

Thesis Title

Institution Name

Author Name

Day Month Year



# 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  $\phi, \psi$  and  $\gamma$ . 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
```

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
```