Computational Fluid Dynamics Modelling of Cerebrospinal Fluid Flow In Patient Specific Geometries
Comparisons With in vivo Measurements

Jakob Emil Schreiner
Master’s Thesis, Spring 2016
Acknowledgements

I wish to thank my supervisors Kent-Andre Mardal and Erika Kristina Lindsrøm for invaluable advice, support and encouragement. Thanks to Victor Haughton for medical advice. I also wish to express my gratitude to Benjamin Kehlet for help with msh, Øyvind Evju for help with cbcflow and the Abel computer cluster, and Jostein Kolaas for his script for normalized cross correlation.
### Contents

1 Introduction ................................................. 7  
1.1 Introduction ............................................... 7  

2 Medical Background ........................................ 12  
2.1 Anatomy of the Human Brain and Spine .................... 12  
2.2 Cerebrospinal Fluid ....................................... 12  
2.2.1 Pathophysiology of CSF ................................. 14  
2.2.2 Effect of Elastic Walls ................................. 14  

3 Model Generation ........................................... 15  
3.1 Image Segmentation ....................................... 15  
3.1.1 Surface Generation Using VMTK ......................... 20  
3.2 Surface Preparation ...................................... 20  
3.2.1 Polygonal Surface Preparation ......................... 20  
3.3 mshr ....................................................... 21  
3.3.1 Including mshr in the VMTK pipeline ................. 21  
3.3.2 Taubin smoothing .................................... 22  
3.3.3 Plane Intersection ................................... 23  
3.3.4 Mesh Generation ...................................... 24  
3.3.5 Resolution Control ................................... 25  
3.4 Generating the Mesh ..................................... 25  
3.4.1 Surface Smoothing .................................... 26  
3.4.2 Surface Clipping ..................................... 27  
3.4.3 Surface Capping ..................................... 27  
3.4.4 Surface Remeshing .................................... 28  
3.4.5 The Final Mesh ....................................... 28  

4 Mathematical Methods ..................................... 30  
4.1 Navier-Stokes Equations .................................. 30  
4.2 The Incremental Pressure Correction Scheme ............ 31  
4.3 The Finite Element Method and the Weak Form of IPCS .... 32  
4.3.1 The Weak Formulation ................................ 33  
4.3.2 The Weak Form of IPCS ............................... 35  
4.4 The Finite Element ...................................... 36  
4.4.1 Discretization ....................................... 36  
4.5 Boundary Conditions .................................... 37
List of Figures 120
List of Tables 122
List of Listings 124
Bibliography 125
Chapter 1

Introduction

1.1 Introduction

Cerebrospinal fluid (CSF) is a clear bodily fluid found in the ventricular system and subarachnoid spaces of the brain. The CSF flows in a pulsating manner, balancing the change in intra cranial blood volume during the cardiac cycle with approximately no net flux during each cycle [25].

CSF dynamics are associated with various pathological conditions. Increased knowledge of the CSF flow dynamics will help better our understanding of many of these conditions. Different methods have been used to investigate this connection. Martin et al., 2012, created a coupled hydrodynamic model of the cardiovascular and cerebrospinal fluid system, investigating the relation of this coupling to disorders such as Chiari malformation, syringomyelia and tumours of the spinal cord and brain [24]. Martin & Loth, 2009, investigated the influence of coughing on CSF fluid pressure in an in vitro syringomyelia model [23].

Magnetic resonance imaging (MRI) and phase contrast MRI (PC-MRI) allows for in vivo measurements of CSF flow. Hofmann et al., 2000, analysed volumetric motion data of the spinal cord and CSF over an entire axial cross section in Chiari patients [13]. Bhadelia et al., 1995, determined the features of the CSF flow in Chiari patients before and after decompressive surgery [4]. Brugéries et al., 2000, determined the CSF flow patterns in spinal cord cysts as well as modifications of the CSF flow resulting from surgery [5]. Wolpert et al., 1994, assessed the motion of the medulla, cerebellar tonsils and the upper cervical cord, as well as CSF flow in normal subjects and Chiari patients using PC-MRI [32].

It is difficult to measure CSF flow because it surrounds the brain and spinal canal, which are sensitive to invasive measurements. This is especially true of the intra cranial pressure, which cannot directly be measured using MRI. One of the main advantages to using computers in the simulation of CSF flow is that it can be done causing minimal disturbance to the patient, as only MRI is required. Alperin et al., 2000, calculated the intra cranial pressure from PC-MRI measurements of CSF and blood flow [2]. Another
Figure 1.1: A sagittal view of the human brain.

option is to use computational fluid dynamics (CFD).

Computational geometries based on patient specific MRI scans coupled with CFD provides information of both velocity and pressure in three spatial dimensions and time. With the pressure and velocity computed at every grid point, the resolution offered by CFD is much higher than PC-MRI. Roldan et al., 2009, simulated CSF flow in a healthy volunteer and a Chiari patient based on high resolution MRI scans. Clarke et al. developed CFD models for a normal participant, two Chiari patients with and without syringomyelia. They also performed geometry and flow substitution to investigate the effect of either on the CSF flow [8]. Idealized CFD models has also been used. Linge et al., 2010, used an idealized model capturing the main characteristics of the posterior fossa and anatomy of the cervical spine [19]. Idealized geometries can be changed freely to investigate the impact of different anatomies.

This study will investigate the CSF flow field using CFD based on patient specific MRI
scans. Volumetric flow rates computed from four PC-MRI scans of each volunteer will be used as boundary conditions. Image segmentation of the MRI scans and mesh generation is accomplished with VMTK [17]. It is a collection of libraries and tools for 3D reconstruction, geometric analysis and mesh generation for blood vessels. The CSF is modelled by the incompressible Navier-Stokes equations for Newtonian fluids. They are discretized using the incremental pressure correction scheme (IPCS) and the finite element method (FEM) [20] and solved using the software collection FEniCS [20].

To assess the accuracy of the model, calculated mean and peak velocities are compared to the ones measured in the PC-MRI series. We find that measured velocities tend to be larger than the calculated ones. The difference seems to vary with the cross sectional area. Furthermore, the largest computed velocities are found in a different area of the cross section than the measured ones.

Chapter 3 will outline the steps from a MRI scan to a computational mesh. An outline of the mathematical framework is given in chapter 4, including the derivation of the incremental pressure correction scheme (IPCS) from Navier-Stokes equations. A brief introduction to FEM will be given and how it is used to discretize IPCS in space. The implementation is verified by comparison with known solutions of Navier-Stokes in chapter 5. Here, the effects of spatial and temporal discretization parameters are investigated, as well as the effects of the initial condition and different boundary conditions. The results of the simulations are presented in chapter 7. A discussion of the presented results is found in the final chapter.
Unites and Notation

In this thesis, cephalad velocities are defined as positive, while caudad velocities are negative.

<table>
<thead>
<tr>
<th>physical quantity</th>
<th>symbol</th>
<th>Unit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Velocity</td>
<td>$u$</td>
<td>$\frac{mm}{s}$</td>
<td>To be computed</td>
</tr>
<tr>
<td>Pressure</td>
<td>$p$</td>
<td>$Pa$</td>
<td>To Be computed</td>
</tr>
<tr>
<td>CSF Density</td>
<td>$\rho$</td>
<td>$\frac{g}{mm^3}$</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>CSF Dynamic Viscosity</td>
<td>$\mu$</td>
<td>$Pa \cdot s$</td>
<td>$0.7 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

Table 1.1: Units and values of physical parameters
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS</td>
<td>Sub Arachnoid Space</td>
</tr>
<tr>
<td>PC-MRI</td>
<td>Phase contrast MRI</td>
</tr>
<tr>
<td>CSF</td>
<td>Cerebrospinal Fluid</td>
</tr>
<tr>
<td>CFD</td>
<td>Computational Fluid Dynamics</td>
</tr>
<tr>
<td>FEM</td>
<td>Finite Element Method</td>
</tr>
<tr>
<td>IPCS</td>
<td>Incremental Pressure Correction Scheme</td>
</tr>
<tr>
<td>DICOM</td>
<td>Digital Imaging and Communications in Medicine</td>
</tr>
</tbody>
</table>

Table 1.2: List of abbreviations

<table>
<thead>
<tr>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caudad</td>
<td>Toward the tail</td>
</tr>
<tr>
<td>Cephalad</td>
<td>Toward the head</td>
</tr>
</tbody>
</table>
Chapter 2

Medical Background

2.1 Anatomy of the Human Brain and Spine

The brain is the central organ of the nervous system and is located in the head, protected by the skull. Most of the brain is divided into the two cerebral hemispheres, which are covered by the cerebral cortex. The cortex is folded in a way that greatly increases the surface area that can fit inside the skull. The Cerebellum is a region of the brain important in motor control. Anatomically, it is a separate structure in the posterior part of the brain, located underneath the two hemispheres, as seen in figure 1.1. The cerebellum is connected to the rest of the brain through the brain stem. For further information, see e.g [29].

The brain stem is in the posterior part of the brain. It is continuous with the spinal cord and connected to the sensory nerves of the face and the neck, as well as the motor and sensory connections to the rest of the body. The brain stem also plays an important role in regulating the cardiac cycle and respiration. It is a long thin tube of nervous tissue that extends from the brain stem to the lower parts of the spine. It passes through the spinal canal in the vertebral column in figure 2.1, formed by holes in the vertebrae, as seen in figure 2.2. Together with the brain, it makes up the central nervous system.

2.2 Cerebrospinal Fluid

Cerebrospinal fluid (CSF) is a colourless clear bodily fluid found in the ventricular system and subarachnoid spaces (SAS) of the brain, as well as the spinal SAS, as shown in figures 2.2 and 1.1. The SAS is found between the pia mater and the dura mater. The SAS is occupied by spongy tissue consisting of delicate tissue filaments and intercommunicating channels of CSF. In the upper parts of figure 1.1, the CSF in the folds of the cerebral cortex is visible. In certain parts, the interval between the pia and dura mater is greater, and the SAS forms cisterns. The subarachnoid tissue is less abundant in these cisterns. We see for instance the pontine cistern in the lower part of figure 1.1.
Figure 2.1: Vertebrae numbers 3, 4 and 5.

The CSF helps cushion the brain from mechanical impacts as well as transporting nutrients and waste products [21]. It moves in a pulsating manner with the primary driving mechanism behind the pulsation being the change in intra cranial blood volume during cardiac cycles. This change in blood volume is a result of the phase difference of blood flux to and from the brain. As the cranium is rigid, the CSF flows through the SAS and around the spinal cord, with approximately no net flux between systole and diastole [25].

As the flow oscillates with each cardiac cycle, it changes direction at slightly different times throughout the domain. Inertia and geometric features may also cause flow separation, which results in eddies and vortices. Synchronous bidirectional flow, i.e. both caudad and cephalad flow at the same time, can be observed [12], [10]. Quigley et al.,
2004, demonstrated synchronous bidirectional flow in the anterior part of the spinal canal [28].

Figure 2.2: The spinal canal with the spinal cord and nerve roots [27].

The cervical SAS and the spinal cord is depicted in figure 2.2. The subarachnoid cavity is a space around the spinal cord filled with CSF. Between the vertebrae, nerve roots penetrate the subarachnoid cavity. The spinal cord itself moves a small distance during the cardiac cycle [21], which, together with the nerve roots complicate the flow conditions.

2.2.1 Pathophysiology of CSF

A number of pathological conditions are associated with CSF flow. Theses include Chiari malformation, syringomyelia and hydrocephalus [25]. Elevated CSF pressures and result in increased forces acting on neural tissue, which can cause neurological issues [23] [10]. Increased knowledge of the CSF flow conditions will help better our understanding of many of these conditions.

2.2.2 Effect of Elastic Walls

In our simulations we will assume that the SAS walls are rigid. This is a simplifying assumption that will make the computations considerably easier. The main effect of using solid rather than elastic walls is higher velocities and steeper pressure gradients. Martin and Loth, 2009, found this in an invitro syringomyelia model of the spinal SAS. However, Cheng et al., 2014, found that the assumption of rigid spinal cord does not have a significant effect on the CSF pressures [6]. This was further confirmed by Clarke et al. 2013 [8].
Chapter 3

Model Generation

In this chapter the path from MRI scan to a computational mesh is outlined. Some of the difficulties, assumptions and solutions will be discussed. The outline of the chapter is as follows: First the segmentation process will be explained, then we will discuss how to extract a surface from the segmentation, and prepare it for use in CFD. Finally, the way to create a mesh from a surface is explained.

3.1 Image Segmentation

Patient specific anatomical models is the basis for the simulations. The starting point is a set of MRI scans of four healthy volunteers. These scans are of the whole head and down to the level of C4, see figure 2.1. The surface of the cranial and cervical subarachnoid spaces is extracted from the scans. The surface will then be used to generate a 3D model which we can use for simulations of CSF flow in the SAS.

VMTK [17] is used to process and create a surface representing the relevant parts of the SAS. It is a collection of libraries and tools for 3D reconstruction, geometric analysis and mesh generation for blood vessels. Starting off with a DICOM directory containing the MRI scans, we convert the DICOM images to VTK XML files, which VMTK can work with. This is done with the script

\[
\text{vmtkimagereader -ifile path_to_first_dicom_file.dcm} \ \ \text{--pipe vmtkimagewriter -ofile image_volume.vti}
\]

In addition, this script will rotate the image volume correctly based on its orientation relative to the patient.

The surfaces of the vascular segments are extracted from the MRI scans using image segmentation, where the extracted surface is represented by a level set function. The surface is then located at level zero of the level set function. This level corresponds to the peak gradient modulus of the image intensity within the selected volume. This is an objective criterion for the surface, as the greatest change in image intensity is across vessel walls. The VMTK script used in the segmentation process is:
The level set file level_1.vti is the output the first time we run this script. It is advisable to save level set segments in such files often, in case of any errors done in the segmentation process.

![Sagittal, Frontal, and Axial Planes](image.jpg)

(a) The sagittal plane  (b) The frontal plane  (c) The axial plane

Figure 3.1: Three cross sections of an MRI scan. This is the initial view in the VMTK renderer. Cross sections can be rotated and translated to display the underlying 3D anatomy. In this image, the CSF is found in the light areas of the images.

The script utilizes four different level set methods. Among these, the colliding fronts method has proved to be the most efficacious when segmenting the cervical SAS. It is summed up in two steps. First we determine a lower and optionally an upper bound on pixel intensities, then we place two seeds on the image. Two fronts will then propagate towards each other, one from each seed. The regions covered by both fronts will be segmented. This method is ideal for duct shaped vascular segments, because the propagation speed is proportional to the pixel intensity. By placing one seed at each end of the duct, it is unlikely that side branches with lower intensity will be covered by both fronts. This allows us much greater flexibility and control over which parts to segment.

Having decided on a region to segment, we use the colliding fronts method. The first step is to determine the lower and or upper bounds on the pixel intensity. The intensities are found moving the cursor over the image. The high intensity regions are the ones representing CSF, so the upper bound is left empty. Choosing the lower bound is more tricky. It should be chosen so that only the SAS is segmented, and nothing else. In figure 3.1, for example, the pixel values of the light areas are in the range of 500 to 1000. In the lower region of the image, the SAS might have pixel values of 200 to 300 and up. The pixel value of the black parts vary from 0 to 100 in the lower parts, and increases slowly moving cephalad. This example demonstrates the difficulties in the segmentation of such complex geometries as the SAS.
The volume is segmented placing two seeds and choosing the appropriate intensity threshold values.

The posterior part is segmented placing two more seeds. The segmented regions are then merged.

Figure 3.2: Illustration of the use of the colliding fronts method. Notice how large parts of the image quickly and objectively can be segmented.

The colliding fronts method is illustrated in figure 3.2. In this figure, the CSF has a lighter colour than the surrounding anatomy. In figure 3.2a, all of the light area directly between the two red seeds has been segmented, and some on the opposite side of the spinal canal. In figure 3.2b, two more seeds has been placed, and the rest of the light region has been segmented. The images shown here are two dimensional slices of a three dimensional image. The image has to be inspected further to verify that the segmentation is complete and that no errors has been performed. The lower part of the image has poorer resolution, and the difference in intensity between the SAS and everything else decreases. Figure 3.3 illustrates how we close a hole using the colliding fronts method. Sometimes the colliding fronts method fails. For instance, the geometry might be too complex, the resolution too poor. In these cases manual segmentation can be used. An example is shown in figure 3.4.

Manual segmentation involves manually placing seeds on the image. The regions around the seed with similar intensity will then be segmented. The use of this method should be kept to a minimum, because it lacks the objective criteria of the colliding fronts method and relies to a great extent on visual interpretation of the images, leading to poor reproducibility. Still, the error introduced when using this method should be negligible.
(a) A hole left after segmenting a large region with colliding fronts. It can be filled by placing the seeds much closer and picking the pixel threshold values more carefully.

(b) The hole is now filled using the colliding fronts method

Figure 3.3: Illustration of segmentation of smaller low resolution areas using the colliding fronts method.

After a sufficient number of deformation steps has been performed on the level set [18].

(a) A hole on the level set. The low pixel values in the hole excludes the possibility using the colliding fronts method.

(b) The hole filled with manually placed seeds. The region around the seed will be segmented. Any errors introduced should be minimized after deformation of the level set.

Figure 3.4: Illustration of the use of manual segmentation when colliding fronts fail or can not be used

After a sufficiently large volume is segmented, the level set is put through a series of deformation steps. There are four parameters controlling this procedure. i) Number of iterations is the number of deformation steps performed. ii) Propagation scaling is
the weight assigned to model inflation. iii) Curvature scaling is the weight assigned to surface regularization. iv) Advection scaling regulates the attraction of the level set to image gradient modulus ridges.

The VMTK tutorial recommends setting propagation and curvature scaling to zero and advection scaling to 1. This should lead to more reproducible results. If the segmented region is not too big, we choose 300 iterations. Fig 3.5 show the result before and after deformation with the parameters suggested above.

![Segmentation](image1.png)  ![Segmentation after deformation](image2.png)

(a) A segmentation of the upper cervical SAS before any deformation steps. (b) The same segmentation after deformation is performed. Note that the surface is much smoother.

Figure 3.5: An illustration of the effect of deformation of the level set. After the deformation, the level set should conform to the maximum change in modulus of pixel gradients.

The SAS is a very complex anatomical domain. There are numerous blood vessels and nerve roots penetrating the SAS, illustrated in figure 2.2. The nerve roots is ignored in the segmentation, and the smaller of these blood vessels are so minor that their effect on the overall flow is assumed to be negligible. To segment these regions, threshold values has to be set much lower than the surrounding SAS, meaning that there no longer is a clearly defined border marking the vascular wall. Manual segmentation is used to achieve this.
3.1.1 Surface Generation Using VMTK

When the patient anatomy has been segmented and a sufficient number of deformation steps has been performed, the surface can be extracted using the script

```
 vmtk marches cubes -ifile level_n.vti \
   -ofile surfacemodel.vtp
```

The surface corresponds to the level zero of the level set file.

3.2 Surface Preparation

Image segmentation can result in surfaces with geometric features not found in the real anatomy. This is because of the resolution of the MRI scans. The size of a voxel in the MRI scans considered here is about $0.5\, \text{mm} \times 0.5\, \text{mm} \times 1\, \text{mm}$. It varies from series to series. As a result of the resolution, the surface extracted using the VMTK script will have high frequent features not found in real life. After the surface has been smoothed it will be remeshed. This involves resizing the surface triangles to make them as even as possible.

3.2.1 Polygonal Surface Preparation

The VMTK scripts used in this chapter operates on polygonal surfaces, where the polygons are triangles. A VTK polygonal surface is a surface consisting of interconnected triangles. The data structure is represented by two arrays. One array with all the coordinates of each vertex. Each vertex appears only once. The other array is a connectivity array. It represents a triangle using four integers. The first integers defines the number of vertices in the polygon, three in our case. The other three integers are the indices of the vertices making up the triangle. The indices refer to the array of vertices.

Mesh generation will sometimes fail because there is something wrong with the polygonal surface. There are a number of potential problems: i) Holes in the surface ii) Edges with used by more than two polygons iii) Polygons with zero area iv) Self intersections v) Duplicate points or edges. Manual inspection might reveal the error, but is impractical on a surface consisting of hundreds of thousands of triangles. The VTK library, on which VMTK is built, offers several functions for investigating these problems.

