Optimization-Based Geometry Correction Of Blood Flow CFD Simulations Using 4D-Flow Data

Fritiof Hegardt

Thesis for the degree of Master of Science in Engineering
Division of Fluid Mechanics
Department of Energy Sciences
Faculty of Engineering | Lund University

Optimization-Based Geometry Correction Of Blood Flow CFD Simulations Using 4D-Flow Data

Fritiof Hegardt

April 2021, Lund

This degree project for the degree of Master of Science in Engineering has been conducted at the Division of Fluid Dynamics, Department of Energy Sciences, Faculty of Engineering, Lund University, and at the Cardiac MR-group, Department of Clinical Sciences, Clinical Physiology, Lund University

Supervisor at the Division of Fluid Dynamics was Professor/Dr Johan Revstedt
Supervisor at the Cardiac MR Group, Sweden was Dr Johannes Töger
Examiner at Lund University was Professor/Dr Christer Fureby

Thesis for the Degree of Master of Science in Engineering

ISRN LUTMDN/TMHP-21/5472-SE

ISSN 0282-1990

Fluid Mechanics
Department of Energy Sciences
Faculty of Engineering, Lund University
Box 118, 221 00 Lund
Sweden

www.energy.lth.se

Abstract

Background 4D-flow is a powerful tool capable of capturing 3-dimensional, time-resolved flow measurements of blood flow in the body. Their current use is limited by resolution and scan times. A proposed solution is to use Simulation Based Imaging (SBI). This combines lower resolution 4D-flow scans with CFD simulations to improve resolution and reduce scan times. Previous work has focused on inlet and outlet conditions. This thesis explores the possibility of adding geometry correction to an optimization-based framework for SBI.

Methods Three different ways of deforming the mesh were implemented to explore geometry optimization. First finding the correct the diameter of a simple channel, second finding an initial rotation and translation error, and finally small boundary perturbations using Radial Basis Functions. The CFD simulations were performed using higher order finite element discretization. To compare the CFD simulations the MR images a forward function was used and the optimization was performed using gradients calculated using the adjoint method as well as finite differences.

Results All three cases managed to correct the geometry errors both with and without noise in the MR-image, but the errors increased with increased noise levels.

Conclusion The results shows that all three approaches worked in 2D but was sensitive to noise and flow conditions.

Acknowledgments

First of all I would like to send a huge thank you to my two supervisors Johannes Töger from the Cardiac MR group and Johan Revstedt from the Division of Fluid Dynamic. Thank you for your tireless support, knowledge and countless hours spent helping me.

I would also like to thank Matthew Zahr from University of Notre Dame for all of his help, technical support and invaluable knowledge.

Finally I would like to thank all of my friends and family who have helped and supported me during my studies.

Without all of you non of this would have been possible.

Contents

Abbreviations

CFD	Computational Fluid Dynamics
CMR	Cardiovascular Magnetic Resonance
FEM	Finite Element Method
HIKING	High-resolution Imaging with a priori Knowledge Incorporating the Navier-Stokes equation and the discontinuous Galerkin method
MRI	Magnetic Resonance Imaging
PC-MR	Phase Contrast Velocity Measurements
RBF	Radial Basis Function
SBI	Simulation Based Imaging
Voxel	The three dimensional pixels of the 4D-flow image
WSS	Wall Shear Stress

Chapter 1

Introduction

The leading cause of deaths today is cardiovascular diseases. According to one study [1] these account for 31.8% of total deaths in the world. For comparison, neoplasms (e.g. tumors and cancers) accounts for 17.1% of total deaths. It is therefore important to further our understanding of how to diagnose, treat and prevent these illnesses.

The cardiovascular system refers to the organs responsible for pumping and transporting blood throughout the body, i.e. the heart and the blood vessels. Through the advancement of Magnetic Resonance Imaging (MRI) several non-invasive ways of studying the blood flow have become available for physicians and researchers and one of these is the ability to take 3-dimensional time-resolved images, so-called 4D-flow, of the blood flow. These images enable both measurements of flow parameters such as stroke volume as well as visualization of flow. The ability to acquire data over an entire volume in just one scan is also potentially advantageous.

However, the use of 4D-flow in clinical applications is currently limited. The main reason for this the long time required to acquire a single 4D-Flow scan. Shorter scan times enables more patients to be scanned with a single MR machine. It can also be unpleasant or sometimes impossible for patients to remain stationary for the required time to take the images. It is therefore desirable to minimize the time in an MR machine. Reducing the time required for a scan is possible but this will result in either lower resolution or more noisy images.[2]

One proposed solution to this is to combine lower resolution 4D-flow images with computational fluid dynamics (CFD) into so-called Simulation-Based Imaging (SBI). There are several researchers [3, 4] that have used the data from 4D-flow scans to acquire boundary conditions for CFD simulations and there has also been some previous researchers [5, 6] that have used optimization-based approaches for improving SBI that uses fluid simulations that are optimally matched to the 4D-flow data.

Simulation Based Imaging has shown some promising results[5]. However, there are some limitations to these methods. One of these limitations is that the geometry's influence on the overall results is unknown as the current framework only optimizes over inlet and outlet conditions. It has been hypothesized that the framework is potentially sensitive to variations in the geometry. For example the walls of the vessels are not rigid and so could potentially move during the cardiac cycle, the scan has a limited resolution and the method for extracting the geometry is subject to errors as well. All of these can have an effect on the flow but so far this effect has not been properly evaluated.

1.1 Purpose

The purpose of this thesis is to explore if different errors in the initial geometry can be automatically corrected using 4D-flow data and optimization.

1.2 Aims

The aim of this master thesis is to implement and evaluate how optimization over different geometry parameters performs. This will be performed in steps of increasing complexity starting with very simple optimization over only one geometry parameter and working towards more and more complex problems with multiple geometry parameters as well as inlet flow conditions.
To achieve these aims we will need to meet the following goals:

- Implement functions for moving and deforming the mesh of the geometry to enable optimization over different geometry parameters.
- Evaluate performance of these parameters in isolation as well as simultaneously with optimizing over inlet velocity.
- Evaluate performance with varying noise levels.

1.3 Limitations

- Simple 2-dimensional steady-state simulations will be used, this enables faster implementations and reduces calculation times.
- Synthetic data will be used for the 4D-flow data which enables more control over establishing a ground truth for the simulations.

Chapter 2

Background

2.1 The Cardiovascular System

The cardiovascular system, sometimes called the circulatory system, refers to the heart and blood vessels that are the organs responsible for pumping and transporting blood throughout the body (see Figure 2.1). These can be compared to a system where the blood vessels are channels, the heart is the motor/pump and the blood is the transport medium responsible for supplying the body with oxygen, vital nutrients and the removal of waste products.

The cardiovascular system is a looping system [7], beginning with the heart pumping the deoxygenated blood returning from the body through the pulmonary circulation system, also known as the lungs. Through breathing, the blood is oxygenated in the lungs and returns to the heart where it is pumped out to the rest of the body, initiating the loop again. [8]

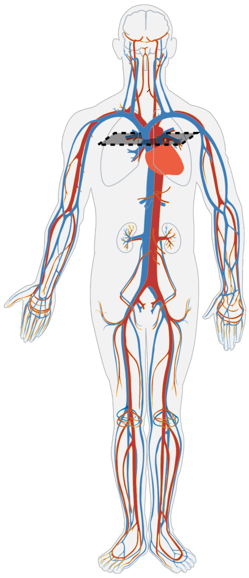


Figure 2.1: The cardiovascular system, the major blood vessels in the body together with the heart. Red is oxygenated blood and blue is the deoxygenated blood.

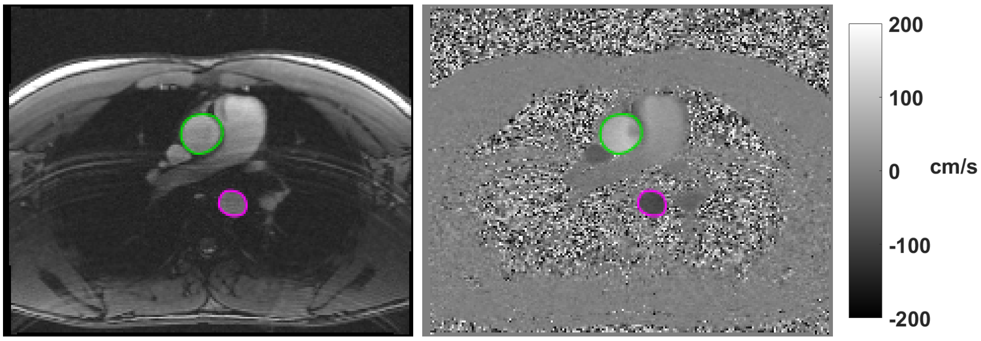
2.1.1 Cardiovascular Disease

Maintaining blood flow in the cardiovascular system is vital for the functioning of the body. Any defects that limits the cardiovascular systems ability to do so can have severe consequences. An example of such a heart disease that effects the flow in the circulatory system is Aortic coarctation. This is a narrowing at the beginning of the descending aorta and is a congenital birth defect that occurs in approximately 4 out of every 1000 births [9, 10]. However the most common cause of death due to cardiovascular disease is coronary heart disease (heart attack) and cerebrovascular disease (stroke). [11] These are often a consequence of atherosclerosis [12], a buildup of plaques in the blood vessel walls that can lead to disrupted blood flow as well as blood clots.

There have been multiple studies on the importance of flow when assessing cardiac health and diagnosing patients [13, 14]. Stroke volume is one that is commonly used when looking at cardiac heath. Another parameter that has been investigated us wall shear stress and its influence on blood vessels. One study [13] have linked atherosclerosis to low wall shear stresses in the blood vessel. To detect these symptoms early and furthering our knowledge of the blood flow has become an increasingly important part of understanding how to treat and prevent these illnesses.

2.2 Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) is a non-invasive technique for producing images of internal organs and soft tissue. An MRI scanner is essentially a large magnet with a very powerful magnetic field, usually in the magnitude around $1.5 - 3$T. To produce the image it uses the magnetic properties of hydrogen nuclei that becomes aligned with the magnetic field of the MRI scanner[15]. The scanner also includes elements that can introduce gradients to the magnetic field along all three spatial dimensions as well as a radio frequency transmitter and receiver. As hydrogen nuclei are present in varying amounts in all water and organic molecules in the body and different tissues responds differently to these inputs a detectable signal can be produced by exposing the nuclei to different combinations of gradients and frequencies.



(a) Example of MR-image. This figure shows plane through the ascending and descending aorta.

(b) Here can be seen the same plane through the ascending and descending aorta but with velocity encoding.

