

Contents

1	Introduction	3
2	Medical Background	4
2.1	Anatomy of the central nervous system	4
2.2	The Spinal Cord	4
2.3	The Chiari malformation	5
2.3.1	Syringomelia	6
3	Mathematical background	7
3.1	Fluid flow	7
3.1.1	Reynolds Transport Theorem	8
3.1.2	Conservation of mass and momentum	8
3.1.3	Incompressible Newtonian fluids	9
3.1.4	Navier-Stokes equations for incompressible flow	10
3.2	Linear Elasticity	10
3.3	Linear Poroelasticity	11
3.3.1	Biot's Equations	11
3.4	Descriptions of Motion	11
3.4.1	The arbitrary Lagrangian-Eulerian description	12
4	Numerical Methods	15
4.1	The Finite Element Method	15
4.1.1	Variational formulation	15
4.1.2	Finite elements	17
4.1.3	Implementation in FEniCS	18
4.2	A benchmark FSI-problem	20
4.2.1	Domain, Initial- and boundary conditions	21
4.2.2	CFD tests	22
4.2.3	CSM tests	22
4.2.4	FSI tests	23
4.3	Fluid Structure Interaction using the Finite Element method	24
4.3.1	A note on temporal discretization	25
4.3.2	Spatial discretization	25
4.3.3	Treatment of boundary conditions	26
4.3.4	A FSI solver in FEniCS	27
4.4	A Discontinuous Galerkin method	30
4.4.1	Stokes flow	31

5	Material parameters	32
6	Simulating interaction between CSF and the Spinal Cord	33
6.1	Overview of previous studies	33
6.2	Assessment of CSF velocities before and after decompression surgery	34

Chapter 1

Introduction

The Cerebrospinal Fluid (CSF) surrounds the brain and acts as a protection to the brain inside the skull. As a result of the cardiac cycle the CSF will flow up and down the subarachnoid space (SAS) surrounding the spinal cord. The Chiari malformation is a displacement of the cerebellar tonsils that partially blocks CSF flow entering the SAS around the spinal cord. This causes abnormal CSF flows which sometimes results in a syringomyelia inside the spinal cord filled with fluid. Treatment may include surgery to remove parts of the bones of the skull to relieve pressure. Studies have shown that the syrinx gradually vanishes after surgery. The mechanisms behind this are not yet fully understood. Many researchers have suggested Computational Fluid Dynamics (CFD) to give useful insight, as experiments are very difficult and expensive.

Chapter 2

Medical Background

In this chapter, relevant background information from medicine is presented. The information provided here is similar to previous studies focusing on the pathogenesis of the Chiari malformation at Simula Research Laboratory and the University of Oslo. Example of studies with relevance include the works of Støverud, Kylstad and Drosdal. [\[1\]](#) [\[2\]](#) [\[3\]](#).

2.1 Anatomy of the central nervous system

The human nervous system consists of peripheral nervous system, and the central nervous system (CNS). The former consists of spinal and cranial nerves and sensory receptor organs while the latter consists of two parts: the brain and the spinal cord. The CNS receives and process information from all parts of the body. Consequently, studies on the CNS are crucial for our understanding of the human anatomy.

2.2 The Spinal Cord

The spinal cord carries information between the body and the brain and is divided into four regions from top to bottom: Cervical (C), thoracic (T), lumbar (L) and sacral (S). Along the cord, there are 31 pairs of nerves exiting the cord: 8 in the cervical region, 12 in the thoracic region, 5 in the lumbar region and 5 in the sacral region. These segments of the cord are given names C1-C8, T1-T12, L1-L5, S1-S5. The tissue within the spinal cord consists of nervous tissue in form of white and grey matter. These two are different in both structure and function. In the center of the spinal cord, lies a tiny central spinal canal where CSF can flow. This channel closes off with age. Surrounding the central canal lies the gray matter in an H-shape, similar to a butterfly. The rest consists of white matter, and the ratio between white and gray matter differs along the spinal cord.

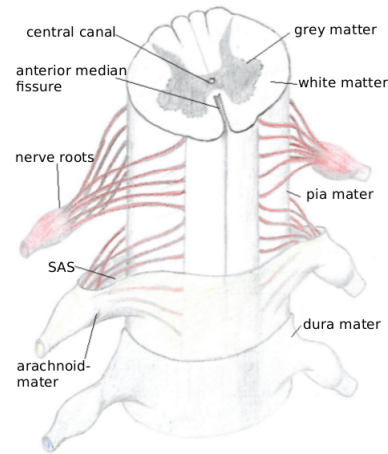


Figure 2.1: Schematic figure of the spinal cord

2.3 The Chiari malformation

The Chiari malformation, also known as Arnold-Chiari Malformation is a neurological condition where a displacement of the cerebellum, or more precisely the cerebellar tonsils, down through the foramen magnum occurs. The condition is classified into four stages I-IV where IV is the most severe. (see figure ??) In a Chiari patient, the tonsils obstruct the CSF and even patients with Chiari I have shown to have greater CSF velocities and a more complex flow pattern than healthy subjects. [4] These patients could also experience severe headache, dizziness, tinnitus and muscle weakness. As the cerebellum is part of the brain that controls balance, loss of coordination has also been reported.

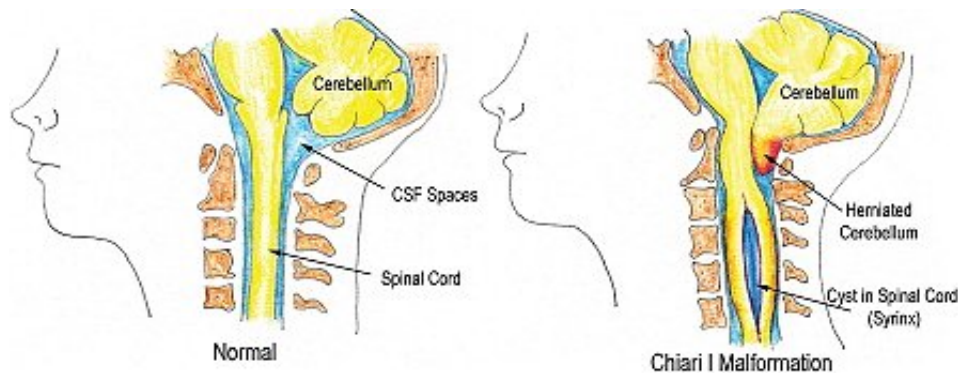


Figure 2.2: hei

2.3.1 Syringomelia

In some cases, Chiari patients develop a fluid cavity, known as a syrinx within the spinal cord. Some of the symptoms are similar to the Chiari patients in general, but include muscle and back pain, weakness, numbness and inabilities to feel temperature changes. Many theories on the pathogenesis of syringomelia have been proposed, but the details are not yet fully understood. In patients diagnosed with Chiari I, about 2/3 develop syrinxes within the spinal cord tissue. Thus, not all Chiari patients have syringomelia and in addition patients with syringomelia does not necessarily have the Chiari malformation. Thompson et al. have in a recent study suggested the anatomy of the cervical (upper part) spinal canal plays a role in the formation of syrinxes. [5] The study showed that Chiari patients with syringomelia had greater $C4-C7$ -tapering of the cervical spinal canal, but no significant difference on $C1-C4$ -tapering. Tapering is the narrowing of the spinal canal in the downwards, or caudal direction.

Most researchers seem to agree on the fact that altered CSF dynamics is associated with the formation of the syrinx.

Chapter 3

Mathematical background

The flow of CSF around the spinal cord requires equations for fluid flow to be coupled with equations for elasticity or in the optimal case, poroelasticity. The underlying concepts of these kinds of problems were originally developed somewhat independently within petroleum engineering, geomechanics and hydrogeology. The equations will first be presented separately. Later in the chapter, coupling conditions will be discussed. Several quantities will be discussed and we try to use a consistent notation for each quantity throughout the study.

In this thesis we use the following notation for the physical quantities:

\mathbf{v} - Velocity of the material. In the fluid, \mathbf{v} represents fluid velocity, and in the solid \mathbf{v} denotes the velocity of the solid.

\mathbf{U} - Total displacement of the solid. In the fluid, this quantity represents the total mesh displacement.

p - Pressure in the fluid. In the case of elasticity, the same pressure does not exist in the solid, but in the case of an (almost) incompressible solid, an extra equation involving p can be set up in the solid as well.

\mathbf{w} - Domain velocity, (i.e, all material points within the domain moves with velocity \mathbf{w}). In the solid, the domain changes with the velocity of the solid, \mathbf{v} , so in the solid we have $\mathbf{v} = \mathbf{w}$ which will not necessary hold in the fluid.

Subscripts f and s are used to denote fluid and solid quantities, respectively

3.1 Fluid flow

The most fundamental equations in fluid flow are conservation laws. These equations are based on classical mechanics and states conservation of mass, momentum and energy. In the literature, these are often referred to as balance equations.

3.1.1 Reynolds Transport Theorem

The famous engineer and scientist Osbourne Reynold stated the general conservation law the following way [6]:

Any change whatsoever in the quantity of any entity within a closed surface can only be effected in one or other of two distinct ways:

1. it may be effected by the production or destruction of the entity within the surface, or
2. by the passage of the entity across the surface.