There is a function for computing how many triangles an edge belongs to. This function can also be used to find holes, as an edge around a hole belongs to only one triangle. Edges are not necessary for representing the surface, but can be derived from the connectivity array. There is also a function which automatically triangulates and fills holes. In order to find duplicate points or degenerate cells, i.e. triangles with zero area, one can iterate over the hole point set. Degenerate cells can be deleted, and a function used to remove duplicate points. These procedures were used in the process of mesh generation. An example of a polygonal surface file is shown in 3.3.2.
3.3 mshr

Originally, we hoped to use a much larger part of the SAS in our simulations. VMTK has support for mesh generation of annular geometries, but in order to include blood vessels we had to use mshr [14]. mshr is the mesh generation backend of FEniCS. It generates simplicial 2D or 3D meshes from surfaces described by by Constructive Solid Geometry (CSG) or form surface files. It utilizes CGAL and TETGEN as mesh generation backends. mshr can create meshes from much more diverse surfaces than VMTK. The goal was to implement methods for surface smoothing and adding geometric flow extensions in mshr, but only the smoothing method ended up being useful.

3.3.1 Including mshr in the VMTK pipeline

In order to include mshr in the VMTK pipeline, I wrote a function converting surfaces represented by halfedges as polygonal surfaces, as described in section 3.2.1. Figure 3.6 show a polygonal triangle represented by three halfedges. The three halfedges together define a facet. Algorithm 1 describes how to create the two arrays with points and connectivity described in section 3.2.1. The algorithm takes two arrays as input, one with facets and with halfedges. Together they represent a polynomial surface in CGAL. The mshr scripts listed in this thesis relies on this method.

There are two steps performed by mshr. First, the surface was reconstructed from the point set. This will hopefully remove any problems with the polygonal connectivity. Then the surface is smoothed using an implementation of Taubin’s non shrinking smoothing algorithm proposed in [30]. This step is described further in 3.3.2.

Algorithm 1 Algorithm for converting halfedges to polydata

```plaintext
for all vertices in surface do
    vertexMap[vertex] = vertexCounter++
    points.InsertNextPoint(vertex)
end for

for all facets in surface do
    halfedge ← facet.getFirstHalfedge()
    poly.SetPoint(3) ← Set number of points in the polygon
    poly.setPoint(vertexMap[halfedge.vertex()]) ← Add the index of the first point
    poly.setPoint(vertexMap[halfedge.next().vertex()])
    poly.setPoint(vertexMap[halfedge.next().next().vertex()])
    polygons.InsertNextCell(polygon)
end for

Write points and polygons to a file
```
3.3.2 Taubin smoothing

One method for smoothing a surface is moving each point by a vector average of the distance to each of its neighbours. This is known as Gaussian smoothing, and can be described as follows. For each vertex $v_i$ compute a vector average.

$$
\Delta v_i = \sum_{j=0}^{N} v_j - v_i
$$

Where $N$ is the number of adjacent vertices to $v_i$. Then a new position for $v_i$ is computed by $v'_i = v_i + \lambda \Delta v_i$. Where $\lambda$ is a suitable scaling factor. This process is the carried out iteratively, but will produce shrinkage. Applied enough times, the surface will eventually shrink to a point. The result of successive Gaussian smoothing steps is shown in 3.9

Taubin smoothing [30] is a modification to this method. The basic idea is to use two scaling factors, $\lambda$ and $\kappa$, where $0 < \lambda < -\kappa$. He showed that this would produce a low pass filter effect where surface curvature would take the place of frequency. In the case of Gaussian smoothing, the filter transfer function is $f(k) = (1 - \lambda k)(1 - \kappa k)$. Because this is a low-pass filter, there will be a region $[\alpha, k_p]$ in which $f(k)^N \approx 1$. The value of
\( k_{pb} \) is

\[
k_{pb} = \frac{1}{\lambda} + \frac{1}{\kappa}.
\]

(3.1)

Taubin suggests in his paper [30] that choosing values for \( k_{pb} \) in the range 0.1 to 0.01 produce good results, and further that \( f(1) = -f(2) \). Combining this with equation 3.1 and choosing \( k_{pb} \), we find both \( \lambda \) and \( \kappa \).

\[
(1 - \lambda) \left( 1 - \frac{1}{k_{pb} - \frac{1}{\lambda}} \right) (1 - 2\lambda) \left( 1 - \frac{2}{k_{pb} - \frac{1}{\lambda}} \right) = 0
\]

(3.2)

Choosing \( k_{pb} = 0.1 \) as suggested by Taubin, we can solve equations 3.2 and (3.1) to and get

\[
\lambda = 0.63 \\
\kappa = -0.67
\]

The volume changes should be monitored for each non shrinking smoothing step.

### 3.3.3 Plane Intersection

As mentioned in section 3.3, we worked on implementing a method for adding geometric extensions to a geometry. Given a point inside the geometry and a normal vector. Then let this point and normal vector define a plane. The goal is to find a new plane whose normal vector is as parallel as possible to the tangents of the surface of the geometry where it is intersected by the plane. This is illustrated in figure 3.7. The black line is the initial plane, and the red has been adjusted. Once the appropriate intersection has been found, the polygon in the intersection can be extruded in 3D. The implementation of the method for finding the polygon in the intersection between the geometry and the plane is still in progress.

![Figure 3.7: An illustration of algorithm 2](image)

The algorithm for adjusting the plane is described in algorithm 2. Given the triangles in the intersection between the plane and the geometry, project the normal vector of the plane into each of these triangles, and compute the average of all the projections. Let each triangle in the cut plane be defined by its two edges, \( t_1, t_2 \), and the intersecting
plane be defined by a point and its normal vector $n$. The idea is to get a new normal vector by computing the projection of $n$ into $\text{Span}\{t_1, t_2\}$. A new intersecting plane is then defined by the same point and a new normal vector which is the average of all the projections.

**Algorithm 2** Algorithm for adjusting the intersecting plane

```plaintext
new_n ← 0
for all t1, t2 in Triangles do
  proj1 ← inner_product(n, t1)/inner_product(t1, t1) × t1
  proj2 ← inner_product(n, t2)/inner_product(t2, t2) × t2
  new_n ← proj1 + proj2
end for
new_n ← new_n / number_of_triangles
if new_n is parallel to n then
  return
end if
Normalize new_n
n ← $\alpha$ (new_n - n)
Normalize n
return n
```

### 3.3.4 Mesh Generation

We chose to use VMTK rather than mshr when creating the meshes. There are two reasons for this. First, there is a VMTK script for remeshing the surface, and second, VMTK offers resolution control. As computer memory is a limited resource, user control over which parts of the mesh that should have higher resolution is important. If, for instance, the thinner parts of the mesh are only one or two cells across, the velocity would be near zero due to no-slip boundary conditions. Attempts to use mshr to create a mesh after remeshing the surface in VMTK failed. The problem was not identified. The VTK functions discussed in section 3.2.1 on page 20 did not provide a solution.
3.3.5 Resolution Control

The Courant number puts a restriction on how large the time step can be in a simulation. Looking at it as an Heuristic, it describes the relationship between velocities, grid spacing and time stepping. If a signal is moving across a discrete spatial grid and we want to simulate this at discrete time steps, then this time step must be less than the time it takes the signal to travel from grid point to grid point. In the nD case, the Courant number takes to form

\[ C = \sum_{i=1}^{n} \frac{u_i \Delta t}{\Delta x_i} \leq C_{\text{max}} \]

where \( u_i \) is the velocity in the \( i \)th direction, and \( \Delta x_i \) is the \( i \)th spatial variable. Given a mesh, using an a priori estimate of the fluid velocities, the Courant number can be used to estimate the required time step. After a simulation, the mesh can be re-evaluated and adjustments to the time step if necessary.

Pulsating flows, like CSF flow, form a boundary layer, when the flow reverses. In order to capture this, a boundary layer should be added to the mesh in order to capture the rapid variation in the flow near the wall. VMTK is able to add boundary layers in the mesh generation algorithm, but I was unable to either create a mesh using boundary layers, or get FEniCS to recognize the boundaries of the mesh when using boundary layers.

Another issue related to mesh generation is backflow divergence. Numerical backflow divergence can occur as a result of partial or bulk flow reversal over and outlet. Divergence due to flow separation or flow recirculation is often caused by complex geometries or increased cross sectional area near the outlet. These numerical instabilities emanate from the use of Neumann boundary conditions on the outlet face, where velocity profile information is not specified [26].

One often used method for avoiding backflow divergence is to add flow extensions to the mesh. This will allow any vortices to dissipate before they reach the outlet. As discussed in 3.3.3 on page 23, an attempt to include functionality for geometric extensions in mshr is as yet incomplete. At the very least, the mesh should be so that no spurious artificial flow features appear near the Neumann boundary. In the end, this problem was avoided by using Dirichlet conditions, but with the cost of an artificial velocity profile. The effect of these two boundary conditions is investigated in section 4.5.

3.4 Generating the Mesh

The first step is to use mshr to reconstruct the surface from the points described in section 3.3 on page 21. The next step is to smooth the surface using the method described in section 3.3.2 on page 22. We then clip the surface, leaving only the surface relevant anatomy. Before remeshing the surface, we append end caps to the surface, making it a closed volume. Surface remeshing allows us to control the resolution of the mesh. Finally, the surface mesh is populated with tetrahedrons creating a volume mesh.
3.4.1 Surface Smoothing

Listing 1: Python script for Taubin smoothing using mshr

```python
import mshr as m

def smooth(ifname, ofname, iterations=5, lam=0.63, kappa=-0.67):
    s = m.Surface3D(ifname)
    d = m.CSGCGALDomain3D(s)

    print "Initial volume", d.volume(), "\n"
    for i in range(iterations):
        d.smooth_laplacian(lam)
        d.smooth_laplacian(kappa)

        print i, d.volume(), "\n"

    d.save(ofname)
```

The parameters $\lambda$ and $\mu$ refer to the scaling factors $\lambda$ and $\mu$ described in section 3.3.2 on page 22. The images in figure 3.9 are the original surface, the smoothed surface and a surface smoothed using Gaussian smoothing. Figure 3.9b, which has been smoothed using Taubin smoothing, has kept more of its features than figure 3.9c, which has been smoothed using only Gaussian smoothing.

![Figure 3.9](image)

(a) Area: 22039 mm$^3$  
(b) Area: 22101 mm$^3$  
(c) Area: 20498 mm$^3$

Figure 3.9: Illustrations of a surface before smoothing, a smoothed surface, and a surface smoothed using Gaussian smoothing. Both surfaces were smoothed using 30 smoothing iterations.
3.4.2 Surface Clipping

When we have a smooth surface file, we use the VMTK surface clipping script to extract the volume we want to mesh.

\[ \text{vmtksurfaceclipper -ifile smooth_surface.vtp} \]
\[ \text{-ofile clipped_surface.vtp} \]

Figure 3.10a illustrates the use of this script. In the renderer, pressing "i", a box opens. The parts of the surface within the box will be clipped. The box can be rotated by dragging it around, and translated by holding down the "shift" key. The box can be resized by moving the balls. Pressing "space" will clip the surface.

(a) The VMTK clipping tool. (b) The resulting smaller surface

Figure 3.10: Illustration of surface clipping in VMTK.

3.4.3 Surface Capping

The clipped surface is open, and has to be closed before meshing. This is done using the VMTK capping script

\[ \text{vmtksurfacecapper -ifile clipped_surface.vtp} \]
\[ \text{-cappingmethod annular -entityidsarray caps} \]
\[ \text{-ofile capped-surface.vtp} \]

The keyword \textit{cappingmethod annular} specifies the use of an annulus as a cap. Using the option \textit{entityidsarray} allows us to reference the polygons in the caps at a later stage in the mesh generation.
3.4.4 Surface Remeshing

The default edge length in VMTK is 1 mm. With the narrowest parts of the cervical SAS is less than 2 mm, some of the narrowest regions might end up with only one or two cells across. Because of the no slip condition on the wall. The resolution on the mesh is controlled by the script

```
vmtkdistancetospheres -ifile capped-surface.vtp \ 
  -ofile surface-dist.vtp
```

As illustrated in figure 3.11a on the next page, spheres are placed where the cursor is pointing by pressing 'space'. By pressing 'd', the surface is coloured by the distance to the spheres. This script creates a distance array. The distances are relative to the spheres depicted in 3.11a on the facing page. Finally, by pressing 'a', a scaling function can be specified by four parameters.

An offset added to the distances, a scale applied to the distances, and a minimum and maximum value for the distances. The distance array can be referenced later during the remeshing to control the edge lengths.

After the distance array is created, the surface can be remeshed with the script

```
vmtksurfaceremeshing -ifile surface-dist.vtp \ 
  -elementsizemode edgelengtharray -edgelengtharray \ 
  DistanceToSpheres -ofile surface-remeshed.vtp \ 
  -entityidsarray caps
```

The option elementsizemode specifies that the edge lengths are to be determined from an array, edgelengtharray DistanceToSpheres tells the script to use the distance array created by the script vmtkdistancetospheres. The entityidsarray is passed to preserve the caps during remeshing. The result is shown in figure 3.11b.

3.4.5 The Final Mesh

After the remeshing, the surface can be populated with tetrahedrons to create a volume mesh. The size of the tetrahedrons is controlled by a proportionality constant volumeelementfactor. The following script generates the mesh:

```
vmtkmeshgenerator -ifile surface-remeshed.vtp -skipremeshing 1 \ 
  -entityidsarray caps \ 
  -ofile surface-mesh.vtu -volumeelementfactor 0.6
```

The file extension '.vtu' signifies a vtkUnstructuredGrid file, and can be opened in Paraview [1]. The mesh can be converted to the 'xml' format for simulations in FEniCS.

```
vmtkmeshwriter -ifile surface-mesh.vtu -ofile surface-mesh.xml
```
(a) The colored distance graph in vmtkdistance spherospheres.  
(b) The resulting smaller surface

Figure 3.11: Illustration on mesh resolution control in VMTK.
Chapter 4
Mathematical Methods

4.1 Navier-Stokes Equations

The incompressible Navier-Stokes equations describe the flow of incompressible viscous fluids, including CSF [12]. They can be derived from the principles of conservation of mass and momentum. Navier-Stokes equations give us a relation between the velocity and pressure of an incompressible fluid, and can be written as:

\[
\frac{\partial u}{\partial t} + u \cdot \nabla u = -\frac{1}{\rho} \nabla p + \nu \nabla^2 u + g, \quad \nabla \cdot u = 0,
\]

in \( \Omega, t > 0 \)

\( \nabla \cdot u = 0 \), in \( \Omega, t > 0 \) \hspace{1cm} (4.1)

With the associated boundary conditions

\[
\begin{align*}
&u = g_D, \quad \text{on } \partial \Omega_D, \; t > 0 \\
&\nu \frac{\partial u}{\partial n} - p n = g_N, \quad \text{on } \partial \Omega_N, \; t > 0 \\
&u = u_0, \quad \text{in } \Omega, \; t = 0
\end{align*}
\]

Here \( \partial \Omega_D \cup \partial \Omega_N \) is the boundary of the flow domain \( \Omega \). \( \nu = \frac{\mu}{\rho} \) where \( \mu \) is the dynamic viscosity and \( \rho \) is the density. \( f \) represents the external forces. Gravity will be neglected since it will be contained in the boundary conditions.

Navier stokes equations (4.1) are a set of non linear partial differential equations. No general analytical method exist. There exists, however, a number of analytical solutions for particular cases and specific geometries [31]. Some of these solutions will be used to verify the that the solver has been implemented correctly.

Many flows in nature and industry are turbulent. The transition from laminar to turbulent flow is governed by the Reynolds number, \( Re = \frac{UL}{\nu} \), where \( U \) is a characteristic velocity, \( L \) a characteristic length scale for the geometry. Navier stokes equations describe both laminar and turbulent flow, but the spatial and temporal resolution required to describe the smallest scales makes it very computationally demanding to solve [16]. In this study, we will assume laminar flow, which is a reasonable assumption [12].
4.2 The Incremental Pressure Correction Scheme

Operator splitting schemes such as increment pressure correction scheme (IPCS) are often used strategies for solving Navier-Stokes equations [9] [3]. The system is split into a series of simpler, well studied equations, such as convection diffusion equations and Poisson equations. Efficient numerical methods are more easily constructed for these well known equations than for the original system 4.1 [16]. The Incremental Pressure Correction Scheme implemented in cbcflow will be used in this study. It can be described in three steps.

1. Replace the unknown pressure with a known approximation, and find a tentative velocity $\tilde{u}^{n+1}$.
2. Solve a Poisson equation for the pressure $p^{n+1}$.
3. Use the updated pressure to find a pressure correction and the velocity $u^{n+1}$.

The first step is to discretize (4.1) in time. The time derivative is handled using Backward Euler, the diffusion is discretized using Crank-Nicholson and the convection is handled explicitly. This gives us the following discretization of Navier-Stokes equations:

$$\frac{1}{\Delta t} \left( u^{n+1} - u^n \right) - \nabla \cdot \nu \nabla u^{n+\frac{1}{2}} + u^n \cdot \nabla u^n + \frac{1}{\rho} \nabla p^{n+1} = f^{n+1},$$

$$\nabla \cdot u^{n+1} = 0.$$  (4.2)

where $u^{n+\frac{1}{2}} = \frac{1}{2} (u^{n+1} - u^n)$.

The problem with (4.2) is that there is no natural update for the pressure. Following the idea of the operator splitting scheme, we use a known approximation for the pressure to get an equation for a tentative velocity $\tilde{u}^{n+1}$.

$$\frac{1}{\Delta t} \left( \tilde{u}^{n+1} - u^n \right) - \nabla \cdot \nu \nabla u^{n+\frac{1}{2}} + u^n \cdot \nabla u^n + \frac{1}{\rho} \nabla p^n = f^{n+1}$$  (4.3)

The tentative velocity is not divergence free, so we define a velocity correction $u^c = u^{n+1} - \tilde{u}^{n+1}$. Subtracting (4.3) from (4.2) we get

$$\frac{1}{\Delta t} u^c - \frac{1}{2} \nabla \cdot \nu \nabla u^c + \frac{1}{\rho} \nabla (p^{n+1} - p^n) = 0,$$

$$\nabla \cdot u^c = -\nabla \cdot \tilde{u}^{n+1}.$$  (4.4)

The operator splitting is $O(\Delta t)$, which is the same as the time-stepping, so without reducing the order of accuracy. Equation 4.4 is then simplified to

$$\frac{1}{\Delta t} u^c + \frac{1}{\rho} \nabla (p^{n+1} - p^n) = 0,$$

$$\nabla \cdot u^c = -\nabla \cdot \tilde{u}^{n+1}.$$  (4.5)

Taking the divergence of (4.5) and using the incompressibility constraint, it is reducible to a Poisson problem:

$$\Delta p^{n+1} = \Delta p^n - \frac{1}{\Delta t} \nabla \cdot \tilde{u}^{n+1}$$  (4.6)
Knowing the pressure correction, we can advance the velocity in time simply using
\[ u^{n+1} = \tilde{u}^{n+1} - \frac{\Delta t}{\rho} \nabla (p^{n+1} - p^n) \]

The accuracy of the scheme outlined above is \( O(\Delta t) \). This can be improved upon with a better estimation of the tentative velocity. This motivates us to make a few changes to (4.3). We allow for semi-implicit handling of both the convection and diffusion terms by adding a parameter \( \theta \) to the discretization of those terms. In addition a projection for the convecting velocity is used, rather than \( u^n \). The new equation for the tentative velocity then becomes
\[ \frac{1}{\Delta t} (\tilde{u}^{n+1} - u^n) - \nabla \cdot \nu \nabla \tilde{u}^{n+\theta} + u^* \cdot \nabla \tilde{u}^{n+\theta} + \frac{1}{\rho} \nabla p^n = f^{n+1}, \]

where
\[ \tilde{u}^{n+\theta} = \theta \tilde{u}^{n+1} + (1 - \theta) u^n \]
\[ u^* = \frac{3}{2} u^n - \frac{1}{2} u^{n-1} \]

With \( \theta = 0.5 \), this scheme is second order in both time and space [3].

### 4.3 The Finite Element Method and the Weak Form of IPCS

The rest of this section builds on theory found in e.g. the FEniCS book [20]. The Finite Element Method (FEM) is a numerical method for finding approximate solutions to partial differential equations (PDEs). It divides the problem domain into simpler parts, called finite elements, which makes it well suited to handle geometrically complicated domains, such as the SAS. It will be used for the spatial discretization in this thesis. The theory of finite elements is substantial, and is beyond the scope of this thesis, but a short introduction, exemplified by the Poisson equation will be given below.

