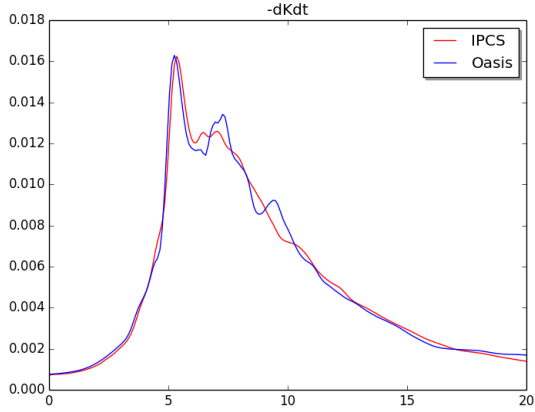


Figure 1.1:  $\epsilon(E_k)$   $N = 32$

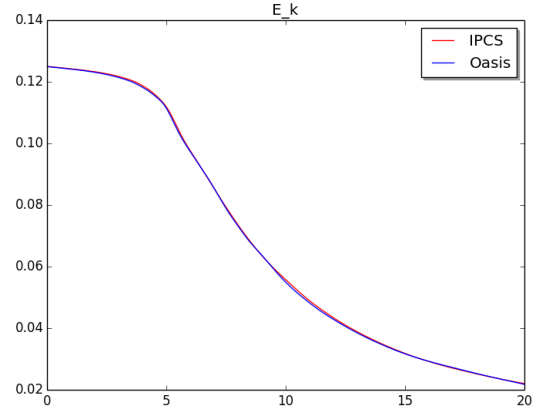


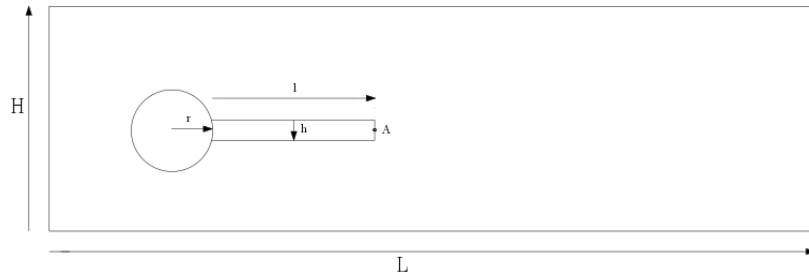
Figure 1.2:  $E_k$   $N = 32$

## 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 [?]. 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.

### Problem Definition

#### 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 "ceilingso to speak.

$$u(x, y, t) = 0 \text{ on}$$

On the fluid solid interface the boundary conditions are set to:

$$\sigma_f n_f = \sigma_s n_s \quad \text{on } \Gamma^0(\text{interface})$$

In our variational form we leave this out and so implying that they are equal.

### 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 each test the numbers with ref are the values taken from the benchmark paper [?]

### 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  $\rho_s$  and  $\mu_s$  to a large value.

Tabell 1.4: CFD parameters

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
$Re = \frac{Ud}{\nu_f}$	20	100	200

Tabell 1.5: CFD 1

elements	dofs	Drag	Lift
6616	32472	14.2439	1.0869
26464	124488	14.2646	1.11085
105856	487152	14.2755	1.11795
ref		<b>14.29</b>	<b>1.119</b>



Tabell 1.6: CFD 2

elements	dofs	Drag	Lift
6616	32472	135.465	6.27158
26464	124488	136.566	9.82166
105856	487152	136.573	10.4441
ref		<b>136.7</b>	<b>10.53</b>

## CSM test

The CSM test are calculated using only the bare and adding a gravity term  $g$  with the same value but changing the parameters of solid. As with the CFD test the first to CSM test cause a steady state solution, and CSM 3 is more slender causing the bar to go up and down in time. Our quantity for comparing there will be the deformation of the point  $A$ . In CSM 3 the energy is conserved by using a Crank-Nicholson scheme as can be seen in the plots fig7 (hvordan citer man et plot?)

Tabell 1.7: Parameters

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
$\nu_s$	0.4	0.4	0.4
$\mu_s [10^6 \frac{m^2}{s}]$	0.5	2.0	0.5
$g$	2	2	2

Tabell 1.8: CSM 1

elements	dofs	ux $[10^{-3}]$	uy $[10^{-3}]$
725	1756	-5.80951654915	-59.4781430115
2900	6408	-6.77960453995	-64.2130757639
11600	24412	-7.08597041285	-65.635825349
46400	95220	-7.11626976966	-65.7456687273
ref	ref	-7.187	-66.10

Tabell 1.9: CSM 2

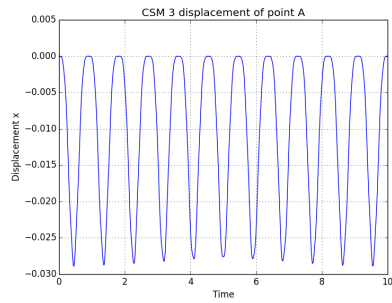
Elements	Dofs	ux $[10^{-3}]$	ux $[10^{-3}]$
725	1756	-0.375962146908	-15.1950342598
2900	6408	-0.441308781709	-16.4643196042
11600	24412	-0.462087305294	-16.8478689583
46400	95220	-0.464128022327	-16.8782135872
ref	ref	-0.4690	-16.97