Figure 2.2: Example of MR image (a) and 2D-flow image (b) with velocity encoding. Approximate position of imaging plane can be seen in Figure 2.1. Green outline marks the ascending aorta and purple outline marks the descending aorta.

2.2.1 Cardiovascular Magnetic Resonance

Cardiovascular Magnetic Resonance (CMR) is a collective term for when MRI is used for imaging the cardiovascular system. One particular type of CMR is Phase Contrast Velocity Measurements (PC-MR) that enables measurement of the velocity of blood flow.

The basic working principle of PC-MR is that the nuclei are exposed to two subsequent and opposite gradients in the magnetic field. For stationary particles this will have a net zero effect but for moving particles this will result in a phase offset in the particles magnetization. This offset is dependent on the velocity component of the particles compared to the direction of the gradients. By measuring this offset it is therefore possible to determine both the direction and the velocity magnitude of the particle compared to the normal of the plane. [16, 17] When applying this sequence along one direction a velocity encoded image of one plane can be acquired, a so called 2D-flow image. It is possible to apply the sequence along all three dimensions. By doing this a time resolved 3-dimensional velocity image over an entire volume can be sampled, a so called 4D-flow image.

Capturing 4D-flow images that give information about flow in an entire volume rather than just a single plane has multiple advantages. With just one data acquisition, flow parameters at multiple different points is possible. By visualizing the flow it is also possible to get insight into how and where the blood flows. [18] There are however some downsides to 4D-flow, the main one being the time required for sampling an image. It is possible to speed up the sampling but this will come at a cost of either lower resolution, increased noise levels, or both.

2.3 Computational Fluid Dynamics

To understand how flow behaves there are two main approaches: experimental and computational. The first method is to set up a controlled experiment on either the real system or a model, where the desired parameters can be measured. For example, building a prototype of a small scale airplane and subject the prototype to wind-tunnel testing. This approach can yield very good results but generally comes with some drawbacks. The downside to using an experimental approach is that getting good and detailed measurements of all quantities of interest, while controlling the environment to get consistent results, can be both costly and time consuming.

The second method, known as Computational Fluid Dynamics (CFD), is to numerically calculate how the flow will behave based on equations called the governing equations derived from fundamental laws of physics. The governing equations chosen for this application are the Navier-Stokes equations, which will be further explained in Chapter 3. [19]. For very simple cases it is possible to solve these equations analytically but for even moderately complex problems a numerical approach is necessary to solve the equations.

One method for solving these equations is the finite element method (FEM). Figure 2.3 shows how first a geometry is created (here a simplified aorta based on the cardiovascular anatomy). After the geometry is created, boundary conditions are applied to the model. The geometry is then split into multiple tiny elements that together form the *mesh*. By discretizing the equations for each mesh-element and connecting them together builds a system of equations that can be solved to get the final result.

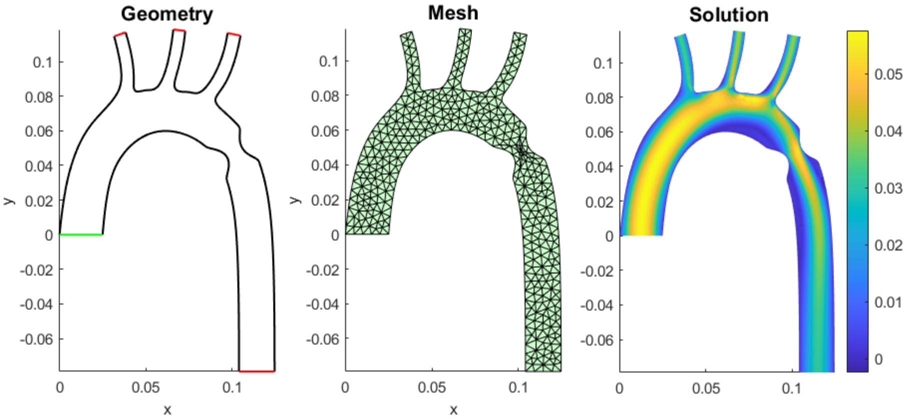


Figure 2.3: Example of how a problem is solved using computational fluid dynamics using the finite element method. 1. Geometry with boundary conditions: walls (black), inlets (green) and outlets (red). 2. Finite Element Mesh 3. Final solution.

There are some advantages of using CFD compared to an experimental procedure. For example, testing new design alterations or applied cases can be done without building new prototypes and evaluating test data can then be done more rapidly. however, the simulation can be complex and it is computationally expensive to solve the exact equations accurately. The computational model is after all trying to emulate reality with all of its irregularities. Therefore many approximations have to be used to simplify the equations. The results are also greatly dependent of the setup, if the setup is inaccurate the results will be in distinctive.

2.4 Simulation Based Imaging

CFD as well as PC-MR are both powerful tools with their respective advantages, but also with significant drawbacks. This is has led several research groups to try and combine these in order to work around their respective shortcomings. In this thesis all techniques combining CFD and PC-MR will be refer to as simulation-based imaging, or SBI for short.

The most common solution [20] has been to use MRI for generating the geometry and 2D flow to get the flow profiles at the inlets and outlets. This data can then be fed into a conventional CFD solver and solved like a regular fluid problem. CFD can give good results and for example, better Wall Shear Stress (WSS) estimations but the results are hard to validate.

Another proposed solution is to use 4D flow instead of 2D-flow [5]. This will use the entire 4D-flow volume to optimize the CFD simulations. The basic operating principle is to have a lower resolution 4D-flow image, a high resolution CFD simulation, and a method for comparing the CFD simulation to the 4D-flow data. [5, 6] Using this method a set of parameters, usually controlling inlet and outlet conditions, are set as free parameters.

By comparing the 4D-flow image to the CFD simulation these free parameters are then adjusted through optimization in order to find the CFD simulation that best agrees with the 4D-flow data. The final optimized CFD simulation can then be used as a higher resolution SBI. So far only parameters governing the inlet and outlet conditions of the flow has been optimized, and the influence of optimizing for other parameters such as geometry has not been explored.

Chapter 3

Methods

In this chapter the underlying methods used for this thesis will be described. By way of introduction, the Computational Fluid Dynamics set up is defined, followed by an explanation of the optimization process. Lastly, the experimental set up, the generation of the synthetic data and the arrange of the different cases is presented.

3.1 Overall Concept

The work presented in this thesis is based on development made for the 4D-flow HIKING framework [5]. The overall goal for this framework is to produce high resolution, 3D images of a patients blood flow from lower resolution 4D-flow data. By using a forward function, $\mathbf{M}$, that mimics the process of capturing 4D-flow data, the difference between the CFD solution ($\mathbf{v}_{CFD}^{\boldsymbol{\mu}}$) and the 4D-flow data ($\mathbf{v}_{MR}$) can be compared using our objective function:

$$F(\boldsymbol{\mu}) = \|\mathbf{v}_{MR} - \mathbf{M}\mathbf{v}_{CFD}^{\boldsymbol{\mu}}\|_2^2 \tag{3.1}$$

$\boldsymbol{\mu}$ is a vector ($\boldsymbol{\mu} \in \mathbb{R}^p$) containing all of the variables to the CFD setup that can be changed. We call these our free parameters. By minimizing the objective function, the best fit of the CFD solution given the set of free parameters $\boldsymbol{\mu}$ is acquired. The parameters $\boldsymbol{\mu}$ that minimize Equation 3.1 are called $\boldsymbol{\mu}^*$. The solution $\mathbf{v}_{CFD}^{\boldsymbol{\mu}^*}$ can then be used as the final image. In Figure 3.1 a flow chart of the framework is presented.

In previous research the parameters $\boldsymbol{\mu}$ only includes the parameters that control the velocities at the inlet and outlet, while the geometry of the blood vessels were assumed to be rigid and known. This thesis will explore the possibility of adding different aspects of the geometry to the free parameters $\boldsymbol{\mu}$ to see if these could be corrected for in the same way as the inlet and outlet conditions.

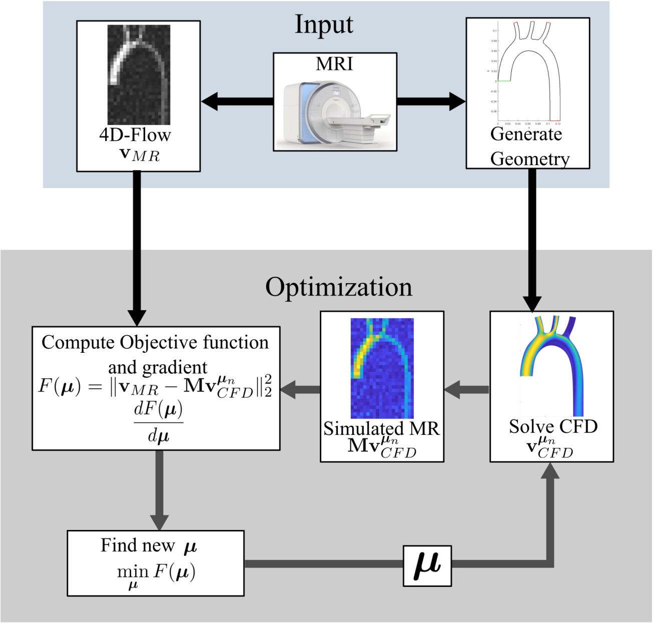

Input
MRI
4D-Flow
$\mathbf{v}_{MR}$
Generate
Geometry
Optimization
Compute Objective function
and gradient
$F(\boldsymbol{\mu}) = \|\mathbf{v}_{MR} - \mathbf{M}\mathbf{v}_{CFD}^{\boldsymbol{\mu}_n}\|_2^2$
$\frac{dF(\boldsymbol{\mu})}{d\boldsymbol{\mu}}$
Simulated MR
$\mathbf{M}\mathbf{v}_{CFD}^{\boldsymbol{\mu}_n}$
Solve CFD
$\mathbf{v}_{CFD}^{\boldsymbol{\mu}_n}$
Find new $\boldsymbol{\mu}$
$\min_{\boldsymbol{\mu}} F(\boldsymbol{\mu})$
$\boldsymbol{\mu}$

Figure 3.1: Flowchart over the program outlining the overall structure of the proposed framework.

All code for this framework was written using Matlab 2019a (MathWorks Inc, Natick, Massachusetts, United States) and uses the Finite Element Solver mCaMO (Copyright ©[2020] [Matthew J. Zahr]).