Introducing the Finite Element Method (FEM), we will consider the pressure correction equation from IPCS, equation (4.6). For now, we will consider it with both Neumann and Dirichlet boundary conditions. The problem then reads Find \( p \in V(\Omega) \) and \( p \in Q(\Omega) \) such that

\[ \Delta p^{n+1} = \Delta p^n - \frac{\rho}{\Delta t} \nabla \cdot \tilde{u}^{n+1}, \text{ in } \Omega, \]
\[ p = g, \text{ on } \partial \Omega_D, \]
\[ \frac{\partial p}{\partial n} = g, \text{ on } \partial \Omega_N \]

Where \( \Omega \in \mathbb{R}^d \) is the domain, and \( \partial \Omega_D \) is the domain boundary with a Dirichlet condition and \( \partial \Omega_N \) is the domain boundary with a Neumann condition.
4.3.1 The Weak Formulation

Equation 4.7 on the preceding page is often referred to as the strong formulation of the problem. Enforcing the boundary conditions is difficult, especially on complex domain. Furthermore, the double derivative itself is problematic. Instead of working directly with the strong form, we will formulate a weak variational form of the problem.

To obtain the weak form, multiply (4.7) with a test function \( q \in Q \) and integrate over \( \Omega \). Without loss of generality, let \( p \in Q \).

\[
\int_{\Omega} \Delta p^{n+1} q \, dx = -\int_{\Omega} q \nabla \cdot \hat{u}^{n+1} \, dx - \int_{\Omega} \frac{q}{\Delta t} \nabla \cdot \hat{u}^{n+1} \, dx, \quad \forall \, p, q \in Q \tag{4.8}
\]

Assuming the functions \( p \) and \( q \) are sufficiently smooth, integration by parts gives the weak form of the problem

\[
\int_{\Omega} \nabla p^{n+1} \cdot \nabla q \, dx = \int_{\partial \Omega} q \nabla \left( p^{n+1} - p^n \right) \cdot n \, ds \\
+ \int_{\Omega} \nabla p^n \cdot \nabla q \, dx \\
+ \int_{\Omega} \frac{q}{\Delta t} \nabla \cdot \hat{u}^{n+1} \, dx
\]

where \( n \) is the normal vector on the surface of the domain, and the dot indicates a scalar product. The test function \( q \) is required to vanish on the parts of the boundary where \( p^{n+1} \) is known, i.e. \( \partial \Omega_D \).

Now let \( L^2(\Omega) = \{ p : \Omega \to \mathbb{R} | \int_{\Omega} p^2 \, dx < \infty \} \), and let \( H^1_0 \) be a subspace of function \( v \) in \( L^2 \) which possess first order weak derivatives and where \( p|_{\partial \Omega_D} = 0 \). A weak derivative is a function \( p \in L^2(\Omega) \) with derivatives \( \frac{\partial p}{\partial x_i} \in L^2(\Omega) \) satisfying

\[
\int_{\Omega} \frac{\partial p}{\partial x_i} \varphi \, dx = -\int_{\Omega} p \frac{\partial \varphi}{\partial x_i} \, dx, \quad \forall \varphi \in C^\infty
\]

If we equip this space with the inner product

\[
(p, q)_{1, \Omega} = \int_{\Omega} pq \, dx + \int_{\Omega} \nabla p \cdot \nabla q \, dx,
\]

and the norm

\[
||v||_{1, \Omega} = \left( \int_{\Omega} |p|^2 + |\nabla p|^2 \, dx \right)^{\frac{1}{2}}
\]

Then \( H^1_0(\Omega) \) is the Sobolev space \( W^{1,2}(\Omega) \). In the following we will omit the \( \Omega \) subscript.

Now let \( p \in Q = H^1_0 \). Then 4.8 can be written as follows: Find \( p \in Q \) such that

\[
a(p, q) = f(q), \quad \forall q \in Q \tag{4.9}
\]
where
\[ a(p, q) = \int_{\Omega} \nabla p^{n+1} \cdot \nabla q \, dx \]
\[ f(q) = \int_{\partial \Omega} q \nabla p^{n+1} - p^n \cdot n \, ds + \int_{\Omega} \nabla p^n \cdot \nabla q \, dx + \int_{\Omega} \frac{\partial}{\partial t} \nabla \cdot \tilde{u}^{n+1} \, dx \]

We would like our problem to be well-posed, meaning that a solution exists, the solution is unique and the solution’s behaviour changes continuously with the initial conditions. Equation (4.9) is well posed if it satisfies the Lax-Milgram theorem. For equation (4.9), which is a Poisson equation, the theorem states that, if \( V(\Omega) \) is a Hilbert space and, \( a(\cdot, \cdot) \) is a symmetric bilinear form which is

(i) Bounded: \[ |a(u, v)| \leq C_1 ||u||_1 ||v||_1, \forall u, v \in V \]

(ii) Coercive: \[ a(u, u) \geq C_2 ||u||_1^2 \forall u \in V \]

For some constants \( C \) and \( D \). Then for any bounded linear functional \( f : V \rightarrow \mathbb{R} \) and all \( v \in V \), the equation
\[ a(u, v) = f(v) \]
has a unique solution.

In the following, assume \( u, v \in H^1_0 \). To show (i):
\[ ||u||_1 ||v||_1 = \left( \int_{\Omega} u^2 \, dx + \int_{\Omega} (\nabla u)^2 \, dx \right) \left( \int_{\Omega} v^2 \, dx + \int_{\Omega} (\nabla v)^2 \, dx \right) \]
\[ \geq \int_{\Omega} (\nabla v)^2 \, dx \int_{\Omega} (\nabla v)^2 \, dx \]
\[ \geq | \int_{\Omega} \nabla u \cdot \nabla v \, dx|^2 \]
\[ = |a(u, v)|^2 \]
which implies that
\[ a(u, v) \leq C_1 ||u||_1 ||v||_1, \forall u, v \in H^1_0 \]

In order to show (ii) we need the Poincaré’s inequality: Let \( v \in H^1_0(\Omega) \). Then
\[ ||v||_{L^2(\Omega)} \leq C_0 ||v||_{L^2(\Omega)} \]
where \( C_0 \) depends only on \( \Omega \). Note that
\[ ||u||_1^2 = \int_{\Omega} |u|^2 + |\nabla u|^2 \, dx \leq (1 + C_0) \int_{\Omega} |\nabla u|^2 \, dx \]
which implies
\[ a(u, u) = \int_{\Omega} |\nabla u|^2 \, dx \geq \frac{1}{1 + C_0} ||u||_1^2 \]
which shows the coercivity of \( a \), letting \( D = \frac{1}{1 + C_0} \). Because \( a \) is both bounded and coercive, Lax-Milgram guarantees us that (4.9) has a unique solution given a \( f \in V' \). Of the strong formulation (4.7) has a solution, that solution will correspond with the solution of the weak formulation (4.8).
4.3.2 The Weak Form of IPCS

We discretize IPCS following the ideas of the previous section. Below are the weak form of the equation for the tentative velocity (4.3), the pressure correction equation (4.6) and the velocity correction (4.5). First the weak form of the equation for the tentative velocity $\tilde{u}^{n+1}$. Given two test functions, $v \in V$ and $q \in Q$,

$$\int_{\Omega} \tilde{u}^{n+1} v \, dx = \int_{\Omega} u^n v \, dx$$

$$- \Delta t \int_{\Omega} \nu \nabla \tilde{u}^{n+\theta} : \nabla v \, dx$$

$$+ \Delta t \int_{\partial\Omega} \nu \nabla \tilde{u}^{n+\theta} \cdot \mathbf{n} v \, ds$$

$$- \Delta t \int_{\Omega} u^* \cdot \nabla \tilde{u}^{n+\theta} v \, dx$$

$$+ \Delta t \int_{\Omega} f^{n+1} v \, dx$$

$$- \int_{\Omega} \frac{\Delta t}{\rho} \nabla v \cdot p^n v \, dx$$

$$- \int_{\partial\Omega} \frac{\Delta t}{\rho} p^n v \cdot \mathbf{n} \, ds.$$

Equation 4.10 holds for all $v \in V$ and $q \in Q$. And then the pressure correction equation:

$$\int_{\Omega} \nabla p^{n+1} \cdot \nabla q \, dx = \int_{\partial\Omega} q \nabla \left( p^{n+1} - p^n \right) \cdot \mathbf{n} \, ds$$

$$+ \int_{\Omega} \nabla p^n \nabla q \, dx$$

$$+ \int_{\Omega} \frac{\rho}{\Delta t} \nabla \cdot \tilde{u}^{n+1} q \, dx, \quad \forall q \in Q$$

And finally the velocity correction equation:

$$\int_{\Omega} u^{n+1} v \, dx = \int_{\Omega} \tilde{u}^{n+1} v \, dx - \int_{\partial\Omega} \frac{\Delta t}{\rho} \nabla \left( p^{n+1} - p \right) v \, ds, \quad \forall v \in V \quad (4.11)$$

Figure 4.1: A picture of 1D linear Lagrange finite elements, from [15]
4.4 The Finite Element

We will follow Ciarlet’s definition of a finite element [7].

**Definition 4.4.1.** Let

1. The domain $T$ is a bounded, closed subset of $\mathbb{R}^d$ with nonempty interior and piecewise smooth boundary.

2. The space $\mathcal{V} = \mathcal{V}(T)$ is a finite dimensional function space on $T$ of dimension $n$.

3. The set of degrees of freedom (nodes) $L = l_1, l_2, \ldots, l_n$ is a basis for the dual space $\mathcal{V}'$; that is, the space of bounded linear functionals on $\mathcal{V}$.

Then $(K, \mathcal{V}, L)$ is called a finite element.

**Definition 4.4.2.** Let $(K, \mathcal{V}, L)$ be a finite element. The basis $\{\varphi_1, \varphi_2, \ldots, \varphi_k\}$ of $\mathcal{V}$ dual to $L$ (i.e. $l_i(\varphi_j) = \delta_{ij}$) is called the nodal basis of $\mathcal{V}$.

The domain is discretized $\Omega$ using $N$ non overlapping intervals in 1D, triangles in 2D and tetrahedrons in 3D, such that $\bigcup_{i=0}^N K_i = \Omega$, as seen in figure 3.11b on page 29. There are many kinds of elements. Among the most common are Lagrange or continuous Galerkin (CG) elements. We will denote the CG element of polynomial order $q$ by $P^q$. We discretize equation (4.8) by inserting the ansatz $u_h = \sum U_i \varphi_i$. Then let the test function be an arbitrary basis function, $v = \varphi_j$. Now the problem (4.8) can be rewritten Find $u_h \in v_h$ such that

$$a(u_h, \varphi_j) = f(\varphi_j), \quad \text{for } j = 0, 1, \ldots, N$$

One of the advantages with FEM is the partition of $\Omega$. The division of the computation domain into non-overlapping simplices allows us to approximate complex domains. By then approximating the solution on each simplex, we have a very general framework for solving equation on complex domains. Another advantage is the inclusion of Neumann type boundary conditions in the weak form, making them easy to implement.

### 4.4.1 Discretization

We can discretize the weak form of the pressure correction equation ((4.8)) using homogeneous Dirichlet conditions by introducing the discrete function space $Q_h$. The discretized problem then reads: Find $p_h \in Q_h$ such that

$$\int_{\Omega} \nabla p_h^{n+1} \cdot \nabla q_h \, dx = \int_{\Omega} f v_h \, dx, \quad \forall q_h \in Q_h$$

Where $q_h$ is the space spanned by our basis, and chosen to be 0 on $\partial \Omega$. We then write $p_h$ as a linear combinations of the basis functions, $p_h = \sum_{j=1}^N c_j \phi_j$. Without loss of generality, we let $q_h = \phi_i$. Now

$$\int_{\Omega} (\nabla \sum_{j=1}^N c_j \phi_j) \cdot \nabla \phi_i \, dx = \int_{\Omega} f \phi_i \, dx, \quad \text{for } i = 1, 2, \ldots, N$$
This is equivalent to the linear system

\[ Ap^{n+1} = b \]  \hspace{1cm} (4.12)

where

\[ A = \int_{\Omega} \nabla \phi_i \cdot \nabla \phi_j \, dx \]

\[ b = \int_{\Omega} f \nabla \phi_i \, dx \]

Here \( p^{n+1} \) is the vector of the unknown coefficients, that is \( p^{n+1} = c_j \). Equation (4.12) is a linear system and can be solved with a number of effective numerical methods, or directly with Gaussian elimination.

### 4.5 Boundary Conditions

The mesh in shown in figure 3.11b. We divide its boundaries into three. First there are the two ends, one cephalad (top) and one caudad (bottom). We call them \( \partial \Omega_{Ca} \) and \( \partial \Omega_{Ce} \), respectively. \( Ca \) represents caudad and \( Ce \) cephalad. The remaining boundary is the wall of the SAS. We call it \( \partial \Omega_{wall} \). We do not know the velocity field at either the caudad or cephalad end. As a substitute, we will use the volumetric flow rate from the PC-MRI scans instead. Because mass is conserved and we model CSF as incompressible, the volumetric flow rate over any cross section of the mesh is the same. We can therefore divide the flow rate by the area of the relevant boundary to get the bulk velocity per unit area. Let \( Q_{MRI} \) be the measured volumetric flow rate, \( A_{Ca} \) and \( A_{Ce} \) be the area of the caudad and cephalad boundaries respectively. Then

\[ \bar{u}_{caudaCad} = \frac{Q_{MRI}}{A_{Ca}} \]

where \( \bar{u}_{Ca} \) is the bulk velocity at the caudad boundary, and \( Q_{MRI} \) is the measured volumetric flow rate. It is common to use the bulk velocity as a boundary condition \([8][12]\). The correct velocity profile will develop some distance from \( \partial \Omega_{Ca} \) and \( \partial \Omega_{Ce} \). For stationary duct flows, the velocity profile is parabolic. This is discussed in section 5.2. Womersely flow, discussed in section 5.3, is more complex, and exhibits boundary layers. There is a module for imposing time dependent parabolic velocity profiles for circular shaped boundaries in \( \text{cbbcflow} \), but not for annular shaped ones. This is why we used bulk velocities.

Two different sets of conditions will be considered for \( \partial \omega_{Ce} \). The first we consider is

\[ p = 0 \] on \( \partial \Omega_{Ce} \), and use homogeneous Neumann conditions for the velocity. We will look at how this cause flow instabilities. next we investigate the consequences of stabilizing the flow by also prescribing the bulk velocity at \( \partial \Omega_{cephalad} \). Let us first consider the case where \( p = 0 \). In this case, the boundary condition for the equation 4.10 for the tentative
velocity becomes

\begin{align*}
\tilde{u}^{n+1} &= \bar{u}_{Ca}, \quad \text{on } \partial \Omega_{Ca} \\
\tilde{u}^{n+1} &= 0 \quad \text{on } \partial \Omega_{wall} \\
\nabla \tilde{u}^{n+1} \cdot \mathbf{n} + p^{n} \mathbf{n} &= 0, \quad \text{on } \partial \Omega_{Ce}
\end{align*}

As \( p = 0 \) on \( \partial \Omega_{Ce} \), this last condition is an homogeneous Neumann condition for the velocity, meaning that all the boundary integrals in equation 4.10 vanish, as the test functions are required to vanish at boundaries where the solution is known.

The boundary conditions for the pressure correction equation (4.8) are

\begin{align*}
p^{n+1} &= 0, \quad \text{on } \partial \Omega_{Ce} \\
\nabla p^{n+1} \cdot \mathbf{n} &= 0 \quad \text{on } \partial \Omega_{Ce} \cup \partial \Omega_{wall}
\end{align*}

Again, these conditions imply that all boundary integrals in the pressure correction equation vanish as \( p \) is either known, hence the test function vanish, or the last condition force it to vanish.

and finally, the boundary conditions for the velocity update (4.11) are

\begin{align*}
u^{n+1} &= \bar{u}_{caudad}, \quad \text{on } \partial \Omega_{caudad} \\
u^{n+1} &= 0, \quad \text{on } \partial \Omega_{wall}
\end{align*}

Figure 4.2: The left image shows backflow instabilities resulting from lack of flow information outside the domain. The right image shows the more stable flow after setting the velocity at the boundary.
4.5.1 Dirichlet Boundary Conditions

We will now consider the case where we specify the velocity at the cephalad boundary. Now let

\[ u^{n+1} = \bar{u}_{Ce}, \text{ on } \partial \Omega_{Ce} \]

Where \( \bar{u}_{Ce} = \frac{Q_{MRI}}{A_{Ce}} \).

With this new condition, some changes has to be made to the mathematical formulation of the boundary conditions in the previous section. For the equation for the tentative velocity (4.10)

\[ \tilde{u}^{n+1} = \bar{u}_{Ca}, \text{ on } \partial \Omega_{Ca} \]
\[ \tilde{u}^{n+1} = \bar{u}_{Ce}, \text{ on } \partial \Omega_{Ce} \]
\[ \tilde{u}^{n+1} = 0 \text{ on } \partial \Omega_{wall} \]

All of the boundary integrals vanish with the test function, because the solution is known on the whole of \( \partial \Omega \).

The pressure correction equation now only has one condition:

\[ \nabla p^{n+1} \cdot \mathbf{n} = 0, \text{ on } \partial \Omega \]

Which means we can only solve the equation up to a constant. But as only \( \nabla p \) appears in Navier-Stokes equation, the pressure can only be computed up to a constant. Finally, the conditions for the velocity correction equation are the same as ones for the tentative velocity.

\[ u^{n+1} = \bar{u}_{caudad}, \text{ on } \partial \Omega_{caudad} \]
\[ u^{n+1} = \bar{u}_{cephalad}, \text{ on } \partial \Omega_{cephalad} \]
\[ u^{n+1} = 0 \text{ on } \partial \Omega_{wall} \]

4.5.2 Pressure Drop and Boundary Conditions

Figure 4.3 shows the pressure difference between the cephalad and caudad ends. There are no visible differences.

4.5.3 FEniCS

We use FEniCS [20] for the simulations of CSF flow. The FEniCS project is a collection of free software with an extensive list of features for automated, efficient solution of differential equations (http://fenicsproject.org). One of the advantages of FEniCS is how close in syntax it lies to the mathematical variational formulation. For our implementation is based on the tutorials for cbcflow. cbcflow is a collection of solvers for
Figure 4.3: The pressure difference between $y = 280$ and $y = 330$ with the two different boundary conditions. The pressure is averaged over one plane and subtracted.

The incompressible Navier Stokes equations, built on the FEniCS project. It is originally based on solvers from the NSbench project. (https://bitbucket.org/simula_cbc/cbcflow/). It is written using Dolfin, the python interface of FEniCS.
Chapter 5

Verification

The performance and correctness of the solver will be investigated in this section. Two exact solutions to the Navier-Stokes equation 4.1 on page 30 will be used to investigate convergence rates, Poiseuille flow and Womersley flow. The effect of different time steps and mesh resolution will be checked, as well as the number of cardiac cycles needed for the effects of the initial conditions to wear off.

5.0.4 Numerical Errors

We assume that there are two sources of discretization errors in our problem. The time stepping and the spatial discretization. That is, we assume the error can be split into

\[ e(h, \Delta t) = ch^\alpha + d\Delta t^\beta. \]

Here \( c \) and \( d \) are constants, and \( \alpha \) and \( \beta \) are the convergence rates in space and time respectively. In order for us to get an idea of \( c \) and \( d \) are constants, and the values of \( \alpha \) and \( \beta \), we have to separate them somehow. When \( h \) is small, we expect the error to be dominated by the time stepping error, and conversely for \( \Delta t \) [22].

<table>
<thead>
<tr>
<th></th>
<th>Before Smoothing</th>
<th>After Smoothing</th>
<th>Relative Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volunteer 1</td>
<td>38617 mm³</td>
<td>38720 mm³</td>
<td>−0.3%</td>
</tr>
<tr>
<td>Volunteer 2</td>
<td>22039 mm³</td>
<td>22101 mm³</td>
<td>−0.3%</td>
</tr>
<tr>
<td>Volunteer 3</td>
<td>21958 mm³</td>
<td>22010 mm³</td>
<td>−0.2%</td>
</tr>
<tr>
<td>Volunteer 4</td>
<td>21425 mm³</td>
<td>21460 mm³</td>
<td>−0.2%</td>
</tr>
</tbody>
</table>

Table 5.1: The volumes of each of the models before and after the Taubin smoothing.
5.1 Surface Smoothening

The differences in volume before and after smoothing are shown in table 5.1 on the previous page. The relative difference is computed by

\[
\text{Difference} = \frac{V_{\text{Before}} - V_{\text{After}}}{V_{\text{Before}}}
\]

The relative difference for the model shown in figure 3.9c on page 26 is 7%. This model has lost many of its defining features and has become very thin in some regions. Figures 3.9a on page 26 and 3.9b on page 26, show how the Taubin non-shrinking algorithm smooths the surface while keeping most of the features intact. It should be noted that table 5.1 on the previous page report small volume increases. This could be explained if the inner walls shrunk faster than the outer walls, marginally thus increasing the volume. Another explanation is that the parameter \(\mu\) in equation 3.1 on page 23 is too large.

