Simulating Heart Valve Dynamics in FEniCS

Kristoffer Selim
Center for Biomedical Computing
Simula School of Research and Innovation
e–mail: selim@simula.no

Anders Logg
Center for Biomedical Computing
Simula Research Laboratory
Dept. of Informatics University of Oslo

Summary The present paper addresses the implementation of a coupled fluid–structure interaction problem in the free/open source software FEniCS. We demonstrate the ease by which the coupled fluid–structure problem may be implemented in FEniCS. As an example, we consider a simple two–dimensional model problem for the time–dependent displacement of a beam ("heart valve") in a pulsative flow.

Introduction

Recent years have seen an increase in the use of computer simulations for biomedical applications. Many of these simulations include demanding fluid-structure interaction (FSI) problems coupled with complex constitutive relations for the fluid and the surrounding biological tissue. Examples include [8, 16, 17, 4, 1].

FSI occurs when a fluid interacts with a solid structure, exerting stress that may cause deformation in the structure and, thus, alter the flow of the fluid itself. This category of problems are among the most important and, with respect to both modelling and computation, among the most challenging multi-physics problems in biomedicine.

Simulating the blood flow in the aortic valve is a true FSI problem, with large forces and two-way exchange of energy between the heart valve and the blood flowing through it. Pressure and shear stress on the heart valve cause the aortic valve cusps to open and close, resulting in a modified geometry for the blood flowing through the aortic valve.

Formulation of the Fluid–Structure Interaction Problem

Let Ω be a bounded polygonal subset of $\mathbb{R}^d$ containing an elastic solid immersed in an incompressible viscous fluid flow. We denote the time–dependent fluid domain by $\Omega_F(t)$ and the time–dependent structure domain by $\Omega_S(t)$. It is assumed, for all time t, that $\Omega_F(t)$ is completely occupied by the fluid and that $\Omega_S(t)$ is completely occupied by the solid. Further, $\bar{\Omega} = \bar{\Omega}_F(t) \cup \bar{\Omega}_S(t)$ and $\Omega_F(t) \cap \Omega_S(t) = \emptyset$, for all time t.

The Fluid Problem

We consider the simple case when the fluid flow is described by the time–dependent Stokes problem in an Arbitrary Lagrangian–Eulerian (ALE) setting. We define the fluid domain deformation mapping $F : \Omega_F(0) \times [0,T] \to \Omega_F(t)$ and $(X,t) \mapsto x = F(X,t)$ where X denotes the material position in the Lagrangian variables and x denotes the spatial position in Eulerian variables. We thus define the fluid–domain velocity as

$$w(x,t) = \frac{d(F(X,t),\ t)}{dt}|_{x=F(X,t)}.$$

The fluid problem in the ALE framework reads: Find the velocity $u_F(\cdot,t) : \Omega_F(t) \to \mathbb{R}^d$ and the pressure $p_F(\cdot,t) : \Omega_F(t) \to \mathbb{R}$ such that

$$\left\{ \begin{array}{rcll} \dot{u}_F - w \cdot \nabla u_F - \nabla \cdot \sigma_F(u_F, p_F) &=& f_F(\cdot,t) & \text{in } \Omega_F(t), \\ \nabla \cdot u_F &=& 0 & \text{in } \Omega_F(t), \\ u_F &=& g_{F,D}(\cdot,t) & \text{on } \Gamma_{F,D}(t), \\ \sigma_F(u_F,p_F)n_F &=& g_{F,N}(\cdot,t) & \text{on } \Gamma_{F,N}(t), \\ u_F(\cdot,0) &=& u_F^0 & \text{in } \Omega_F(0), \\ p_F(\cdot,0) &=& p_F^0 & \text{in } \Omega_F(0), \end{array} \right. \tag{1}$$

for $0 < t \leq T$. Here, f_F is a given body force and σ_F is the stress tensor defined as

$$\sigma_F(u_F,p_F) = 2\eta\varepsilon(u_F) - p_F I,$$

where ε is the strain tensor defined as

$$\varepsilon(v) = \frac{1}{2}(\nabla v + \nabla v^\top).$$

Furthermore, η denotes the dynamic viscosity and I the $d \times d$ identity matrix. The boundary $\partial\Omega_F(t)$ is divided into two parts, $\Gamma_{F,D}(t)$ and $\Gamma_{F,N}(t)$, which are associated with the Dirichlet and Neumann boundary conditions.

