Thesis Title

Institution Name

Author Name

Day Month Year

The Fluid-Structure Interaction problem can be observed all around us in nature, from large industrial engineering complexes to the smallest blood vessels in the human body. A large scale example is the collapse of the Tacoma Narrows Bridge that collapsed in 1940 only two months after being opened. The collapse was due to aero-elastic fluttering from strong winds. No human life was lost in the collapse, but a cocker spaniel name Tubby left in a car was not so lucky. The construction of windmills are a second example of the Fluid-Structure Interaction problem. Todays windmills are rigid and hence giving a big difference in density between fluid and structure, $\rho_s >> \rho_f$. The structure will therefore only give rise to small deformations. However applying FSI to hemodynamics(dynamics of blood flow) deems more challenging. One FSI hemodynamic problem are inter-cranial aneurysms, which are balloon shaped geometries often occurring where a blood vessel splits into two parts, due to weak vessel walls. Bursting of one of these aneurysms in the skull can have fatale consequences. With fluid and structure densities more equal than the previous example, the structure has an elastic character giving under the right circumstances large deformations. The blood flow also transitions to turbulent flow. This combination gives the need for a rigid stabile solver. Therefore the main goal of this master thesis is to build a framework to solve the FSI problem, investigating different approaches and schemes. The framework will be validated and verified using MMS, companying a wide range of benchmarks.

Before considering the full FSI problem will discuss the fluid and solid equations separately. These equations will be stated in different frameworks. The fluid is described in an Eulerian framework, where points in the domain are fixed and the fluid flow passes through these points. This reasoning can be understood if one were to stand by the side of river and watch the water flow down. The continuous deformation of fluid flow makes this the most sensible approach, the water becomes the cupso to speak.

Fluid equations

The Navier-Stokes equations are derived using principles of mass and momentum conservation. These equations describes the velocity and pressure in a given fluid continuum. They are here written in the time domain $\mathcal{F}$:

$$\rho\frac{\partial u}{\partial t} + \rho u \cdot \nabla u = \nabla \cdot \sigma_f + f \tag{1}$$

$$\nabla \cdot u = 0 \tag{2}$$

where u is the fluid velocity, p is the fluid pressure, ρ stands for constant density, f is body force and $\sigma_f = \mu_f(\nabla u + \nabla u^T) - pI$
We will only compute incompressible fluids.
There does not yet exist an analytical solutions to the N-S equations, only simplified problems can be solved [4]. But this does not stop us from discretizing and solving them numerically.
Before these equations can be solved we need to impose boundary conditions.

Boundary conditions

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 needs to be defined for both u and p

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

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

The forces on the boundaries need to equal an eventual external force $\mathbf{f}$

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

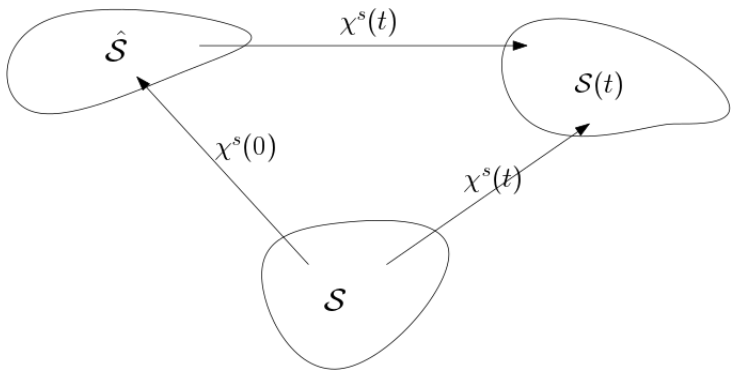
Solid Equations

The solid equation is most commonly described and solved in a Lagrangian description. This description fits a solid problem as the material particles are fixed with grid points. This we will see later is an important property when tracking the solid domain. The displacement vector will be the quantity describing the motion of solid.

Reference domain

Mapping and identites

We will start by providing a short introduction to Lagrangian physics for the sake of completness.



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

$$\chi^s(t) : \hat{\mathcal{S}} \rightarrow \mathcal{S}(t)$$

Following the notation of [3],where $\mathbf{X}$ denote a material point in the reference domain and χ^s denotes the mapping from the reference configuration. $d^s(\mathbf{X}, t)$ denotes the displacement field and w($\mathbf{X}$,t) is the domain velocity. we set the mapping

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

where $d^s(\mathbf{X}, t)$ represents displacement field

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

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

Next we will need a function that describes the rate of deformation in the solid.