5.2 Poseuille Flow

The laminar flow through a cylindrical pipe with uniform cross section is known as Hagen-Poseuille flow. It can be derived from the Navier-Stokes equations under the following set of assumptions:

(i) The flow is steady \(\left(\frac{\partial u}{\partial t} = 0\right)\)

(ii) The radial and swirl components are zero \((u_r = u_\theta = 0)\).

(iii) The flow is axisymmetric \(\left(\frac{\partial u_z}{\partial \theta}\right)\)

(iv) The flow is fully developed. \(\left(\frac{\partial u_z}{\partial z} = 0\right)\)

The angular component of the momentum equation is identically satisfied, and the radial component is \(\frac{\partial p}{\partial r} = 0\). The axial part of the momentum equation reduces to

\[
\frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial u_z}{\partial r} \right) = \frac{1}{\mu} \frac{\partial p}{\partial z}, \tag{5.1}
\]

With the solution

\[
u_z(r) = \frac{1}{4\mu} \frac{\partial p}{\partial z} r^2 + c_1 \ln r + c_2.
\]

Since \(u_z\) is finite at \(r = 0\), \(c_1 = 0\). The no slip condition, \(u_z = 0\) at \(r = R\), yields \(c_2 = -\frac{1}{4\mu} \frac{\partial p}{\partial z} R^2\). Our final velocity profile then is

\[
u_z(r) = -\frac{1}{4\mu} \frac{\partial p}{\partial z}(R^2 - r^2)
\]

In the bottom row of table 5.2 where the temporal error is small, we see almost linear convergence in \(h\). In the leftmost column we see almost no convergence in \(\Delta t\), But that is to be expected as Poseuille flow assumes fully developed flow. The only temporal variations are when we let the flow develop from rest. But if \(\Delta t\) is too large, the solution might not converge.
Womersley Flow

The driving force behind Womersley flow is a fluctuating pressure gradient. This type of flow appears in biomedical flow, such as CSF flow. The governing equation is similar to equation 5.1 on the preceding page, only now we let the pressure gradient be a periodic function in time with frequency \( f = \frac{n \pi}{2} \). The pressure gradient can be represented by a Fourier series

\[
\frac{\partial P}{\partial z} = Re \left[ \sum_{n=0}^{\infty} a_n e^{i\pi n t} \right].
\]

The governing equation now is

\[
\frac{\partial^2 u_z}{\partial r^2} + \frac{1}{r} \frac{\partial u_z}{\partial r} - \frac{\rho}{\mu} \frac{\partial u_z}{\partial t} = -a_n e^{i\pi n t} \mu
\]

(5.2)

Because \( u \) changes periodically with time, we look for a solution of the form

\[
u(r, t) = \omega(r) e^{i\pi n t}\]

Inserting this into equation (5.2), and rearranging, we get a Bessel zero order differential equation.

\[
\frac{d^2 \omega}{dr^2} + \frac{1}{r} \frac{d\omega}{dr} + \lambda^2 u = 0
\]

(5.3)

with

\[
\lambda^2 = \frac{i^3 \pi n}{\nu}
\]

Equation (5.3) has the general solution

\[
\omega_n(r) = c_1 J_0(\lambda r) + c_2 Y_0(\lambda r)
\]

subject to the conditions that \( \omega_n(r) \) is finite at the centre of the pipe \( r = 0 \), and no-slip conditions at the walls, i.e. \( \omega_n(r) = 0 \) at \( r = R \). \( J_0 \) and \( Y_0 \) are Bessel functions of first and second kind, respectively. Using these conditions, we find

\[
\omega_n(r) = \frac{a_n}{ip fn} \left( \frac{J_0(\lambda r)}{J_0(\lambda R)} - 1 \right)
\]

Summing all the frequencies, and adding a steady flow component, we finally get

\[
\omega(r) = \omega_0 + \sum_{n=1}^{\infty} \omega_n(r)
\]
<table>
<thead>
<tr>
<th>dt(s) \ h(mm)</th>
<th>3.54e-1</th>
<th>1.77e-1</th>
<th>8.84e-2</th>
<th>4.42e-2</th>
<th>2.21e-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00e-1</td>
<td>152.912</td>
<td>125.112</td>
<td>118.297</td>
<td>115.493</td>
<td>114.52</td>
</tr>
<tr>
<td>2.50e-2</td>
<td>3.227</td>
<td>2.625</td>
<td>2.832</td>
<td>2.996</td>
<td>3.234</td>
</tr>
<tr>
<td>1.25e-2</td>
<td>0.893</td>
<td>0.599</td>
<td>0.545</td>
<td>0.532</td>
<td>0.574</td>
</tr>
<tr>
<td>6.25e-3</td>
<td>0.363</td>
<td>0.165</td>
<td>0.140</td>
<td>0.107</td>
<td>0.104</td>
</tr>
<tr>
<td>3.125e-3</td>
<td>0.310</td>
<td>0.106</td>
<td>0.074</td>
<td>0.037</td>
<td>0.028</td>
</tr>
</tbody>
</table>

Table 5.3: The $L^2$ error obtained using P1 - P1 elements and IPCS

Table 5.3 was produced following the cbcflow demo for 3D Womersley flow. The convergence rates are quite good for the time step, but roughly doubling the number of cells has little effect.

## 5.4 Cycle repeatability

One source of errors is associated with the initial conditions. In this thesis we will consistently use $u = 0, 0, 0$ and $p = 0$ as initial conditions. But as illustrated in section 5.2, this does not matter once we reach a steady state.

CSF flow is simulated for 6 cardiac cycles. We look at the variation from cycle to cycle to determine how many cycles it is necessary to simulate. For each cycle, maximum and minimum velocities are computed and compared. Spatial velocity distributions are computed and visually compared for each cycle. The maximum velocity is defined as discrete vertical velocity component which has greater magnitude than every other discrete velocity for 50 time steps in either direction.

Figure 5.1 on the next page shows the maximum caudad velocity in an axial plane computed after a different number of cardiac cycles. The velocities shown are almost equal are marginally larger in the last cycle than the first. This is in accordance with the velocities reported in table 5.4.

<table>
<thead>
<tr>
<th>cycle</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{\text{max}}$</td>
<td>-49.22</td>
<td>-49.26</td>
<td>-49.27</td>
<td>-49.27</td>
</tr>
<tr>
<td>Difference</td>
<td>-0.12%</td>
<td>-0.03%</td>
<td>0.01%</td>
<td>— %</td>
</tr>
</tbody>
</table>

Table 5.4: The maximum cephalad velocity in each cycle. The difference is computed with reference to cycle 6

Let $V_6$ be the maximum velocity at the 6th cycle. Then the difference is computed as

$$\text{difference}_i = \frac{V_i - V_6}{V_6} \times 100, \text{ for } i = 3, 4, 5, 6$$

Then we look at the pressure difference.
From figure 5.1 and from tables 5.4 on the facing page and 5.5 we conclude that the flow is fully developed after the third cycle. We will therefore satisfy ourselves to simulating three cycles, saving lots of time.

5.5 Time Stepping

In section 5.3 the importance of time stepping to the discretization error was investigated. Another feature of the time stepping is that it has to resolve the temporal scales, and capture the transient behaviour of the flow. To confirm that the time step is small enough, two cases are simulated with two different time steps. To determine whether the chosen time step is small enough, maximum velocities, velocity profiles and the pressure drop are
compared using different time steps. Two time steps were chosen based on the Courant number described in section 3.3.5 on page 25. Haughton et al., 2003, report peak CSF velocities in normal volunteers up to $48 \text{mm/s}$ in the foramen magnum [11]. We estimate that velocities in inferior regions of the SAS will be higher as the cross sectional area decreases. We made an a priori estimate of the peak velocity at $100 \text{mm/s}$. The minimum distance between two nodes in the mesh is $0.2 \text{mm}$. We then found an estimate for the maximum time step:

$$\Delta t = \frac{100 \text{mm/s}}{0.2 \text{mm}} = 2 \times 10^{-3}$$

We then chose two time steps to compare: $\Delta t = 10^{-3}$ and $\Delta t = 5 \times 10^{-4}$.

**Velocity distribution**

![Computed Velocity: level: 331, time: 0.115386](image1) ![Computed Velocity: level: 331, time: 0.108174](image2)

(a) Cephalad flow, $\Delta t = 10^{-3}$

(b) Cephalad flow, $\Delta t = 0.5 \times 10^{-4}$

![Computed Velocity: level: 331, time: 0.259618](image3) ![Computed Velocity: level: 331, time: 0.259618](image4)

(c) Caudad flow, $\Delta t = 10^{-3}$

(d) Caudad flow, $\Delta t = 0.5 \times 10^{-4}$

Figure 5.2: Comparing velocity distribution at the time of maximum cephalad and caudad velocity. The figures in each row has been computed using a different time step.

Figure 5.2 show the spatial variation in maximum caudad and cephalad velocity over an axial plane. The plots in each row has been computed with a different time step. There are no visible differences in the velocity distribution. Table 5.6 on the facing page lists maximum velocities in both directions, and lists the differences.
\[ \Delta t = 10^{-3} \quad \Delta t = 5 \times 10^{-4} \quad \text{Difference} \]

<table>
<thead>
<tr>
<th>Velocity</th>
<th>( \Delta t = 10^{-3} )</th>
<th>( \Delta t = 5 \times 10^{-4} )</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_{Ce} )</td>
<td>54.14</td>
<td>53.97</td>
<td>-0.32%</td>
</tr>
<tr>
<td>( V_{Ca} )</td>
<td>-24.87</td>
<td>-24.86</td>
<td>-0.05%</td>
</tr>
</tbody>
</table>

Table 5.6: The differences in maximum velocities computed with two different time steps.

The difference in table 5.6 is computed as

\[
\frac{V_f - V_c}{V_f} \times 100.
\]

To further investigate the effect of the time step of the time stepping on the pressure drop, the pressure drop for both cases is graphed for one cardiac cycle. The result is shown in figure 5.3.

![Pressure difference for two different time steps](image)

Figure 5.3: A comparing pressure drop over one cardiac cycle for two different time steps.

## 5.6 Mesh Convergence

The effect of mesh resolution is related to that of time stepping. Similar tests were performed to investigate this effect. Even though we assume laminar flow, there might still be spatially high frequent features of the flow which requires a much finer grid.

The velocities reported in table 5.8 on the next page are the maximum caudad and cephalad velocities. The number of cells in the three meshes are given in table 5.7 on the following page. Table 5.8 on the next page show maximum caudad and cephalad velocity across an axial cross section. The velocities are calculated on three different meshes, one coarse, one medium and one fine.
Table 5.7: The number of cells used in the different meshes.

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Coarse</th>
<th>Medium</th>
<th>Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of cells</td>
<td>1539071</td>
<td>3909930</td>
<td>8033644</td>
</tr>
</tbody>
</table>

Table 5.8: Comparing maximum velocities is two meshed with different spatial resolution

<table>
<thead>
<tr>
<th>Velocity</th>
<th>coarse</th>
<th>medium</th>
<th>fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{Ce}$</td>
<td>25.89</td>
<td>25.68</td>
<td>24.51</td>
</tr>
<tr>
<td>$V_{Ca}$</td>
<td>-55.24</td>
<td>-47.32</td>
<td>-44.78</td>
</tr>
</tbody>
</table>

Table 5.9: Comparing mean of the 1% largest velocities by magnitude is two meshed with different spatial resolution

<table>
<thead>
<tr>
<th>Velocity</th>
<th>coarse</th>
<th>medium</th>
<th>fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{Ce}$</td>
<td>24.71</td>
<td>23.81</td>
<td>24.29</td>
</tr>
<tr>
<td>$V_{Ca}$</td>
<td>-49.21</td>
<td>-43.10</td>
<td>-42.77</td>
</tr>
</tbody>
</table>

Table 5.9, the mean of the 1% largest velocities are much more similar. Indeed, the difference between the mean of the 10% largest velocities is even smaller between the meshes. These further investigations point to two things. First, increasing the mesh resolution does not lead to any qualitative changes in the solution, and second, the coarse mesh is satisfactory.

Figure 5.4 on the next page show the spatial variation in velocity at the time of maximum velocities. We see that maximum velocity occurs earlier the finer the mesh is. Apart from that, they look qualitatively the same. Table 5.8 shows as somewhat discouraging result. It seems unreasonable to expect that the lack of resolution can explain the 50% difference to the MRI data in table 5.8, given that they look qualitatively the same.
Figure 5.4: Comparing velocity distribution at the time of maximum cephalad and caudad velocity computed on different meshes.
Chapter 6

Results

In the figures presented in this thesis, cephalad velocities are defined as positive, while caudad velocities are negative. We have MRI scans of four volunteers. For each volunteer, we have four PC-MRI series. One PC-MRI series consists of 32 images. The value of each pixel in the image corresponds to the velocity in that pixel. Each image has associated scaling factors, which relates the pixels to a physical length in each dimension. The volumetric flow rates derived from these PC-MRI scans are used as boundary conditions at both $\partial \Omega_{Ce}$ and $\partial \Omega_{Ca}$, as described in section 4.5. In this section we will present the results of the simulations and compare them to the measured velocities.

Each PC-MRI measurement reports the CSF velocities in a cross section of the SAS. We need to know where these cross sections are to make meaningful comparisons. The coordinate systems used in the PC-MRI series differ from the other MRI scans, but we have pictures of where these cross sections are. To find the right coordinates we will compare them visually. The PC-MRI series are all from slightly different cross sections, and because of the uncertainty about the exact locations, we will investigate how the simulated velocity magnitudes and profiles vary between different cross sections.

Before comparing computed and measured velocities, we will look at how the cross sectional area of the mesh varies and how it relates to the cross sectional area reported by the PC-MRI. We will then compare peak and mean velocities during one cardiac cycle. When investigating peak velocities we will calculate the mean of the 10% largest velocities by magnitude. This method is less sensitive to noise in the PC-MRI series than simply looking at maximum velocities. Finally we will plot the velocity profile over an axial cross section at selected time steps. When referring to these cross sections, we will give the vertical coordinate, which is along the y-axis. The highest values are caudad, and then decrease as we move cephalad. More generally, the cross sections will be referred to as cephalad, middle and caudad, describing their position relative to each other. In the tables, the abbreviations Ce, Mi and Ca will be used, standing for cephalad, middle and caudad respectively. All plots of mean and peak velocities have been phase-shifted so that maximum caudad flow occurs halfway through the cardiac cycle. This is done for ease of comparison.

Starting off with volunteer 1, we will investigate how the areas reported in the PC-MRI
match the cross sections of the mesh. We will explain how we determine where it is most probable the measurements were made, and the cross sectional area is determined. We will then look at the maximum and mean velocities and what the velocity profile is like in an axial cross section. We will investigate the effect of increasing the volumetric flow rate on all the quantities mentioned above. Finally, we will compare the pressure drop and how it relates to the mean velocity. We do not have pressure measurements, so we use pressure gradients computed from the PC-MRI velocity measurements. Having explained in detail how the comparisons are made, we will present the results for volunteers 2, 3 and 4 in a more condensed form.

6.1 Volunteer 1

6.1.1 Cross Sectional Areas and PC-MRI Measurement Locations

The line in picture in figure 6.1a show the location where the PC-MRI scans were made. The neighbouring figure 6.1b shows our best estimate of the corresponding location in the MRI used in our simulations. From figure 6.1a, it is clear that the PC-MRI is taken around C3/C4.

![An image of the PC-MRI measurement location](image1)

![An interval containing the estimated corresponding CFD sampling location](image2)

Figure 6.1: Two images of different MRI scans of volunteer 1. The left has a line showing where the PC-MRI measurements were made. The right image shows an interval with our estimate of the corresponding location. We will compare the measured velocities to CFD velocities sampled from three cross sections in this interval.

The plot in figure 6.2a show how the cross sectional area of the mesh varies with the distance from the caudad end. The horizontal lines are the cross sectional areas of each
PC-MRI series. The cross sectional area of the mesh does not equal those of the PC-MRI series. Investigating this further, we looked directly at the level set file, described in section 3.1. In the process of creating the mesh, it was smoothed and remeshed, as discussed in section 3.3.2. Both of these procedures could have changed the cross sectional area. Computing the cross sectional area directly from the level set will rule out changes to the volume of the model from these two steps. The red line in figure 6.2b show that the cross sectional areas do match, but in a location not consistent with the picture in figure 6.1a.

![Variation in Area by Height](image1)

(a) The red dotted line is the cross sectional area of the mesh as it varies with the distance to the caudad end.

![Levelset Area](image2)

(b) The red line is the cross sectional area of the level set as it varies with the distance to the caudad end. The vertical black line is the middle of the interval shown in figure 6.1b. The areas match in a location not in agreement with figure 6.1a.

Figure 6.2: Two plots of the cross sectional area of the computational model of volunteer 1 and PC-MRI cross sectional areas. The left figure shows the variation in cross sectional area of the mesh, and the right of the level set file. The green vertical line in the left figure corresponds to $x = 0$ in the left image. In neither of the figures do the cross sectional areas match in a location agreeing with figure 6.1. The horizontal lines are the cross sectional area of the PC-MRI series.

The red line in figure 6.2a were computed by the code snippet below.

```python
def find_area(mesh, y, eps=.25):
    c = Expression("x[1] < y + eps && x[1] > y - eps",
                   y=y, eps=eps)
    a = assemble(c*ds(mesh))/(2*eps)
    return a
```

The expression $c$ evaluates to 1 in in the box between $y + \epsilon$ and $y - \epsilon$. The volume of this box is computed by the call to `assemble`. Dividing by the height of the box, $2\epsilon$ we then
have the average cross sectional area around y. The red line in figure 6.2a is computed by letting y vary with the height of the mesh. This code uses FEniCS. We chose $\epsilon = 0.5$ because this is approximately the vertical resolution of the MRI scans.

The plot in figure 6.2b was made by the code following two code snippets. It iterates over the number of pixels in each level and counts how many are inside the surface. Pixels with value less than zero are inside the surface, which is located at the value zero.

```python
xmin, xmax, ymin, ymax, zmin, zmax = image.GetExtent()
for j in xrange(y):
    counter = 0  # number of pixels inside surface
    for i in xrange(nx):
        for k in xrange(nz):
            a = image.GetScalarComponentAsFloat(xmin + i, ymax - j, zmin + k, 0)  # ymax is caudadd
            if a <= 0:  # Surface is at level 0
                counter += 1
    yield counter
```

When we know how many pixels there are in the cross section, we multiply with the scaling factors, converting the number of pixels to physical units. In this case, the scaling factors, $dx$ and $dz$ are 0.488mm and 1.000mm respectively. In the PC-MRI they are 0.556mm and 0.556mm.

```python
dx, dy, dz = image.GetSpacing()
numpix = np.array([num for num in count_pixels(image, 100)])
area = numpix * dx * dz
x = np.arange(len(numpix)) * dy
```

Both these code snippets use the *vtk* module.

In table 6.1 the cross sectional areas of the PC-MRI series are listed and compared to our estimated cross section located at $y = 331$, marked by the black line in figure 6.2b. The difference is computed as follows:

$$\text{Difference} = \frac{\text{MRI Area} - A_{331}}{A_{331}} \times 100.$$  

The variation in cross sectional area of the mesh can be read from figure 6.2. This discrepancy in cross sectional area is a source of errors when we compare velocities later. It should also be noted that the cross sectional area from series to series vary, as seen on table 6.1. Even though the PC-MRI series are from different cross sections, we will not vary our estimated $y$ coordinate. We will, however, investigate the effect of sampling the CFD velocities 5mm caudad and cephalad. The required number of pixels for the cross sectional areas in table 6.1 to match, range from 130.56 to 33.55 pixels. The numbers are decimals because we cannot expect the physical cross sectional area to be divisible by the resolution of the image.
Table 6.1: Table of the relative differences in area between the mesh and PC-MRI series of volunteer 1. The differences are computed relative to the cross sections of the mesh.