The Structure Problem

For the structure, we consider a quasi–static, isotropic, homogeneous body that undergoes small deformations. At each time t, the structure problem reads: Find the displacement $u_S : \Omega_S(t) \to \mathbb{R}^d$ such that

$$\left\{ \begin{array}{rcll} -\nabla \cdot \sigma_S(u_S) &=& f_S(\cdot,t) & \text{in } \Omega_S(t) \\ u_S &=& g_{S,D}(\cdot,t) & \text{on } \Gamma_{S,D}(t) \\ \sigma_S(u_S)n_S &=& -\sigma_F(u_F,p_F)n_F & \text{on } \Gamma_{S,N}(t) = \Gamma_{SF}(t). \end{array} \right. \tag{2}$$

Here, f_S is a given body force and σ_S is the Cauchy stress tensor defined as

$$\sigma_S(u_S) = 2\mu\varepsilon(u_S) + \lambda(\nabla \cdot u_S)I.$$

The Lamé parameters μ and λ are given by $\mu = E/(2(1+\nu))$ and $\lambda = E\nu/((1+\nu)(1-2\nu))$ where E is the Young's modulus and ν the Poisson ratio. The FSI occurs at the interface $\Gamma_{SF}(t)$ via the stress boundary condition in (2). We also notice that on the boundary $\Gamma_{SF}(t)$ we have $n_F = -n_S$, for all time t.

Weak Formulation

The Fluid Problem

The weak formulation of the fluid problem (1) reads: Find $(u_F,p_F) \in V_F = \{(u_F,p_F) \ : \ u_F \in L^2(0,T;H^1(\Omega_F)) \ : \ u_F \,|_{\Gamma_{F,D}} = g_{F,D}, \ p_F \in L^2(0,T;L^2(\Omega_F))\}$ such that

$$a_F((v,q),(u_F,p_F)) = L_F(v), \quad \forall (v,q) \in \hat{V}_F, \tag{3}$$

where $\hat{V}_F = \{(v,q)\ :\ v \in L^2(0,T;H^1(\Omega_F))\ :\ v\,|_{\Gamma_{F,D}} = 0,\ q \in L^2(0,T;L^2(\Omega_F))\}$ and where the bilinear form $a_F : \hat{V}_F \times V_F \to \mathbb{R}$ and the linear form $L_F : \hat{V}_F \to \mathbb{R}$ are defined as

$$\begin{aligned} a_F((v,q),(u_F,p_F)) &= \int_0^T (v, \dot{u}_F - w \cdot \nabla u_F)\, dt + \int_0^T (\varepsilon(v), \sigma_F(u_F,p_F))\, dt \\ &- \int_0^T (q, \nabla \cdot u_F)\, dt, \\ L_F((v,q)) &= \int_0^T (v, f_F)\, dt + \int_0^T (v, g_{N,F})_{\Gamma_{F,N}}\, dt. \end{aligned}$$

Here, $(\cdot,\cdot)$ denotes the L^2 inner product on Ω_F and $(\cdot,\cdot)_{\Gamma_{F,N}}$ denotes the L^2 inner product on $\Gamma_{F,N}$.