Table 3.1: *Important symbols*

Important symbols	
$\boldsymbol{\mu}$	The *free parameters*, the variables for each CFD simulation that can be changed.
$\mathbf{v}_{MR}$	The velocity data from the 4D-flow scan.
$\mathbf{v}_{CFD}^{\boldsymbol{\mu}}$	the velocity data from the CFD simulations.
$\mathbf{u}$	The state variables for of the PDE. Contains the velocity $\mathbf{v}_{CFD}^{\boldsymbol{\mu}}$ and pressure $\boldsymbol{p}$
$\mathbf{M}$	The forward function, Transforms the CFD data to the same dimension as the MRI data. (essentially aims to mimics the 4D-flow scan)
$F(\boldsymbol{\mu})$	The objective function

3.2 Experimental Setup

For this thesis three main cases using parameters affecting different aspects of the geometry was investigated. This section will describe the three different test Cases and the different methods used to optimize the geometries.

3.2.1 Geometries and Meshes

Two main geometries were used, that will be referred to as the *simple stenosis* geometry and the *Aorta* geometry, respectively (see Figure3.2 and 3.3).

The simple stenosis was used for initial development as its simplicity enabled quick simulation times and easy modifications to the geometry. The Aorta geometry was used for the late stages of development as it more complex shape enabled more investigations of how the framework handled different aspects.

For each geometry, three different size settings for the element size was chosen and for each element size one second order ($p = 2$) and one third order ($p = 3$) mesh was generated resulting in a total of six different meshes for each geometry. Multiple meshes were generated for two primary reasons. One, to be able to simulate our reference solution $\mathbf{v}_{ref}^{\boldsymbol{\mu}^*}$ (see Section 3.3.2) on a different mesh than our SBI flow $\mathbf{v}_{CFD}$. As $\mathbf{v}_{ref}^{\boldsymbol{\mu}^*}$ is calculated only once but $\mathbf{v}_{CFD}$ is calculated multiple times this will save some computing power. And two, to have the possibility of comparing the performance of the framework for different mesh resolutions. The simple stenosis geometry and mesh was made using the mCaMO application and the Aorta geometry was created using the free software Gmsh 4.8.2 [21].

The goal of this thesis was to investigate possibilities of optimizing for different geometry parameters and not the flow itself. Therefore the speed of the CFD simulations was much more important than mesh independence. A mesh sensitivity analysis was still performed to find the most suitable meshes to use for the simulations and the results for this can be seen in Section 4.1.

3.2.2 Cases

Table 3.2 shows an overview of the three main Cases. Each Case was first tested with static inlet conditions, i.e. the free parameters $\boldsymbol{\mu}$ only contained the geometry parameters being explored. Once this was working, the parameters controlling the inlet velocities was added to the parameters $\boldsymbol{\mu}$ for a second evaluation.

Table 3.2: *Overview of the different test cases.*

Case 1: Diameter		
	Free parameters $\boldsymbol{\mu}$	Geometry
1	v_x^{inlet}, v_y^{inlet}, D_y	Simple Stenosis

Case 2: Rotation and translation		
	Free parameters $\boldsymbol{\mu}$	Geometry
2	v_x^{inlet}, v_y^{inlet}, $d\theta$, dx, dy	Aorta

Case 3: Boundary Deformation		
	Free parameters $\boldsymbol{\mu}$	*Geometry*
3a	dx_n, dy_n; $n = 5$	Aorta
3b	v_x^{inlet}, v_y^{inlet}, dx_1, dy_1, dx_2, dy_2	Aorta

3.2.3 Case 1, Diameter

Case 1 was the first and simplest Case run. It used a simple stenosis geometry (see Figure 3.2) to investigate the possibility of finding the correct diameter of the geometry. Due to the simplicity of the model, this could by done by simply scaling the mesh in the y-direction with the scaling factor D_y.

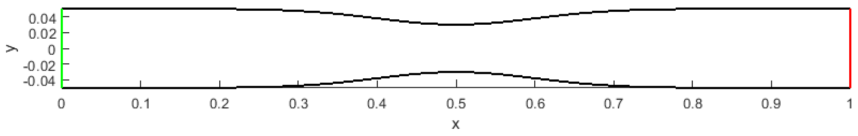


Figure 3.2: Simple Stenosis geometry. Green represents inlet, red outlets and black is the walls.

3.2.4 Case 2, Translation and Rotation

During MRI scans it is possible that the patients will move slightly. This could result in rotation and translation error in the geometry compared to the 4D-flow image. Case 2 used the aorta geometry (see Figure 3.3) to investigate if the framework could correct for small errors in rotation and translation.

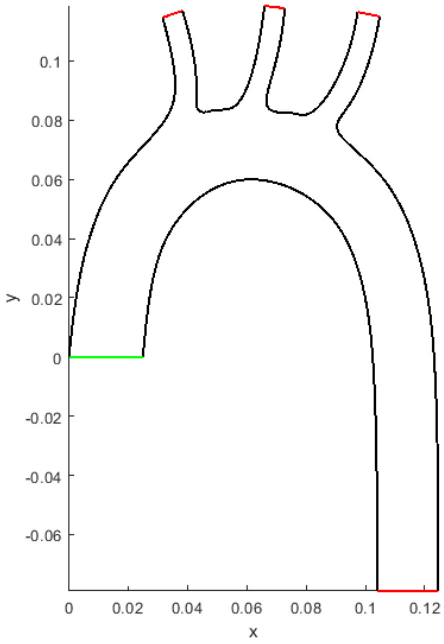


Figure 3.3: Aorta geometry. Green represents inlet, red outlets and black is the walls.

For rotating the geometry all mesh nodes coordinates were multiplied by the rotation matrix

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{3.2}$$

and the offset was a simple addition of the desired offset in x-and y-direction to the mesh nodes.

3.2.5 Case 3, Boundary Deformation

The aim for Case 3 was to explore more fine grained deformations of the boundaries. There are several methods to parameterize a geometry so that it can be deformed in a uniform manner. The method chosen for deforming the boundary is to use radial basis functions (RBF), and specifically a version usually referred to as a *bump function*, see Equation (3.3).

$$\varphi(x) = \begin{cases} e^{-\frac{1}{1-((x-x_0)r)^2}} & ,r < \frac{1}{(x-x_0)} \\ 0 & ,r \geq \frac{1}{(x-x_0)} \end{cases} \tag{3.3}$$

Each RBF is defined by a predetermined radius r and center x_0. The mesh motion can then be calculated by multiplying the basis function with the desired motion in x- and y-direction (dx and dy) respectively and moving the affected mesh nodes by that amount. (see Figure 3.5) In total each RBF will add 2 parameters (dx and dy) to the free parameters $\boldsymbol{\mu}$.

To get a smoother transition from the bump function the square of the bump function was used instead in order to get a softer transition (see Figure 3.4).

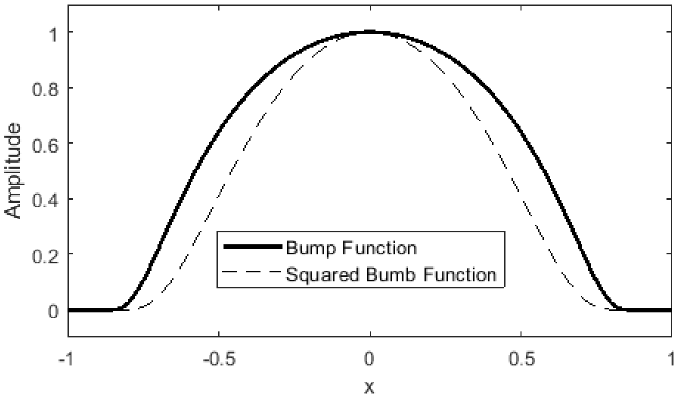


Figure 3.4: The two versions of the radial basis function used plotted in 1D.

As each RBF is independent from each other they can be used either in isolation, or ad multiple together with some overlap to parameterize a larger continuous part of the boundary. This makes them great for early development as one can start simple and gradually increase complexity of the problem by adding more parameters.

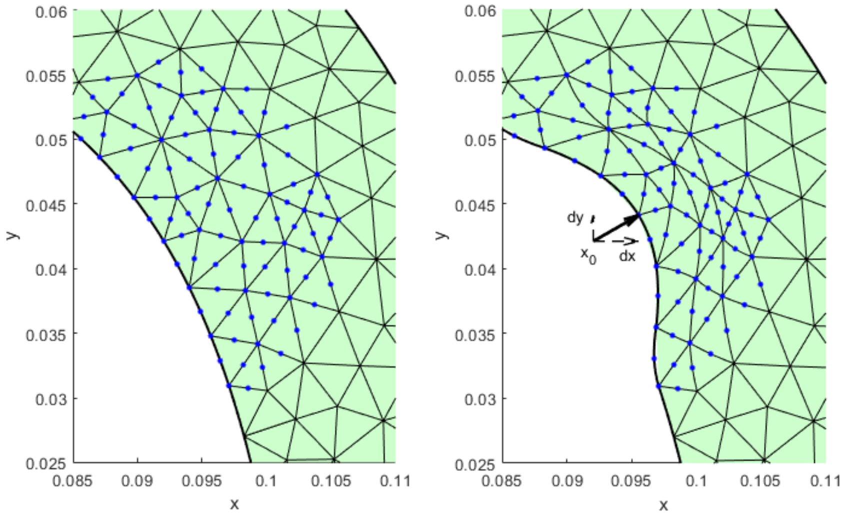


Figure 3.5: Example of how the Radial Basis function moves the mesh. Left is before and right is after the mesh motion was applied. Large arrow is total movement and dashed arrows are the *dx* and *dy* offset respectively and x_0 is the center of the RBF. Blue dots are all nodes within the radius *r*.

3.3 4D-flow

As mentioned in Section 2.2.1 4D-flow images is the name for velocity encoded, 3-dimensional, time resolved images captured using Magnetic Resonance Imaging.

The 4D-flow data is sampled in slices with consistent spacing along each spatial dimension resulting in a structured mesh of cuboid elements of size $\Delta x \times \Delta y \times \Delta z$. These elements are called *voxels* and for each voxel we will get the time resolved velocity data for each spatial dimension(v_x, v_y and v_z). Figure3.6 shows an example of a 2 dimensional voxel grid and the corresponding velocity magnitude image.

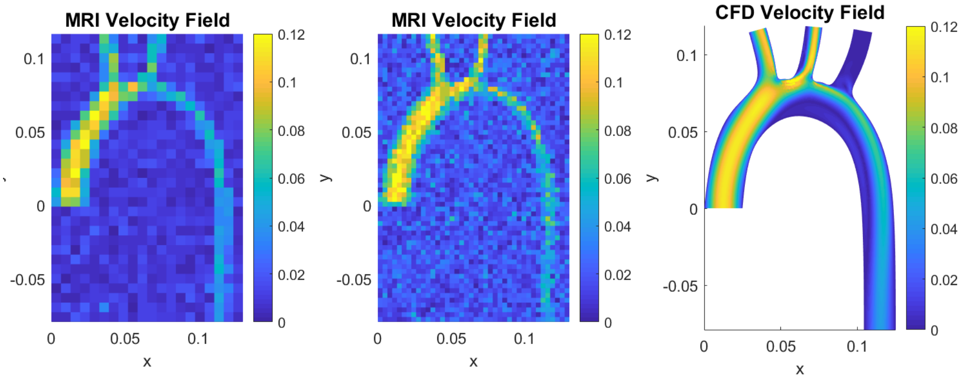


Figure 3.6: Example of simulated MRI image with 5% noise for two different voxel resolutions, and the CFD simulation using the aorta geometry.