Now, consider a control volume, V_0 and some fluid property $Q(\mathbf{x}, t)$. The rate of change of Q within the control volume can be written

$$\frac{d}{dt} \int_{V_0} Q(\mathbf{x}, t) dV$$

The net change of Q must be equal the rate of change in Q within the control volume plus the net rate of mass flow out of the volume. In other words

$$\frac{d}{dt} \int_{V_0} Q(\mathbf{x}, t) dV = \int_{V_0} \frac{\partial Q(\mathbf{x}, t)}{\partial t} dV + \int_{S_0} Q(\mathbf{x}, t) \mathbf{v} \cdot \mathbf{n} dS \quad (3.1)$$

Here, \mathbf{v} denotes fluid velocity, and \mathbf{n} denotes the outward pointing unit-normal, i.e \mathbf{n} points *out* of the fluid. This equation is known as the Reynold's transport theorem. The right hand side could be rewritten by using Gauss' theorem on the last term.

$$\frac{d}{dt} \int_{V_0} Q(\mathbf{x}, t) dV = \int_{V_0} \left[\frac{\partial Q(\mathbf{x}, t)}{\partial t} + \nabla \cdot (Q(\mathbf{x}, t) \mathbf{v}) \right] dV \quad (3.2)$$

where

$$\nabla = \mathbf{i}_j \frac{\partial}{\partial x_j}$$

3.1.2 Conservation of mass and momentum

Choose $Q(\mathbf{x}, t) = \rho$. Conservation of mass means that

$$\frac{d}{dt} \int_{V_0} \rho dV = 0$$

And by using the transport theorem (3.2)

$$\int_{V_0} \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) \right] dV = 0$$

This should hold for any volume V_0 , hence the integrand has to be zero:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (3.3)$$

(3.3) is known as the continuity equation and states conservation of mass.

To derive a similar property for the momentum, Newton's second law of motion can be used. The net change of momentum must be equal to the applied forces to the system. The forces can be divided into volume forces, acting on the entire control volume, and forces acting only on the control surface. The forces acting on the surface can be written $\sigma_f \cdot \mathbf{n}$, where $\sigma_f = \sigma_f(\mathbf{v}, p)$ is the tensor denoting the total stress.

This can be written

$$\frac{d}{dt} \int_{V_0} \rho \mathbf{v}(\mathbf{x}, t) dV = \int_{\partial V_0} \sigma_f \cdot \mathbf{n} dS + \int_{V_0} \mathbf{F}_v dV$$

By using the Transport Theorem on the left hand side and with Gauss' theorem on the right hand side we end up with

$$\int_{V_0} \left[\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) - \nabla \cdot \sigma_f - \mathbf{F}_v \right] dV = 0$$

With the same argument as before the integrand has to be zero, and with some rearrangement:

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) = \nabla \cdot \sigma_f + \mathbf{F}_v \quad (3.4)$$

The stress tensor, σ_f , depends on fluid properties and will be defined in the next subsection. Equation (3.4) is known as the momentum equation as it states conservation of momentum.

3.1.3 Incompressible Newtonian fluids

In this text we will only consider incompressible fluid flow for a Newtonian fluid. The assumption of a Newtonian fluid requires the viscous stresses to be linear functions of the components of the strain-rate tensor, denoted by ϵ . These assumptions were first made by Stokes in 1845. Stokes' assumptions have later proven to be quite accurate for all gases and most common fluids. Stokes' three postulates regarding the deformation laws are: [7]

1. The fluid is continuous, and its stress tensor, $\sigma_{f_{ij}}$ is at most a linear function of the strain rates, ϵ_{ij}
2. The fluid is isotropic, i.e., its properties are independent of direction, and therefore the deformation law is independent of the coordinate axes in which it is expressed.
3. When the strain rates are zero, the deformation law must reduce to the hydrostatic pressure condition, $\sigma_{f_{ij}} = -p\delta_{ij}$, where δ_{ij} is the Kroenecker delta function.

From the first and third condition the following assumption can be made

$$\sigma_{f_{ij}} = -p\delta_{ij} + M_{ijkl}\epsilon_{kl} \quad (3.5)$$

It can be shown that symmetry of σ_f and ϵ also requires symmetry of M . This assumption reduces the number of coefficients in (3.5) from 36 to 21. If Stokes'

second condition is also taken into account and the fluid properties are identical in each direction the number of coefficients are further reduced to 2. These simplifications allow us to denote the stress tensor the following way:

$$\sigma_{f_{ij}} = -p\delta_{ij} + 2\mu\epsilon_{ij} + \lambda\nabla \cdot \mathbf{v} \quad (3.6)$$

where $\epsilon = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})$, p is the fluid pressure and μ and λ are known as Lamé's constants. In the present study we only consider incompressible flow where ρ is constant. From the continuity equation (3.3), this implies $\nabla \cdot \mathbf{v} = 0$ and the last term in (3.6) vanishes. Furthermore,

$$\nabla \cdot \epsilon = \frac{\partial}{\partial x_j}(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j})\mathbf{i}_i = (\frac{\partial}{\partial x_i}\frac{\partial u_j}{\partial x_j} + \frac{\partial u_i}{\partial x_j\partial x_j})\mathbf{i}_i = \frac{\partial u_i}{\partial x_j\partial x_j}\mathbf{i}_i$$

Which simplifies the representation of $\nabla \cdot \sigma_f$ in (3.4) for an incompressible fluid

3.1.4 Navier-Stokes equations for incompressible flow

Stating both conservation of mass and momentum of a fluid together with suitable boundary conditions gives us all the information we need to be able to define the flow field and the corresponding pressure. This requires a solution to the system (3.3)-(3.4), equations which are commonly referred to as the Navier-Stokes equations. With the simplifications described in the previous section the system of equations can be written

$$\rho(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v}) = -\nabla p + \mu\nabla^2\mathbf{v} + \mathbf{F}_v$$

$$\nabla \cdot \mathbf{v} = 0$$

The parameters ρ and μ describe density and dynamic viscosity. Often, the momentum equation is written in terms of the kinematic viscosity $\nu = \frac{\mu}{\rho}$ by dividing the momentum equation with ρ .

These equations are coupled and non-linear and can generally not be solved analytically. However, numerous analytical solutions have been carried out for different specific problems, see e.g. White [7] pp...xxx. These problems are often very simple and idealized. Hence, numerical solutions are a necessity to obtain useful solutions to real-life problems. Such methods will be discussed in chapter xxx.

3.2 Linear Elasticity

For an elastic medium the stress tensor has no pressure term, and the stress is related to the total displacement \mathbf{U} rather than the velocity \mathbf{v} . The stress tensor for such a material reads: $\sigma_s = 2\mu\epsilon(\mathbf{U}) + \lambda\nabla \cdot \mathbf{U}$

$$\rho\frac{\partial^2\mathbf{U}}{\partial t^2} = \nabla \cdot \sigma_s$$

The left hand side can also be expressed from the solid velocity \mathbf{v} using that $\frac{\partial^2\mathbf{U}}{\partial t^2} = \frac{\partial \mathbf{v}}{\partial t}$. The advantage of using \mathbf{v} as the unknown will be discussed in chapter xxx

3.3 Linear Poroelasticity

In this section, the equations describing fluid flowing through a elastic medium is presented. For a more detailed discussion, derivation and history within the field we refer to [8] on Linear Poroelasticity. To keep the mathematics as similar to the fluid case as possible, we use μ and λ instead of the poisson ratio.

3.3.1 Biot's Equations

The stress tensor for the Biot problem is

$$\sigma_s = -\alpha p I + 2\mu \epsilon(\mathbf{U}) + (\mu + \lambda) \text{tr}(\epsilon(\mathbf{U})) I$$

Here μ and λ are Lamé's parameters for the solid. The parameter $\alpha = \frac{K}{H}$ is known as the Biot-Willis coefficient. K is known as the drained bulk modulus, and $\frac{1}{K}$ denotes compressibility. H is a poroelastic parameter describing how much the bulk volume changes due to a change in pore pressure while holding the applied stress constant. Again, conservation of momentum and mass, respectively, yields

$$-\mu \nabla^2 \mathbf{U} - \lambda \nabla \nabla \cdot \mathbf{U} + \nabla p = \mathbf{F}_v \quad (3.7)$$

$$(\nabla \cdot \mathbf{U})_t - \nabla \cdot (\mu_f^{-1} \mathbf{K} \nabla p) = 0 \quad (3.8)$$

Where μ_f is the dynamic viscosity of the fluid and the t-subscript denotes time derivative. As described in [9], $-\mu^{-1} \mathbf{K} \nabla p$ represents the fluid velocity in the porous medium relative to the solid movement. In other words, the total fluid movement in the poroelastic medium is $\mathbf{U}_t - \mu_f^{-1} \mathbf{K} \nabla p$. \mathbf{K} is known as the permeability matrix. In an isotropic medium we can assume that \mathbf{K} is a scalar constant, K .

Equations (3.7)-(3.8) are nothing more than superpositioning of the two phases. The material derivatives in (3.7) has been dropped under the assumption that these terms are small. This assumption is known as a quasistatic approximation.

3.4 Descriptions of Motion

In the previous section, we saw that the stress tensor for elastic solids were linked to the total displacement from the stress-free configuration of the material. The stresses and velocity in the material will depend on the current deformation of the material with respect to the stress-free configuration. To this end, it will be convenient to provide the reader with two classical descriptions of a continuum in motion.

We consider a domain $\Omega_{\mathbf{X}} \in \mathbb{R}^3$ consisting of material particles \mathbf{X} . The domain can undergo deformations, and the deformed domain, $\Omega_{\mathbf{x}}$, is the current configuration at time t . We define the mapping:

$$\begin{aligned} \beta : \Omega_{\mathbf{X}} \times [0, T] &\rightarrow \Omega_{\mathbf{x}} \times [0, T] \\ (\mathbf{X}, t) &\rightarrow \beta(\mathbf{X}, t) = (\mathbf{x}, t) \end{aligned}$$

Which takes any point \mathbf{X} in the reference configuration to a new position $\mathbf{x} = \beta(\mathbf{X}, t)$ at time t . The time is measured with the same variable, t , in both domains. The gradient of β with respect to (\mathbf{X}, t) can be written in matrix form as:

$$\frac{\partial \beta}{\partial (\mathbf{X}, t)} = \begin{pmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{X}} & \mathbf{v} \\ 0^T & 1 \end{pmatrix}$$

where the material velocity

$$\mathbf{v}(\mathbf{X}, t) = \left. \frac{\partial \mathbf{x}}{\partial t} \right|_{\mathbf{X}} \quad (3.9)$$

is the temporal change in the spatial variable \mathbf{x} while holding \mathbf{X} fixed. 0^T denotes a null vector.