Deformation gradient

If $d(\mathbf{X}, t)$ is differentiable deformation field in a given body. We define the deformation gradient as:

$$F = \frac{\partial \chi}{\partial \mathbf{X}} = I + \nabla d(\mathbf{X}, t)$$

which denotes relative change of position under deformation in a Lagrangian frame of reference. The similar Eulerian viewpoint is defined as the inverse deformation gradient

$$\hat{\mathbf{F}} = I - \nabla d(\mathbf{X}, t)$$

J is det(F). In continuum mechanics relative change of location of particles is called strain and this is the fundamental quality that causes stress in a material. [Godboka]. We say that stress is the internal forces between neighboring particles. E denotes the Green-Lagrangian strain tensor $E = F^T F - I$. This measures the squared length change under deformation.

Solid equation

From the principles of conservation of mass and momentum, we get the solid equation stated in the Lagrangian reference system (Following the notation and theory from [Godboka]:

$$\rho_s J \frac{\partial d^2}{\partial t^2} = \nabla \cdot (F\Sigma) + J\rho_s f \tag{3}$$

where f is the body force and Σ denotes the St. Venant Kirchhoff material law:

$$\Sigma = 2\mu_s E + \lambda_s tr(E) I$$

Multiplying Σ with F we get the 2nd Piola Krichhoff stress tensor. This gives us a non-linear stress tensor.

Locking

The problem og shear locking can happen FEM computations with certain elements. [mek4250 Kent] - Locking occurs if $\lambda >> \nu$ that is, the material is nearly incompressible. The reason is that all the elements discussed in this course are poor at approximating the divergence. Locking refers to the case where the displacement is to small because the divergence term essentially lock the displacement. It is a numerical artifact not a physical feature. [Verbatum]

Fluid Structure Interaction Problem in ALE

The ALE formulation stands for Arbitrary Lagrangian Eulerian. This entails formulating the fluid equations in an Eulerian and the solid in a Lagrangian framework. This is as previously stated a very nice approach as it describes both fluid velocity and displacement in known configurations. The ALE method can be solved by moving the mesh for each time step following the structure body movements, and calculating on a new"mesh every time. This approach gives advantages as we can explicitly represent the fluid-structure interface, but problems arise when there are large deformations in the solid giving large deformations to the fluid domain. We will therefore approach the ALE from a reference frame. That is we solve the equation on a initial, stress free domain, and use a series of mappings to account for the movements of the domain. From a technical point of view, both formulations are equivalent. But the ease of computing on fixed mesh does so that we chose fixed approach.
Since the domain is moving with the structure movements, we will need to state the fluid in a little different.

Notation

u - Velocity in fluid and structure.
w- Velocity in the domain. It is the velocity of the mesh in the calculations. This will also be the velocity in structure when defined in the Lagrangian formulation.
d - Displacement of the solid. The time derivative of the displacement will be the domain velocity.
p - Pressure in the fluid.
$\hat{\mathcal{S}}$ - Solid reference domain
$\mathcal{S}$ - Solid current domain
$\hat{\mathcal{F}}$ - Fluid reference domain
$\mathcal{F}$ - Fluid reference domain

Before we can state the full FSI problem we need to define the mappings to be used when computing from a reference frame.

Mapping

Let $\hat{\mathcal{V}}$ be a reference volume and $\mathcal{V}(t)$ be the current time volume using [Godboka]. Then using the determinant of the deformation gradient $F = I + \nabla d$, $J = det(F)$ we define a mapping between the volumes

$$\int_{\mathcal{V}(t)} 1 dx = \int_{\hat{\mathcal{V}}} J dx$$

The gradients acting on a vector $\mathbf{u}$ will also be mapped between current and reference configurations

$$\int_{\mathcal{V}(t)} \nabla \mathbf{u} dx = \int_{\hat{\mathcal{V}}} J \mathbf{u} F^{-1} dx$$

Balance laws

We will formulate the equations in the Eulerian, Lagrangian and the ALE description.