The Structure Problem

The weak formulation for the structure problem (2) reads: Find $u_S \in V_S = \{u_S \in H^1(\Omega_S) :\ u_S\,|_{\Gamma_{S,D}} = g_{S,D}\}$ such that

$$a_S(v,u_S) = L_S(v), \quad \forall v \in \hat{V}_S, \tag{4}$$

where $\hat{V}_S = \{v \in H^1(\Omega_S) :\ v\,|_{\Gamma_{S,D}} = 0\}$ and where the bilinear form $a_S : \hat{V}_S \times V_S \to \mathbb{R}$ and the linear form $L_S : \hat{V}_S \to \mathbb{R}$ are defined as

$$\begin{aligned} a_S(v,u_S) &= \int_0^T (\varepsilon(v), \sigma_S(u_S))\, dt, \\ L_S(v) &= \int_0^T (v, f_S)\, dt - \int_0^T (v, \sigma_F(u_F,p_F) n_F)_{\Gamma_{SF}}\, dt. \end{aligned}$$

Here, $(\cdot,\cdot)$ denotes the L^2 inner product on Ω_S and $(\cdot,\cdot)_{\Gamma_{SF}}$ denotes the L^2 inner product on Γ_{SF}.

The Finite Element Method

We consider a family $\{\mathcal{T}\}$ of meshes $\mathcal{T} = \{K\}$ of simplicial elements K. We denote the mesh on the sub domain on $\Omega_F(t)$ by $\mathcal{T}_F(t)$ and on Ω_S by $\mathcal{T}_S(t)$. Further, we assume that all elements along the interface between $\mathcal{T}_F(t)$ and $\mathcal{T}_S(t)$ are matching for $0 < t \leq T$.

The Fluid Problem

To discretize (3), we use Taylor–Hood elements [15] in space. We seek our approximate velocity and pressure $(U_F, P_F) \in W_F^h = \{(v,q) :\ v \in V_F^h,\ q \in Q_F^h\}$. Here, $V_F^h \subset V_F$ is the space of functions that are continuous piecewise quadratic vector–valued on $\mathcal{T}_F$, and piecewise constant in time. The space $Q_F^h \subset Q_F$ is the space of functions that are continuous piecewise linear on $\mathcal{T}_F$, and piecewise constant in time. Let $0 = t_0 < t_1 < ... < t_N = T$ be a partition of $[0,T]$ into time intervals $I_n \in (t_{n-1}, t_n]$ of length $k_n = t_n - t_{n-1}$.

The finite element method for (3) then reads: Find $(U_F^n, P_F^n) \equiv (U_F,(t_n), P_F(t_n))$ with $(U_F^n, P_F^n) \in W_F^h$ for $n = 1, ..., N$ such that

$$a_F((v,q),(U_F^n,P_F^n)) = L_F((v,q)), \quad \forall (v,q) \in \hat{W}_F^h. \tag{5}$$

The test space $\hat{W}_F^h$ is given by the set of piecewise constant functions in time with the same variation as for W_F^h in space [2]. Hence, the bilinear form $a_F(\,\cdot,\cdot\,)$ and linear form $L_F(\,\cdot\,)$ are then defined by

$$\begin{aligned} a_F((v,q),(U_F^n,P_F^n)) &= (v,(U_F^n-U_F^{n-1})k_n^{-1}-w\cdot\nabla U_F^n)+(\varepsilon(v),\sigma_F(U_F^n,P_F^n)) \\ &- (q,\nabla\cdot U_F^n), \end{aligned}$$

$$L_F((v,q)) = (v,f_F)+(v,g_{F,N})_{\partial K_{F,N}}.$$

The Structure Problem

For the structure problem (4), we simply seek the approximate displacement $U_S^n \in W_S^h(t_n) \subset W_S(t_n)$ where W_S is the space of continuous piecewise linear vector–valued function on $\mathcal{T}_S$, satisfying the Dirichlet boundary condition. Hence, the finite element method for (4) reads: For $n = 1, ..., N$ find $U_S^n \in W_S^h(t_n)$ for $n = 1, ..., N$ such that

$$a_S(v,U_S^n) = L_S(v), \quad \forall v \in \hat{W}_S^h(t_n), \tag{6}$$

where $\hat{W}_S^h(t_n)$ denotes the test space of continuous piecewise linear vector–valued functions on $\mathcal{T}_S$ vanishing on the Dirichlet boundary. The bilinear form $a_S(\,\cdot,\cdot\,)$ and linear form $L_S(\,\cdot\,)$ are defined by

$$a_S(v,U_S^n) = (\varepsilon(v),\sigma_S(U_S^n)),$$

$$L_S(v) = (v,f_F)-(v,\sigma_F(U_F^n,P_F^n)n_F)_{\partial K_{SF}}.$$

Model Problem

As a first step towards simulating the dynamic fluid–structure interaction in a heart valve, we consider a simple two dimensional channel flow containing a single leaflet. As a model problem, we consider an immersed elastic beam in a pressure–driven pulsative flow in the two–dimensional channel shown in Figure 1.