The *Lagrangian* description, where we follow a fixed set of material particles as suggested by the mapping β , is often used. In the Lagrangian description all quantities are expressed in terms of the reference configuration $\Omega_{\mathbf{X}}$ and time. In other words, even though the material is deformed, we can still compute displacements and particle velocities using the material coordinates \mathbf{X} . For instance, the displacement from the starting material configuration will be given as $\beta(\mathbf{X}, t) - \mathbf{X}$ and the velocity as given in (3.9).

Because the grid coincides with the material coordinates, there are no convective terms in the Lagrangian description. When a material undergoes large deformations or for instance vortices or turbulence occur, the material velocity from the Lagrangian point of view becomes difficult to handle.

In fluid mechanics the *Eulerian* description is the most used, which means that fluid flows through a fixed region in space and in each point we can measure various properties or quantities such as velocity, pressure and temperature. The conservation equations in the Eulerian description are expressed in terms of the spatial coordinates \mathbf{x} and time, and are neither connected to a reference configuration nor the material coordinates. Compared to the Lagrangian approach, large material deformations is not a problem, as material can enter and leave the fixed domain. This movement of a material through a fixed region results in convective effects, and convection operators can often be problematic in computational fluid dynamics due to their non-symmetric nature.

To be able to couple the two, we still need some adjustment from the usual Eulerian description. The region is no longer fixed, and we will now describe what is known as an *arbitrary Lagrangian-Eulerian* (ALE) description.

3.4.1 The arbitrary Lagrangian-Eulerian description

The following derivation is inspired by the the works on Arbitrary Lagrangian-Eulerian methods by Donea et. al in [10].

We consider a domain $\Omega_{\mathbf{X}} \in \mathbb{R}^3$ consisting of material particles \mathbf{X} . The domain can undergo deformations, and in the spatial domain \mathbf{x} , we refer to the current configuration at time t as $\Omega_{\mathbf{x}}$. We define the mapping:

$$\begin{aligned} \beta : \Omega_{\mathbf{X}} \times [0, T] &\rightarrow \Omega_{\mathbf{x}} \times [0, T] \\ (\mathbf{X}, t) &\rightarrow \beta(\mathbf{X}, t) = (\mathbf{x}, t) \end{aligned}$$

Which takes any point \mathbf{X} in the reference configuration to a new position $\mathbf{x} = \beta(\mathbf{X}, t)$ at time t . The time is measured with the same variable, t , in both domains. The gradient of β with respect to (\mathbf{X}, t) can be written in matrix form as:

$$\frac{\partial \beta}{\partial(\mathbf{X}, t)} = \begin{pmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{X}} & \mathbf{v} \\ 0^T & 1 \end{pmatrix}$$

where the material velocity

$$\mathbf{v}(\mathbf{X}, t) = \left. \frac{\partial \mathbf{x}}{\partial t} \right|_{\mathbf{X}}$$

is the temporal change in the spatial variable \mathbf{x} while holding \mathbf{X} fixed. 0^T denotes a null vector.

$$\mathbf{U}(\mathbf{X}, t) = \chi(\mathbf{X}, t) - \mathbf{X} \quad \mathbf{w} = \frac{\partial \chi}{\partial t} \quad \mathbf{F} = \frac{\partial \chi}{\partial \mathbf{X}}, \quad J = \det \mathbf{F}$$

Relative to some fixed point in space, the fluid moves with velocity \mathbf{v} and the domain moves with velocity \mathbf{w} .

Because the domain itself is moving, we need to recall the balance equation from the Transport Theorem in the form (3.1). The velocity carrying the net flux of mass across the boundary will now be $\mathbf{v} - \mathbf{w}$

As in the benchmark case, consider a compressible structure defined by the region Ω_s^t immersed in an otherwise fixed region Ω_f^t containing an incompressible fluid. The outer boundaries of the fluid is named $\partial\Omega_f^t$. The interface is named Γ .

$$\frac{d}{dt} \int_{\Omega_f^t} \rho_f dV = \frac{\partial}{\partial t} \int_{\Omega_f^t} \rho_f dV + \int_{\partial\Omega_f^t} \rho_f (\mathbf{v} - \mathbf{w}) \cdot \mathbf{n}_{\Omega_f^t} dS = 0$$

If the solid expands or contracts with speed \mathbf{w} , the rate of change of its volume will be

$$- \int_{\Gamma} \mathbf{w} \cdot \mathbf{n} dS$$

Because of the rigid outer boundaries, the fluid will have the same rate of change, but with opposite sign. The fluid has a constant density, ρ_f , so the rate of change in mass in the fluid domain can be written

$$\frac{d}{dt} \int_{\Omega_f^t} \rho_f dV = \int_{\Gamma} \rho_f \mathbf{w} \cdot \mathbf{n} dS$$

The left hand side can be rewritten by the transport theorem.

$$\int_{\Omega_f^t} \frac{\partial \rho_f}{\partial t} dV + \int_{\partial\Omega_f^t} \rho_f (\mathbf{v} - \mathbf{w}) \cdot \mathbf{n} dS + \int_{\Gamma} \rho_f (\mathbf{v} - \mathbf{w}) \cdot \mathbf{n} dS = \int_{\Gamma} \rho_f \mathbf{w} \cdot \mathbf{n} dS$$

Where the boundary integral have been split up into two parts. If the boundary conditions $\mathbf{v} = \mathbf{w}$ on Γ , and $\mathbf{w} = 0$ on $\partial\Omega_f^t$ are taken into account we end up with:

$$\int_{\Omega_f^t} \frac{\partial \rho_f}{\partial t} dV + \int_{\partial\Omega_f^t} \rho_f \mathbf{v} \cdot \mathbf{n} dS + \int_{\Gamma} \rho_f \mathbf{v} \cdot \mathbf{n} dS = 0$$

Or with the use of Gauss theorem, one can conclude that the local form

$$\frac{\partial \rho_f}{\partial d} + \nabla \cdot \rho_f \mathbf{v} = 0$$

also holds in the ALE formulation. Therefore, independendtly of the compressibility of the solid, an incompressible fluid will still satisfy $\nabla \cdot \mathbf{v} = 0$

xx Read Donea paper [\[10\]](#)

Both equations need appropriate boundary conditions on the interface as well as the boundary of the domain. For the interface we have to ensure mass conservation and balance of the forces. These two conditions can be written as:

$$\left. \begin{array}{l} \sigma_f \cdot \mathbf{n} = \sigma_s \cdot \mathbf{n} \\ \mathbf{v}_f = \mathbf{v}_s \end{array} \right\} \text{ on } \Gamma_t$$

Chapter 4

Numerical Methods

4.1 The Finite Element Method

The theory presented in this section is inspired by the great work of Langtangen in "Finite Element Method - INF5620 lecture notes" [11]

Consider the Poisson-equation

$$-\nabla^2 v = f \quad \text{in } \Omega \quad (4.1)$$

$$v = v_0 \quad \text{on } \partial\Omega_D$$

$$\frac{\partial v}{\partial n} = g \quad \text{on } \partial\Omega_N \quad (4.2)$$

where $\Omega \in \mathbb{R}^d$ is a domain, $v = v(x)$ is an unknown function and f is a source function. The boundary, $\partial\Omega$ is divided into two parts. $\partial\Omega_D$ for the Dirichlet boundary condition, and $\partial\Omega_N$ for the Neumann condition.

4.1.1 Variational formulation

(4.1) is known as the strong form of the equation. To reformulate the problem and state a weak formulation we multiply the equation with a test function, $\phi \in \hat{V}$, where \hat{V} is some function space, and integrate over the domain. Weak formulations are important in the sense that differential equations can be transformed into systems of linear equations. In the rest of this text the following notation is used for the inner product of two functions

$$(v, \phi)_\Omega = \int_\Omega v \phi \, dx$$

By multiplying (4.1) with a test function, ϕ and integrating over the domain, the weak form is obtained

$$(\nabla^2 v + f, \phi)_\Omega = 0 \quad \forall \phi \in \hat{V}$$

We are now searching for a v to satisfy the weak form instead of the strong. This equation should hold for all ϕ in the function space \hat{V} . The trial function does not necessarily have to lie in the same function space, in general $v \in V$.

In this thesis we will use two Sobolov spaces (named after the Russian mathematician Sergei Sobolov) widely used in Finite Element computing. For these definitions to be valid, we assume that the functions v are all **locally integrable** and in the case of definition (4.2), has one **weak derivative**. For more details on weak derivatives and the generalized concept of Sobolev spaces and functional analysis, we refer to the textbook by Brenner and Scott [12].

Definition 4.1 *Let Ω be an open subset of \mathbb{R} with a piecewise smooth boundary. We then define the L^2 -norm as follows*

$$\|v\|_{L^2(\Omega)} = (\int_{\Omega} v^2 dx)^{\frac{1}{2}}$$

The corresponding L^2 -space is defined via

$$L^2(\Omega) = \{v : \Omega \rightarrow \mathbb{R} \mid \int_{\Omega} v^2 dx < \infty\}$$

Definition 4.2 *Let Ω be an open subset of \mathbb{R} with a piecewise smooth boundary. We then define the H^1 -norm as follows*

$$\|v\|_{H^1(\Omega)} = (\int_{\Omega} [v^2 + (\nabla v)^2] dx)^{\frac{1}{2}}$$

The corresponding H^1 -space is defined via

$$H^1(\Omega) = \{v : \Omega \rightarrow \mathbb{R} \mid \int_{\Omega} [v^2 + (\nabla v)^2] dx < \infty\}$$

In other words, using functions from these spaces, we can have some assurance that the integrals involved in the variational form are bounded. By the divergence theorem, a generalized concept of integration by parts we can state the variational problem as follows: find $v \in V$ such that

$$(\nabla v, \nabla \phi)_{\Omega} = (f, \phi)_{\Omega} + (g, \phi)_{\partial\Omega_N} \quad \forall \phi \in \hat{V} \quad (4.3)$$

Where we have used that $\frac{\partial v}{\partial n} = g$ on $\partial\Omega_N$. (4.3) is known as the variational formulation of the Poisson problem. The right hand side is known as the bilinear form while the left hand side is the linear form. In generic form the equation can be written

$$a(v, \phi) = L(\phi) \quad (4.4)$$

The first derivative of v appears in the variational form. A common choice is then

$$\begin{aligned} V &:= \{v \in H^1(\Omega) : v = v_0 \text{ on } \partial\Omega_D\} \\ \hat{V} &:= \{v \in H^1(\Omega) : v = 0 \text{ on } \partial\Omega_D\} \end{aligned}$$

, where the subscript v_0 denotes that $v = v_0$ on the Dirichlet part of the boundary.

