Thesis Title

Institution Name

Author Name

Day Month Year

Kapittel 1

Verification and validation.

When we set out to solve 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 this solutions to extract data that will answer the questions of the problem we set out so 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. [3] The goal of this section will hence be to verify and validate the different numerical schemes.
Verification is process of assessing numerical correctness and accuracy of a computed solution. Validation is assessing physical accuracy of the numerical model, a process which is done by comparing numerical simulation with experimental data.

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 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. In verifying the code, order of convergence tests will be the most rigorous. Will will here compare an analytical solutions to the computed numerical solution. To do this test we will use the method of manufactured solutions (MMS) [2]. This method entails manufacturing an exact solution that is non trivial but analytic. 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 give 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 than be more confident that our computation is correct. This will also be done in time, by reducing the time steps and looking at 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.

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:

$$u = w = \tag{1.1}$$

$$d = \tag{1.2}$$

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:

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

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. Its the rate of this error that lets us now wether the solution is converging 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:

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

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

1.1.2 Structure MMS

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

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

We want to test the variational formulation:

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

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

Results: with solution

Tabell 1.1: My caption

N	Δt	m degree	E_u	k_u	E_d	k_d
4	1x10^{-8}	2	0.000762421947222	-	0.000762421932529	-
8	1x10^{-8}	2	9.53027534024e-05	2.99999984862	9.5302746523e-05	2.99999992496
16	1x10^{-8}	2	1.1912850306e-05	2.99999925755	1.1912885838e-05	2.99999485034
32	1x10^{-8}	2	1.48910976236e-06	2.99999663417	1.48946247368e-06	2.99965926019
64	1x10^{-8}	2	1.86142797916e-07	2.99996839616	1.8900667118e-07	2.97828271418

$$u_x = x[0] * x[0] * x[0] + 3 * t * t$$

$$u_y = x[1] * x[1] * x[1] + 3 * t * t$$

$$d_x = x[0] * x[0] * x[0] + t * t * t$$

$$d_y = x[1] * x[1] * x[1] + t * t * t$$

1.2 Validation

After the code has been verified to see that we are indeed computing in the right fashion. We have to see that it is the right equations that are being solved. This is achieved using known benchmark tests. These tests supply us with a problem setup, initial and boundary conditions, and lastly results that we can compare with. We can then determine the accuracy of the computational model, and see if these meet the requirement needed to solve the problem. [3]
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 [1].

Problem definition

Using a cube with sides 2π.
We have an initial distribution of velocity $\bar{u} = (u, v, w)$:

$$u(x, y, z) = V_0 sin(x) cos(y) cos(z) \tag{1.7}$$

$$v(x, y, z) = -V_0 cos(x) sin(y) cos(z) \tag{1.8}$$

$$w(x, y, z) = 0 \tag{1.9}$$

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

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

Problem Defintion

Domain

The computational domain resembles the classic cfd benchmark with an added bar, with dimensions:
The box: L = 2.5, H = 0.41
The bar: l = 0.35, h = 0.02
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.
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 "ceilingso to speak.
On the fluid solid interface the boundary conditions are set to:

$$\sigma_f n_f = \sigma_s n_s \quad on \;\; \Gamma^0(interface)$$

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

CSM test

Parameters

Tabell 1.2: My caption

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

FSI test

Tabell 1.3: 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
$\mathrm{Re} = \frac{Ud}{\nu_f}$	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

Results: In my monolithic

Tabell 1.4: 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
ref	**ref**	**0.0227**	**0.8209**	**14.295**	**0.7638**	**ref**

Bibliografi

[1] James DeBonis. Solutions of the Taylor-Green Vortex Problem Using High-Resolution Explicit Finite Difference Methods. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, (February):1–9, 2013.

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

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