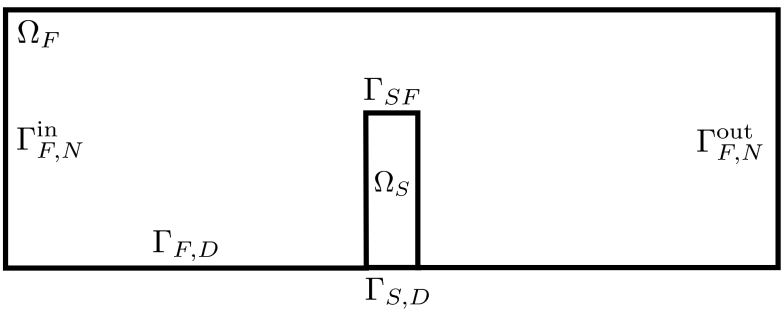


Figure 1: The two-dimensional channel with the immersed beam.

Boundary Conditions

For boundary conditions, we consider a pressure driven flow. To simulate a time–dependent pulsative blood flow, we let the inlet pressure be given by the ODE

$$\begin{cases} \dot{p}_F^{\text{in}}(t) = & c_1\ \theta_\tau(t) - c_2\ \chi(p_F^{\text{in}}(t))\ Q(t), \\ p_F^{\text{in}}(0) = & 0. \end{cases} \tag{7}$$

Here, θ_τ is a Heaviside step function with period τ and

$$Q(t) = \int_{\Gamma_{F,N}^{\text{out}}} u_F \cdot n_F\ ds.$$

Furthermore, χ is the characteristic function such that

$$\chi(p_F^{\text{in}}) = \begin{cases} 1, & p_F^{\text{in}} > 0, \\ 0, & p_F^{\text{in}} \leq 0, \end{cases}$$

and c_1, c_2 some given positive constants.

The outlet pressure is set to

$$p_F^{\text{out}}(t) = c_3\ Q(t)$$

for a given positive constant c_3.

These pressure boundary conditions are imposed weakly. The boundary integral of the Neumann boundary is split up into

$$\begin{aligned} -\int_0^T (v, \sigma_F(u_F, p_F) n_F)_{\Gamma_{F,N}}\ dt \quad = \quad & -\eta \int_0^T (v, \nabla u_F \cdot n_F)_{\Gamma_{F,N}}\ dt - \eta \int_0^T (v, \nabla u_F^\top \cdot n_F)_{\Gamma_{F,N}}\ dt \\ & + \int_0^T (v, \bar{p}_F n_F)_{\Gamma_{F,N}}\ dt, \end{aligned}$$

where $\bar{p}_F$ is the solution to the above described ODE for the pressure. This term is denoted `bcp` in Figure 3. Further, we assume that the term $\eta \int_0^T (v, \nabla u_F \cdot n_F)_{\Gamma_{F,N}}\ dt = 0$, i.e., the flow is considered fully developed. For the Dirichlet part of the boundary, we consider pure homogeneous boundary conditions, i.e., no–slip condition for the fluid and the structure is attached to the channel wall.

Implementation in FEniCS

FEniCS [3, 7, 9, 14, 10, 13, 12] is a collection of several stand-alone software components which together form a program environment for solving ordinary and partial differential equations. FEniCS automates central aspects of the finite element method. One of these components is the form compiler FFC [11] which takes as input a variational problem together with a set of finite elements and generates low-level code for the automatic computation of the discrete system of equations. FFC is available as a Python module allowing variational problems to be specified and computed within the Python scripting environment.