This project will be working exclusively in 2D with no time component. This means the MR images used are technically not 4D-flow. For this reason the images used in this thesis will therefore be called MR-flow instead of 4D-flow.

3.3.1 Forward Function M

In order to be able to accurately compare the MR-flow data to the CFD data a forward function is needed. This forward function **M** has two main purposes. First of all it needs to transfer the higher resolution CFD solution on to the lower resolution voxel mesh so that they can be compared one to one. Second, it should do this in such a way that it models some of the physical properties of the MR scanner. The way this is modeled for a two dimensional voxel mesh of size $n \times m$ can be seen in Equation (3.4).

$$\mathbf{Mv} = \sum_{i=1}^{n}\sum_{j=1}^{m} \int_{\Omega} W_{ij} \; \mathbf{v} \; d\Omega \; \boldsymbol{e}_{ij} \tag{3.4}$$

$\boldsymbol{e}_{ij}$ is a matrix $\boldsymbol{e}_{ij} \in \mathbb{R}^{n\times m}$ where all entries equals zero except the ij-th entry that equals one. By choosing the weight W_{ij} the desired spatial smoothing properties of the forward function can be acquired.

It has been shown [22] that the spatial smoothing of the velocity for each voxel can be modeled by

$$\psi(x,y) = \text{sinc}\left(\frac{x}{\Delta x}\right) \times \text{sinc}\left(\frac{y}{\Delta y}\right) \tag{3.5}$$

where Δx and Δy are the voxel size in x- and y-direction. The sinc-function is defined as:

$$\operatorname{sinc}\left(\frac{x}{\Delta x}\right) = \frac{\Delta x}{\pi x} \cdot \sin\left(\frac{\pi x}{\Delta x}\right)$$

Although Equation (3.5) is the theoretically most accurate way of modeling the spatial smoothing of the voxels its main drawback is that it approaches zero very slowly. To keep down the computation time for calculating the weights for each voxel the smooth box function is therefore used instead. The one-dimensional version of the smooth box function can be seen in Equation (3.6).

$$\chi\left(x, x_0\right) = \alpha\left(\frac{1}{1+e^{-(x-x_0+w)/\sigma}} - \frac{1}{1+e^{-(x-x_0-w)/\sigma}}\right) \tag{3.6}$$

The smooth box function can be tuned to different shapes where w determines the width of the function, σ determines how sharp the bounds of the function is, and α determines the height of the function. The parameters used for this thesis is $w = \Delta x/2$, $\sigma = \Delta x/10$ and $\alpha = 1/\int_{-\infty}^{\infty} \chi dx$ (normalized so the area under the graph is one). This results in the shape seen in Figure (3.7).

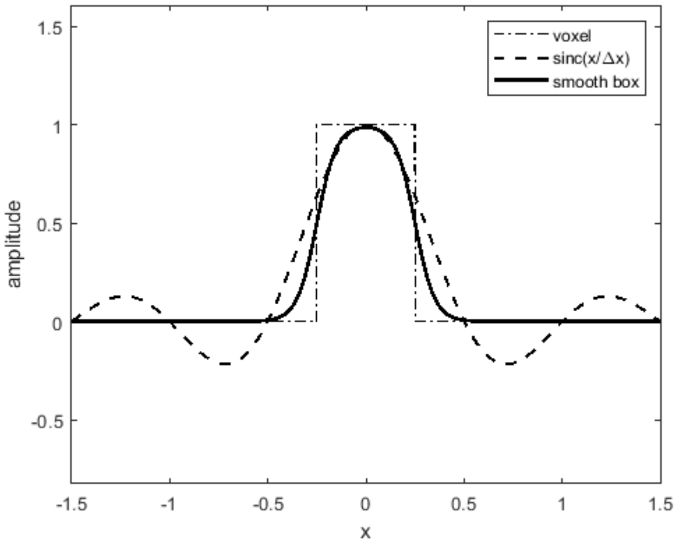


Figure 3.7: Comparison of a voxel of width $\Delta x = 0.5$ in one dimension with the $\operatorname{sinc}\left(x/\Delta x\right)$ function and the *smooth box* function as described in Equation (3.6) and (3.5) respectively

To visualize some of the similarities and differences between using the smooth box function and the sinc function for the weight W_{ij}, a simple 1D test was done as shown in Figure (3.8). This was done to validate that the sinc function could be substituted with smooth box function in order to speeds up calculations of $\mathbf{M}\mathbf{v}_{CFD}$ and still provide a good enough approximation of the MRI physics.

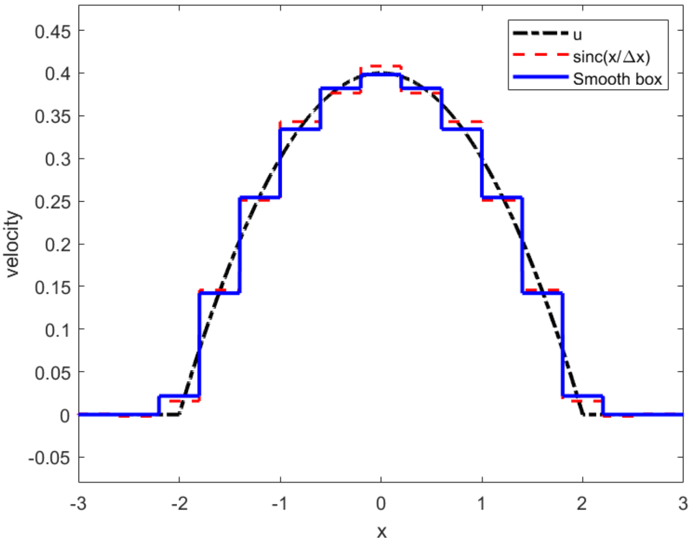


Figure 3.8: Example of how using the sinc-function versus the smooth box function for the spatial smoothing translates to voxel velocities for a 1D parabolic velocity distribution with a voxel width $\Delta x = 0.4$

3.3.2 Generating the MR-flow data $\mathbf{v}_{MR}$

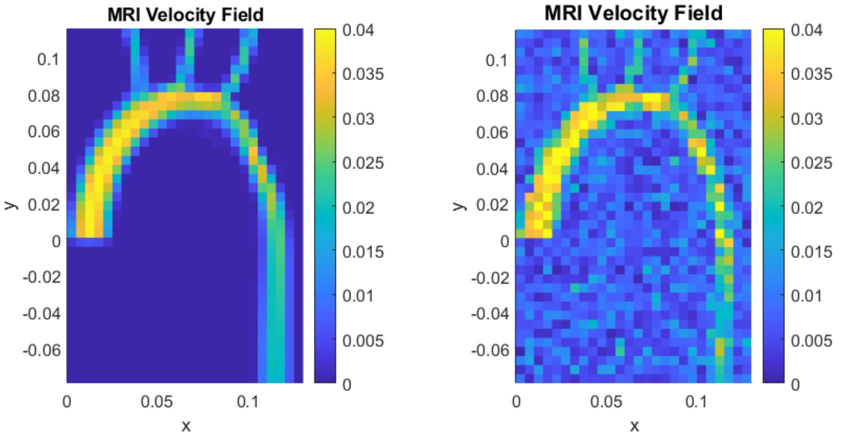
To get the MR-flow data for the development of the framework there are three main options. Full 4D-flow scans of healthy volunteers, phantom measurements of controlled flow experiments, and synthetic data.

The option used for this thesis was to use synthetic data. The downside of using synthetic data for the MR-flow is that it could miss some of the physical phenomena seen in real 4D-flow scans that could potentially influence the results. The advantage is that it provides the most control over the reference flow used to establish a *ground truth* that the final SBI flow can be compared to. This enables quicker changes to the experimental setup compared to the other two methods and easier evaluation of the performance of the framework. The synthetic data is generated by using the same forward function already constructed for the objective function from an initial *truth* CFD simulation.

$$\mathbf{v}_{MR} = \mathbf{M}\mathbf{v}_{ref}^{\boldsymbol{\mu}^*} \tag{3.7}$$

$\mathbf{v}_{MR}$ is the MR velocity field, $\mathbf{v}_{ref}^{\boldsymbol{\mu}^*}$ is the velocity field from the CFD simulation using the *truth* setting $\boldsymbol{\mu}^*$ (see Section 3.4) and $\mathbf{M}$ is the forward function (see Section 3.3.1).

Another benefit of using synthetic data is that artificial noise of varying levels can be added to the MR-flow. The noise level is defined as a percentage of the maximum velocity amplitude in the MR-flow image. Figure 3.9 shows an example of a MR-flow and an example where normally distributed noise was added to the image.



(a) MRI velocity field, no noise.

(b) MRI velocity field, 10% noise.

Figure 3.9: Example MR-flow image: Example of the same MR-flow image with and without 10% noise. The color scale shows the velocity magnitude for each voxel.

3.4 Computational Fluid Dynamics

In order to acquire the higher resolution velocity field $\mathbf{v}^{\boldsymbol{\mu}}_{CFD}$ this framework uses a finite element based fluid solver for the incompressible Navier-Stokes equations capable of using higher order elements. In this section some of the underlying theory and implementation of the CFD solver will be described.

3.4.1 Navier-Stokes

Computational Fluid Mechanics (CFD) uses the Navier-Stokes equations to compute the fluid flow. In its most general form the Navier-Stokes equations consists of three main parts, conservation of mass, conservation of momentum, and conservation of energy.

This thesis uses the incompressible, steady state version of the Navier-Stokes equations meaning the density of the fluid is assumed to be constant and the flow is constant over time. Using these assumptions the energy equation can be removed and simplify the Navier-Stokes equations to:

$$\nabla \cdot \mathbf{v} = 0 \tag{3.8}$$

$$\rho(\mathbf{v} \cdot \nabla)\mathbf{v} = \nabla p + \nabla \cdot \tau_{ij} \tag{3.9}$$

There have been studies done highlighting the problems associated with modeling blood as a Newtonian fluid [23] but for simplicity of the simulations the fluid was assumed to be a Newtonian fluid giving us the following expression for shear stress:

$$\nabla \cdot \tau_{ij} = \mu \nabla^2 \mathbf{v} \tag{3.10}$$

Table 3.3 shows the values used to model the flow the density and viscosity for blood.