<table>
<thead>
<tr>
<th>Series</th>
<th>PC-MRI</th>
<th>Ce area</th>
<th>area diff</th>
<th>Mi area</th>
<th>area diff</th>
<th>Ca area</th>
<th>area diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>139.72</td>
<td>157.36</td>
<td>11.21</td>
<td>150.02</td>
<td>6.87</td>
<td>156.17</td>
<td>10.54</td>
</tr>
<tr>
<td>S2</td>
<td>125.51</td>
<td>157.36</td>
<td>20.24</td>
<td>150.02</td>
<td>16.34</td>
<td>156.17</td>
<td>19.63</td>
</tr>
<tr>
<td>S3</td>
<td>132.62</td>
<td>157.36</td>
<td>15.72</td>
<td>150.02</td>
<td>11.60</td>
<td>156.17</td>
<td>15.08</td>
</tr>
<tr>
<td>S4</td>
<td>109.74</td>
<td>157.36</td>
<td>30.26</td>
<td>150.02</td>
<td>26.85</td>
<td>156.17</td>
<td>29.73</td>
</tr>
</tbody>
</table>

6.1.2 Volumetric Flow Rates and PC-MRI Series

The measured volumetric flow rates for each of the four series are plotted in figure 6.3. The four MRI series show the same behaviour. The amplitudes are similar, but the periods vary from 0.85s to 1.12s. The volumetric flow rate is computed as the product of the mean velocity and the cross sectional area.

Figure 6.3: The volumetric flow rate for four PC-MRI series of volunteer 1. The graphs has been phase shifted so the time of maximum caudad flow is the same.

6.1.3 Comparing Velocities

Mean Velocities

Figure 6.4 compares the simulated and measured mean velocities at three different cross sections. The cross sections used in figures 6.4 and 6.5 are shown in figure 6.1. The cross sections are all 5mm apart. The cross section at y = 336 is most caudal. The measured
Figure 6.4: The dotted line is the mean of the measured velocities in volunteer 1. The solid lines is the mean of the sampled CFD velocities at the three different cross sections. The one at $y = 326$ is cephalad.

Velocity magnitudes are higher than the simulated ones. The velocity waveforms are also similar, but there is a phase difference. This is true for all three series, even though the waveforms vary between them, is similar to the flow rates in figure 6.3. There is little variation from cross section to cross section.

**Peak Velocities**

Figure 6.5: The mean of 10% largest velocities by magnitude for velocities at the three different cross sections in volunteer 1. The one at $y = 326$ is cephalad.

The discrepancy in magnitude between the computed peak velocities and the measured ones are even larger than with the mean velocity, but still the waveforms match well. The most cephalad cross section, $y = 336$ report lower peak velocities for all three series. This is the cross section with the largest area. Tables 6.2 6.3 tabulate the range of the mean and peak velocities. The range of e.g. the mean velocity is the difference between the maximum and minimum mean velocities. The difference is computed relative the the computed velocities.

```python
perc = 0.1*z.size
maxind = np.argpartition(-np.abs(z), perc)[:perc]
return z[maxind]
```

The code snippet above computes the 10% largest elements in an array by absolute value.
In our case the elements are velocities computed at a node in the mesh or given as the value of a pixel in a PC-MRI. The first line computed the number of elements corresponds to 10% of the total number of elements. We then perform a partial sorting of the \textit{perc} smallest elements, and return the corresponding indices.

<table>
<thead>
<tr>
<th>–</th>
<th>Ce</th>
<th>Mi</th>
<th>Ca</th>
</tr>
</thead>
<tbody>
<tr>
<td>–</td>
<td>(u_{\text{CFD}})</td>
<td>(u_{\text{MRI}})</td>
<td>(\text{diff})</td>
</tr>
<tr>
<td>S2</td>
<td>36.26</td>
<td>44.32</td>
<td>-22.22</td>
</tr>
<tr>
<td>S3</td>
<td>42.54</td>
<td>52.18</td>
<td>-19.67</td>
</tr>
<tr>
<td>S3</td>
<td>44.02</td>
<td>61.38</td>
<td>-17.36</td>
</tr>
</tbody>
</table>

Table 6.2: The range of mean velocities in volunteer 1. The range is computed as the difference between the largest and smallest velocity found during a cardiac cycle. The difference is computed relative to the CFD range.

<table>
<thead>
<tr>
<th>–</th>
<th>Ce</th>
<th>Mi</th>
<th>Ca</th>
</tr>
</thead>
<tbody>
<tr>
<td>–</td>
<td>(u_{\text{CFD}})</td>
<td>(u_{\text{MRI}})</td>
<td>(\text{diff})</td>
</tr>
<tr>
<td>S2</td>
<td>55.26</td>
<td>82.71</td>
<td>-49.68</td>
</tr>
<tr>
<td>S3</td>
<td>66.98</td>
<td>86.52</td>
<td>-29.18</td>
</tr>
<tr>
<td>S4</td>
<td>66.19</td>
<td>100.40</td>
<td>-34.21</td>
</tr>
</tbody>
</table>

Table 6.3: The range of peak velocities in volunteer 1. The range is computed as the difference between the largest and smallest velocity found during a cardiac cycle. The difference is computed relative to the CFD range.
Comparison of computed and measured peak cephalad velocity profiles

Figure 6.6: Volunteer 1: PC-MRI of peak cephalad flow

Figure 6.7: Cephalad cross section

Figure 6.8: Middle cross section

Figure 6.9: Caudad cross section
Comparison of computed and measured peak caudad velocity profiles

(a) Series 2  
(b) Series 3  
(c) Series 4

Figure 6.10: Volunteer 1: PC-MRI of peak caudad flow

Figure 6.11: Cephalad cross section

Figure 6.12: Middle cross section

Figure 6.13: Caudad cross section
Comparison of measured and computed velocities profiles at flow reversal

Figure 6.14: Volunteer 1: PC-MRI of flow reversal

Figure 6.15: Cephalad cross section

Figure 6.16: Middle cross section

Figure 6.17: caudad cross section

The contour plots in figures 6.6 to 6.17 show the velocity profile across three different axial cross sections from the interval shown in figure 6.1b. There are contour plots
of the velocity profile at three different times: Peak cephalad flow, peak caudad flow and flow reversal. Table 6.4 lists the L2 norm of the difference between the computed velocity at three different cross sections and the PC-MRI measurements. Peak cephalad flow, peak caudad flow and flow reversal are referred to as T1, T2 and T3 respectively. Flow reversal is defined as the first time step after peak cephalad flow where the mean velocity is negative (caudad). The plots in figure 6.6 are contour plots of the PC-MRI scans. The contour plots in each column use the same volumetric flow rate as boundary conditions. They are shown in figure 6.3. All contour plots from the same series show the same instance in time. As can be seen from figures 6.4 and 6.5, the time of e.g. peak caudad flow can vary between cross sections and from CFD simulations to PC-MRI measurements.

<table>
<thead>
<tr>
<th>Location</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce</td>
<td>0.64</td>
<td>0.63</td>
<td>0.88</td>
</tr>
<tr>
<td>Mi</td>
<td>0.69</td>
<td>0.67</td>
<td>0.94</td>
</tr>
<tr>
<td>Ca</td>
<td>0.85</td>
<td>0.84</td>
<td>1.03</td>
</tr>
<tr>
<td>T2</td>
<td>0.62</td>
<td>0.57</td>
<td>0.87</td>
</tr>
<tr>
<td>Ce</td>
<td>0.64</td>
<td>0.59</td>
<td>0.91</td>
</tr>
<tr>
<td>Mi</td>
<td>0.80</td>
<td>0.76</td>
<td>0.96</td>
</tr>
<tr>
<td>Ca</td>
<td>0.87</td>
<td>0.91</td>
<td>0.96</td>
</tr>
<tr>
<td>T3</td>
<td>1.46</td>
<td>1.28</td>
<td>1.70</td>
</tr>
<tr>
<td>Ce</td>
<td>1.47</td>
<td>1.32</td>
<td>1.70</td>
</tr>
<tr>
<td>Mi</td>
<td>1.49</td>
<td>1.37</td>
<td>1.68</td>
</tr>
<tr>
<td>Ca</td>
<td>1.28</td>
<td>1.37</td>
<td>1.68</td>
</tr>
</tbody>
</table>

Table 6.4: A table of the L2 norm of the difference between the PC-MRI and simulated velocities at each of the three cross section and at three time steps. T1 refers to peak cephalad flow, T2 to peak caudad flow and T3 to flow reversal. The L2 norm takes both the velocity and geometry into account.

In the contour plots for peak cephalad flow, we find the largest velocities in the widest areas of the cross section. This is especially true of series 2, but a similar pattern can be seen in series 3 and 4. For peak caudad flow, we find the opposite behaviour. The PC-MRI velocities still display the largest velocities in the widest areas. At flow reversal the PC-MRIs exhibit a complex velocity profile, with much greater variety than we see in the computed velocities. The PC-MRIs report velocities with greater magnitude in both directions, while the computed velocities have a more homogeneous profile. In the CFD contour plots, we see that the flow first reverses close to the walls.

The code in listing 2 interpolates the CFD data and MRI data to a common mesh. It is then possible to take the L2 norm of the difference velocities. Even though the PC-MRIs and the cross sections of the mesh have different geometries, this L2 norm will help us differentiate between the cross sections.

### 6.2 Volunteer 1 with Increased Volumetric Flow Rate

The maximum measured velocities reported in the last section are about 1.5 times as high as the computed ones. The simulations were performed using the measured volumetric flow rate as input. Aiming to match the measured velocities, we increased the volumetric flow rate by a factor 1.5. The volumetric flow rate of series 1 with normal flow rate is shown in figure 6.3.
6.2.1 Comparison of computed and measured velocity profiles with increased Volumetric Flow Rate

Figure 6.18: Measured and computed peak cephalad velocities with a higher volumetric flow rate in volunteer 1 with increased volumetric flow rate.

Figure 6.19: Measured and computed peak caudad velocities with a higher volumetric flow rate in volunteer 1 with increased volumetric flow rate.

Figure 6.20: Measured and computed velocities at flow reversal in Volunteer 1 with increased volumetric flow rate.

Figures 6.18 to 6.20 show the velocity profile in three axial cross sections compared with the PC-MRI measurements. With increased low rate, the computed velocities are now much higher, as can also be seen in figures 6.21.

The erratic behaviour of the velocity in the caudad cross section can be explained by it being so close to the caudad boundary, only 2mm. With higher velocity at the inlet, it may take longer for a correct velocity profile to develop.

6.2.2 Mean and Peak Velocities with Increased Volumetric Flow Rate

Increasing the flow rate has an effect on both the mean and peak velocities. But figure 6.21 it indicates that also the velocity waveforms change. Increasing the flow rate by a factor 1.5 increased the mean and peak velocities by more than that factor. These differences are listed in table 6.6.
### Table 6.5: A Table of the L2 norm of the difference between the PC-MRI series and the computed velocities at each of the three cross sections in volunteer 1 with increased flow rate.

<table>
<thead>
<tr>
<th></th>
<th>Series 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ce</td>
</tr>
<tr>
<td>T1</td>
<td>1.27</td>
</tr>
<tr>
<td>T2</td>
<td>1.10</td>
</tr>
<tr>
<td>T3</td>
<td>1.78</td>
</tr>
</tbody>
</table>

Figure 6.21: The dotted line is the peak and mean of the measured velocities in volunteer 1 with increased volumetric flow rates. The solid lines is the mean of the sampled CFD velocities at the three different cross sections. The one at $y = 326$ is cephalad.

### Table 6.6: The range of mean and peak velocities in volunteer 1 with increased volumetric flow rate. The range is computed as the difference between the largest and smallest velocity found during a cardiac cycle. The difference is computed relative to the CFD range.

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>peak</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce</td>
<td>$u_{\text{CFD}}$</td>
<td>$u_{\text{MRI}}$</td>
</tr>
<tr>
<td>mean</td>
<td>169.99</td>
<td>48.59</td>
</tr>
<tr>
<td>peak</td>
<td>256.86</td>
<td>82.70</td>
</tr>
</tbody>
</table>

---

### 6.2.3 The Effect of increased Volumetric Flow rate on the Pressure Drop

One quantity of interest not immediately measurable is the pressure drop. The pressure drop $\Delta p$ is computed by taking the difference $p_{\text{Ce}} - p_{\text{Ca}}$, where Ce denote cephalad and Ca denotes Caudad.

Figure 6.22 shows the pressure drop and mean velocity through a cardiac cycle. The dotted lines are the PC-MRI velocities and pressure gradient. The pressure gradient is
computed from Navier-Stokes equations assuming a one componential velocity field.

\[
\frac{\partial p}{\partial y} = -\mu \frac{\partial u}{\partial t} + \mu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial x^2} \right)
\]

In series 2, 3 and four, with normal flow rate, the waveforms of the pressure drops and pressure gradients are similar. We see a spike in pressure right before the mean velocity falls sharply. In series 1, with increased flow rate, the pressure drop and pressure gradient are no longer similar. The phase difference between the pressure drop and the computed mean velocity has become so large that it is difficult to explain the relationship. Table 6.7 lists the difference rang of the mean velocities and pressure drop and pressure gradient, as well as the phase differences. For series 2, 3 and 4, the phase difference is about 6% of the cardiac cycle while the phase difference between the pressure drop and pressure gradient is 1 to 2 percentage points larger. The range of a quantity is the difference between the maximum and minimum value over a cardiac cycle. The phase difference was computed using a cross correlation of the two signals.
Table 6.7: A table of the range in mean velocities for measured and computed velocities for volunteer 1. The range of $u$ is the difference between maximum and minimum mean velocities found in the cross section at $y = 331$. The pressure drop is the difference of the mean pressure 5 mm cephalad of $y = 331$ and 5 mm caudad of $y = 331$. The range is computed the same way as with the velocities. The phase difference is computed using a cross correlation of the measured and computed velocities.
def make_grid(cx, cz, mx, mz, n=1000):
    """ Return common grid for MRI and CFD """
    x = np.concatenate((cx, mx))
    z = np.concatenate((cz, mz))
    return np.mgrid[np.min(x):np.max(x):n*1j,
                    np.min(z):np.max(z):n*1j]

def l2(mri_path, cfdpath):
    c = np.loadtxt(cfdpath, skiprows=1,
                   delimiter=" ", unpack=True)
    v0, v1, v2, cx, cy, cz = c
    cx -= np.min(cx)  # translate to (0, 0)
    cz -= np.min(cz)
    
    # convert to mm
    mri = np.loadtxt(mri_path, delimiter=" ")*10
    
    # remove rows and columns with only 0
    mri = mri[~(mri == 0).all(1)]
    mri = mri.T[~(mri == 0).all(0)].T
    
    dx_mri = 0.556  # mri pixel scale factor
    
    # MRI grid
    x = np.arange(mri.shape[0])*dx_mri
    z = (np.arange(mri.shape[1]))*dx_mri
    grid_x, grid_z = make_grid(cx, cz, mx, mz, n=1000)

    # interpolate irregular CFD data to common regular grid
    cfd = -scint.griddata((cx, cz), v1, (grid_x, grid_z),
                          fill_value=0)

    # Create spline representation of
    # MRI data interolate to common grid
    znew = scint.interp2d(mx, mz, mri.T, fill_value=0)
    data = np.fliplr(znew(grid_x[:, 0], grid_z[0, :]))

    # compute L2 norm
    x = grid_x[:, 0]
    z = grid_z[0, :]
    return np.sqrt(np.trapz(np.trapz(np.power(cfd - data, 2)
                                      , x, axis=1), z))
6.3 Volunteer 2

6.3.1 Cross Sectional Areas and PC-MRI Measurement Locations

Figure 6.23: Two images of different MRI scans of volunteer 2. The left has a line showing where the PC-MRI measurements were made. The right image shows an interval with our estimate of the corresponding location. We will compare the measured velocities to CFD velocities sampled from three cross sections in this interval.

The location of the PC-MRI measurements are shown in figure 6.23a. Our best estimate of the corresponding location in the MRI used for segmentation is shown in figure 6.23b. From figures 6.23 and 6.1 we see that the measurement location for both volunteer 1 and 2 is between vertebrae C3 and C4.

6.3.2 Cross Sectional Area

Figure 6.24 shows how the area in the level set file vary vertically. The vertical black line marks the coordinate of the estimated measurement location. It is made on the basis of the picture in figure 6.23a. We will investigate the sensitivity to the exact location by also sampling the computed velocities at two cross sections 5 mm caudad and cephalad of this. As seen from table 6.8, the cross sectional areas between the series are similar. On the other hand, the cross section of the mesh are about 50% larger than the PC-MRI cross section.
Figure 6.24: The cross sectional area of the level set for volunteer 2. The horizontal lines are the cross sectional areas of the PC-MRI series. The vertical line denote the vertical coordinate of the centre of the interval shown in figure 6.23b.

Table 6.8: Table of the relative differences in cross sectional areas between the mesh and PC-MRI series of volunteer 2. The differences are computed relative to the cross sections of the mesh.

<table>
<thead>
<tr>
<th>Series</th>
<th>PC-MRI area</th>
<th>Ce area</th>
<th>Mi area diff</th>
<th>Ca area</th>
<th>diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>78.83</td>
<td>146.13</td>
<td>46.06</td>
<td>122.25</td>
<td>35.51</td>
</tr>
<tr>
<td>S2</td>
<td>72.34</td>
<td>146.13</td>
<td>50.50</td>
<td>122.25</td>
<td>40.82</td>
</tr>
<tr>
<td>S3</td>
<td>79.45</td>
<td>146.13</td>
<td>45.63</td>
<td>122.25</td>
<td>35.01</td>
</tr>
<tr>
<td>S4</td>
<td>75.74</td>
<td>146.13</td>
<td>48.17</td>
<td>122.25</td>
<td>38.04</td>
</tr>
</tbody>
</table>

6.3.3 Volumetric Flow Rate

Figure 6.25 shows the volumetric flow rates computed from the PC-MRI series. The series are almost identical. They all lack the bump we find in the other three volunteers at around 0.6s.

Comparing Mean Velocities

Figure 6.26 shows the computed and measured mean velocities. All four series look very similar, as expected from the similarity of the volumetric flow rates in figure 6.25. The cross section at $y = 355$ consistently report lower velocities, which can be explained by it having a larger cross sectional area. This is confirmed by tables 6.9 and 6.10, which list the differences between the measured and computed velocities for each of the three cross sections.
Figure 6.25: The volumetric flow rate for four PC-MRI series of volunteer 2. The graphs have been phase shifted, so that the time of maximum caudad flow is the same.

**Peak Velocities**

The velocities reported in figure 6.27 are the mean of the 10% largest velocities by magnitude. The peak computed velocities are smaller than the measured ones. The most cephalad cross section, with the largest area, report smaller peak velocities. There is a delay in the cardiac cycle for the computed mean velocities. We see the same here, but the delay is smaller. Both the mean velocities in figure 6.26 and peak velocities in figure 6.27 have similar waveforms to the volumetric flow rates in figure 6.25

<table>
<thead>
<tr>
<th></th>
<th>Ce</th>
<th>Mi</th>
<th>Ca</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$u_{CFD}$</td>
<td>$u_{MRI}$</td>
<td>diff</td>
</tr>
<tr>
<td>S1</td>
<td>34.83</td>
<td>66.40</td>
<td>-90.64</td>
</tr>
<tr>
<td>S2</td>
<td>31.85</td>
<td>65.58</td>
<td>-105.90</td>
</tr>
<tr>
<td>S3</td>
<td>33.34</td>
<td>62.66</td>
<td>-87.93</td>
</tr>
<tr>
<td>S4</td>
<td>34.10</td>
<td>67.63</td>
<td>-98.30</td>
</tr>
</tbody>
</table>

Table 6.9: The range of mean velocities in volunteer 2. The range is computed as the difference between the largest and smallest velocity found during a cardiac cycle. The difference is computed relative to the CFD range.
Figure 6.26: The dotted line is the mean of the measured velocities in volunteer 2. The solid lines is the mean of the sampled CFD velocities at the three different cross sections. The one at $y = 355$ is cephalad.