4.1.2 Finite elements

The next step is to approximate v with a sum of basisfunctions in the finite-dimensional function space, $V = \text{span}\{\phi_0, \phi_1, \dots, \phi_N\}$. Here, ϕ_i represents the basis functions and we search for a solution $v_h \in V$ such that v_h can be written as a linear combination of the basis functions. The first step in the finite element method consists of dividing the domain into smaller parts

$$\Omega = \Omega_0 \cup \Omega_1 \cup \dots \cup \Omega_{N_e}$$

where N_e is the number of elements. Each element have a number of nodes within them depending on what type of basis functions to be used. Let's first consider the continuous Galerkin basis functions, in a one-dimensional domain. There is exactly one basisfunction for each node located at x_i . These basis functions have the property that

$$\phi_i(x_j) = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}$$

That is, the basis functions ϕ_i are zero on all nodes except at node i . Each basis function is constructed by taking the Lagrange-polynomial which is 1 at the given node and 0 on all other nodes within the same element. Note that the basis functions for a node on the boundary of an element will have two Lagrange-polynomials "pieced together" depending on at which element the basis function is considered. For the rest of the domain the basis functions are defined to be 0.

Now, let's return to the original problem (4.1)-(4.2) in scalar form. We start by approximating v as a linear combination of all the basis functions.

$$v_h = \sum_{i=0}^N c_i \phi_i$$

The definitions of v_h and V now give rise to a linear system. Using the Einstein summation convention, $x_i y_i = \sum_0^N x_i y_i$, (4.3) is now written

$$-c_i (\nabla \phi_i, \phi_j)_\Omega = (f, \phi_j)_\Omega - (g, \phi_j)_{\partial\Omega_N}$$

In the case of Dirichlet boundary conditions all test functions ϕ_j will take the value 0 on $\partial\Omega_D$ and the linear system will be adjusted to take these boundary conditions into account.

The system can be written in matrix form, and in the end the problem consists of solving the linear system

$$A_{i,j} c_i = b_j$$

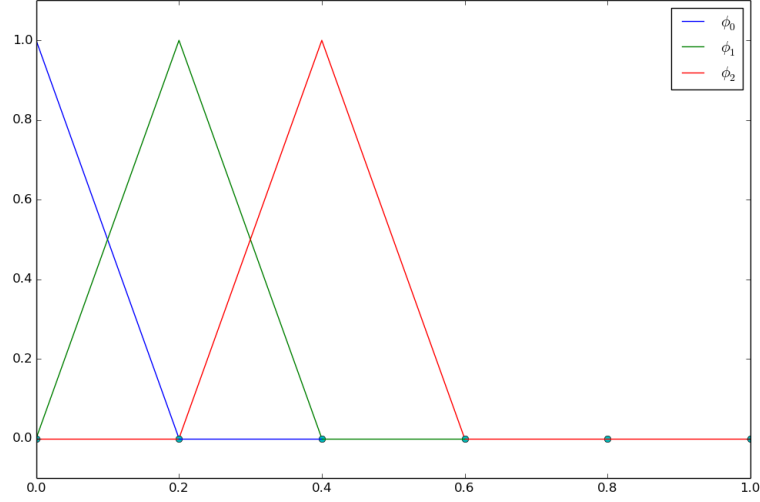


Figure 4.1: The three first linear basis functions on the unit interval divided uniformly into 5 elements

4.1.3 Implementation in FEniCS

When the variational form has been carried out, implementation in FEniCS is relatively simple. The programs in this study are written in the Python programming language. When programming with Python, we first need to import `dolfin` to access the DOLFIN library, containing classes convenient and efficient for finite element computing. In Python the full library can be imported as simple as

```
from dolfin import *
```

Now, let's focus our attention on solving the following problem:

$$\begin{aligned} \nabla^2 v &= 20x \quad \text{in } \Omega \\ v(0, y) &= 0, \quad v(1, y) = 1 \\ \frac{\partial v(x, 0)}{\partial n} &= \frac{\partial v(x, 1)}{\partial n} = 0 \end{aligned}$$

Where Ω is the unit square, $\Omega = [0, 1] \times [0, 1]$

First of all we need to define the computational mesh.

```
mesh = UnitSquareMesh(10,10)
```

The class `UnitSquareMesh` initializes a mesh with triangular cells. The mesh consists of $n \times m$ squares depending on the arguments, n and m , sent into the constructor. Each of these squares are divided on the diagonal to form two

triangles, and these triangles are the computational cells. In this case we get the unit square divided into 10×10 smaller squares and thus the total number of triangles, or cells, will be 200.

The next thing to do is to define an appropriate function space for the test functions. The solution will be a linear combination of these functions and will be in (almost) the same function space.

```
V = FunctionSpace(mesh, 'CG', 1)
```

The function space needs a domain, type of element, and the degree of the element. In this case we use Continuous Galerkin elements ('CG') with degree 1. These basis functions are visualized in Figure 4.1

We can then define our test- and trial functions v and ϕ

```
v = TrialFunction(V)
phi = TestFunction(V)
```

Note that the test- and trial function seem to be in the exact same function space. This is the case except when imposing Dirichlet boundary conditions. The functions f , v_0 and g can be defined by using the 'Constant' or 'Expression' classes. We set $f = 20x$ and use Dirichlet boundary conditions, $v(0, y) = 0$, $v(1, y) = 1$ and Neumann conditions $\frac{\partial v(x, 0)}{\partial n} = \frac{\partial v(x, 1)}{\partial n} = 0$. The homogenous Neumann condition is simple in the finite element method as the terms appearing after integration by parts can be dropped. If this is not the case, we can insert g for $\frac{\partial u}{\partial n}$ on the boundary integral appearing in the variational form. When the Neumann conditions are incorporated this way we say that the boundary conditions are weakly imposed. Functions (or classes) describing the boundaries must also be defined:

```
f = Expression('20*x[0]')
def boundary0(x, on_bnd):
    return on_bnd and near(x[0], 0.0)
def boundary1(x, on_bnd):
    return on_bnd and near(x[0], 1.0)

bc0 = DirichletBC(V, 0.0, boundary0)
bc1 = DirichletBC(V, 1.0, boundary1)
bcs = [bc0, bc1]
```

Note that 'x[0]' means first dimension in space, 'x[1]' means second dimension and so on. The Dirichlet conditions are put in a list. Next, the variational form is defined, and when solving for a function v , the boundary conditions are added to the "magic" solve function.

```
F = inner(grad(v), grad(phi))*dx - inner(f, phi)*dx
v = Function(V)
solve(lhs(F)==rhs(F), v, bcs)
plot(v)
```

The functions lhs and rhs separates the form F into the left hand side, equivalent to the bilinear form and to the right hand side, equivalent to the linear form.

Specifying the full form F is simple and convenient when the equations are short and simple. If we want to relate the code to the mathematics as written in (4.4) we can, define these forms manually.

```
a = inner(grad(v),grad(phi))*dx
L = inner(f,phi)*dx
v = Function(V)
solve(a==L,v,bcs)
```

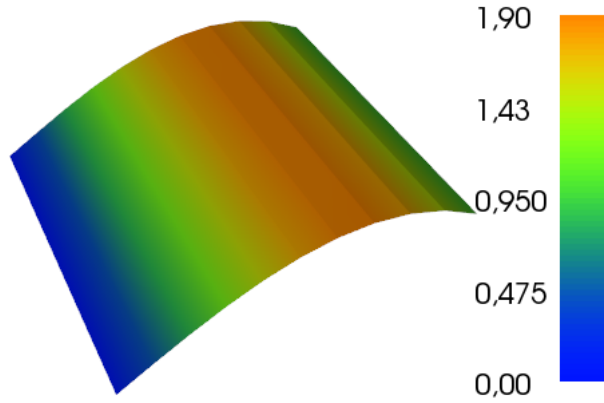


Figure 4.2: Plot of the computed solution

The plot in the figure is slightly rotated. The solution is independent of y -position, as expected.

4.2 A benchmark FSI-problem

Within CFD, a benchmark is a configuration or a test case which should help test and compare different numerical methods and code implementations. A classical Fluid Dynamics problem regarding flow around a circular cylinder has been under vast research the last 50 years, working as a test case for both laminar and turbulent flows. One of the most cited benchmark proposals for this case is the problem described by Michael Schafer et. al in 1996 [13]. The research group still focus on these kinds of problems and one of the co-authors of the 1996 paper, Stefan Turek, together with Jaroslav Hron has proposed a similar benchmark for FSI solvers, consisting of the exact same domain and rigid cylinder, but now with an elastic flag attached to it [14].

The first results presented will contain a validation of the present FSI-solver implemented in FEniCS compared to the results of Turek and Hron in their benchmark proposal.