To specify the weak form for the fluid problem (1) in FEniCS, one must first specify the function space `W` along with a set of test functions `(v, q)` and trial functions `(u, p)` as shown in Figure 2. The differential operators $\nabla\cdot$ and ∇, are available as `div` and `grad` respectively. The bilinear form `a` and the linear form `L` for the fluid may then be defined as shown in Figure 3. The forms for the structure problem are defined in a similar manner. The coupled FSI problem (1)–(2) is solved by simple fixed point iteration as shown in Figure 4.

```
# Create function spaces
V = VectorFunctionSpace(mesh, "CG", 2)
Q = FunctionSpace(mesh, "CG", 1)
W = V + Q

# Create test and trial functions
(v, q) = TestFunctions(W)
(u, p) = TrialFunctions(W)
```

Figure 2: Definition of function spaces and test and trial functions for the fluid problem.

```
# Define epsilon
def epsilon(u):
    return 0.5 * (grad(u) + transp(grad(u)))

# Define sigma
def sigma(u, p):
    return 2.0 * eta * epsilon(u) - mult(I, p)

# Create forms
a = dot(v, u) * dx \
  - dot(v, mult(grad(u), X1 - X0)) * dx \
  + k * dot(epsilon(v), sigma(u, p)) * dx \
  - k * q * div(u) * dx \
  - k * eta * dot(v, mult(transp(grad(u)), n))*ds

L = dot(v, u0) * dx \
  - k * bcp * dot(v, n)*ds \
  + k * dot(v, f) * dx
```