<table>
<thead>
<tr>
<th>Series</th>
<th>$u_{\text{CFD}}$</th>
<th>$u_{\text{MRI}}$</th>
<th>diff</th>
<th>$u_{\text{CFD}}$</th>
<th>$u_{\text{MRI}}$</th>
<th>diff</th>
<th>$u_{\text{CFD}}$</th>
<th>$u_{\text{MRI}}$</th>
<th>diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>56.06</td>
<td>125.10</td>
<td>-123.14</td>
<td>74.47</td>
<td>125.10</td>
<td>-67.98</td>
<td>76.37</td>
<td>125.10</td>
<td>-63.81</td>
</tr>
<tr>
<td>S2</td>
<td>51.59</td>
<td>131.58</td>
<td>-155.03</td>
<td>68.49</td>
<td>131.58</td>
<td>-92.12</td>
<td>70.00</td>
<td>131.58</td>
<td>-87.9</td>
</tr>
<tr>
<td>S3</td>
<td>54.17</td>
<td>123.03</td>
<td>-127.14</td>
<td>71.36</td>
<td>123.03</td>
<td>-72.43</td>
<td>73.18</td>
<td>123.03</td>
<td>-86.12</td>
</tr>
<tr>
<td>S4</td>
<td>55.52</td>
<td>133.46</td>
<td>-140.38</td>
<td>72.52</td>
<td>133.46</td>
<td>-84.02</td>
<td>73.99</td>
<td>133.46</td>
<td>-80.37</td>
</tr>
</tbody>
</table>

Table 6.10: The range of peak velocities in volunteer 2. The range is computed as the difference between the largest and smallest velocity found during a cardiac cycle. The difference is computed relative to the CFD range.
Figure 6.27: The mean of 10% largest velocities by magnitude for velocities at the three different cross sections in volunteer 2. The one at $y = 355$ is cephalad.
6.3.4 Comparison of Computed and Measured Peak Cephalad Velocity Distributions

PC-MRI Series 1

![Contour plots of PC-MRI series 1 velocities and CFD velocities at three axial cross sections at peak caudad flow in volunteer 2.](image)

Figure 6.28: Contour plots of PC-MRI series 1 velocities and CFD velocities at three axial cross sections at peak caudad flow in volunteer 2.

<table>
<thead>
<tr>
<th></th>
<th>Series 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ce</td>
</tr>
<tr>
<td>T1</td>
<td>0.99</td>
</tr>
<tr>
<td>T2</td>
<td>1.55</td>
</tr>
</tbody>
</table>

Table 6.11: A Table of the L2 norm of the difference between the PC-MRI series and the computed velocities at each of the three cross sections in volunteer 2.
Figure 6.29: Contour plots of PC-MRI series 1 velocities and CFD velocities at three axial cross section at flow reversal in volunteer 2
PC-MRI Series 2

(a) Volunteer 2: PC-MRI series 2 at peak caudad flow

(b) Cephalad cross section

(c) Middle cross section

(d) Caudad cross section

Figure 6.30: Contour plots of PC-MRI series 2 velocities and CFD velocities at three axial cross section at peak caudad flow in volunteer 2.

<table>
<thead>
<tr>
<th></th>
<th>Series 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ce</td>
</tr>
<tr>
<td>T1</td>
<td>1.04</td>
</tr>
<tr>
<td>T2</td>
<td>1.51</td>
</tr>
</tbody>
</table>

Table 6.12: A Table of the L2 norm of the difference between the PC-MRI series and the computed velocities at each of the three cross sections in volunteer 2.
Figure 6.31: Contour plots of PC-MRI series 2 velocities and CFD velocities at three axial cross section at flow reversal in volunteer 2.
PC-MRI Series 3

(a) Volunteer 2: PC-MRI series 3 at peak caudad flow  
(b) Cephalad cross section

(c) Middle cross section  
(d) Caudad cross section

Figure 6.32: Contour plots of PC-MRI series 3 velocities and CFD velocities at three axial cross section at peak caudad flow in volunteer 2.

<table>
<thead>
<tr>
<th></th>
<th>Series 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ce</td>
</tr>
<tr>
<td>T1</td>
<td>1.09</td>
</tr>
<tr>
<td>T2</td>
<td>1.06</td>
</tr>
</tbody>
</table>

Table 6.13: A Table of the L2 norm of the difference between the PC-MRI series and the computed velocities at each of the three cross sections volunteer 2.
Figure 6.33: Contour plots of PC-MRI series 3 velocities and CFD velocities at three axial cross section at flow reversal in volunteer 2.
PC-MRI Series 4

(a) Volunteer 2: PC-MRI series 4 at peak caudad flow

(b) Cephalad cross section

(c) Middle cross section

(d) Caudad cross section

Figure 6.34: Contour plots of PC-MRI series 3 velocities and CFD velocities at three axial cross section at peak caudad flow in volunteer 2.

|  | Series 4 |
|---|---|---|---|
| Ce | Mi | Ca |
| T1 | 1.06 | 0.82 | 0.69 |
| T2 | 1.44 | 1.40 | 1.34 |

Table 6.14: A Table of the L2 norm of the difference between the PC-MRI series and the computed velocities at each of the three cross sections in volunteer 2.

The contours plots in figure 6.28 to 6.35 show the velocity profile at the three axial cross section shown in figure 6.23. The mean and peak velocities from figures 6.26 and 6.27 show that the measured velocities are higher than the computed ones. This is also reflected in the contour plots. At peak caudad flow, we find the greatest velocities in the widest regions in the MRIs, but in the most narrow regions in the CFD cross sections. At flow reversal we see traces of jets in the MRIs. The simulated flows reverses first close to the walls, leading to jet like structures in the widest regions, but without the same
magnitude as in the MRI. Some of the differences evident in the contour plots can be explained by the different waveforms of the mean and peak velocities in figure 6.26 and 6.27. Figure 6.36 shows the computed pressure drops and the PC-MRI pressure gradients. The differences are listed in table 6.15. The differences between computed pressure drop and PC-MRI pressure gradients are about half that of the computed and measured mean velocity relative to the computed mean velocity.

Table 6.11 to 6.14 list the L2 norm of the difference between the computed and measured velocities. In all these tables, the caudad cross section has the lowest difference for peak caudad flow, while the middle cross section has the lowest difference for flow reversal. The most cephalad cross section has consistently the greatest difference.

6.4 Volunteer 3

In the PC-MRI data in this section there is aliasing that has not been filtered away. As a result of this, the colour schemes in the contour plots has been adjusted and does not
span the whole range of velocities reported in the PC-MRI.

### 6.4.1 Cross Sectional Areas and PC-MRI Measurement Locations

The location where the PC-MRI measurements were made is shown in figure 6.37a. Our best estimation of the corresponding location in the MRI used for segmentation is shown in figure 6.37b. The line in figure 6.37a is one vertex cephalad of the measurement location for volunteers 1 and 2. It is comparable the location for volunteer 4, as can be seen in

<table>
<thead>
<tr>
<th>Series</th>
<th>u</th>
<th>PD</th>
<th>CFD</th>
<th>PC-MRI</th>
<th>Diff</th>
<th>PD</th>
<th>CFD</th>
<th>PC-MRI</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>6.29</td>
<td>42.87</td>
<td>66.40</td>
<td>-54.89</td>
<td>8.39</td>
<td>1.10</td>
<td>1.33</td>
<td>-20.87</td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td>5.59</td>
<td>39.22</td>
<td>65.58</td>
<td>-67.20</td>
<td>8.39</td>
<td>0.96</td>
<td>1.25</td>
<td>-29.41</td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>6.29</td>
<td>41.04</td>
<td>62.66</td>
<td>-52.67</td>
<td>7.69</td>
<td>1.04</td>
<td>1.24</td>
<td>-19.20</td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td>6.00</td>
<td>42.00</td>
<td>67.63</td>
<td>-61.02</td>
<td>8.00</td>
<td>0.96</td>
<td>1.23</td>
<td>-27.84</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.15: A table of the range in mean velocities for measured and computed velocities for volunteer 2. The range of u is the difference between maximum and minimum mean velocities found in the cross section at $y = 360$. The pressure drop is the difference of the mean pressure 5 mm cephalad of $y = 360$ and 5 mm caudad of $y = 360$. The range is computed the same way as with the velocities. The phase difference is computed using a cross correlation of the measured and computed velocities.
Figure 6.37: Two images of different MRI scans of volunteer 3. The left has a line showing where the PC-MRI measurements were made. The right image shows an interval with our estimate of the corresponding location. We will compare the measured velocities to CFD velocities sampled from three cross sections in this interval.

6.4.2 Cross Sectional Area

The cross sectional area of the level set for volunteer 3 undulates up to the level of C1/C2, where it begins to increase more rapidly. The cross sectional area of the level set is close to the cross sectional areas of the PC-MRI series in several locations, but none of which are consistent with the image in figure 6.37. The middle of the interval in figure 6.37b is shown as a vertical black line.

<table>
<thead>
<tr>
<th>Series</th>
<th>PC-MRI</th>
<th>Ce</th>
<th>Mi</th>
<th>Ca</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>area</td>
<td>area diff</td>
<td>area diff</td>
<td>area diff</td>
</tr>
<tr>
<td>S1</td>
<td>122.11</td>
<td>171.45</td>
<td>28.78</td>
<td>150.91</td>
</tr>
<tr>
<td>S2</td>
<td>144.37</td>
<td>171.45</td>
<td>15.79</td>
<td>150.91</td>
</tr>
<tr>
<td>S3</td>
<td>142.51</td>
<td>171.45</td>
<td>16.88</td>
<td>150.91</td>
</tr>
<tr>
<td>S4</td>
<td>154.88</td>
<td>171.45</td>
<td>9.66</td>
<td>150.91</td>
</tr>
</tbody>
</table>

Table 6.16: Table of the relative differences in area between the mesh and PC-MRI series of volunteer 3. The differences are computed relative to the cross sections of the mesh.
Figure 6.38: The cross sectional area of the level set for volunteer 3. The horizontal lines are the cross sectional areas of the PC-MRI series. The vertical line denote the vertical coordinate of the centre of the interval shown in figure 6.37b.

Figure 6.39: The volumetric flow rate for four PC-MRI series of volunteer 3. The graphs have been phase shifted, so that the time of maximum caudad flow is the same.
6.4.3 Volumetric Flow Rates

The volumetric flow rates shown in figure 6.39 are all similar, but series 4 is greater in magnitude, and has a shorter period.

Mean Velocities

![Graphs showing mean velocities for Series 1, 2, 3, and 4.]

Figure 6.40: The dotted line is the mean of the measured velocities in volunteer 3. The solid lines is the mean of the sampled CFD velocities at the three different cross sections. The one at $y = 315$ is cephalad.

The mean of the computed and measured velocities are shown in figure 6.40. In all the four series the waveforms agree very well, but the computed magnitudes are a bit smaller. The most cephalad cross section consistently report lower velocities, but this can readily be explained by it larger cross sectional area, as seen in figure 6.38 and table 6.16. We do not see the same variation in the mean velocities as we saw in the volumetric flow rates.
Figure 6.41: The mean of 10% largest velocities by magnitude for velocities at the three different cross sections in volunteer 3. The one at $y = 315$ is cephalad.

**Peak Velocities**

The maximum velocities shown in figure 6.41 differ from each other. In series 1 and 2, the waveforms agree, but around 80% through the cardiac cycle, the measurements and computations show flow in different directions. In series three, the waveforms are very similar, but the magnitudes are different. Series 4 is much more even than the three others. The differences between the mean and peak velocities for volunteer 3 are larger than for the other volunteers, which might be explained by noise in the data. The differences in the range of mean and peak velocities are listed in tables 6.17 and 6.18.
Table 6.17: The range of mean velocities in volunteer 3. The range is computed as the difference between the largest and smallest velocity found during a cardiac cycle. The difference is computed relative to the CFD range.

<table>
<thead>
<tr>
<th></th>
<th>Ce</th>
<th>MRI</th>
<th>diff</th>
<th>Ce</th>
<th>MRI</th>
<th>diff</th>
<th>Ce</th>
<th>MRI</th>
<th>diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>13.52</td>
<td>29.29</td>
<td>-116.71</td>
<td>20.24</td>
<td>29.29</td>
<td>-44.73</td>
<td>20.45</td>
<td>29.29</td>
<td>-43.21</td>
</tr>
<tr>
<td>S2</td>
<td>17.14</td>
<td>31.78</td>
<td>-85.39</td>
<td>25.70</td>
<td>31.78</td>
<td>-23.68</td>
<td>25.98</td>
<td>31.78</td>
<td>-22.34</td>
</tr>
<tr>
<td>S3</td>
<td>16.69</td>
<td>31.09</td>
<td>-86.28</td>
<td>25.02</td>
<td>31.09</td>
<td>-24.28</td>
<td>25.29</td>
<td>31.09</td>
<td>-22.92</td>
</tr>
<tr>
<td>S4</td>
<td>20.52</td>
<td>35.10</td>
<td>-71.06</td>
<td>30.70</td>
<td>35.10</td>
<td>-14.30</td>
<td>31.03</td>
<td>35.10</td>
<td>-13.09</td>
</tr>
</tbody>
</table>

Table 6.18: The range of peak velocities in volunteer 3. The range is computed as the difference between the largest and smallest velocity found during a cardiac cycle. The difference is computed relative to the CFD range.

<table>
<thead>
<tr>
<th></th>
<th>Ce</th>
<th>MRI</th>
<th>diff</th>
<th>Ce</th>
<th>MRI</th>
<th>diff</th>
<th>Ce</th>
<th>MRI</th>
<th>diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2</td>
<td>24.92</td>
<td>56.98</td>
<td>-128.62</td>
<td>32.61</td>
<td>56.98</td>
<td>-74.73</td>
<td>33.18</td>
<td>56.98</td>
<td>-71.73</td>
</tr>
<tr>
<td>S3</td>
<td>24.31</td>
<td>53.87</td>
<td>-121.57</td>
<td>31.79</td>
<td>53.87</td>
<td>-69.44</td>
<td>32.36</td>
<td>53.87</td>
<td>-66.47</td>
</tr>
<tr>
<td>S4</td>
<td>29.81</td>
<td>64.13</td>
<td>-115.12</td>
<td>39.19</td>
<td>64.13</td>
<td>-63.66</td>
<td>39.77</td>
<td>64.13</td>
<td>-61.27</td>
</tr>
</tbody>
</table>
6.4.4 Comparison of Computed and Measured Peak Cephalad Velocity Profiles

PC-MRI Series 1

(a) Volunteer 3: PC-MRI series 1 at peak caudad flow

(b) Cephalad cross section

(c) Middle cross section

(d) Caudad cross section

Figure 6.42: Contour plots of PC-MRI series 1 velocities and CFD velocities at three axial cross section at peak caudad flow in volunteer 3.

<table>
<thead>
<tr>
<th></th>
<th>Series 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce</td>
<td>0.91</td>
</tr>
<tr>
<td>Mi</td>
<td>0.84</td>
</tr>
<tr>
<td>Ca</td>
<td>0.68</td>
</tr>
<tr>
<td>T1</td>
<td>0.85</td>
</tr>
<tr>
<td>T2</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>0.78</td>
</tr>
</tbody>
</table>

Table 6.19: A Table of the L2 norm of the difference between the PC-MRI series and the computed velocities at each of the three cross sections in volunteer 3. The differences are normalized using the computed velocities.
Figure 6.43: Contour plots of PC-MRI series 1 velocities and CFD velocities at three axial cross section at flow reversal in volunteer 3.
PC-MRI Series 2

(a) Volunteer 3: PC-MRI series 2 at peak caudad flow

(b) Cephalad cross section

(c) Middle cross section

(d) Caudad cross section

Figure 6.44: Contour plots of PC-MRI series 2 velocities and CFD velocities at three axial cross section at peak caudad flow in volunteer 3.

<table>
<thead>
<tr>
<th></th>
<th>Series 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ce</td>
<td>Mi</td>
</tr>
<tr>
<td>T1</td>
<td>0.79</td>
<td>0.77</td>
</tr>
<tr>
<td>T2</td>
<td>0.95</td>
<td>0.90</td>
</tr>
</tbody>
</table>

Table 6.20: A Table of the L2 norm of the difference between the PC-MRI series and the computed velocities at each of the three cross sections in volunteer 3. The differences are normalized by the computed velocities.
Figure 6.45: Contour plots of PC-MRI series 2 velocities and CFD velocities at three axial cross section at flow reversal in volunteer 3.
PC-MRI Series 3

(a) Volunteer 3: PC-MRI series 3 at peak caudad flow

(b) Cephalad cross section

(c) Cross section

(d) Caudad cross section

Figure 6.46: Contour plots of PC-MRI series 3 velocities and CFD velocities at three axial cross section at peak caudad flow in volunteer 3.

<table>
<thead>
<tr>
<th>Series 3</th>
<th>Ce</th>
<th>Mi</th>
<th>Ca</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>0.88</td>
<td>0.76</td>
<td>0.58</td>
</tr>
<tr>
<td>T2</td>
<td>1.09</td>
<td>1.11</td>
<td>1.08</td>
</tr>
</tbody>
</table>

Table 6.21: A Table of the L2 norm of the difference between the PC-MRI series and the computed velocities at each of the three cross sections in volunteer 3. The differences are normalized by the computed velocities.
Figure 6.47: Contour plots of PC-MRI series 3 velocities and CFD velocities at three axial cross section at flow reversal in volunteer 3.
Figure 6.48: Contour plots of PC-MRI series 3 velocities and CFD velocities at three axial cross section at peak caudad flow in volunteer 3.

<table>
<thead>
<tr>
<th></th>
<th>Series 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ce</td>
</tr>
<tr>
<td>T1</td>
<td>0.74</td>
</tr>
<tr>
<td>T2</td>
<td>1.24</td>
</tr>
</tbody>
</table>

Table 6.22: The L2 norm of the difference between the PC-MRI series and the computed velocities at each of the three cross sections in volunteer 3. The differences are normalized using the computed velocities.

Figures 6.43, 6.45, 6.47 and 6.49 show the velocity profiles at flow reversal. Due to noise in the MRI, the color scale does not span the whole range of velocities in the PC-MRI plots. As with the maximum caudad flow, the velocity profile is flat. Judging from the slightly lower velocities near the wall, the flow first reverses there.

Figures 6.42, 6.42, 6.42 and 6.48 show the velocity profiles of peak caudad flow. Due to noise in the MRI, it is difficult to judge exactly when the flow is at its maximum. These
Figure 6.49: Contour plots of PC-MRI series 3 velocities and CFD velocities at three axial cross section at flow reversal in volunteer 3.

plots are different from the ones of the other volunteers. The computed velocity profile is flatter, with slightly lower velocities where the cross section is wider. There is some similarity between the geometries, with the two most caudad looking most similar.

Tables 6.19 to 6.22 show the L2 norm of the difference between the measured and simulated velocities. The tables show that the caudad cross section has the lowest error, but at flow reversal it is the most cephalad that has the lowest errors for series 3 and 4. For series 4, all the cross sections show the same numbers.

Figures 6.50 shows a more uplifting picture than the contour plots. Despite the differences in velocity profiles seen in figures 6.42 to 6.49, the computed pressure drop is similar to the PC-MRI pressure gradients, with the relative differences being smaller than the differences between the computed and measured mean velocities relative to the computed mean velocities, as seen in table 6.23.
Figure 6.50: The effect of increased volumetric flow rate on pressure drop in volunteer 3 plotted against mean velocity during one cardiac cycle.

6.5  Healthy Volunteer 4

6.5.1  Cross Sectional Areas and PC-MRI Measurement Locations

The picture of the measurement location shown in figure 6.51 show the same vertebrae as in volunteer 3, namely C1/C2. The cross sectional areas, shown in figure 6.52, are similar, but they do not match in a location consistent with the picture in figure 6.51a. This is the same case as with all the other volunteers.

6.5.2  Cross Sectional Area

The simulated flow is sampled at three axial cross sections in the interval shown in figure 6.51b. The middle of this interval is denoted by a black dotted line in figure 6.52. The cross sectional area increases steeply moving cephalad. The discrepancies in area are listed in table 6.24.

6.5.3  Volumetric Flow Rate

The volumetric flow rates used as boundary conditions in the simulations are graphed in figure 6.53 on page 96. The graphs has been phase shifted so they align with the time of maximum caudad flow. The different series are similar in both magnitude and duration.
<table>
<thead>
<tr>
<th>Series</th>
<th>PD</th>
<th>CFD</th>
<th>PC-MRI</th>
<th>Diff</th>
<th>PD</th>
<th>CFD</th>
<th>PC-MRI</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>6.54</td>
<td>20.24</td>
<td>29.29</td>
<td>-44.73</td>
<td>8.50</td>
<td>0.31</td>
<td>0.45</td>
<td>-43.23</td>
</tr>
<tr>
<td>S2</td>
<td>6.08</td>
<td>25.70</td>
<td>31.78</td>
<td>-23.68</td>
<td>7.43</td>
<td>0.50</td>
<td>0.57</td>
<td>-12.15</td>
</tr>
<tr>
<td>S3</td>
<td>6.08</td>
<td>25.02</td>
<td>31.09</td>
<td>-24.28</td>
<td>7.43</td>
<td>0.54</td>
<td>0.65</td>
<td>-19.53</td>
</tr>
<tr>
<td>S4</td>
<td>8.70</td>
<td>30.70</td>
<td>35.10</td>
<td>-14.30</td>
<td>10.87</td>
<td>0.62</td>
<td>0.67</td>
<td>-8.30</td>
</tr>
</tbody>
</table>

Table 6.23: A table of the range of mean velocities for measured and computed velocities for volunteer 3. The range of $u$ is the difference between maximum and minimum mean velocities found in the cross section at $y = 320$. The pressure drop is the difference of the mean pressure 5 mm cephalad of $y = 320$ and 5 mm caudad of $y = 320$. The range is computed the same way as with the velocities. The phase difference is computed using a cross correlation of the measured and computed velocities.