A proper validation of a FSI solver requires separate verification of the fluid and structural parts as well as coupled tests. In the present study we first solve

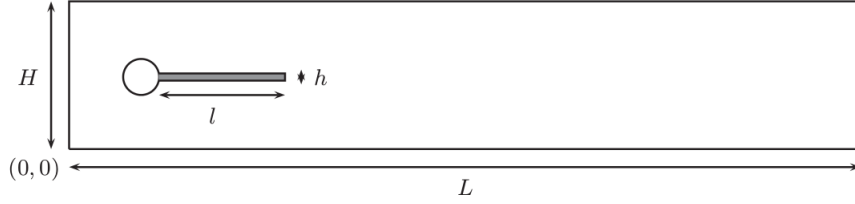


Fig. 1. Computational domain

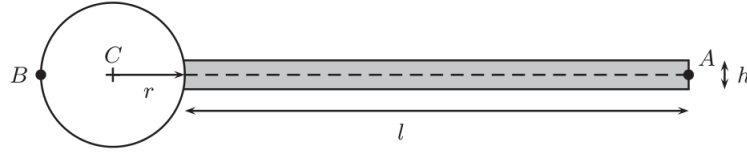


Fig. 2. Detail of the structure part

Figure 4.3: The Domain as published in [14]

the equations with a *monolithic* approach, i.e fully coupling between the fluid and solid. The alternative would be a *partitioned* approach, where the fluid and solid equations are solved separately. For instance, one can solve the fluid equations independently and then proceed by solving the solid equation with prescribed stress on the interface computed from the fluid solution. Iteration back and forth would be needed until convergence.

The fully coupled monolithic scheme is usually preferred with respect to accuracy and stability. Also, when the systems are strongly coupled in nature, i.e. the solid movement is affected by the fluid movement and vice versa, a monolithic scheme would be advantageous. The partitioned approach, on the other hand, can benefit from numerous previous studies where efficiency and stability for various solution techniques have been investigated. See e.g. [15] for a short review.

4.2.1 Domain, Initial- and boundary conditions

The origin is set at the bottom left corner. We also set:

- The channel height, $H = 0.41$
- The channel length, $L = 2.5$
- The circle center $C = (0.2, 0.2)$
- The right bottom corner of the elastic structure has position $(0.6, 0.19)$
- The elastic structure has length, $l=0.35$ and height $h=0.02$
- At the left boundary, the inlet, of the channel, we set a prescribed parabolic velocity profile

$$\mathbf{v}_{\text{in}}(0, y) = 1.5\bar{v}_0 \frac{y(H-y)}{(\frac{H}{2})^2}$$

- In the case of unsteady flow a smooth increase in time is used:

$$\mathbf{v}_{\text{in}}(t, 0, y) = \begin{cases} \mathbf{v}_{\text{in}}(0, y) \frac{1 - \cos(\frac{\pi}{2}t)}{2} & \text{if } t < 2.0 \\ 0 & \text{otherwise} \end{cases}$$

- On the outlet, the condition $\sigma \cdot \mathbf{n} = 0$ is applied - On rigid walls the no-slip condition is used - On the interface, Γ^t , the previously described coupling conditions are applied:

$$\left. \begin{aligned} \sigma_f \cdot \mathbf{n} &= \sigma_s \cdot \mathbf{n} \\ \mathbf{v}_f &= \mathbf{v}_s \end{aligned} \right\} \text{ on } \Gamma^t$$

In addition to the interface, Γ^t , it may be convenient to define the fluid boundary, $\partial\Omega_f^t$ consisting of the outer rectangle and the part of the circle in contact with the fluid, and the solid boundary, $\partial\Omega_s^t$ consisting of the circle in contact with the solid.

4.2.2 CFD tests

For the CFD tests we perform tests treating the flag as a rigid object. This can be done by changing the structural parameters, or simply by adjust the mesh to include the fluid domain only. In this validation we choose the latter.

Parameter	CFD1	CFD2	CFD3
$\rho_f [10^3 \frac{\text{kg}}{\text{m}^3}]$	1	1	1
$\nu_f [10^{-3} \frac{\text{m}^2}{\text{s}}]$	1	1	1
\bar{v}_0	0.2	1	2
$\text{Re} = \frac{Ud}{\nu_f}$	20	100	200

Table 4.1: Parameters for the CFD test cases

Mesh refinement	Cells	Drag	Lift
0	1334	13.9344	1.0980
1	5336	14.1165	1.0836
2	21334	14.1865	1.0944
Ref. (Hron, Turek)		14.29	1.119

Table 4.2: Results for CFD1

4.2.3 CSM tests

The structural tests are performed by adding the gravitational force to the structural part only. The CSM3 test is computed as a time-dependent case, starting from the initial position while CSM1 and CSM2 are Steady State (SS) solutions.

Mesh refinement	Cells	Drag	Lift
0	1	1	1
1	1	1	1
2	.2	1	2
Ref. (Hron, Turek)		136.7	10.53

Table 4.3: Results for CFD2

Parameter	CSM1	CSM2	CSM3
$\rho_s [10^3 \frac{\text{kg}}{\text{m}^3}]$	1	1	1
ν_s	0.4	0.4	0.4
$\mu_s [10^6 \frac{\text{m}^2}{\text{s}}]$	0.5	2	0.5
$g [\frac{\text{m}^2}{\text{s}}]$	2	2	2

Table 4.4: Parameters for the CSM test cases

Mesh refinement	Cells	U_x of A $[10^{-3}]$	U_y of A $[10^{-3}]$
0	738	-12.410569	-60.599246
1	2952	-12.419505	-60.622920
2	11808	-12.422290	-60.630433
Ref. (Hron, Turek)		-7.187	-66.10

Table 4.5: Results for CSM1

Mesh refinement	Cells	U_x of A $[10^{-3}]$	U_y of A $[10^{-3}]$
0	738	-0.92479395	-16.853778
1	2952	-0.92558954	-16.861757
2	11808	-0.92583356	-16.864252
Ref. (Hron, Turek)		-0.4690	-16.97

Table 4.6: Results for CSM2

4.2.4 FSI tests

Parameter	FS1	FSI2	FSI3
$\rho_f [10^3 \frac{\text{kg}}{\text{m}^3}]$	1	1	1
$\nu_f [10^{-3} \frac{\text{m}^2}{\text{s}}]$	1	1	1
\bar{v}_0	0.2	1	2
$\text{Re} = \frac{Ud}{\nu_f}$	20	100	200

- note: Hron, Turek, $\Delta t = 0.0005$, we used $\Delta t = 0.0003$.

Parameter	FSI1	FSI2	FSI3
$\rho_s [10^3 \frac{\text{kg}}{\text{m}^3}]$	1	10	1
ν_s	0.4	0.4	0.4
$\mu_s [10^6 \frac{\text{m}^2}{\text{s}}]$	0.5	0.5	2

Mesh refinement	Cells	U_x of A [10^{-3}]	U_y of A [10^{-3}]	Drag	Lift
0	2698	0.015596	0.74221	14.0876279441	0.756130219216
1	10792	0.017738	0.77686	14.1777783843	0.763145083966
2					
Ref. (Hron, Turek)		0.0227	0.8209	14.295	0.7638

Table 4.7: Results for FSI1

Mesh refinement	Cells	U_x of A [10^{-3}]	U_y of A [10^{-3}]	Drag	Lift
0	2698	-4.33 ± 4.54	1.40 ± 29.96	441.45 ± 33.15	-2.30 ± 178.00
1	10792				
2					
Ref. (Hron, Turek)		-2.69 ± 2.53	1.48 ± 34.38	457.3 ± 22.66	2.22 ± 149.78

Table 4.8: Results for FSI3

Numbers for FSI3: U_x : Max = 0.0002099, Min = -0.008879
 U_y : Max = 0.03136, Min = -0.02856
Drag: Max = 474.6, Min = 408.3
Lift: Max = 175.7, Min = -180.3 (174.4, -177.0)

4.3 Fluid Structure Interaction using the Finite Element method

The solver was implemented from scratch in Python using the DOLFIN library. FSI solvers within the FEniCS framework exists, e.g. under Unicorn or CBC.twist. For instance, Selin [16] implemented a partitioned solver in FEniCS in his PhD-thesis, using the already existing modules for solving fluid flow and structural deformations separately (CBC.Flow and CBC.Twist). However, the benefits of being in full control of the equations and all boundary conditions involved are obvious, especially in the process of understanding the complexity of the physical problem. In addition to this, it will be easier to adjust the solver with respect to the model. The equations can be changed whether we want the spinal cord to be porous, elastic, poroelastic or viscoelastic. The use of two separate solvers is also problematic for the monolithic approach, when all equations are solved simultaneously. On the other hand, solvers implemented by experienced and skilled research groups will probably be a lot more efficient and should already have been validated.

In the rest of this section, we give a brief explanation to the mathematics and implementation in FEniCS. In the previous example we saw the close link between code and mathematics. :

$$\begin{aligned} a(\mathbf{v}, \Phi) &= (\nabla \mathbf{v}, \nabla \Phi)_\Omega \\ L(\mathbf{v}) &= (\mathbf{f}, \Phi)_\Omega \end{aligned}$$

Translates to

```
a = inner(grad(v), grad(phi)*dx)
L = inner(f, phi)*dx
```

In deriving variational forms, we try to keep this close link by recalling the symbols used for velocity (\mathbf{v}), total displacement (\mathbf{U}), pressure (p), and mesh velocity, (\mathbf{w}). Also recall that in the solid the mesh moves exactly with the velocity of the structure so $\mathbf{w}_s = \mathbf{u}_s$

The total displacement \mathbf{U} can now be expressed as a function of the displacement from the previous time step, $\mathbf{U}^{(1)}$, and the mesh velocity \mathbf{w} . In an implicit scheme, for instance, $\mathbf{U} = \mathbf{U}^{(1)} + \Delta t \mathbf{w}$

4.3.1 A note on temporal discretization

Throughout the thesis we have used the notation $\mathbf{v} = \mathbf{v}^{n+1}$ to denote the unknown functions at the time step we want to proceed to. If we use an implicit scheme in time, i.e, $(\frac{\partial \mathbf{v}}{\partial t})^{n+1} \approx \frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\Delta t}$, and $\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \mathbf{w}^{n+1}$ where n denotes time step number. For the known time step n , we have used $\mathbf{v}^{(1)} = \mathbf{v}^n$ and for the guess in the linearized form, $v^{(0)}$.