Table 3.3: *Values used for simulations.*

Name	Symbol	Value
Density	ρ	1060 kg/m^3
Viscosity	μ	0.003 kg/(m·s)

Reynolds Number

The Reynolds number the was defined as

$$Re = \frac{\rho v D}{\mu}, \tag{3.11}$$

where v is the inlet velocity and D is the inlet diameter. As there is no support for turbulence modeling in the mCaMO solver all simulations where done with a Reynolds number of maximum 1500 to keep it in the laminar domain. For the aorta geometry used in Case 2 and 3 a Reynolds number of 1500 resulted in a inlet velocity of $v_{in} = 0.1698$m/s. For comparison peak velocity in the human aorta vary between $0.5 - 2$m/s with a mean velocity around 0.5m/s [24].

3.4.2 Finite Element Formulation

To numerically discretize the Navier-Stokes equations a continuous Galerkin Finite Element method was used. To approximate the solution u curved tetrahedral elements of second ($p = 2$) and third ($p = 3$) order was used. The advantage of higher order mesh elements is that they provide higher accuracy for the same mesh size as the error is proportional to $\mathcal{O}(h^{p+1})$, where h is the mesh size and p is the element order. The elements ability to curve also enables them to better follow the curvature of the model compared to first order elements as illustrated by Figure 3.10.

In following sections the resulting fully discretized equation system will be written in its residual form as $\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}) = 0$, where $\boldsymbol{u}$ are the state variables (velocity $\mathbf{v}^{\boldsymbol{\mu}}_{CFD}$ and pressure $\boldsymbol{p}$) subject to the free parameters $\boldsymbol{\mu}$.

we use two notations for the

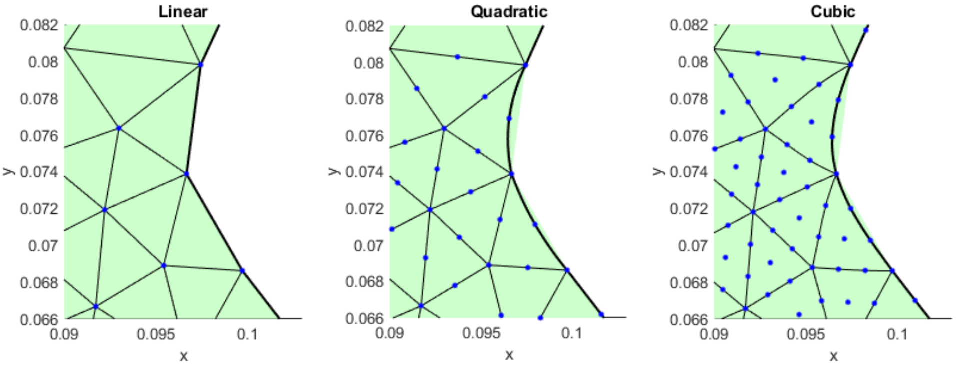


Figure 3.10: Comparison of linear, quadratic and cubic mesh elements. Each element is denoted by the thin line and the element nodes are plotted as the blue dots.

The equation system $\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu})$ is solved using the Newton-Raphson method. This is a iterative method for finding the roots to a system of non linear functions, i.e find $\boldsymbol{u}^*$ such that $\boldsymbol{r}(\boldsymbol{u}^*, \boldsymbol{\mu}) = \mathbf{0}$. [25] It works by linearizing the equation system using a Taylor expansion around the current estimate u_n.

$$\boldsymbol{r}(\boldsymbol{u}) \approx \boldsymbol{r}(\boldsymbol{u}_n) + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}_n)(\boldsymbol{u} - \boldsymbol{u}_n) \tag{3.12}$$

By using the root to this approximation as a guess for the new root and repeating the process we get an iterative method of better and better approximations of the root $\boldsymbol{u}_n$.

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n - \left[\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}_n)\right]^{-1} \boldsymbol{r}(\boldsymbol{u}_n) \tag{3.13}$$

it can be shown that this method converges quadratically assuming the initial point $\boldsymbol{u}_o$ is chosen *sufficiently close* to $\boldsymbol{u}^*$.

Solving Flows 500-1500 Re

The Newton-Raphson method was generally only stable for Reynolds numbers below 500. To mitigate this a second outer loop was implemented that initialized the simulations at a lower inlet velocity such that the Reynolds number was below 500. The solution to this simulation was then used as an initial guess for a new simulation with higher inlet velocity. This process was then repeated until the desired inlet velocity was reached. By tuning the increase in velocity for each loop flows of Reynolds numbers up to 1500 Re could be simulated.

3.4.3 Boundary conditions

For the walls and inlet, Dirichlet boundary conditions was used. At the walls the velocity was set to $v_x = v_y = 0$. For the inlet a normalized parabolic profile was fitted that could then be scaled by the factor μ_0 to get the x-velocity and μ_1 to get the y-velocity as illustrated by Figure 3.11.The outlets are pressure outlets modeled as Robin boundary conditions with the traction ($\frac{1}{2}(\nabla(\boldsymbol{v}) - \nabla(\boldsymbol{v})^T) \cdot n$), set to zero.

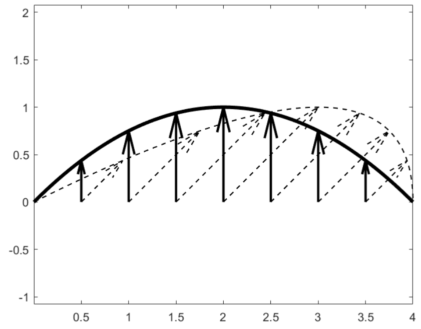


Figure 3.11: Illustration of how the velocities are prescribed at the inlet boundary. Here the solid line is using $\mu_0 = 0$ and $\mu_1 = 1$ and the dashed line is using $\mu_0 = \mu_1 = 1$

3.5 Optimization

The CFD simulations are set up so that they will be dependent on a set of free parameters $\boldsymbol{\mu}$, and an objective function that compares the difference between the MR-flow data and the CFD solution have been defined (see equations (3.1)). The goal of the framework is to find the set of parameters $\boldsymbol{\mu}$ that minimizes the objective function $F(\boldsymbol{\mu})$ with the assumption that this will result in the CFD solution that best matches the real flow.

There are several methods to solving these problems that all follow the same overall approach:

- Find a direction where the objective function decreases
- Determine the length of the step along that direction
- Repeat until a minimum is found

The parameters that solve (3.16) as denoted $\boldsymbol{\mu}^*$ and thus $\mathbf{v}_{CFD}^{\boldsymbol{\mu}^*}$ will be the final solution.

3.5.1 PDE constrained optimization

In addition to the objective function the optimization problem can be subjected to additional constraints. If at least one of these additional constraints is a Partial Differential Equation (PDE) this is called a PDE constrained optimization.

General Form:

The generalized form for our PDE constrained optimization problem will be:

$$\min_{\boldsymbol{u},\boldsymbol{\mu}} J(\boldsymbol{u},\boldsymbol{\mu}) \tag{3.14}$$

subject to

$$\boldsymbol{r}(\boldsymbol{u},\boldsymbol{\mu}) = 0 \tag{3.15}$$

where J is our objective function, $\boldsymbol{r}$ is our PDE problem in its residual form and $\boldsymbol{u}$ is the state variables.

By introducing a new variable $u^*(\boldsymbol{\mu})$ that is defined as the solution of the PDE for the given parameters $\boldsymbol{\mu}$ the constraint 3.15 can be removed as $\boldsymbol{r}(\boldsymbol{u}^*(\boldsymbol{\mu}),\boldsymbol{\mu}) = 0$ will always be true by definition. From this a new function

$$F(\boldsymbol{\mu}) := J(u^*(\boldsymbol{\mu}),\boldsymbol{\mu})$$

is introduced and the new optimization problem becomes

$$\min_{\boldsymbol{\mu}} F(\boldsymbol{\mu}). \tag{3.16}$$

This approach where the constraint of the PDE is remove from the optimization problem is called a reduced space approach.

Quasi-Newton method

The basic idea behind the Newton method for optimization is to approximate the function as a second degree polynomial using Taylor expansion and finding the minimum of this new function. By using this minimum as a new guess we get the iterative Newton method:

$$\boldsymbol{\mu}_n + 1 = \boldsymbol{\mu}_n - \boldsymbol{H}(\boldsymbol{\mu}_n)^{-1}\nabla F(\boldsymbol{\mu}_n). \tag{3.17}$$

However, computing the inverse of the hessian matrix $-\boldsymbol{H}(\boldsymbol{\mu}_n)^{-1}$ is both expensive computationally and requires a lot of memory. For this reason the a quasi-Newton method is often used instead. In the quasi-Newton method we replace the hessian matrix $\boldsymbol{H}_n^{-1}$ with $d_n\boldsymbol{B}_n$ where $\boldsymbol{B}$ is an approximation of the inverse hessian matrix computed using the gradient $\nabla F(\boldsymbol{\mu})$ and d is a distance computed using a one dimensional line search.

$$\boldsymbol{\mu}_{n+1} = \boldsymbol{\mu}_n - d_n\boldsymbol{B}_n\nabla F(\boldsymbol{\mu}_n) \tag{3.18}$$

The Matlab function `fminunc.m` was used to solve the optimization. This uses the *Broyden–Fletcher–Goldfarb–Shanno algorithm* (BFGS) [26] to calculate the B matrix and a cubic line search for finding the step size d [27]. In order to calculate the gradients ∇F both finite difference and the adjoint method was used.

3.5.2 Finite Differences

There are many ways of numerically calculating the gradient of a function. One of the simplest methods is the finite differences method. For this application a version of the finite difference method called the central difference method was used. This calculates the gradient as described in Equation (3.19).

$$\frac{dF(\boldsymbol{\mu})}{d\mu_i} = \frac{f(\boldsymbol{\mu} + \boldsymbol{\delta}_i\varepsilon) - f(\boldsymbol{\mu} - \boldsymbol{\delta}_i\varepsilon)}{2\varepsilon} \tag{3.19}$$

Here $\boldsymbol{\delta}_i$ is a vector that is 1 at index i and zero otherwise. ε was chosen between 10^{-6} and 10^{-8} depending on the problem. Central difference was simple to implement but computationally expensive as each derivative requires two function evaluations. To perform each function evaluation, in addition to solving the Navier-Stokes equation, a complete set of new weights for the forward model had to be computed to account for any changes caused by the mesh motions. Two measures was taken to somewhat shorten the additional time required for this calculations. First, as ε is very small the solver could be initialized with the original solution to significantly shorten the time it took for the Newton-Raphson method to converge. Second, each parameter μ_i could be calculated independently from each other enabling great parallelization of the gradient calculations.