(a) An image of the PC-MRI measurement location  
(b) An interval containing the estimated corresponding CFD sampling location

Figure 6.51: Two images of different MRI scans of volunteer 4. The left has a line showing where the PC-MRI measurements were made. The right image shows an interval with our estimate of the corresponding location. We will compare the measured velocities to CFD velocities sampled from three cross sections in this interval.

Mean Velocities

The computed mean velocities in figure 6.54 vary from cross section to cross section. The most caudad reports mean velocities very close to the measured ones, while the cephalad cross section, which has greater cross sectional area, as seen in figure 6.52.
Figure 6.52: The cross sectional area of the level set. The horizontal lines are the cross sectional areas of the PC-MRI series. The vertical line denote the vertical coordinate of the centre of the interval shown in figure 6.51b.

Table 6.24: Table of the relative differences in area between the mesh and PC-MRI series of volunteer 4. The differences are computed relative to the cross sections of the mesh.

<table>
<thead>
<tr>
<th>Series</th>
<th>PC-MRI area</th>
<th>Ce area</th>
<th>diff</th>
<th>Mi area</th>
<th>diff</th>
<th>Ca area</th>
<th>diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>170.02</td>
<td>259.12</td>
<td>34.39</td>
<td>184.13</td>
<td>7.66</td>
<td>171.06</td>
<td>0.61</td>
</tr>
<tr>
<td>S2</td>
<td>140.35</td>
<td>259.12</td>
<td>45.84</td>
<td>184.13</td>
<td>23.78</td>
<td>171.06</td>
<td>17.95</td>
</tr>
<tr>
<td>S3</td>
<td>139.11</td>
<td>259.12</td>
<td>46.32</td>
<td>184.13</td>
<td>24.45</td>
<td>171.06</td>
<td>18.68</td>
</tr>
<tr>
<td>S4</td>
<td>153.64</td>
<td>259.12</td>
<td>40.71</td>
<td>184.13</td>
<td>16.56</td>
<td>171.06</td>
<td>10.18</td>
</tr>
</tbody>
</table>

Peakvelocities

The peak computed velocities in the caudad cross section follow the measured ones as with the mean velocities, except for the bump at around 80% through the cardiac cycle. As with the mean velocities, the cephalad cross section reports lower velocities. The differences between the computed and measured mean velocities and the computed and measured peak velocities relative to the computed velocities are listed in table 6.25 and 6.26.
Figure 6.53: The volumetric flow rate for four PC-MRI series of volunteer 4. The graphs have been phase shifted, so that the time of maximum caudad flow is the same.

Table 6.25: The range of mean velocities in volunteer 4. The range is computed as the difference between the largest and smallest velocity found during a cardiac cycle. The difference is computed relative to the CFD range.

<table>
<thead>
<tr>
<th></th>
<th>Ce</th>
<th>u_{CFD}</th>
<th>u_{MRI}</th>
<th>diff</th>
<th>Mi</th>
<th>u_{CFD}</th>
<th>u_{MRI}</th>
<th>diff</th>
<th>Ca</th>
<th>u_{CFD}</th>
<th>u_{MRI}</th>
<th>diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>S3</td>
<td></td>
<td>15.63</td>
<td>29.77</td>
<td>-90.47</td>
<td></td>
<td>22.63</td>
<td>29.77</td>
<td>-31.53</td>
<td></td>
<td>28.29</td>
<td>29.77</td>
<td>-5.24</td>
</tr>
</tbody>
</table>

Table 6.26: The range of peak velocities in volunteer 4. The range is computed as the difference between the largest and smallest velocity found during a cardiac cycle. The difference is computed relative to the CFD range.

<table>
<thead>
<tr>
<th></th>
<th>Ce</th>
<th>u_{CFD}</th>
<th>u_{MRI}</th>
<th>diff</th>
<th>Mi</th>
<th>u_{CFD}</th>
<th>u_{MRI}</th>
<th>diff</th>
<th>Ca</th>
<th>u_{CFD}</th>
<th>u_{MRI}</th>
<th>diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td></td>
<td>24.92</td>
<td>41.19</td>
<td>-65.24</td>
<td></td>
<td>32.61</td>
<td>41.19</td>
<td>-26.30</td>
<td></td>
<td>44.89</td>
<td>41.19</td>
<td>8.24</td>
</tr>
<tr>
<td>S2</td>
<td></td>
<td>22.48</td>
<td>44.26</td>
<td>-96.91</td>
<td></td>
<td>29.36</td>
<td>44.26</td>
<td>-15.01</td>
<td></td>
<td>39.51</td>
<td>44.26</td>
<td>-12.01</td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td>23.55</td>
<td>46.75</td>
<td>-98.57</td>
<td></td>
<td>30.78</td>
<td>46.75</td>
<td>-27.98</td>
<td></td>
<td>41.06</td>
<td>46.75</td>
<td>-13.86</td>
</tr>
<tr>
<td>S4</td>
<td></td>
<td>20.50</td>
<td>37.13</td>
<td>-81.14</td>
<td></td>
<td>26.73</td>
<td>37.13</td>
<td>-8.41</td>
<td></td>
<td>37.29</td>
<td>37.13</td>
<td>0.44</td>
</tr>
</tbody>
</table>
Figure 6.54: The dotted line is the mean of the measured velocities in volunteer 4. The solid lines is the mean of the sampled CFD velocities at the three different cross sections. The one at \( y = 305 \) is cephalad.
Figure 6.55: The mean of 10% largest velocities by magnitude for velocities at the three different cross sections in volunteer 4. The one at $y = 305$ is cephalad.
6.5.4 Comparison of Computed and Measured Peak Cephalad Velocity Profiles

Series 1, Maximum Caudad Flow

Figure 6.56: Contour plots of PC-MRI series 1 velocities and CFD velocities at three axial cross section at peak caudad flow in volunteer 4.

Series 1, Flow Reversal

<table>
<thead>
<tr>
<th></th>
<th>Ce</th>
<th>Mi</th>
<th>Ca</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>0.67</td>
<td>0.51</td>
<td>0.52</td>
</tr>
<tr>
<td>T2</td>
<td>1.50</td>
<td>1.50</td>
<td>1.46</td>
</tr>
</tbody>
</table>

Table 6.27: A Table of the L2 norm of the difference between the PC-MRI series and the computed velocities at each of the three cross sections in volunteer 3. The differences are normalized using the computed velocities.
Figure 6.57: Contour plots of PC-MRI series 1 velocities and CFD velocities at three axial cross section at flow reversal in volunteer 4
Series 2, Peak Caudad Flow

(a) Volunteer 4: PC-MRI series 2 at peak caudad flow

(b) Cephalad cross section

(c) Middle cross section

(d) Caudad cross section

Figure 6.58: Contour plots of PC-MRI series 2 velocities and CFD velocities at three axial cross section at peak caudad flow in volunteer 4.

Series 2, Flow Reversal

<table>
<thead>
<tr>
<th></th>
<th>Ce</th>
<th>Mi</th>
<th>Ca</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>0.80</td>
<td>0.59</td>
<td>0.64</td>
</tr>
<tr>
<td>T2</td>
<td>1.62</td>
<td>1.61</td>
<td>1.61</td>
</tr>
</tbody>
</table>

Table 6.28: A Table of the L2 norm of the difference between the PC-MRI series and the computed velocities at each of the three cross sections in volunteer 4. The difference are normalized using the computed velocities.
Figure 6.59: Contour plots of PC-MRI series 2 velocities and CFD velocities at three axial cross section at flow reversal in volunteer 4.
Series 3, Maximum Caudad Flow

Figure 6.60: Contour plots of PC-MRI series 3 velocities and CFD velocities at three axial cross section at peak caudad flow in volunteer 4.

Series 3, Flow Reversal

In the figures previously presented, the time match. From the plot of maximum and mean velocity in figures 6.55 on page 98 and 6.54 on page 97, we see that there is a certain lag between the time of maximum caudad flow in the PC-MRI and the CFD velocities. The contour plots in 6.60 have been selected based on the time of maximum caudad flow of the most caudad cross section. The difference in time is 0.0675s. The difference in time is 0.0675s.
Figure 6.61: Contour plots of PC-MRI series 3 velocities and CFD velocities at three axial cross section at flow reversal in volunteer 4.

Table 6.29: A Table of the L2 norm of the difference between the PC-MRI series and the computed velocities at each of the three cross sections in volunteer 4. The differences are normalized using the computed velocities.
Series 4, Maximum Caudad Flow

(a) Volunteer 3: PC-MRI series 4 at peak caudad flow

(b) Cephalad cross section

(c) Middle cross section

(d) Caudad cross section

Figure 6.62: Contour plots of PC-MRI series 3 velocities and CFD velocities at three axial cross section at peak caudad flow in volunteer 4.

Series 4, Flow Reversal

<table>
<thead>
<tr>
<th></th>
<th>Series 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce</td>
<td>Mi</td>
</tr>
<tr>
<td>T1</td>
<td>0.78</td>
</tr>
<tr>
<td>T2</td>
<td>1.36</td>
</tr>
</tbody>
</table>

Table 6.30: A Table of the L2 norm of the difference between the PC-MRI series and the computed velocities at each of the three cross sections in volunteer 4.

Figures 6.56, 6.56, 6.56 and 6.62 show the velocity profiles at flow reversal. As with the other volunteers, the CFD cross sections report the greatest velocities where the cross section is narrowest. The PC-MRI contour plots show the opposite.
Figures 6.57, 6.59, 6.61 and 6.63 are contour plots of the flow reversing. The peak and mean velocities are similar, as testified by figures 6.54 and 6.55, but the MRI contour plots show a greater degree of variability, while the CFD velocity profiles are more plug like. The differences between the computed pressure drop and PC-MRI pressure gradients are shown in figure 6.64. These differences are listed in table 6.31. While the computed mean velocities for volunteer 4 are more similar to the measurements than for the other volunteers, the differences between the computed pressure drop and PC-MRI pressure gradients are about the same. Tables 6.27 to 6.30 show the L2 norm of the difference between the measured and computed velocities. From these tables we see no great difference between the different cross sections.
Figure 6.64: The effect of increased volumetric flow rate on pressure drop in volunteer 4 plotted against mean velocity during one cardiac cycle.

<table>
<thead>
<tr>
<th></th>
<th>PD</th>
<th>CFD</th>
<th>PC-MRI</th>
<th>Diff</th>
<th>PD</th>
<th>CFD</th>
<th>PC-MRI</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>6.25</td>
<td>24.27</td>
<td>26.83</td>
<td>-10.54</td>
<td>7.64</td>
<td>0.37</td>
<td>0.41</td>
<td>-9.95</td>
</tr>
<tr>
<td>S2</td>
<td>6.38</td>
<td>21.44</td>
<td>28.57</td>
<td>-33.25</td>
<td>7.80</td>
<td>0.35</td>
<td>0.43</td>
<td>-23.32</td>
</tr>
<tr>
<td>S3</td>
<td>5.56</td>
<td>22.63</td>
<td>29.77</td>
<td>-31.53</td>
<td>4.86</td>
<td>0.42</td>
<td>0.51</td>
<td>-22.41</td>
</tr>
<tr>
<td>S4</td>
<td>6.34</td>
<td>20.01</td>
<td>24.12</td>
<td>-20.57</td>
<td>8.45</td>
<td>0.32</td>
<td>0.36</td>
<td>-12.59</td>
</tr>
</tbody>
</table>

Table 6.31: The range of mean velocities for measured and computed velocities for volunteer 4. The range of u is the difference between maximum and minimum mean velocities found in the cross section at y = 300. The pressure drop is the difference of the mean pressure 5 mm cephalad of y = 300 and 5 mm caudal of y = 300. The range is computed the same way as with the velocities. The phase difference is computed using a cross correlation of the measured and computed velocities.
Chapter 7

Discussion

7.1 Summary

In this thesis we have simulated CSF flow in patient specific geometries and compared the results to PC-MRI measurements. Simulations were conducted on computational geometries of four healthy volunteers, where each geometry was segmented from an MRI scan. We have four PC-MRI measurement series of each volunteer, and one simulation was conducted for each series. Bulk velocity profiles were used as boundary conditions, where the volumetric flow rate was scaled to match the volumetric flow rate of the PC-MRI series. The results were compared to the measurements by calculating mean and peak velocities. Finally, the relationship between the pressure drop and mean velocity was investigated. We have no measurements of the pressure, therefore pressure gradients were computed from the measured velocities and used for comparisons.

The first step was to segment surfaces from the MRI scan of each of the four volunteers using VMTK. The surfaces were then smoothed and remeshed before mesh generation. The surface smoothing required changes to mshr, and the meshes were generated using VMTK. The CSF was modelled using Navier-Stokes equations for incompressible fluids, and simulations were carried out using a solver from cbcflow, which is built on FEniCS. Scripts for visualization and comparisons were developed as part of this thesis using the python interface to Paraview and the plotting library matplotlib.

A lot of effort went into the mesh generation and especially trying to integrate mshr and VMTK, as well as a method for geometric extension. We wanted to generate the mesh in mshr after remeshing in VMTK. In the end, surface smoothing was done in mshr, and surface remeshing and mesh generation was accomplished with VMTK.

7.2 Results

The vertical velocity component of the computed and measured velocities were compared in axial cross sections at peak cephalad flow, peak caudal flow and flow reversal using
contour plots. The contour plots show marked differences between the CFD and PC-MRI cross sections. The computed velocity field is more homogeneous than that found in the PC-MRI measurements. Furthermore, the largest computed velocities are found in a different area of the cross section than the measured ones. The PC-MRI reports largest magnitudes where the SAS is widest. This is also the case for the computed velocities, but only at peak cephalad flow. At peak caudad flow and flow reversal, the computed velocities are largest where the SAS is narrowest.

The L2 norm of the difference between the velocities in the different cross sections and the PC-MRI measurements takes both the geometry and the magnitude of the velocity into account. Even though it is difficult to differentiate between the influence of the geometry and velocity, this L2 norm nonetheless serves to distinguish the different cross sections. The differences range from 0.9 to 1.7 between peak cephalad flow and flow reversal depending on cross section. Between cross sections, the differences typically range from 0.7 to 1.0. The L2 norms are lower for peak caudad flow than for flow reversal. This can be explained by the phase lag between the computed and measured velocities since the velocities vary more rapidly in time at flow reversal. This is also shown in the plots of mean velocity. Cross sections with more similar mean velocities also have a lower L2 difference.

The cross sections where the PC-MRI measurements were made were identified visually from images. As a result, there is uncertainty around the exact locations. To account for this, three cross sections were selected, each 5 mm apart. The cross sections from the mesh varied from 110.47 mm$^2$ to 259.12 mm$^2$. The cross sectional areas computed from the PC-MRI series ranged from being 26.94% smaller, relative to the mesh, to 50.5% greater, relative to the mesh.

Mean computed velocities range from around 10 mm/s cephalad to −20 mm/s caudad, where cephalad is positive. The measured velocities tend to be 50% larger in magnitude. These differences in velocity might be connected to the cross sectional area. Where the difference in cross sectional area is greater, the greater the difference in mean velocity. This does not mean that the differences in velocity would vanish if the cross sectional areas were equal. The most caudad cross sections in volunteers 3 and 4, have almost equal cross sectional area as PC-MRI series 1, but the measured mean velocity is 40% greater than the computed mean velocity and 10% lower than the computed mean velocity respectively.

Looking at velocities in each of the three cross sections of the mesh, the ones with largest velocities tend to have smallest cross sectional areas. However, this correlation is far from perfect. Examples are the middle and caudad cross sections of volunteers 2 and 3. In both cases, the cross sectional areas are different, but the mean velocities are similar.

The peak velocities display the same pattern as the mean velocities. The peak velocities are generally highest where the mean velocities are highest. However, the difference between the computed and measured velocities as a percentage of computed velocities are higher when we compare peak velocities than when we compare mean velocities.

The waveforms of the computed pressure drop and the pressure gradient from the PC-MRI are similar and comparable to Alperin et al. (2000) [2]. That is also the case with the mean velocities. We find the same rise in pressure drop right before the velocity
reverses. There is a phase difference between the measured and computed quantities. The phase difference is about 6 percent of a cardiac cycle for the mean velocities, and 1 or 2 percentage points more for the pressure. In both cases, the computed quantities lags behind the PC-MRI series.

Small blood vessels and branching nerve roots were ignored in the segmentation process. We do not know what effect they have on the CSF flow field, but they are to small to be captured by the MRI scans. Moreover, the SAS walls and the spinal cord are assumed rigid. This should not have a significant effect on the CSF pressure. When comparing velocities with the PC-MRI measurements, there was some uncertainty concerning the exact location of the measurements. This was addressed by comparing velocities at three different levels. However, the relationship is not exact, although increased cross sectional area seems to be connected to lower velocities. Only horizontal cross sections were considered. Velocities from angled cross section might prove to be more similar to the PC-MRI measurements. To better capture boundary layer effects, a boundary layer could have been added to the mesh. However, this should not impact the solution further away from the wall, and thus have a limited effect on our comparisons. We do see, however, differences between the CFD and PC-MRI in locations where the largest velocities occur.

When discretizing Navier-Stokes equations, errors are introduced. Time stepping and mesh convergence was investigated. The time stepping proved to have little effect, but a finer mesh showed higher peak velocities. However, the mean of the 10% largest velocities by magnitude were similar. The use of bulk velocity profile has an effect on the results close to the boundary. The effects of this were investigated but found to have little impact on the flow field further away from the boundary.

The geometrical differences between our computations and the PC-MRI are considerable. Both the shape and the cross sectional areas were different. Despite this, the computed mean velocities are promising. Cross sections of the mesh with more similar shapes tend to have more similar cross sectional areas. Larger cross sectional area is connected to both smaller mean and peak velocities. 16 PC-MRI series from four healthy volunteers were compared and the differences between the computed pressure drop and the PC-MRI pressure gradients relative to the computed pressure drop is smaller than the difference between computed mean velocities and measured mean velocities relative to the computed velocities. Despite the differences in velocity, CFD can be a way to compute the pressure drop.