4.3.2 Spatial discretization

Since we have three unknown functions, we use a mixed function space with three function spaces, Φ , η and Ψ . In the fluid we multiply the momentum equation with Φ , the continuity equation with η and the equation for mesh velocity with Ψ and integrate over the domain in its *current* configuration, Ω^t . In the fluid, this yields

$$\frac{\rho_f}{\Delta t}(\mathbf{v}, \Phi)_{\Omega_f} + \rho_f((\mathbf{v} - \mathbf{w}) \cdot \nabla) \mathbf{v}^{(0)}, \Phi)_{\Omega_f^t} - (p, \nabla \cdot \Phi)_{\Omega_f^t} + 2\mu_f(\epsilon(\mathbf{v}), \epsilon(\Phi))_{\Omega_f^t} =$$

$$\frac{\rho_f}{\Delta t}(\mathbf{v}^{(1)}, \Phi)_{\Omega_f^t} - (\sigma_f(p, \mathbf{v}) \cdot \mathbf{n}, \Phi)_{\partial \Omega_f^t} - (\sigma_f(p, \mathbf{v}) \cdot \mathbf{n}_f, \Phi)_{\Gamma^t}$$

$$-(\nabla \cdot \mathbf{v}, \eta)_{\Omega_f^t} = 0$$

$$\begin{aligned} \Delta t(\nabla \mathbf{w}, \nabla \Psi)_{\Omega_f^t} &= -(\nabla \mathbf{U}^{(1)}, \nabla \Psi)_{\Omega_f^t} + ([\nabla \mathbf{U}^{(1)} + \Delta t \nabla \mathbf{w}] \cdot \mathbf{n}, \Psi)_{\Omega_f^t} \\ &\quad + ([\nabla \mathbf{U}^{(1)} + \Delta t \nabla \mathbf{w}] \cdot \mathbf{n}_f, \Psi)_{\Gamma^t} \end{aligned}$$

And in the solid

$$\begin{aligned} \frac{\rho_s}{\Delta t}(\mathbf{v}, \Phi)_{\Omega_s} + \Delta t(\sigma_s(\mathbf{v}), \nabla(\Phi))_{\Omega_s} &= \frac{\rho_s}{\Delta t}(\mathbf{v}^{(1)}, \Phi)_{\Omega_s} - (\sigma_s(\mathbf{U}^{(1)}), \nabla\Phi)_{\Omega_s} \\ &\quad - ([\sigma_s(\mathbf{U}^{(1)}) + \Delta t\sigma_s(\mathbf{w})] \cdot \mathbf{n}, \Phi)_{\partial\Omega_s^t} \\ &\quad - ([\sigma_s(\mathbf{U}^{(1)}) + \Delta t\sigma_s(\mathbf{w})] \cdot \mathbf{n}_s, \Phi)_{\Gamma^t} \end{aligned}$$

$$\frac{1}{\delta}(\mathbf{v}, \Psi)_{\Omega_s} - \frac{1}{\delta}(\mathbf{w}, \Psi)_{\Omega_s} = 0$$

The parameter δ should be small and ensures the importance of $\mathbf{v}_s = \mathbf{w}_s$ inside the solid. On the interface, we have distinguished between the normal vector with respect to the fluid and solid domain. In general $\mathbf{n}_f = -\mathbf{n}_s$. To be able to set up and assemble the matrices for this system, the equations should be added to form one bilinear form and one linear form.

4.3.3 Treatment of boundary conditions

In addition to the boundary conditions described in the original benchmark paper from Turek and Hron, homogenous Dirichlet conditions are prescribed to the mesh displacement velocity on the domain boundary, i.e

$$\mathbf{w} = 0 \text{ on } \partial\Omega_f^t \cup \partial\Omega_s^t$$

Except for the fluid velocity on the outlet, the domain boundaries (not interface) have prescribed Dirichlet conditions on both \mathbf{u} and \mathbf{w} . Therefore the test functions Φ and Ψ will be zero on these boundaries.

If we add all the equations in the previous section together, the contributions to the boundary integral on the interface gives:

$$-(\sigma_f(p, \mathbf{v}) \cdot \mathbf{n}_f, \Phi)_{\Gamma^t} - ([\sigma_s(\mathbf{U}^{(1)}) + \Delta t\sigma_s(\mathbf{w})] \cdot \mathbf{n}_s, \Phi)_{\Gamma^t}$$

Dropping this contribution from the variational form gives

$$\sigma_f(p, \mathbf{v}) \cdot \mathbf{n} = [\sigma_s(\mathbf{U}^{(1)}) + \Delta t\sigma_s(\mathbf{w})] \cdot \mathbf{n} = \sigma_s(\mathbf{U}) \cdot \mathbf{n}$$

where the choice of \mathbf{n} ($\mathbf{n} = \mathbf{n}_f$ or $\mathbf{n} = \mathbf{n}_s$) is arbitrary, but the same for each side of the equation.

Because we use the same function for fluid velocity and solid velocity we will also achieve the no-slip condition for the fluid on the structure

$$\mathbf{v}_f = \mathbf{v}_s \text{ on } \Gamma^t$$

Because the functions \mathbf{v}_f and \mathbf{v}_s share nodes on the interface.

The additional equation for \mathbf{w} in the fluid also gives rise to boundary conditions on $\nabla\mathbf{U} \cdot \mathbf{n}$ on the interface. To this end we set

$$\nabla\mathbf{U} \cdot \mathbf{n} = 0$$

and let the parameter δ underline the importance of $\mathbf{w} = \mathbf{v}$ *inside* the solid, whereas \mathbf{w} in the fluid should just be smoothed out.

On the outlet, we assign the stress-free condition $\sigma_f(p, \mathbf{v}) \cdot \mathbf{n} = 0$ so the boundary integral also vanish on the outlet for the momentum equation in the fluid.

This means that all integrals involving boundaries will vanish in the variational form. The Dirichlet conditions are imposed in FEniCS as previously described.

4.3.4 A FSI solver in FEniCS

There will be some changes and a great leap in complexity compared to the previous example using FEniCS. The main differences and additions are explained here. The computational mesh is constructed in gmsh with a straight boundary dividing the fluid and the solid. This way, the class MeshFunction can be utilized by dividing the mesh in two subdomains. We now assume we have classes describing the solid and fluid region, implemented with functions simliary to the boundary functions in the Poisson example.

```
mesh = Mesh('FSI_mesh.xml')
SD = MeshFunction('uint', mesh, mesh.topology().dim())
SD.set_all(0)
Elastic().mark(SD,1)
```

where

```
class Elastic(SubDomain):
    def inside(self,x,on_bnd):
        # returns True if vector x in solid.
```

'uint' means that the MeshFunction has values of nonnegative integers. The last argument ensures the MeshFunction to have the same dimension as the mesh. Using the MeshFunction, the fluid domain have been marked 0, and the solid domain have been marked 1. Integration over the two domains can be separated by passing this number to dx in the variational formulation. A similar class, the FacetFunction

```
boundaries = FacetFunction("size_t",mesh)
```

is used to mark the boundaries and, if needed, separate integration over specific parts of the boundary.

'uint' means that the MeshFunction has values of nonnegative integers. 'size_t' means the same for the FacetFunction. The last argument to MeshFunction ensures the MeshFunction to have the same dimension as the mesh.

We need a function spaces for all three testfunctions, corresponding to \mathbf{v}, p and \mathbf{w} , and in this case we can use a handy FEniCS class to create a mixed function space. Test -and trial functions should also be created from this mixed space.

```
V = VectorFunctionSpace(mesh, 'CG', 2)
P = FunctionSpace(mesh, 'CG', 1)
W = FunctionSpace(mesh, 'CG', 2)
VPW = MixedFunctionSpace([V, P, V])
v, p, w = TrialFunctions(VPW)
phi, eta, psi = TestFunctions(VPW)
```

All Dirichlet boundary conditions need to be specified, and the functions need to be in the space of the respective trial function where the condition is set. For instance, the top boundary of the domain have been marked 2 with the `FacetFunction`, and we want to prescribe the no-slip condition on the fluid velocity.

```
noslip = Constant((0.0, 0.0))
bcv2 = DirichletBC(VPW.sub(0), noslip, boundaries, 2) # Top
```

All the Dirichlet boundary conditions are put together in a list, `bcs`.

When the `Mesh` -and `FacetFunctions` have been properly marked, we need to map the information from these classes to the different measures, `dx`, `ds` and `dS` representing integration over cells, exterior facets and interior facets, respectively. This is done by:

```
dS = Measure('dS')[boundaries]
dx = Measure('dx')[SD]
ds = Measure('ds')[boundaries]

dx_f = dx(0, subdomain_data=SD)
dx_s = dx(1, subdomain_data=SD)
```

Sometimes, it can be convenient to define the constants used in FEniCS as instances of the class `Constant`, to avoid re-compiling if the value of the constant is changed. e.g.

```
dt = 0.0003
k = Constant(dt)
```

We can now attention our focus to the variational form. Regular Python functions can be used in the variational formulation, and by defining these two

```
def sigma_s(U):
    return 2*mu_s*sym(grad(U)) + lamda*tr(sym(grad(U)))*Identity(2)

def eps(v):
    return 2*mu_f*sym(grad(v))
```

The variational form is very similar to the mathematics. We define the bilinear and linear forms, `a` and `L`, for each separate equation, momentum, continuity and the movement of the domain in both the fluid and solid domain (except for continuity in the solid). For instance, `aMF` will denote the bilinear form, `a`, for the momentum equation in the fluid.

```

# FLUID
aMF = rho_f/k*inner(v,phi)*dx_f + \
      rho_f*inner(grad(v0)*(v-w),phi)*dx_f - \
      inner(p,div(phi))*dx_f + \
      2*mu_f*inner(eps(v),eps(phi))*dx_f

LMF = rho_f/k*inner(v1,phi)*dx_f

aCF = -inner(div(v),eta)*dx_f

aDF = k*inner(grad(w),grad(psi))*dx_f
LDF = -inner(grad(U),grad(psi))*dx_f

aF = aMF + aCF + aDF
LF = LMF + LDF

# SOLID
aMS = rho_s/k*inner(v,phi)*dx_f + \
      k*inner(sigma_s(u),grad(phi))*dx_f

LMS = rho_s/k*inner(v1,phi)*dx_f - \
      inner(sigma_s(U),grad(phi))*dx_f

aDS = 1/delta*inner(u,w)*dx_f - \
      1/delta*inner(d,w)*dx_f

aS = aMS + aDS
LS = LMS

```

We can now add the forms together to obtain one bilinear and one linear form

```

a = aS + aF
L = LS + LF

```

Before the time loop starts we define a function for holding the solution:

```

VPW_ = Function(VPW)

```

This function will consist of all values for \mathbf{v} , p and \mathbf{w} .