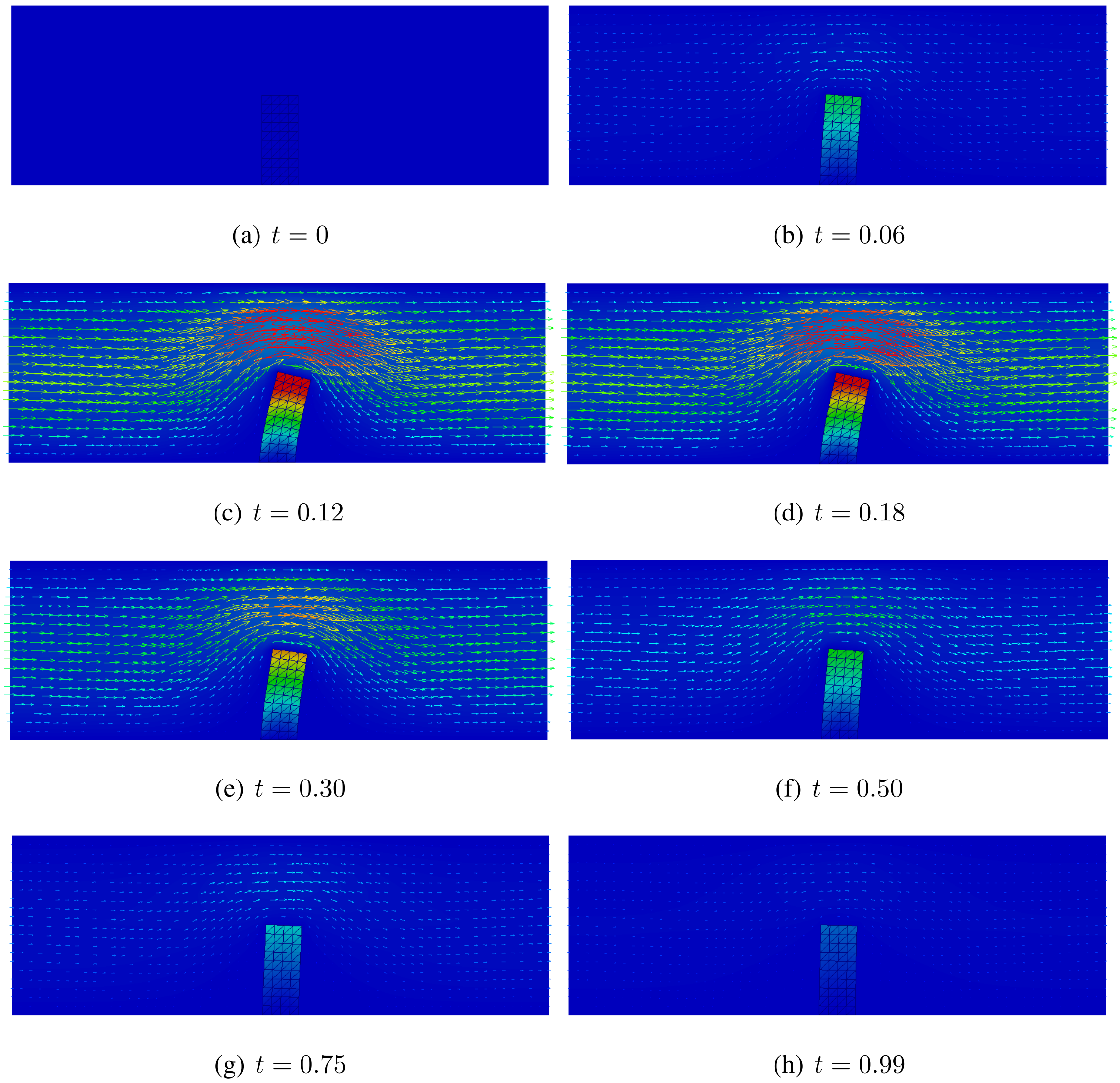
Figure 3: Definition of the bilinear and the linear forms and for the fluid problem. Here, `u0` and `p0` are solutions from previous time steps and `X1 - X0` is the velocity of the fluid coordinate system scaled by the time step k_n. Further, `dot(v,u) * dx` denotes the inner product between two functions `v` and `u`.

```
def solve(problem):

while t < T:

# FSI iteration
for i in range(maxiter):

    # Solve fluid problem
    (u_F, p_F) = fluid_solver.solve()

    # Solve structure problem
    (u_S, du_S) = structure_solver.solve(u_F, p_F)

    # Move structure mesh
    problem.structure_mesh.move(du_S)

    # Move fluid mesh
    problem.fluid_mesh.move(problem.structure_mesh)

    # Smooth mesh
     problem.fluid_mesh.smooth(50)

    # Check for convergence
    if norm(du_S) < TOL:
        break

# Check that the problem converged
if i == maxiter - 1:
   raise RuntimeError, "FSI iteration did not converge."

# Update problem
problem.update(t, u_F, p_F, u_S)

# Move to next time interval
t += k
```

Figure 4: The FSI problem is solved using a simple fixed point algorithm. In each FSI iteration, the mesh is moved according to the displacement given from the structure problem.

Results

In the simulations, shown in Figure 5, the structure responses to the pressure driven flow. Our given ODE (7), pictured in Figure 6, yields a pulsative flow. The displacement of the beam reaches its maximum when the inlet pressure hits its peak value. Also, the beam reaches its original position when the inlet pressure is zero. The method presented in this paper works for small deformations on the structure. However, if we impose larger deformations, the mesh becomes folded and we cannot obtain a solution due to bad mesh quality.



(a) $t = 0$ (b) $t = 0.06$

(c) $t = 0.12$ (d) $t = 0.18$

(e) $t = 0.30$ (f) $t = 0.50$

(g) $t = 0.75$ (h) $t = 0.99$

Figure 5: The solution at different times t during one period.