3.5.3 Adjoint method

There are however more efficient ways of calculating the gradients of the function than using finite differences. Two such methods are the sensitivity approach and the adjoint method. These methods are quite similar and both uses the residual of the PDE for calculating the gradient of the function $F(\boldsymbol{\mu})$. A detailed derivation of these formulations can be found in [28] but a summary will be done below. First using the definitions from Equation (3.14) the gradient can be written as

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}(\boldsymbol{\mu}) = \frac{\partial J}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) + \frac{\partial J}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) \tag{3.20}$$

Because $\boldsymbol{u}(\boldsymbol{\mu})$ is defined so that $\boldsymbol{r}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) = 0$ is true, it will also be true that

$$\frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}\boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) = 0.$$

By applying the chain rule and rearranging the equation this can be used to calculate

$$\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) = -\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu})^{-1} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}). \tag{3.21}$$

This is called the sensitivity and by entering this into Equation 3.20 the gradient of the objective function can be calculated as:

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial J}{\partial \boldsymbol{\mu}} - \frac{\partial J}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^{-1} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}. \tag{3.22}$$

By solving this linear equation system for each free parameter $\boldsymbol{\mu}$ the sensitivity method can calculate the exact gradient from p linear solves of size $N \times N$ where p is the total number of parameters $\boldsymbol{\mu}$, i.e. $\boldsymbol{\mu} \in \mathbb{R}^p$ and N is the number of state variables of our PDE, i.e. $\boldsymbol{u} \in \mathbb{R}^N$. Even though this is much faster and more accurate than the finite difference method as seen in Section 3.5.2 it can still require a lot of calculations if the parameter set $\boldsymbol{\mu}$ is large. If that is the case the adjoint formulation can be used instead. This is acquired by transposing the term $\frac{\partial J}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^{-1}$ to get:

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial J}{\partial \boldsymbol{\mu}} - \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^{-T} \frac{\partial J}{\partial \boldsymbol{u}} \right)^{T} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}} = \frac{\partial J}{\partial \boldsymbol{\mu}} - \boldsymbol{\lambda}^{T} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}} \tag{3.23}$$

λ can then be calculated using Equation (3.24).

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^{T} \boldsymbol{\lambda} = \frac{\partial J}{\partial \boldsymbol{u}}^{T} \tag{3.24}$$

Equation (3.24) is called the adjoint equation and to get this solution we only have to solve the linear system once per constraint to our minimization problem. Unfortunately, due to the way that the forward function is calculated it proved quite difficult using adjoints when including mesh motions. Due to this the finite difference method, more specifically the central difference method shown in Equation (3.19), was used to calculate the gradients for all parameters $\boldsymbol{\mu}$ that would move the mesh.

3.5.4 Cost of Gradient calculations

In summary, three different ways of calculating the gradient of the objective function has been presented; the sensitivity method, the adjoint method, and the finite difference method. The sensitivity and adjoint method both have the advantage of theoretically calculating the exact gradient as well as only needing to solve linear equation systems which greatly improves performance. The finite difference method in contrast requires solutions of the non-linear equation system of the CFD simulation, which is significantly slower. In Table 3.4 is a summary of the number of solves required

Table 3.4: *Comparison of the cost of the different gradient calculations, p is the number of parameters $\boldsymbol{\mu}$, $\boldsymbol{\mu} \in \mathbb{R}^p$ and m is the number of constraints on the optimization problem.*

Method	operations
Adjoint	m-linear
Sensitivity	p-linear
Central Difference	2p-non-linear

As can be seen, if $p > m$ the adjoint method is faster than the sensitivity method, otherwise the sensitivity method is faster while central difference is theoretically always slower.

Chapter 4

Results

4.1 Mesh Sensitivity

The results from the mesh sensitivity analysis can be viewed below. The mesh sensitivity analysis was only done for the Aorta geometry and an overview of the different meshes can be seen in Table 4.1.

Table 4.1: *mesh statistics for the three meshes generated from the Aorta geometry and the number of nodes for the 2:nd and 3:rd order elements respectively.*

Mesh	**Elements**	**Mesh nodes**	
		2:nd order	**3:rd order**
1	718	1595	3469
2	1057	2312	4914
3	2316	4914	10846

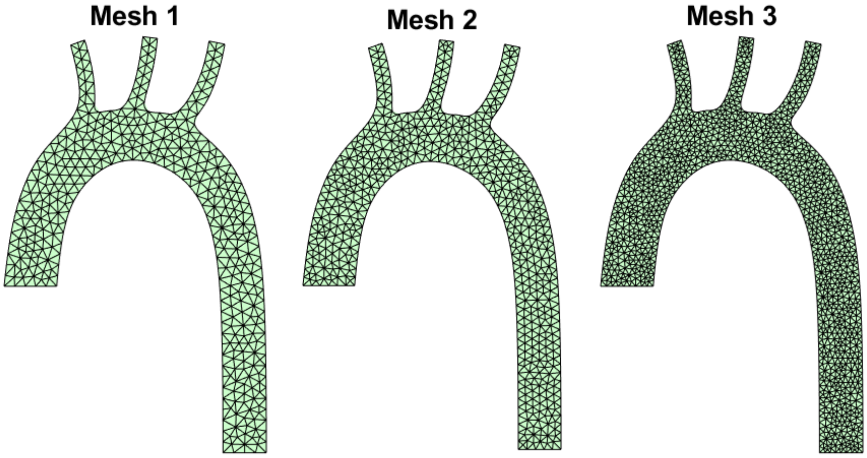


Figure 4.1: Comparison of the three different meshes used

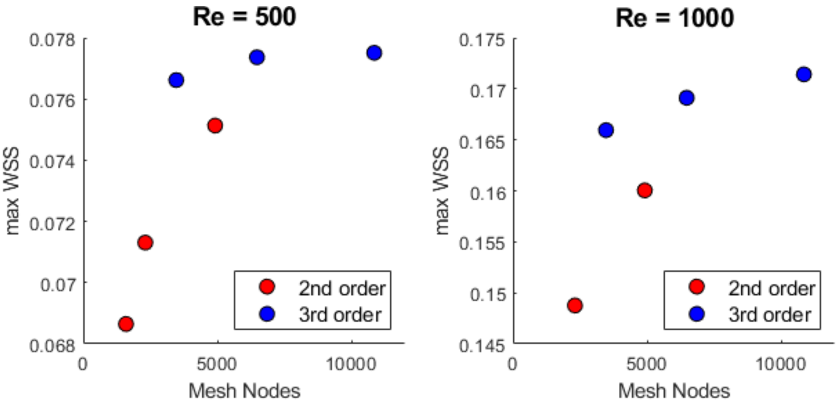


Figure 4.2: Max wall shear stress for each element order of each mesh. Done for Reynolds number of 500 and 1000.

Figure 4.2 shows the max wall shear stress for the different meshes as a function of the total number of mesh nodes in each mesh. The coarsest second order mesh did not converge for Re = 1000 and is therefore not included.

For the simulations with a Reynolds number of 500 the mesh appears to be very well matched between the finest (mesh 3) and coarsest mesh (mesh 1). At Reynolds number of 1500 some differentiation between the two meshes can be seen between mesh 1 and 3.

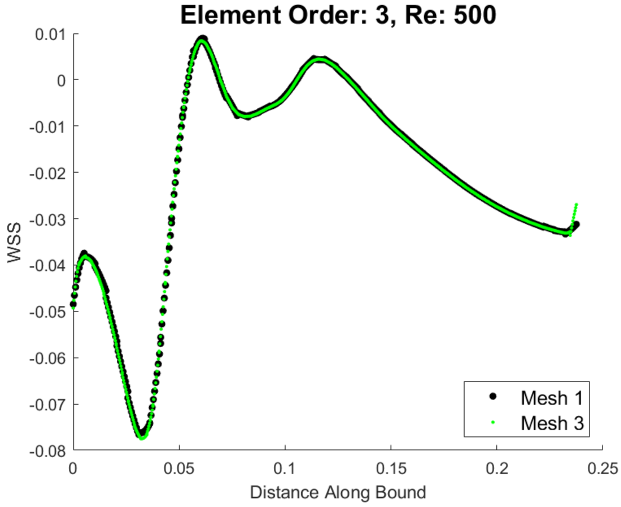


Figure 4.3: Comparison of the wall shear stress along the lower wall of the aorta mesh for the coarsest mesh (mesh 1) and the finest mesh (Mesh 3) using an element order of 3 and Reynolds number of 500

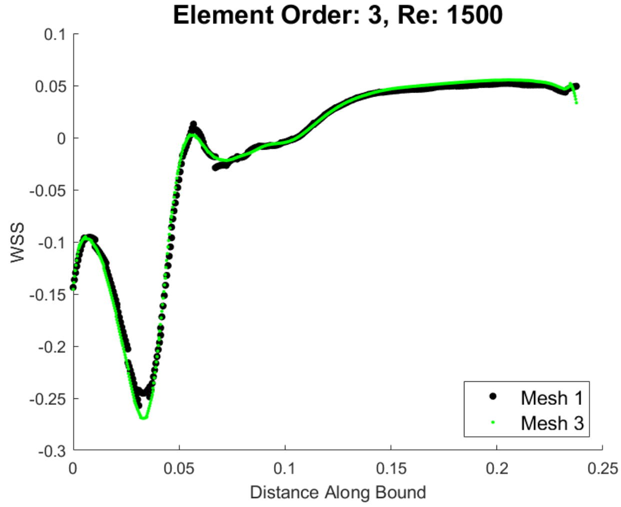


Figure 4.4: Comparison of the wall shear stress along the lower wall of the aorta mesh for the coarsest mesh (mesh 1) and the finest mesh (Mesh 3) using an element order of 3 and Reynolds number of 1500

As mentioned in Section 3.2.1 the focus is not on accuracy of the CFD simulations so the meshes used was generally chosen based on performance rather than results. In general mesh one with 3:rd order elements and mesh 2 with 2:nd order elements was used for SBI simulations for performance reasons while mesh 3 of corresponding element order was used for the reference flow. Because mesh 1 using 2:nd order elements did not converge for Reynolds number higher than 500 this mesh was not used for any simulations.