The time loop runs until the current time exceeds the specified end time, T . The forms change in time, and thus needs to be assembled to be updated to use the correct values for $\mathbf{v}^{(1)}$, $\mathbf{w}^{(1)}$, $\mathbf{U}^{(1)}$ and $\mathbf{v}^{(0)}$. The linear form needs an update each time step, while the bilinear form needs to be updated every single iteration inside the time loop. For the iterative method, we have (for now) chosen the Picard iteration based on the simplicity of the algorithm compared to Newton's method, especially when dealing with a mixed function space consisting of three separate spaces. The iteration runs until the L^2 norm of $(\mathbf{v} - \mathbf{v}^{(0)})$ is less than a given number, α , or if the number of iterations becomes to large. [4.3](#)

```

while t < T:
    ...
    b = assemble(L)
    ...
    while error > alpha and k_iter < max_iter:
        A = assemble(a)
        A = ident.zeros()
        [bc.apply(A,b) for bc in bcs]
        solve(A,VPW_.vector,b,'lu')
        v_,p_,w_ = VPW_.split(True)
        eps = errornorm(v_,v0,degree_rise=3)
        k_iter += 1

    v0.assign(v_)

```

The second statement within the iteration loop is needed because the lack of an equation for p within the solid. The `ident.zeros()` function replaces zeros with ones on the diagonal of the matrix block, and the solution vector for p will be zero inside the solid. To assign a new value for v_0 , and later be able to calculate drag and lift, we split the solution vector with the argument `True`. The solver is 'lu' by default, but in this case it is written explicitly. When converging iterative solvers are in general way faster, but in this case no Krylov Solver was found to converge.

The next problem to address is how the mesh should be updated. The domain should now move with velocity \mathbf{w} , so we want to move the mesh with $\Delta t \mathbf{w}$ from one time step to the next. For the total displacement, the update $\mathbf{U} = \mathbf{U}^{(1)} + \Delta t \mathbf{w}$ should also be taken into account. The actual update of the mesh is done with the functions `move()` and `bounding_box_tree().build()`

```

w_vector()[:] *= float(k)
U_.vector()[:] += w_vector()[:]
mesh.move(w_)
mesh.bounding_box_tree().build(mesh)

v1.assign(v_)

```

The final line is to update the velocity, so we can move to the next time step. Note that the velocity in both the fluid and the solid is updated by this call.

4.4 A Discontinuous Galerkin method

Discontinuous Galerkin (DG) methods is a relatively new tool for CFD simulations. The method itself was developed during the 1970s and has been used increasingly the last few decades. Unlike with the Continuous Galerkin elements, we now allow the solution to be discontinuous, i.e cells do not share nodes anymore. Instead of solving over the whole domain, we now seek approximate continuous solutions on each cell independently of the others. To this end it is

convenient to define the average and jump of a discontinuous variable

$$\{\mathbf{v}\} = \frac{1}{2}(\mathbf{v}^+ + \mathbf{v}^-) \quad [\mathbf{v}] = \mathbf{v}^+ - \mathbf{v}^-$$

where \mathbf{v}^+ and \mathbf{v}^- is the solution at two neighboring cells at cell E^+ and E^- . The normal vectors are denoted \mathbf{n}^+ and \mathbf{n}^- . If consistent, the choice of \mathbf{n} is arbitrary. [17]

4.4.1 Stokes flow

By integrating Stokes equation by parts, adding symmetry and penalty terms as in [18] we end up with a DG-method ready for use. Since the method involves many terms we derive a weak formulation for each term. All sets of facets are denoted as e , interior facets as Γ and exterior facets as $\partial\Omega$. Starting with the diffusion term $-\mu\nabla^2 u$, we multiply with a test function and integrate by parts:

$$\begin{aligned} \mu \sum_{E \in e} (\nabla \mathbf{v}, \nabla \Phi)_E - \mu \sum_{E \in \Gamma} ([\mathbf{v}], \{\nabla \Phi\} \cdot \mathbf{n}_e)_E - \mu \sum_{E \in \Gamma} ([\mathbf{v}], \{\nabla \Phi\} \cdot \mathbf{n}_e)_E + \frac{\alpha}{h} \sum_{E \in \Gamma} ([\mathbf{v}], [\Phi])_E \\ - \mu \sum_{E \in \partial\Omega} (\Phi, \nabla \mathbf{v} \cdot \mathbf{n}_e)_E - \mu \sum_{E \in \partial\Omega} (\mathbf{v}, \nabla \Phi \cdot \mathbf{n}_e)_E + \frac{\beta}{h} \sum_{E \in \partial\Omega} (\mathbf{v}, \Phi)_E \end{aligned}$$

Similiarly, the contributions from the term ∇p will be

$$- \sum_{E \in e} (p, \nabla \cdot \Phi)_E - \sum_{E \in \Gamma}$$

Chapter 5

Material parameters

In the simulations the units millimeters and grams are used. This combination gives back the SI-unit $\frac{N}{m^2} = \frac{kg \cdot m}{s^2 \cdot m^2} = \frac{g}{s^2 \cdot mm} = Pa$ for pressure, and is also convenient when considering the scale of the spinal cord. CSF is modeled as water at $37^\circ C$, i.e

parameter	value	unit
ρ_f	10^{-3}	$\frac{g}{mm^3}$
ν_f	0.658	$\frac{mm^2}{s}$

For the spinal cord, studies have shown a huge variety in material parameters. One of the most measured properties in the mammalian central nervous system is probably the Young's modulus, E . [Smith, Humphrey 2006]. In addition to this, values for the Poisson ratio, ν_P needs to be found. Smith and Huphrey used the following values for these parameters.

$$E = 5 \cdot 10^4 \text{ dyn/cm}^2 = 5000 \text{ Pa}$$

$$\nu_P = 0.479$$

From this, Lamé's parameters for the spinal cord were determined as

$$\mu_s = \frac{E}{2(1 + \nu_P)} = 1.7 \cdot 10^3 \text{ Pa}$$

$$\lambda_s = \frac{\nu_P E}{(1 + \nu_P)(1 - 2\nu_P)} = 3.9 \cdot 10^4 \text{ Pa}$$

The permeability, κ is used as a measurement for the how fluid flows in a porous medium. A large permeability indicates a pervious medium. We use the value from [Karen, Ida]

$$\kappa = 1.4 \cdot 10^{-15} m^2$$

Chapter 6

Simulating interaction between CSF and the Spinal Cord

The previously described model is used on a different geometry based on idealized versions of the spinal cord. The meshes have the same dimensions as found in previous work by Drøsdal, [3] except for the size of the syringe which is increased from 1 mm to 3 mm in diameter.

6.1 Overview of previous studies

To our knowledge, few studies have investigated fluid-structure interaction modeling of the CSF and the spinal cord including a syringe. The spinal cord has fibres oriented in the axial direction and a direction-dependent Young's modulus would then be expected. In the literature values range between 0.012 to 1.98 MPa. As reported by [19] most of spinal cord experimental studies use a tensile test, and the stress-strain and stress-relaxation responses of the spinal cord are non-linear. Therefore, it will in general be hard to compare results from different studies using different arbitrary levels of strain. Another approach used by Kwon [20], focuses more on spinal cord injuries and are thus more interested in properties during compression. From the modeling point of view, both approaches are of interest, and better constitutive models could be attained by combining results from several of these studies. In this work, we limit Young's modulus to be a constant independent of spatial direction.

Several studies have investigated the effects of FSI on the spinal cord movement and CSF pressure in geometries without a syringe. Clark [21] assumed the Young's modulus to be 1 MPa for the spinal cord, and in their initial tests this choice caused only small displacements. Therefore the conclusion was that FSI had a negligible effect on CSF pressure in the SAS.

Cheng et al. [22] investigated FSI effects on a patient-specific 3D-geometry. In their model they assumed a Youngs modulus of 0.7 MPa, and reached the same conclusion as Clark. As highlighted: *"This study informs that fluid structure interaction has no effect on CSF pressure"*.

Clearly, a too high elastic modulus will undermine the effects of FSI if they do exist. Considering the wide range of material parameters reported in the literature for the spinal cord, we believe further investigation is necessary to be able to make such a statement. In addition, these studies does not seem to investigate syringomelia as a primary target, and therefore important effects of FSI could have been overlooked in that specific case.

In porous models presented by Drøsdal, [3] fluid pressure within the cord was altered by the presence of a syrinx but the CSF pressure was not. velocities up to only $3e-7$ cm/s have been seen inside the syrinx, and therefore there must be some other effects causing the more rapid fluid movement reported.