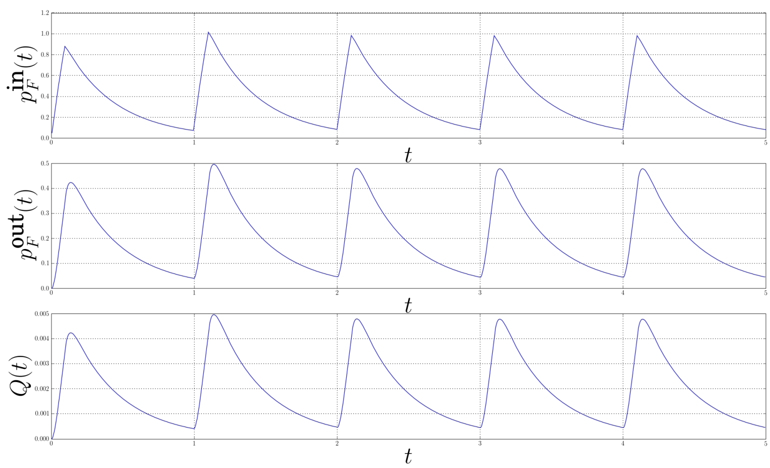


Figure 6: The inlet/outlet pressure and the flux over five periods.

Conclusions

We have demonstrated a simple implementation of a FSI problem in FEniCS. For structures that undergo small deformations, the approach with matching meshes is sufficient. However, for large deformations the mesh folds and a solution cannot be computed. To overcome this we intend to (i) improve the current mesh smoothening algorithms implemented in FEniCS (for example using the variable diffusion method [6]) and (ii) consider Nitsche's method [5] for FSI problems on overlapping non-matching meshes.

References

[1] M.Astorino, J.-F.Gerbeau, O.Pantz and K.-F.Traoré Fluid–structure interaction and multi-body contact: Application to aortic valves *Comp. Meth. Appl. Mech. Engrg*, **vol.doi:10.1016/j.cma.2008.09.012**, 2008.

[2] K.Eriksson, D.Estep and C.Johnson *Applied Mathematics: Body and Soul [volume 3]* Springer, 2004.

[3] FEniCS FEniCS project URL: `http://www.fenics.org/`, 2009.

[4] A.Gerstenberger and W.Wall An extended finite element method / lagrange multiplier based approach for fluid-structure interaction *Comp. Meth. Appl. Mech. Engrg*, **vol.197**, 1699–1741, 2007.

[5] A.Hansbo, P.Hansbo and M.Larsson A finite element method on composite grids based on Nitsche's method *ESIAM:M2AN*, **vol.37**, 495–514, 2003.

[6] P.Hansbo and J.Heramansson A variable diffusion for mesh smoothening *Commun. Nummer. Meth. Engng.*, **vol.19**, 897–908, 2003.

[7] R. C.Kirby and A.Logg Efficient compilation of a class of variational forms *ACM Transactions on Mathematical Software*, **vol.33**(3), 2007.

[8] M. G.Larson and F.Bengzon Adaptive finite element approximation of multiphysics problems: A fluid structure interaction model Technical report, Institutionen för matematik och matematisk statistik, 2009.

[9] A.Logg Automating the finite element method *Arch. Comput. Methods Eng.*, **vol.14**, 93–138, 2007.

[10] A.Logg *FFC User Manual*, 2007 URL: `http://www.fenics.org/ffc/`.

[11] A.Logg et al. *FFC*, 2009 `http://www.fenics.org/ffc/`.

[12] A.Logg and G. N.Wells DOLFIN: Automated finite element computing *Submitted*, 2009.

[13] A.Logg, G. N.Wells, J.Hoffman and J.Jansson *DOLFIN*, 2006 `http://www.fenics.org/dolfin/`.

[14] A.Logg, G. N.Wells, J.Hoffman and J.Jansson *DOLFIN User Manual*, 2007 URL: `http://www.fenics.org/dolfin/`.

[15] C.Taylor and P.Hood A numerical solution of the Navier-Stokes equations using the finite element technique. *Internat. J. Comput. and Fluids*, **vol.1**, 73–100, 1973.

[16] K.van der Zee, E.van Brummelen and R.de Borst Goal-oriented error estimation for Stokes flow interacting with a flexible channel *Int. J. Numer. Meth. Fluids*, **vol.56**, 1551–1557, 2008.

[17] W.Wall and T.Rabczuk Fluid-structure interaction in lower airways of CT– based lung geometries *Int. J. Num. Meth. Fluids*, **vol.57**, 653–675, 2008.