4.1.1 Simulation times

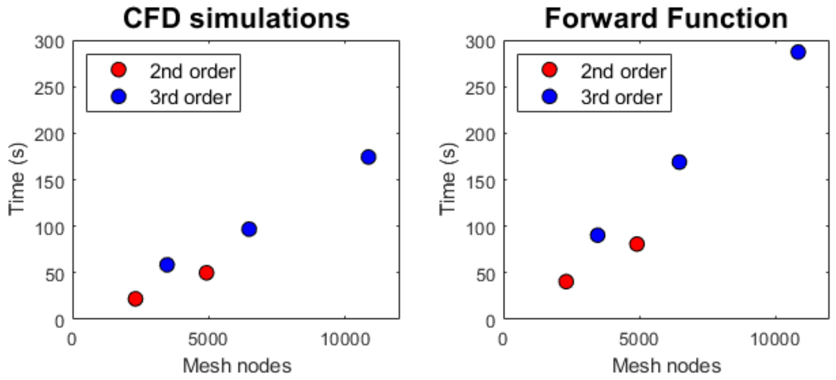


Figure 4.5: Comparison of computing times for CFD solution and forward model using both quadratic and cubic mesh elements. Nodes here refer to the number of mesh nodes.

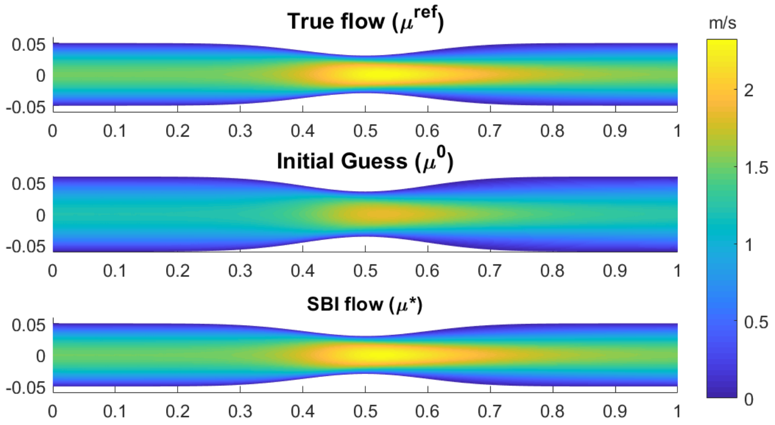
In Figure 4.5 computing times has been plotted for the two most time consuming operations of the framework for the different meshes used. Both the forward function and the CFD solution has to be calculated each time the objective function is evaluated. For the optimization the objective function is calculated once for every function evaluation and twice per gradient evaluation for each parameter μ_n that can not be calculated through the adjoint method. All calculations were done one a compute server with two 22 core Intel Xeon 6152 (Santa Clara, USA) processors and 384GB of RAM. The underlying code was single threaded only using one core for solving solving our CFD simulations but for the gradient calculations using central difference one core was used for each parameter μ, which allowed us to use much more of the available resources and significantly speeding up the optimization process. The total memory required for larger simulations was usually around 100GB.

4.2 Case 1, Diameter

Case 1 was the first and simplest case run. The primary purpose of Case 1 were early development of the basic functions and to validate if simultaneous shape and velocity optimization was possible using the framework. Figure 4.6 shows the reference flow, initial guess and final SBI flow for a run using the initial values and reference values seen in Table 4.6a. Here can be see the difference in inlet velocity and diameter for this simple channel between the initial guess and final SBI flow. Table 4.6a shows some results from repeated runs of the same case but with different noise levels added to the MR-image.

(a) *Case setup for Case 1 and two runs with different noise levels*

Case 1			
General: Re 100, 2:nd order mesh			
$\boldsymbol{\mu}$	v_x^{inlet}	v_y^{inlet}	D_y
$\boldsymbol{\mu}^{\boldsymbol{ref}}$	1.5	0	1
$\boldsymbol{\mu}^0$	1.2	0	1.2
Run 1: 0% MR noise			
$\boldsymbol{\mu}^*$	1.49	0.000754	1.00556
Error	0.00991	-0.000754	-0.00557
Run 2: 15% MR noise			
$\boldsymbol{\mu}^*$	1.536	-0.190	0.954
Error	-0.0362	0.190	0.0460



(b) Comparison of the CFD solution for the true flow, initial guess and final SBI flow for run 1.

Figure 4.6: Setup, reference solution, initial guess and final SBI flow

Figure 4.7 shows the objective function for the parameters D_y and v_x^{inlet} plotted for an area close to the true value of $D_y = 1$ and $v_x^{inlet} = 1.5$. As all of these simulations were run with the y- component of the inlet velocity $v_y^{inlet} = 0$ this parameter was left out of the plot to enable 2D-plotting. It is generally very expensive to calculate all the data points required of the objective function needed for this kind of plot but it was a very useful tool when writing the code as it allowed for easy validation of the methods and solvers.

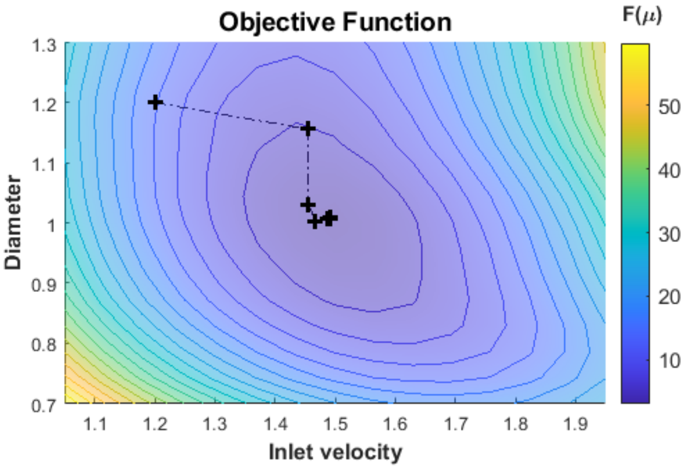


Figure 4.7: Value of objective function (see Equation 3.1) for the inlet velocity and diameter of the simple Stenosis. The target diameter was $D_y = 1$ and the target inlet velocity was $v_x^{inlet} = 1\text{m/s}$. The black + markers are the iterations for the optimization with the initial guess of $D_y = 1.2$ and $v_x^{inlet} = 1.2$.

With the results seen in Table 4.6a together with the information gained from plotting the objective function in Figure 4.7 it was concluded that the initial stage of shape optimization was working good enough. With added noise to the MR-image the final error increases, but for this simple case the results was still quite good.

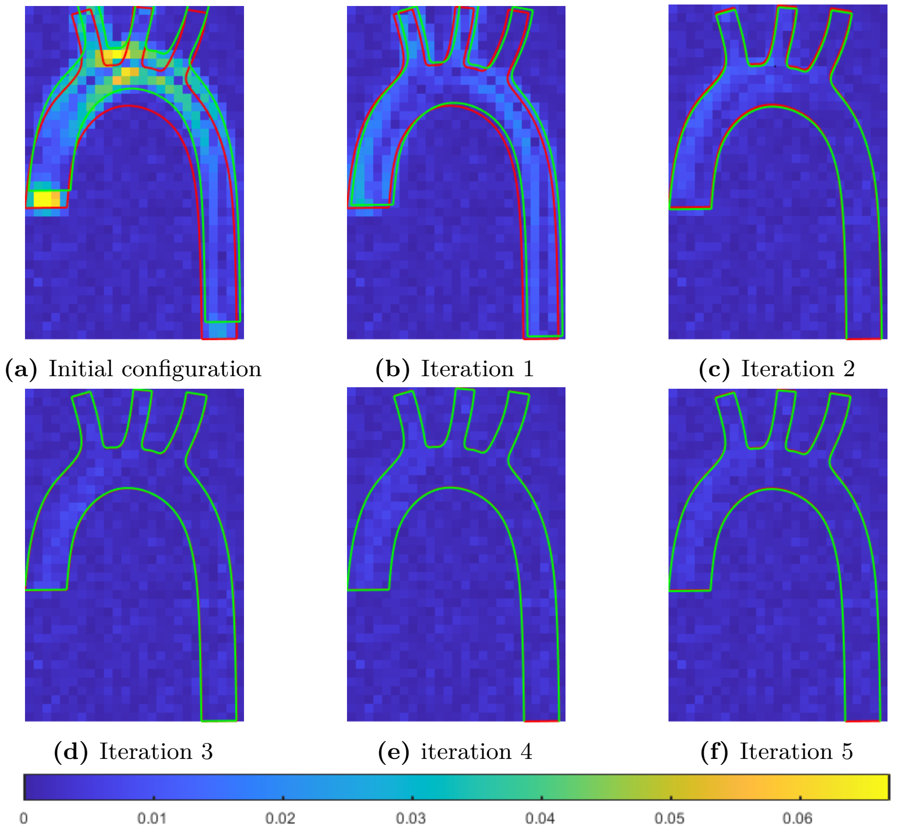
4.3 Case 2, Rotation and Translation

Table 4.2: *Rotation, translation and inlet velocity*

Case 2	
$\boldsymbol{\mu}$	v_x^{inlet}, v_y^{inlet}, $d\theta$, dx, dy
Geometry	Aorta

The second case was an investigation into if an initial offset and rotation of the CFD simulation compared to the ground truth could be found. This was motivated from the idea that patients could move during scanning, creating an error between the geometry and the actual position for the ground truth for example between different time frames.

Figure 4.8 shows the difference in velocity magnitude between the MR-flow image ($\mathbf{v}_{MR}$) and the forward function of the CFD simulation ($\mathbf{M}\mathbf{v}_{CFD}^{\boldsymbol{\mu}^n}$) for each voxel. These plots will be used several times and are called *velocity error plots*. The red lines outline our reference solution and the green lines the outline of the SBI flow. Figure 4.8a shows the configuration for the initial guess with a total offset of 2 voxels (10mm).



(a) Initial configuration **(b)** Iteration 1 **(c)** Iteration 2

(d) Iteration 3 **(e)** iteration 4 **(f)** Iteration 5

(g) velocity error (m/s)

Figure 4.8: progression of the first 5 optimization steps when optimizing over rotation, translation and velocity. Green outline is the CFD geometry and red is the target *Truth* geometry.

Figure 4.9 shows the results from a study to evaluate how the noise level of the MR-flow image $\mathbf{v}_{MR}$ influence the performance of the optimization. Noise levels in the MR-image from 0%-25% were tested. For each noise level, the optimization was run at least 10 times with different initial guesses. The initial offset error was set to two voxels in distance (10mm total) in a random direction. The rotation error was set at random to either 1 or -1 degree and the inlet velocity was set to 5% over the true value.