Solid

We express the solid balance laws in the Lagrangian formulation from the initial configuration

$$\rho_s \frac{\partial^2 d}{\partial t^2} = \nabla \cdot (J \sigma_s F^{-T}) \quad in \quad \hat{\mathcal{S}}$$

Fluid

The fluid domain is moving, and therefore we need to redefine the velocity in the convective term in N-S to account for the moving domain

$$u \cdot \nabla u \rightarrow (u - \frac{\partial d_f}{\partial t}) \cdot \nabla u$$

where d_f is the deformation in the fluid domain. Now the actual fluid velocity will be $u - \frac{\partial d}{\partial t}$
The fluid particles are now moving on a moving domain $\nabla u(u - \frac{\partial d}{\partial t})$. So $u - \frac{\partial d}{\partial t}$ will be the actual fluid velocity.
The fluid equations are denoted from the initial configuration again following [godboka] using the forementioned mappings

$$\int_{\mathcal{V}(t)} \rho_f \frac{\partial u}{\partial t} dx = \int_{\hat{\mathcal{V}}} \rho_f J \frac{\partial u}{\partial t} dx \tag{4}$$

$$\int_{\mathcal{V}(t)} \nabla u(u - \frac{\partial d}{\partial t}) dx = \int_{\hat{\mathcal{V}}} ((\nabla u) F^{-1}(u - \frac{\partial d}{\partial t}) dx \tag{5}$$

$$\int_{\mathcal{V}(t)} \nabla \cdot u dx = \int_{\hat{\mathcal{V}}} \nabla \cdot (JF^{-1}u) dx \tag{6}$$

$$\int_{\mathcal{V}(t)} \nabla \cdot \sigma_f dx = \int_{\hat{\mathcal{V}}} \nabla \cdot (JF^{-1}\hat{\sigma_{\mathbf{f}}}) dx \tag{7}$$

$$\hat{\sigma_{\mathbf{f}}} = -pI + \mu(\nabla u F^{-1} + F^{-T}u^T) \tag{8}$$

Harmonic extension

To bind together the computation of fluid and structure domain, we need a harmonic extension to the boundary values. The solid deformation d is extended from the interface into the fluid domain and is done to help deal with big deformations in fluid domain. These big deformations can then cause several challenges to the ALE mapping. For this purpose define the following harmonic extension equation in the fluid domain:

$$\nabla^2 d^f = 0 \quad in \ \hat{\mathbf{F}}$$

This equation is chosen for its good regularity and smoothing properties.
It is also possible to chose an harmonic extension with stiffening, which can give better control of the deformed meshes. This in practice behaves like a transport problem, transporting the deformation into the fluid domain. Another possibility is extension by pseudo-elasticity which defines the extension operator by means of the Navier-Lame equation. And lastly we can chose a biharmonic extension, that is of fourth order character, and thus will have a high computational cost. For now I will stick with the harmonic extension and maybe look at these in the future. [Godboka.]

Coupled Fluid Structure Interface conditions

This section is based on We introduce a global domain $\Omega \in \mathcal{S} \cup \mathcal{F}$ that is made up of the fluid and the structure and the interface. We define a global velocity function u that is the fluid velocity in the fluid domain and the structure velocity in the structure domain. This can be done due to the interface condition making the velocity field continuous over the entire domain. Then the interface will be $\Gamma \in \mathcal{S} \cap \mathcal{F}$ We need to define conditions that couple that motions of the fluid and structure together. These consist of:

- Kinematic condition: $u_f = u_s \quad on \Gamma$. The fluid and structure velocities need to be equal on the boundary.
  We later realize this later by setting $u - \frac{\partial d}{\partial t} = 0$ on the solid domain to strongly imply that the velocity on the solid is the derivative of the deformation.

- Dynamic condition: $\sigma_f n_f = \sigma_s n_s \quad on \ \Gamma$.
  This relates to Newtons third law of action and reaction. The forces on the interface area, here written as the normal stresses are balanced on the interface. These will be written in a Lagrangian formulation:
  $J\sigma_f F^{-T} n_f = F\Sigma n_s \quad on \ \Gamma$.
  This condition is often added to structure problem, since it is most often in FSI problem the fluid exerting force that cause deformation.

- Geometrical condition: This condition says that the fluid and structure domains do not overlap, but rather that elements connect so the functions needing to transfer force are continouos across the entire domain.