To our knowledge, the most noted group working on FSI effects on Syringomelia include the group of Chris D. Bertram at the University of Sydney. Some of their work include research on pressure waves propagating in the spinal cord in the presence of a syrinx [23]. In this specific paper, Bertram focuses on the overlap between the cervical and thoracic segments of the spinal cord, although the mechanisms of interest remains the same, namely the 'slosh' mechanism proposed by Williams in 1980 [24]. In this theory, increased CSF pressure in the SAS is transmitted to the syrinx generating waves along the walls. Compression of the upper part of the syrinx results in a downwards displacement of the lower part resulting in enlargement of the syrinx.

Even with today's high quality magnetic resonance imaging, (MRI) or phase contrast MR, exact velocities are hard to measure. Healthy subjects also has very complex CSF flow and thus difficult to observe or quantify. Since the Chiari I malformation is associated with abnormal CSF flow, a realistic model simulating the pre-operative case needs abnormal inflow or pressure boundary conditions. The latter is extremely hard to measure exact. Frič and Eide, however, has recently taken methods into use where a sensor measuring intercranial pressure (ICP) is inserted through a cranial burr hole while patients are in local anaesthesia [25],[26]. Conclusions so far is that the pulsatile ICP as well as pulsatile pressure gradients were clearly abnormal or with boarderline values in 69 and 71 % of Chiari I patients, respectively. Without any further speculation, these numbers are very close to the number of Chiari I patients that develops a syrinx.

6.2 Assessment of CSF velocities before and after decompression surgery

We hypothesized that FSI-effects (deformation and pressure wave propagation) was at least partially responsible for the flow up to 3 cm/s within the syrinx as reported by Brucker and Haughton in an assessment of CSF velocities and

Cord Motion Before and After Chiari 1 Decompression. The study focused on the upper region (C1-C5) of the spinal cord.

xxx ALSO READ 5, 13, 21, 45 from Karen xxx

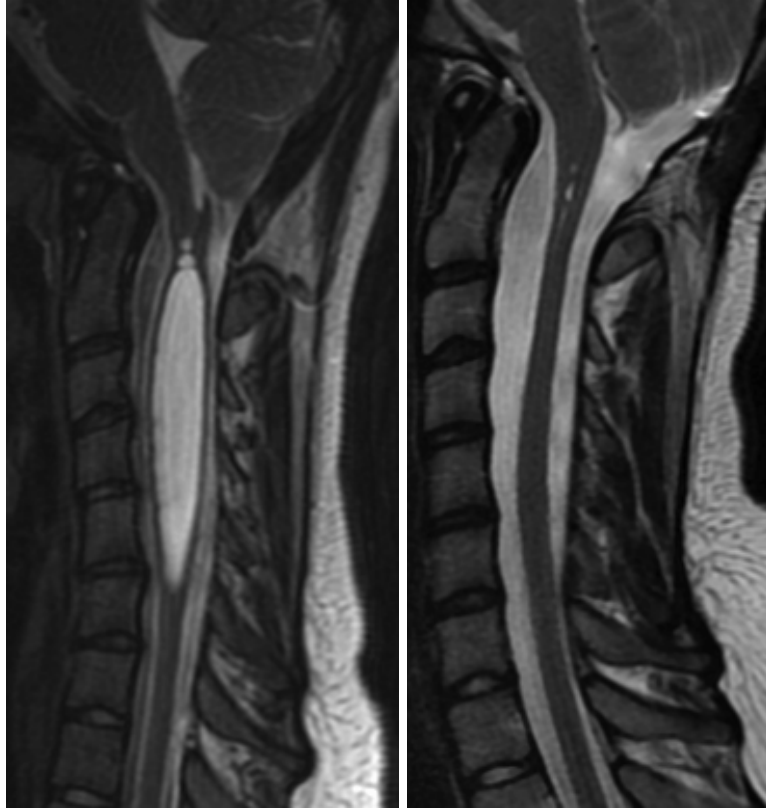


Figure 6.1: MRI image of 14 year old female subject before and 10 months after decompression. Note the withdrawal of the cerebellar tonsils in the post-operative image

CSF flow was measured with PCMR on three different stages: Pre-operative, 2 months post-operative and 10 months post-operative when the patient had no remaining symptoms.

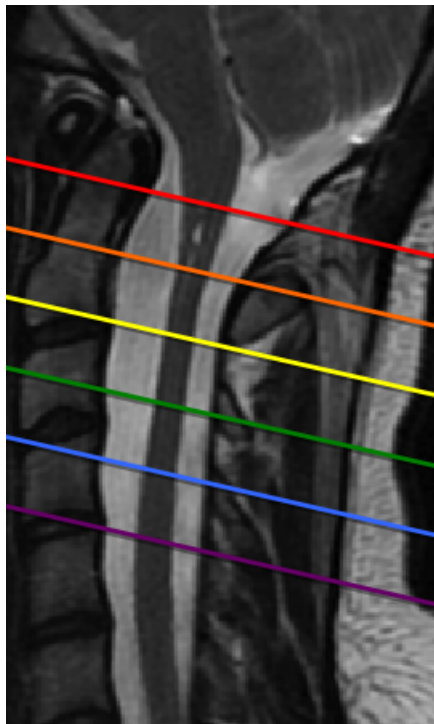


Figure 6.2: Levels C1-C2, C2, C2-C3, C3, C3-C4 and C4-C5

Bibliography

- [1] K.-H. Støverud, “Relation between the chiari i malformation and syringomyelia from a mechanical perspective,” 2014.
- [2] N. K. Kylstad, “Simulating the viscoelastic response of the spinal cord,” 2014.
- [3] I. N. Drøsdal, “Porous and viscous modeling of cerebrospinal fluid flow in the spinal canal associated with syringomyelia,” 2011.
- [4] M. F. Quigley, B. Iskandar, M. A. Quigley, M. Nicosia, and V. Haughton, “Cerebrospinal fluid flow in foramen magnum: temporal and spatial patterns at mr imaging in volunteers and in patients with chiari i malformation 1,” *Radiology*, vol. 232, no. 1, pp. 229–236, 2004.
- [5] A. Thompson, N. Madan, J. Hesselink, G. Weinstein, A. M. del Rio, and V. Haughton, “The cervical spinal canal tapers differently in patients with chiari i with and without syringomyelia,” *American Journal of Neuroradiology*, 2015.
- [6] O. Reynolds, *Papers on Mechanical and Physical Subjects, The Sub-Mechanics of the Universe*, vol. 3. Cambridge University Press, Cambridge, 1903.
- [7] F. White, *Viscous Fluid Flow*. McGraw-Hill series in mechanical engineering, McGraw-Hill, 3 ed., 2006.
- [8] H. F. Wang, *Theory of linear poroelasticity*. 2000.
- [9] D. A. Nield and A. Bejan, *Convection in Porous Media*. Springer, 4 ed., 2013.
- [10] J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodriguez-Ferran, *Encyclopedia of Computational Mechanics Vol. 1: Fundamentals., Chapter 14: Arbitrary Lagrangian-Eulerian Methods*. Wiley & Sons, 2004.
- [11] H. P. Langtangen, *Finite Element Method*. 2015.
- [12] S. Brenner and R. Scott, *The mathematical theory of finite element methods*, vol. 15. Springer Science & Business Media, 2007.
- [13] M. Schäfer, S. Turek, F. Durst, E. Krause, and R. Rannacher, *Benchmark computations of laminar flow around a cylinder*. Springer, 1996.

- [14] S. Turek and J. Hron, *Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow*. Springer, 2006.
- [15] A. Y. Tang and N. Amin, “Some numerical approaches to solve fluid structure interaction problems in blood flow,” in *Abstract and Applied Analysis*, vol. 2014, Hindawi Publishing Corporation, 2014.
- [16] K. Selim, “Adaptive finite element methods for fluidstructure interaction and incompressible flow,” 2011.
- [17] B. Rivière, *Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation*. Society for Industrial and Applied Mathematics, 2008.
- [18] B. Rivière and I. Yotov, “Locally conservative coupling of stokes and darcy flows,” *SIAM Journal on Numerical Analysis*, vol. 42, no. 5, pp. 1959–1977, 2005.
- [19] E. C. Clarke, “Spinal cord mechanical properties,” in *Neural Tissue Biomechanics*, pp. 25–40, Springer, 2010.
- [20] B. K. Kwon, T. R. Oxland, and W. Tetzlaff, “Animal models used in spinal cord regeneration research,” *Spine*, vol. 27, no. 14, pp. 1504–1510, 2002.
- [21] E. C. Clarke, D. F. Fletcher, M. A. Stoodley, and L. E. Bilston, “Computational fluid dynamics modelling of cerebrospinal fluid pressure in chiari malformation and syringomyelia,” *Journal of biomechanics*, vol. 46, no. 11, pp. 1801–1809, 2013.
- [22] S. Cheng, D. Fletcher, S. Hemley, M. Stoodley, and L. Bilston, “Effects of fluid structure interaction in a three dimensional model of the spinal subarachnoid space,” *Journal of biomechanics*, vol. 47, no. 11, pp. 2826–2830, 2014.
- [23] C. Bertram, “A numerical investigation of waves propagating in the spinal cord and subarachnoid space in the presence of a syrinx,” *Journal of Fluids and Structures*, vol. 25, no. 7, pp. 1189–1205, 2009.
- [24] B. Williams, “On the pathogenesis of syringomyelia: a review.,” *Journal of the Royal Society of Medicine*, vol. 73, no. 11, p. 798, 1980.
- [25] R. Frič and P. K. Eide, “Comparison of pulsatile and static pressures within the intracranial and lumbar compartments in patients with chiari malformation type 1: a prospective observational study,” *Acta neurochirurgica*, vol. 157, no. 8, pp. 1411–1423, 2015.
- [26] R. Frič and P. K. Eide, “Perioperative monitoring of pulsatile and static intracranial pressure in patients with chiari malformation type 1 undergoing foramen magnum decompression,” *Acta neurochirurgica*, vol. 158, no. 2, pp. 341–347, 2016.