Expanding the model in the current study might shed light on the different flow patterns with higher anterior velocities than observed with PC-MRI. Further investigations into the reasons behind the differences in cross sectional area might vastly improve the comparisons of CFD and PC-MRI velocities in this study.
Appendix A

Source Code

In this appendix, source code for writing Paraview slice filter output to csv files and computing the contour plots are listed. The source code for creating the tables is also listed. The modifications to mshr is found at the end.

```python
1 import os
2 import numpy as np
3 import matplotlib.pyplot as mpl
4 import sys
5 from scipy.integrate import simps
6 import scipy.interpolate as scint
7 from mpl_toolkits.mplot3d import Axes3D
8 import matplotlib.colors as colors
9 from matplotlib.ticker import LogFormatter
10 import scipy.signal as scig
11 import matplotlib
12 from CSVWriter import VelCSVWriter, PreCSVWriter
13
15 class VelCSV:  # {{
17     def __init__(self, dirpath, prefix):  # {{
18         self.dirpath = dirpath
19         self.prefix = prefix
20         # }}
21
23 def _get_v(self, timestep):  # {{
24     self.vlist = {}
25     if timestep not in self.vlist:
26         fpath = "%s/%s.%d.csv" % (self.dirpath,
27                                      self.prefix, timestep)
28         c = np.loadtxt(fpath, skiprows=1,
29                         delimiter="", unpack = True)
30         v0, v1, v2, x, y, z = c
31         self.vlist[timestep] = [x, y, z, v1]
32         return [x, y, z, v1]
33     else:
34         return self.vlist[timestep]
```
def flux(self, timestep, n=100):
    # {{
    x, y, z, v1 = self._get_v(timestep)
    print "timestep: \%f" % (timestep)
    xi = np.linspace(np.min(x), np.max(x), n)
    zi = np.linspace(np.min(z), np.max(z), n)

    grid_x, grid_z = np.mgrid[np.min(x):np.max(x):n*1j,
        np.min(z):np.max(z):n*1j]
    f = scint.griddata((x, z), v1, (grid_x, grid_z),
        fill_value=0)

    return simps(simps(f, xi), zi)
    # }}

def surfplot(self, timestep, n=10, params=None):
    # {{
    x, y, z, v1 = self._get_v(timestep)
    xi = np.linspace(np.min(x), np.max(x), n)
    zi = np.linspace(np.min(z), np.max(z), n)

    grid_x, grid_z = np.mgrid[np.min(x):np.max(x):n*1j, np.min(z):np.max(z):n*1j]
    f = scint.griddata((x, z), v1, (grid_x, grid_z), fill_value=0)

    fig = mpl.figure()
    ax = fig.add_subplot(111, projection='3d')
    ax.plot_surface(grid_x, grid_z, f, rstride=1, cstride=1, linewidth=0, cmap=mpl.cm.hot)
    ax.set_zlim([0, 80])
    ax.view_init(elev=50., azim=-50)
    mpl.show()
    return ax
    # }}

def contour(self, timestep, n=10, levels=None, odir="contours",
    cmap=None, get_bounds=False, ycor=0):
    # {{
    print "Plotting contourf for time %d" % timestep
    x, y, z, v1 = self._get_v(timestep)
    if get_bounds:
        return np.min(-v1), np.max(-v1)

    if cmap is None:
        cmap = "seismic"

    grid_x, grid_z = np.mgrid[np.min(x):np.max(x):n*1j, np.min(z):np.max(z):n*1j]
    v1 = -scint.griddata((x, z), v1, (grid_x, grid_z), fill_value=0)
    v1[v1 == 0] = 1e9

    fig = mpl.figure()
    ax = fig.add_subplot(111)
if levels is None:
    levels = np.linspace(-90, 90, 181)
CS = ax.contourf(grid_x, grid_z, v1, cmap=cmap, levels=levels)
dt = 0.036058
ax.set_title(r"Computed Velocity: level: %d, time: %f" % (ycor, timestep*dt/5), fontsize=22)
ax.set_xlabel(r"$(mm)$", fontsize=22)
ax.set_ylabel(r"$(mm)$", fontsize=22)
fig.colorbar(CS)
cb.ax.tick_params(labelsize=20)
fig.savefig("%s/jnormal_comp_contourf_%02d.png" % (odir, timestep))
#mpl.show()
mpl.close(fig)
# }}}
def maxvel_slice(self, timestep, perc=None):
    # {{
    x, y, z, v1 = self._get_v(timestep)
    if perc == 100:
        return v1.mean()
    if perc is not None:
        perc = perc*v1.size
        maxind = np.argpartition(-np.abs(v1), perc)[:perc]
        return v1[maxind]
    else:
        maxind = np.argmax(abs(v1))
        return v1[maxind]
    # }}
def maxvel(self, caudaly, timestep, perc=None, where=False):
    # {{
    x, y, z, v1 = self._get_v(timestep)
    ind = y < caudaly
    a = v1[ind]
    if perc is not None:
        perc = perc*a.size
        maxind = np.argpartition(-np.abs(a), perc)[:perc]
        res = a[maxind]
    else:
        maxind = np.argmax(abs(a))
        res = a[maxind]
    if where:
        return res, y[ind][maxind]  # remember, we sliced caudally
    return res
    # }}
def meanvel(self, caudaly, timestep):
    # {{
    x, y, z, v1 = self._get_v(timestep)
    ind = y < caudaly
a = v1[ind]

return np.mean(a)
# }}}
# }}}

class PreCSV:  
# {{{
    def __init__(self, dirpath, prefix):  # {{
        self.dirpath = dirpath
        self.prefix = prefix
        # }}

    def _get_p(self, timestep):  # {{
        self.plist = {}
        if timestep not in self.plist:
            fpath = "%s/%s.%d.csv" % (self.dirpath, self.prefix, timestep)
            c = np.loadtxt(fpath, skiprows=1, delimiter=",", unpack = True)
            p, x, y, z = c
            self.plist[timestep] = [x, y, z, p]
            return [x, y, z, p]
        else:
            return self.plist[timestep]
        # }}

    def dp(self, taily, heady, timestep, eps=3.5):  # {{
        x, y, z, p = self._get_p(timestep)

        # TODO: Do I need y, other than for indices?

        head_ind = np.logical_and(y > heady - eps, y < heady + eps)
        head_p = p[head_ind]
        head_y = y[head_ind]

        tail_ind = np.logical_and(y > taily - eps, y < taily + eps)
        tail_p = p[tail_ind]
        tail_y = y[tail_ind]

        return head_p.mean()- tail_p.mean()
        # }}

    def dp_range(self, taily, heady, timestep):
        dp = []
        try:
            for i in timestep:
                delta = self.dp(taily, heady, i)
                dp.append(delta)
                print "calculating timestep %d, dp = %f" % (i, delta)
        except TypeError:
            for i in xrange(timestep):
                delta = self.dp(taily, heady, i)
                dp.append(delta)
                print "calculating timestep %d, dp = %f" % (i, delta)
```python
import numpy as np
import scipy.interpolate as scint
import sys
import matplotlib.pyplot as mpl

def make_grid(cx, cz, mx, mz, n=1000):
    """Return common grid for MRI and CFD""
    x = np.concatenate((cx, mx))
    z = np.concatenate((cz, mz))
    return np.mgrid[np.min(x):np.max(x):n*1j,
                     np.min(z):np.max(z):n*1j]

def l2(mri_path, cfdpath):
    c = np.loadtxt(cfdpath, skiprows=1,
                   delimiter=',,', unpack=True)
    v0, v1, v2, cx, cy, cz = c
    cx -= np.min(cx)  # translate to (0, 0)
    cz -= np.min(cz)
    mri = np.loadtxt(mri_path, delimiter=',")*10
    mri = mri[~(mri == 0).all(1)]
    mri = mri.T[~(mri == 0).all(0)].T
    dx_mri = 0.556  # mri pixel scale factor
    # MRI grid
    x = np.arange(mri.shape[0])*dx_mri
    z = (np.arange(mri.shape[1]))*dx_mri
    grid_x, grid_z = make_grid(cx, cz, mx, mz, n=1000)
    # interpolate irregular CFD data to common regular grid
    cfd = -scint.griddata((cx, cz), v1, (grid_x, grid_z),
```
fill_value=0)

# Create spline representation of
# MRI data interpolate to common grid
znew = scint.interp2d(mx, mz, mri.T, fill_value=0)
data = np.fliplr(znew(grid_x[:, 0], grid_z[0, :]))

# compute L2 norm
x = grid_x[:, 0]
z = grid_z[0, :]
return np.sqrt(np.trapz(np.trapz(np.power(cfd - data, 2)
    , x, axis=1), z))

#levels = np.linspace(-13, 13, 27)
#mpl.contourf(grid_x, grid_z, cfd - data, levels=levels, cmap="seismic")
#mpl.show()

def interpolate(fpath, n=1000):
c = np.loadtxt(fpath, skiprows=1, delimiter=",", unpack=True)
v0, v1, v2, x, y, z = c
grid_x, grid_z = np.mgrid[np.min(x):np.max(x):n*1j, np.min(z):np.max(z):n*1j]
v1 = -scint.griddata((x, z), v1, (grid_x, grid_z), fill_value=np.nan)
return v1[-np.isnan(v1)]

def table(list_of_data):
    table = []
    for v1 in list_of_data:
        perc = 0.1*v1.size
        max = v1[np.argpartition(-np.abs(v1), perc)[:perc]].mean()
        table.append([max, v1.mean(), v1.min()])
    table = ["V_{max}\)", "V_{mean}\)", "V_{min}\)"] + table
    table = map(list, zip(*table))
    print r"\begin{table}[h!]
    
    \begin{tabular}{c || c c c c}
    
    \hline
    --- & MRI & Cephalad & Middle & Caudad \\
    \hline
    for row in table:
        print r" %s & %.2f & %.2f & %.2f & %.2f \\
    % tuple(row)
    \hline
    \end{tabular}
    \end{table}"

def l2_table(times, ycors = [326, 331, 336]):
data = []
    for t in times:
        mri_path = "flux_matrices/Jakob04time%d.dat" % t
        for y in ycors:
            data.append(l2(mri_path, "velocity_csv_%d/velocity.%d.csv" % (y, t*5)))
```python
    #norms = [l2(mri_path, "velocity_csv_%d/velocity.%d.csv" % (y, time*5)) for y in ycors]
    return data

def create_list_of_data(time, ycors=[326, 331, 336]):
    mri = np.loadtxt("flux_matrices/Jakob04time%d.dat" % time, delimiter=",")
    mri = mri[~(mri == 0)]*10  # convert to mm/s
    data = [mri.flatten()]
    for f in ["velocity_csv_%d/velocity.%d.csv" % (y, time*5) for y in ycors]:
        data.append(interpolate(f).flatten())
    return data

if __name__ == "__main__":
    data = l2_table([12, 21, 19])
    for thing in data:
        print "%.2f & " % thing,
    #table(create_list_of_data(12))
```

```c++
#include <iostream>
#include <dolfin/geometry/Point.h>
#include <array>
#include <vector>
#include <numeric>
#include <math.h>

typedef std::pair<dolfin::Point, dolfin::Point> Segment;
typedef std::array<dolfin::Point, 3> Triangle;

class AdjustPlane // {{
{
    public:
        dolfin::Point adjust_plane(
            const std::vector<std::pair<Segment, Triangle>>& polygon,
            dolfin::Point p,
            dolfin::Point n)
        {
            dolfin::Point new_n(0.0, 0.0, 0.0);
            dolfin::Point v1;  // v1, v2 span the triangle
            dolfin::Point v2;
            double proj_v1, proj_v2, v1_length, v2_length;
            unsigned long int size = polygon.size();
            for (size_t i=0; i < size; ++i)
            {
                Triangle t = polygon[i].second;
                v1 = t[2] - t[0];
                v2 = t[1] - t[0];
```

---

117
proj_v1 = std::inner_product(n.coordinates(), n.coordinates() + 3, v1.coordinates(), 0.0);

proj_v2 = std::inner_product(n.coordinates(), n.coordinates(), v2.coordinates(), 0.0);

v1_length = std::inner_product(v1.coordinates(), v1.coordinates() + 3, v1.coordinates(), 0.0);

v2_length = std::inner_product(v1.coordinates(), v1.coordinates() + 3, v1.coordinates(), 0.0);

// Project n into the plane spanned by v1, v2, then add all the projections.
new_n += v1*proj_v1/v1_length + v2*proj_v2/v2_length;

double n_length = std::inner_product(new_n.coordinates(), new_n.coordinates() + 3, new_n.coordinates(), 0.0);

double parallel = std::inner_product(n.coordinates(), n.coordinates() + 3, new_n.coordinates(), 0.0);

if (std::abs(parallel) < 1e-5)
{
    std::cout << "The vectors are parallel";
    std::cout << "Say we’re happy with the new n and abort the iteration?";
    return n;
}

// alpha <= 1
new_n /= n_length;
double alpha = 0.5;
n += alpha*(new_n - n);

// Normalize n
n_length = std::inner_product(n.coordinates(), n.coordinates() + 3, n.coordinates(), 0.0);

return n/sqrt(n_length);
}; //}}}

void CSGCGALDomain3D::save_vtp(std::string filename) const
#ifdef MSHR_HAS_VTK

std::string message = "Writing to file: " + filename;
log(dolfin::TRACE, message);

std::map<Exact_Polyhedron_3::Vertex_const_handle, std::size_t> vertex_map;
std::vector<std::array<double, 3> > vertices;

std::vector<std::array<std::size_t, 3> > facets;

int vertex_counter = 0;
double tmp_vertex[3];

vtkPoints* points = vtkPoints::New();
vtkCellArray* vtk_facets = vtkCellArray::New();

for (Exact_Polyhedron_3::Vertex_const_iterator vi = impl->p.vertices_begin(), end = impl->p.vertices_end(); vi != end; ++vi)
{
    vertex_map[vi] = vertex_counter;
    tmp_vertex[0] = CGAL::to_double(vi->point().x());
    tmp_vertex[1] = CGAL::to_double(vi->point().y());
    tmp_vertex[2] = CGAL::to_double(vi->point().z());
    points->InsertNextPoint(tmp_vertex);
    vertex_counter++;
}

for (Exact_Polyhedron_3::Facet_const_iterator fi = impl->p.facets_begin(), end = impl->p.facets_end(); fi != end; ++fi)
{
    Exact_Polyhedron_3::Halfedge_const_iterator h = fi->halfedge();
    vtkPolygon* poly = vtkPolygon::New();
poly->GetPointIds()->SetNumberOfIds(3);
poly->GetPointIds()->SetId(0, vertex_map[h->vertex()]);
poly->GetPointIds()->SetId(1, vertex_map[h->next()->vertex()]);
poly->GetPointIds()->SetId(2, vertex_map[h->next()->next()->vertex()]);
    vtk_facets->InsertNextCell(poly);
}

vtkPolyData* polydata = vtkPolyData::New();
polydata->SetPoints(points);
polydata->SetPolys(vtk_facets);
vtkXMLPolyDataWriter* writer = vtkXMLPolyDataWriter::New();
writer->SetFileName(filename.c_str());
writer->SetInput(polydata);
writer->Write();

#else

dolfin::dolfin_error("CSGCGALDomain3D.cpp",
  "writing VTP file",
  "mshr is not built with VTK support");

#endif
}
## List of Figures

1.1 A sagittal view of the human brain. .................................. 8
2.1 Vertebrae numbers 3, 4 and 5. ........................................... 13
2.2 The spinal canal with the spinal cord and nerve roots [27]. .... 14
3.1 Three cross sections of an MRI scan ................................. 16
3.2 Illustration of the colliding fronts method of segmentation ...... 17
3.3 Segmentation of low resolutions areas ............................... 18
3.4 Illustration of the use of manual segmentation .................... 18
3.5 Deformation of the level set ........................................... 19
3.6 Illustration of halfedges ............................................... 22
3.7 An illustration of geometric extensions .............................. 23
3.8 Mesh resolution problems in mshr ................................... 24
3.9 Polygonal surfaces after various degrees of smoothing ........... 26
3.10 An illustration of surface clipping in VMTK ....................... 27
3.11 Illustration of resolution control in VMTK ......................... 29
4.1 A picture of 1D linear Lagrange finite elements, from [15] ....... 35
4.2 Illustration of Neumann and Dirichlet boundary conditions ....... 38
4.3 The pressure drop with Neumann and Dirichlet boundary conditions . 40
5.1 Contour plots of the axial velocity profile in four consecutive cardiac cycles .... 45
5.2 Contour plots of the velocity profile with different time steps .... 46
5.3 The pressure drop with two different time steps .................... 47
5.4 Contour plots of the axial velocity profile in meshes with different resolution 49
6.1 Volunteer 1: PC-MRI measurement and CFD sampling locations ... 51
6.2 Volunteer 1: Cross sectional areas of the mesh and level set .... 52
6.3 Volunteer 1: Volumetric flow rates .................................. 54
6.4 Volunteer 1: Mean velocities ......................................... 55
6.5 Volunteer 1: Peak velocities ......................................... 55
6.6 Volunteer 1: PC-MRI of peak cephalad flow ....................... 57
6.10 Volunteer 1: PC-MRI of peak caudal flow ......................... 58
6.14 Volunteer 1: PC-MRI of flow reversal .............................. 59
6.18 Volunteer 1: Peak cephalad velocities with increased volumetric flow rate 61
6.19 Volunteer 1: Peak caudal velocities ............................... 61
6.20 Volunteer 1: Flow reversal with increased volumetric flow rate .... 61
6.21 Volunteer 1: Mean and peak velocities with increased flow rate .... 62
**List of Tables**

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Units and values of physical parameters</td>
<td>10</td>
</tr>
<tr>
<td>1.2</td>
<td>List of abbreviations</td>
<td>11</td>
</tr>
<tr>
<td>5.1</td>
<td>Volume changes during Taubin smoothing</td>
<td>41</td>
</tr>
<tr>
<td>5.2</td>
<td>A table of Poiseuille flow convergence rates</td>
<td>43</td>
</tr>
<tr>
<td>5.3</td>
<td>A table of convergence rates for Womersley flow</td>
<td>44</td>
</tr>
<tr>
<td>5.4</td>
<td>A table of maximum velocities in four consecutive cardiac cycles</td>
<td>44</td>
</tr>
<tr>
<td>5.5</td>
<td>The pressure drop in four consecutive cardiac cycles</td>
<td>45</td>
</tr>
<tr>
<td>5.6</td>
<td>A table of maximum velocities with different time steps</td>
<td>47</td>
</tr>
<tr>
<td>5.7</td>
<td>A table of cell count for the meshes in the resolution test</td>
<td>48</td>
</tr>
<tr>
<td>5.8</td>
<td>Maximum velocities with different mesh resolution</td>
<td>48</td>
</tr>
<tr>
<td>5.9</td>
<td>A table of peak velocities with different mesh resolution</td>
<td>48</td>
</tr>
<tr>
<td>6.1</td>
<td>Volunteer 1: Table of differences in area</td>
<td>54</td>
</tr>
<tr>
<td>6.2</td>
<td>Volunteer 1: Differences in mean velocity</td>
<td>56</td>
</tr>
<tr>
<td>6.3</td>
<td>Volunteer 1: Differences in peak velocity</td>
<td>56</td>
</tr>
<tr>
<td>6.4</td>
<td>Volunteer 1: Table of L2 errors between PC-MRI and CFD velocities</td>
<td>60</td>
</tr>
<tr>
<td>6.5</td>
<td>Volunteer 1: A table of L2 errors for increased volumetric flow rate</td>
<td>62</td>
</tr>
<tr>
<td>6.6</td>
<td>Volunteer 1: Differences in mean and peak velocity with increased volumetric flow</td>
<td>62</td>
</tr>
<tr>
<td>6.7</td>
<td>Table of differences in velocity and pressure for volunteer 1</td>
<td>64</td>
</tr>
<tr>
<td>6.8</td>
<td>Volunteer 2: Table of differences in area</td>
<td>67</td>
</tr>
<tr>
<td>6.9</td>
<td>Volunteer 2: Differences in mean velocity</td>
<td>68</td>
</tr>
<tr>
<td>6.10</td>
<td>Volunteer 2: Differences in peak velocity</td>
<td>69</td>
</tr>
<tr>
<td>6.11</td>
<td>Volunteer 2: A table of L2 errors for PC-MRI series 1</td>
<td>71</td>
</tr>
<tr>
<td>6.12</td>
<td>Volunteer 2: A table of L2 errors for PC-MRI series 2</td>
<td>73</td>
</tr>
<tr>
<td>6.13</td>
<td>Volunteer 2: A table of L2 errors for PC-MRI series 3</td>
<td>75</td>
</tr>
<tr>
<td>6.14</td>
<td>Volunteer 2: A table of L2 errors for PC-MRI series 4</td>
<td>77</td>
</tr>
<tr>
<td>6.15</td>
<td>Table of differences in velocity and pressure for volunteer 2</td>
<td>79</td>
</tr>
<tr>
<td>6.16</td>
<td>Volunteer 3: Table of differences in area</td>
<td>80</td>
</tr>
<tr>
<td>6.17</td>
<td>Volunteer 3: Differences in mean velocity</td>
<td>84</td>
</tr>
<tr>
<td>6.18</td>
<td>Volunteer 3: Differences in peak velocity</td>
<td>84</td>
</tr>
<tr>
<td>6.19</td>
<td>Volunteer 3: A table of L2 errors for PC-MRI series 1</td>
<td>85</td>
</tr>
<tr>
<td>6.20</td>
<td>Volunteer 3: A table of L2 errors for PC-MRI series 2</td>
<td>87</td>
</tr>
<tr>
<td>6.21</td>
<td>Volunteer 3: A table of L2 errors for PC-MRI series 3</td>
<td>89</td>
</tr>
<tr>
<td>6.22</td>
<td>Volunteer 3: A table of L2 errors for PC-MRI series 4</td>
<td>91</td>
</tr>
<tr>
<td>6.23</td>
<td>Table of differences in velocity and pressure for volunteer 3</td>
<td>94</td>
</tr>
</tbody>
</table>
Listings

1  Python script for Taubin smoothing using mshr  . . . . . . . . . . . . . . 26
2  A python script for interpolation CFD velocities and PC-MRI series  . . 65
   code/postprocessing.py  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 111
   code/table.py  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 115
   code/adjust_plane.cpp  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 117
   code/save_vtp.cpp  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 118
Bibliography