Full FSI Problem in reference domain

Find $u \in \hat{\mathcal{F}}, p \in \hat{\mathcal{F}}$ and $d \in \hat{\mathcal{S}}$ such that :

$$\rho_f J \frac{\partial u}{\partial t} + (\nabla u) F^{-1}(u - \frac{\partial d}{\partial t}) + \nabla \cdot (J\hat{\sigma_\mathbf{f}} F^{-T}) = 0 \text{ on } \hat{\mathcal{F}} \tag{9}$$

$$\nabla \cdot (JuF^{-T})\big) = 0 \text{ on } \hat{\mathcal{F}} \tag{10}$$

$$\rho_s \frac{\partial u}{\partial t} + \nabla \cdot FS_s, = 0 \text{ on } \hat{\mathcal{S}} \tag{11}$$

$$\nabla^2 d = 0 \text{ on } \hat{\mathcal{F}} \tag{12}$$

$$u - \frac{\partial d}{\partial t} = 0 \text{ on } \hat{\mathcal{S}} \tag{13}$$

$$J\sigma_f F^{-T} n_f = \sigma_s n_s \text{ on } \Gamma \tag{14}$$

Finite Element method FSI in ALE

Variational formulation

Reference domain

We use 3 testfunctions, ϕ, ψ, γ. As mentioned before we use a global velocity function u for both the solid and fluid.

$$\rho_f J\big(\frac{\partial u}{\partial t} + (\nabla u) F^{-1}(u - \frac{\partial d}{\partial t}), \phi\big)_{\hat{\mathbf{F}}} + (J\sigma_f F^{-T}, \nabla\phi)_{\hat{\mathbf{F}}} = 0 \tag{15}$$

$$\big(\nabla \cdot (JuF^{-T}), \gamma\big)_{\hat{\mathbf{F}}} = 0 \tag{16}$$

$$\big(\rho_s \frac{\partial u}{\partial t}, \phi\big)_{\hat{\mathbf{S}}} + \big(FS_s, \nabla\phi\big)_{\hat{\mathbf{S}}} = 0 \tag{17}$$

$$\big(\nabla d, \nabla\psi\big)_{\hat{\mathbf{F}}} = 0 \tag{18}$$

$$\big(u - \frac{\partial d}{\partial t}, \psi\big)_{\hat{\mathbf{S}}} = 0 \tag{19}$$

Equation (5) has not been addressed and is added since we use a global function for velocity we need to force that the structure velocity is the time derivative of the deformation in the structure domain.

Spaces and Elements

The velocity and pressure copling in the fluid domain must satisfy the inf-sup condition. If not stabilization has to added. We here need to define some spaces that will have these desired properties. We denote $u_h \in V_h$ and $d_h \in W_h$, here the finite element pair og pressure and velocity

mush satisfy the inf-sup condition given in ALE coordinates:

$$\inf_{p_h \in L_{h,f}} \sup_{v_h \in V_{h,f}} \frac{(p_h, div(J_f F_f^{-1} u_h))_{\mathcal{F}}}{\|\|J^{\frac{1}{2}} p_h\|\|_{\mathcal{F}} \|\|J_f^{\frac{1}{2}} \nabla u_h F_f^{-T}\|\|_{\mathcal{F}}} \geq \gamma$$

A good choice of spaces will be P2-P2-P1 for velocity, displacement and fluid pressure respectively.

Introduction

Here we will look at the partitioned approach to solving the FSI problem. This means splitting our scheme into parts where we solve the fluid, structure and extension problem in different steps. This is to greatly reduce the size of the computational cost, and hopefully increase speed. So far the methods for coupling of the fluid and structure, has led to unconditional numerical instabilites and a large added-mass effect. Here we look at a new approach to explicit coupling, first proposed by Fernadez, which uses a Robin-Neumann explicit treatment of the interface first for thin walled structure [1] but later with an extension to thick walled structures[2]. This combined with a lumped mass approximation of the solid problem ensures added-mass free stability. [Generalized R-N explicit coupling schemes]

Robin-Neumann Interface

The Robin-Neumann treatment of the interface proposed by Fernandez uses a boundary operator $B_h : \Lambda_{\Sigma,h} \to \Lambda_{\Sigma,h}$ which is used together with the known coupling of stresses:

$$J^n \sigma^f(u^n, p^n)(F^n)^{-T} n^f + \frac{\rho^s}{\tau} B_h u^n = \frac{\rho^s}{\tau} B_h(\dot{d}^{n-1} + \tau \partial_t \dot{d}^*) - \Pi^* n^s$$

- The explicit treatment of the solid ensures uncoupling of the fluid and solid computations. Giving a genuine partitioned system.
- Treating the left hand side solid tensor implicitly ensures added-mass free stability