## FSI test

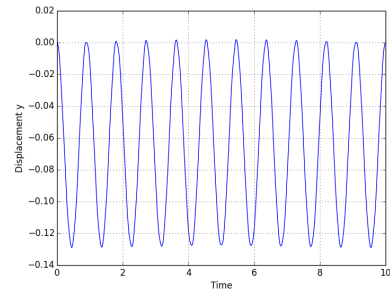
Results:

Tabell 1.10: CSM 3

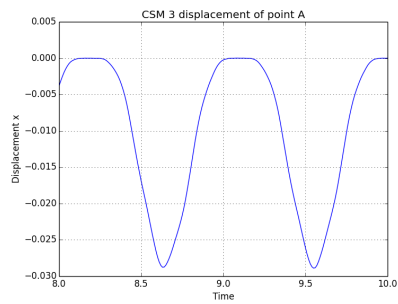
elements	dofs	ux [ $10^3$ ]	uy [ $10^3$ ]
725	1756	$-11.743 \pm 11.744$	$-57.952 \pm 58.940$
2900	6408	$-13.558 \pm 13.559$	$-61.968 \pm 63.440$
11600	24412	$-14.128 \pm 14.127$	$-63.216 \pm 64.744$
46400	95220	$-14.182 \pm 14.181$	$-63.305 \pm 64.843$
ref		$-14.305 \pm 14.305$	$-63.607 \pm 65.160$



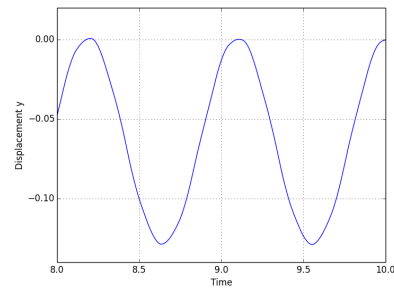
(a) Displacement in x-direction



(b) Displacement in x-direction



(c) Displacement in x-direction



(d) Displacement in x-direction

Figure 1.3: Displacement of point A

OLD SHITZ FSI:

Tabell 1.11: FSI Parameters

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
$Re = \frac{Ud}{\nu_f}$	20	100	200
$\rho_s [10^3 \frac{kg}{m^3}]$	1	10	1
$\nu_s$	0.4	0.4	0.4
$\mu_s [10^6 \frac{m^2}{s}]$	0.5	0.5	2

Tabell 1.12: FSI 1

Cells	Dofs	ux of A [ $x10^{-3}$ ]	uy of A [ $x10^{-3}$ ]	Drag	Lift	Spaces
2698	7095	0.0234594	0.797218	14.4963	0.915801	P1-P1-P1 stab= 0.01
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	0.0227566	0.813184	14.2269	0.771071	P2-P2-P1
<b>ref</b>	<b>ref</b>	<b>0.0227</b>	<b>0.8209</b>	<b>14.295</b>	<b>0.7638</b>	<b>ref</b>

Tabell 1.13: FSI 1

Cells	Dofs	ux of A [ $x10^{-3}$ ]	uy of A [ $x10^{-3}$ ]	Drag	Lift	Spaces
2698	7095	0.0234594	0.797218	14.4963	0.915801	P1-P1-P1 stab= 0.01
2698	23563	0.02271	0.80288	14.1736	0.787891	P2-P2-P1
2698	23563	0.00581116	0.000000738678	12.07	0.02345	P2-P2-P1 without weighting
10792	92992	0.0227341	0.808792	14.1855	0.801044	P2-P2-P1
43168	369448	0.227352	0.812595	14.227	0.797242	P2-P2-P1
<b>ref</b>	<b>ref</b>	<b>0.0227</b>	<b>0.8209</b>	<b>14.295</b>	<b>0.7638</b>	<b>ref</b>



# Bibliografi

- [1] James DeBonis. Solutions of the Taylor-Green Vortex Problem Using High-Resolution Explicit Finite Difference Methods. *51<sup>st</sup> AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, (February):1–9, 2013.
- [2] 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.
- [3] Cm Macal. Proceedings of the 2005 Winter Simulation Conference ME Kuhl, NM Steiger, FB Armstrong, and JA Joines, eds. *Simulation*, pages 1643–1649, 2005.
- [4] Mikael Mortensen and Kristian Valen-Sendstad. Oasis: A high-level/high-performance open source Navier–Stokes solver. *Computer Physics Communications*, 188:177–188, 2015.
- [5] Patrick J. Roache. Code Verification by the Method of Manufactured Solutions. *Journal of Fluids Engineering*, 124(1):4, 2002.
- [6] Noelle Selin. Verification and Validation. (February), 2014.