The fluid domain is computed using a generalized Robin condition on the interface, and the solid is computed with the familiar Neumann condition on the interface, equalling the stresses from the fluid and structure.
The general r-order extrapolation x^* is defined:

$$x^* = \begin{cases} 0, & \text{if } r = 0 \\ x^{n-1}, & \text{if } r = 1 \\ 2x^{n-1} - x^{n-2}, & \text{if } r = 2 \end{cases} \tag{20}$$

Boundary interface operator

Using the notation of [2] We denote $(.,.)_{\mathcal{S},h}$ as the lumped mass approximation of the inner product $(.,.)_{\mathcal{S}}$. We will consider a solid and fluid sided discrete lifting operator $\mathcal{L}_h^s : \Lambda_{\Sigma,h} \to \mathcal{S}$, lifting values from the interface into the solid domain. If $\xi_h, \lambda_h \in \Lambda_{\Sigma,h}$ then $\mathcal{L}_h^s|_\Sigma = \mathcal{L}_h^f|_\Sigma = \xi_h$. We use this to define the boundary operator: $B_h = (\mathcal{L}_h^s)^* \mathcal{L}_h^s$, mapping from interface to interface $B_h : \Lambda_{\Sigma,h} \to \Lambda_{\Sigma,h}$. Where stars stands for the adjoint operator of $\mathcal{L}_h^s$. We can then write:

$$(B_h \xi_h, \lambda_h)_\Sigma = (\mathcal{L}_h^s \xi_h, \mathcal{L}_h^s \lambda_h)_{\mathcal{S},h}$$

Explicit Robin-Neumann scheme:
Step 1: Fluid domain update

$$\begin{aligned} d^{f,n} &= Ext(d^{n-1}) \\ w^n &= \frac{\partial d^{f,n}}{\partial t} \\ with F &= I + \nabla d, J = det(F) \end{aligned}$$

Step 2: Fluid step: find u^n, p^n:

$$\begin{aligned} \rho^f \big(\frac{\partial u^n}{\partial t} + (u^{n-1} - w^n) \cdot \nabla u^n\big) - \nabla \cdot \sigma(u^n, p^n) &= 0 \in \mathcal{F} \\ \nabla \cdot u &= 0 \in \mathcal{F} \\ \sigma(u^n, p^n) n^f &= f \\ J^n \sigma(u^n, p^n)(F^n)^{-T} n^f + \frac{\rho^s}{\tau} B_h u^n &= \frac{\rho^s}{\tau} B_h(\dot{d}^{n-1} + \tau \partial_t \dot{d}) - \Pi^* n^s \end{aligned}$$

Step 3: Solid Step: find d^n

$$\begin{aligned} \rho^s \partial_t \dot{d}^n + \alpha \rho^s \dot{d}^n - \nabla \cdot \Pi^n &= 0 \in \mathcal{S} \\ \dot{d} &= \partial_t d^n \\ d^n = 0, \beta \dot{d}^n &= 0 \in \Gamma^d \\ \Pi^n n^s &= 0 \in \Gamma^n \\ \Pi^n n^s &= -J^n \sigma(u^n, p^n)(F^n)^{-T} n^f \in \Sigma \end{aligned}$$

The solid stress tensor is given as $\Pi^n = \pi(d^n) + \beta \pi^?(d^{n-1}) \dot{d}^n$

Monolithic FSI Code

The first FSI-code we look at is the monolithic version. Where all the equations are solved simultaneously.
We start by looking at the function spaces. We need two vectorfunction spaces for u, d and a function space for p, made from "Continuous Galerkin" elements . These are put together in a mixed space to allow us to solve them all at once.

```
V1 = VectorFunctionSpace(mesh, "CG", v_deg) # Fluid velocity
V2 = VectorFunctionSpace(mesh, "CG", d_deg) # displacement
Q  = FunctionSpace(mesh, "CG", p_deg)       # Fluid Pressure
VVQ = MixedFunctionSpace([V1, V2, Q])
```

From this mixed space we make three testfunctions for u, p and d named ϕ, ψ and γ. We also make the functions u, p and d, these are made in the mixed space and then split up:

```
phi, psi, gamma = TestFunctions(VVQ)
udp = Function(VVQ)
u, d, p  = split(udp)
```

The boundaries are specified with facetfunctions on the boundary and subdomains to specify where the boundaries are, and these are used to make the Dirichlet conditions:

```
Inlet = AutoSubDomain(lambda x: "on_boundary" and near(x[0],0))
boundaries = FacetFunction("size_t",mesh)
Inlet.mark(boundaries, 3)
u_wall    = DirichletBC(VVQ.sub(0), ((0.0, 0.0)), boundaries, 2)
```

Since we are solving an FSI problem we need to specify where the fluid and solid domains exist. This is done using Cellfunctions and marked as before. For ease of writing we make two subdomains that represent the different domains.

```
domains = CellFunction("size_t",mesh)
Bar_area.mark(domains,2)

dx = Measure("dx",subdomain_data=domains)
dx_f = dx(1,subdomain_data=domains)
dx_s = dx(2,subdomain_data=domains)
```

All the mappings and stresstensors are made using functions, I will just show some to understand the code later on:

```
def F_(U):
        return (I + grad(U))
def J_(U):
        return det(F_(U))

def sigma_f(v,p):
        return 2*mu_f*sym(grad(v)) - p*Identity(2)

def sigma_f_hat(v,p,u):
        return J_(u)*sigma_f(v,p)*inv(F_(u)).T
```

The variational form can be written directly into FEniCS. We write all the forms and add them together to make one big form to be calculated in the upcoming timeloop:

```
# Fluid variational form
F_fluid = (rho_f/k)*inner(J_(d)*(u - u0), phi)*dx_f \
        + rho_f*inner(J_(d)*inv(F_(d))*grad(u)*(u - ((d-d0)/k)), phi)*dx_f \
        + inner(sigma_f_hat(u,p,d), grad(phi))*dx_f \
        - inner(div(J_(d)*inv(F_(d).T)*u), gamma)*dx_f

# Structure variational form
F_structure = (rho_s/k)*inner(u-u0,phi)*dx_s + inner(P1(d),grad(phi))*dx_s

# Setting w = u on the structure using (d-d0)/k = w
F_w = delta*((1.0/k)*inner(d-d0,psi)*dx_s - inner(u,psi)*dx_s)

# Laplace
F_laplace =  (1./k)*inner(d-d0,psi)*dx_f +inner(grad(d), grad(psi))*dx_f

F = F_fluid + F_structure + F_w + F_laplace
```

The equation $u = \frac{\partial d}{\partial t} in \mathcal{S}$, is weighted with a delta value which we will see later on is important when solving monolithic.
To solve a non-linear problem we need make a newton solver [Mikael kompendium]
In the time loop we call on the solver and update the functions u, d for each round:

```
while t <= T:
    print "Time t = %.5f" % t
    time_list.append(t)
    if t < 2:
        inlet.t = t;
    if t >= 2:
        inlet.t = 2;

    #Reset counters
    atol = 1e-6;rtol = 1e-6; max_it = 100; lmbda = 1.0;

    udp = Newton_manual(F, udp, bcs, atol, rtol, max_it, lmbda,udp_res,VVQ)

    u,d,p = udp.split(True)

    #plot(u)
    if counter%step==0:
        u_file << u
        d_file << d
        p_file << p

        Dr = -assemble((sigma_f_hat(u,p,d)*n)[0]*ds(6))
        Li = -assemble((sigma_f_hat(u,p,d)*n)[1]*ds(6))
        Dr += -assemble((sigma_f_hat(u('-'),p('-'),d('-'))*n('-'))[0]*dS(5))
        Li += -assemble((sigma_f_hat(u('-'),p('-'),d('-'))*n('-'))[1]*dS(5))
        Drag.append(Dr)
        Lift.append(Li)

        dsx = d(coord)[0]
        dsy = d(coord)[1]
        dis_x.append(dsx)
```

```
        dis_y.append(dsy)

        if MPI.rank(mpi_comm_world()) == 0:
            print "t = %.4f " %(t)
            print 'Drag/Lift : %g %g' %(Dr,Li)
            print "dis_x/dis_y : %g %g "%(dsx,dsy)

    u0.assign(u)
    d0.assign(d)
    p0.assign(p)
    t += dt
    counter +=1
```

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. [**?**] 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) [**?**]. This method entails manufacturing an exact solution that is non trivial but analytic. This solution is passed through the equations giving a source term, usually 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, computer 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.

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. [**?**]
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

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:

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

We set no slip on the floor"and "ceilingso to speak.
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 1.1: 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 1.2: 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
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 1.3: 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] 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] G Holzapfel. Nonlinear solid mechanics: A continuum approach for engineering, 2000.

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