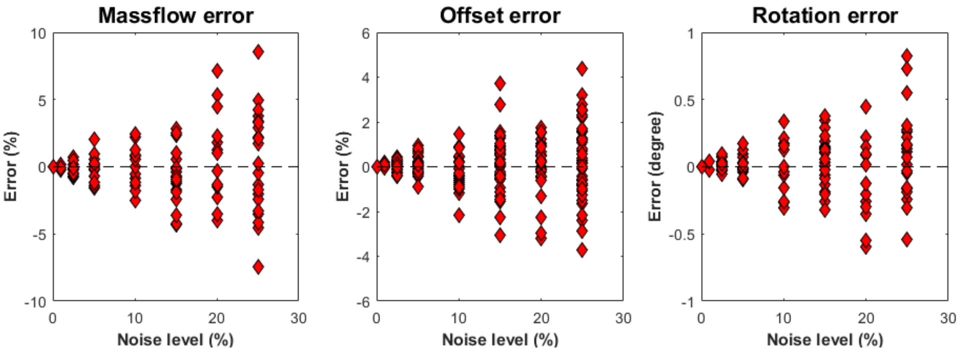


Figure 4.9: Result from study on the influence of noise in the MRI on final error for mass flow, translation (offset), and rotation. Initial mass flow error was 5%, initial offset error was 40%(10mm) and initial rotation error was 1 degree.

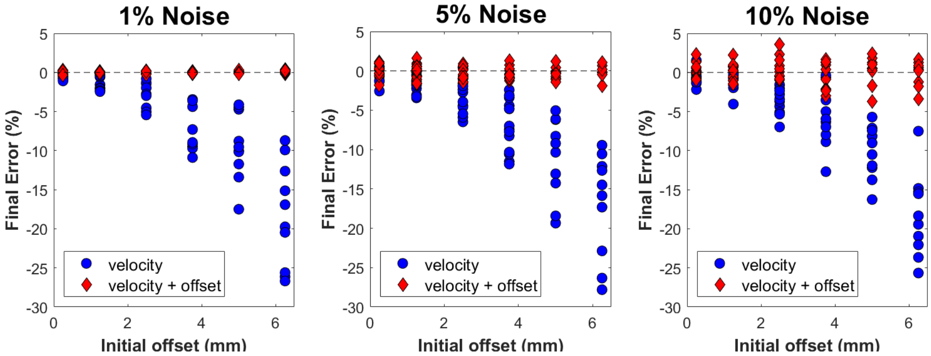


Figure 4.10: Effect of initial offset Comparison of mass flow error on the inlet between optimizing over translation versus not optimizing over translation for noise levels of 1%, 5% and 10%.

Figure 4.9 illustrates how the final error is affected by the noise level when optimizing for both inlet velocity and offset. In order to get a better understanding of how this compared to only optimizing over inlet velocity a second study was performed. This was done by varying the initial offset error between 0.25mm and 6.25mm. Each offset distance was run 10 times with a random direction for the offset. Each direction was then run twice with the same initial guess: first only optimizing for inlet velocity and then optimizing for both inlet and offset. This was repeated for 1, 5 and 10 % noise in the MR-image ($\mathbf{v}_{MR}$). In Figure 4.10 the result from this second study is displayed. Red dots are the cases that were run with optimization over both offset and inlet velocity, and blue dots are the cases that were only optimized for inlet velocity.

The results displayed in Figure 4.9 and 4.10 indicates that optimizing offset in addition to inlet velocity could be beneficial even for quite small initial offsets. Figure 4.10 shows that the mass flow error steadily increases when not optimizing for the offset while the error for the runs optimizing for offset appears to only be dependent on the noise level.

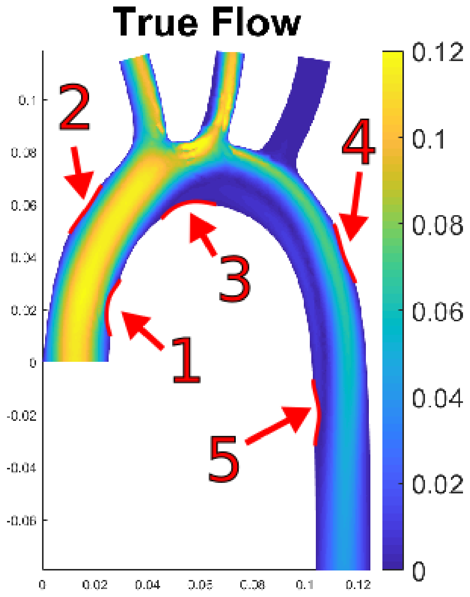
4.4 Case 3

The final stage of the project was to investigate how the framework would handle small perturbations of the boundary. This was done in two steps. The first was to implement Radial Basis Functions (RBFs) that could deform an area of the domain and explore how the framework handles small deformations from 5 RBFs spread evenly along the the boundaries in Case 3a (see Figure 4.11). The second stage was to investigate if it is possible to find the degree of blockage from an Aortic Coarctation using two RBFs to emulate the narrowing of the aorta (see Figure 4.14).

4.4.1 Case 3a: Boundary Perturbation

(a) *Case setup*

Case 3a	
$\boldsymbol{\mu}$	v_x^{inlet}, v_y^{inlet}, dx_n, dy_n; $1 \leq n \leq 5$
Geometry	Aorta



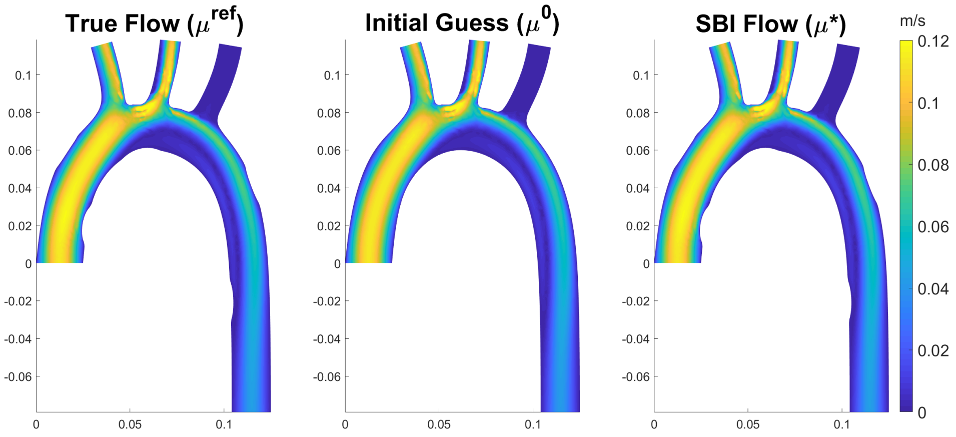
(b)

Figure 4.11: General setup for Case 3a with inlet and outlet velocity and on to five radial basis functions resulting in a total of 4-12 free parameters $\boldsymbol{\mu}$

The overall setup of Case 3a can be seen in Figure 4.11. Different runs using 5 RBFs numbered according to Figure 4.11b was done. The results from this run can be seen in Figure 4.12. This Figure shows a comparison between the true flow, initial guess and final SBI flow. The location of each RBF was spread out to different places along the walls to see how well the optimization would find the displacement for each RBF with varying flow conditions.

(a) *Inlet veloity and a number of radial basis functions*

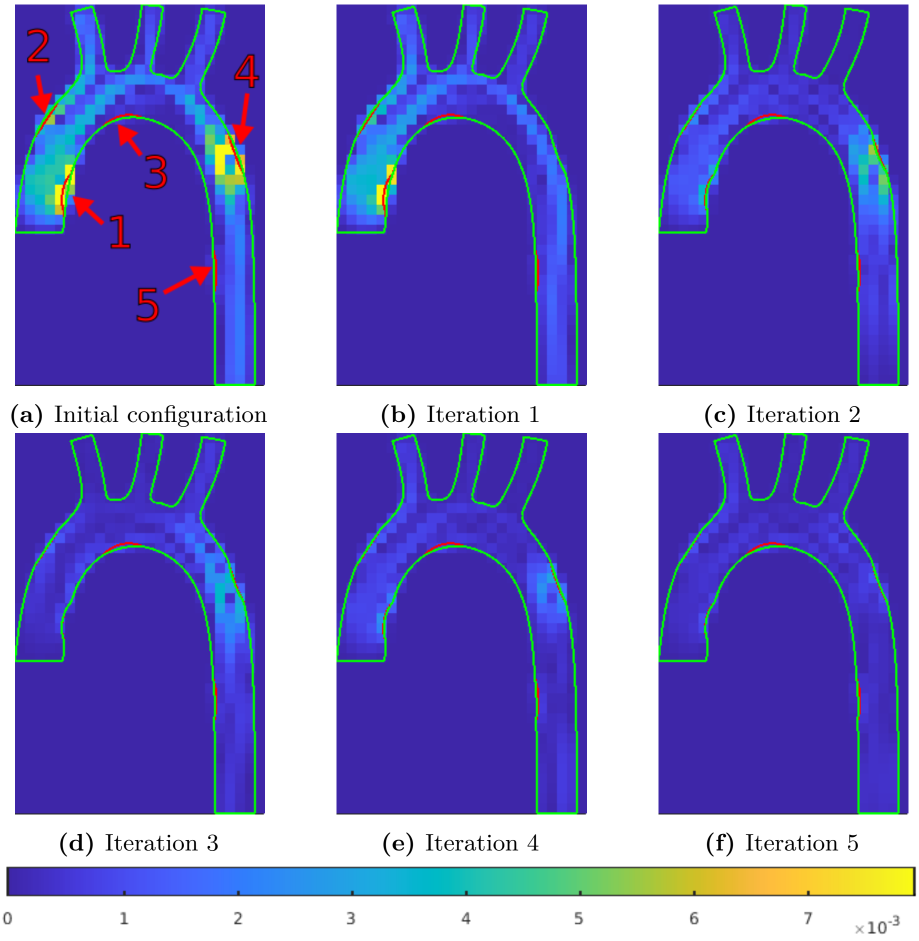
	Case 3a											
General	0% noise, Re 1000, 2:nd oreder mesh											
$\boldsymbol{\mu}$	v_x^{in}	v_y^{in}	dx_1	dy_1	dx_2	dy_2	dx_3	dy_3	dx_4	dy_4	dx_5	dy_5
Units	cm/s		mm	mm	mm	mm	mm	mm	mm	mm	mm	mm
$\boldsymbol{\mu}^{ref}$	0	113	-2.6	0.37	1.5	0	0	1.5	-1.5	0	1.5	0
$\boldsymbol{\mu}^{*}$	0	113	-2.6	0.37	1.5	-2e-3	-4e-3	1.5	-1.5	-1e-3	1.5	-0.1
error	0	0	6e-6	1e-4	2e-3	2e-3	4e-3	8e-4	2e-4	1e-3	7e-3	0.1



(b) Comparison between the CFD solution for the True flow, initial guess and final SBI flow. Displayed is the velocity magnitude in m/s.

Figure 4.12: General setup for Case 3a with inlet and outlet velocity and on to five radial basis functions resulting in a total of 4-12 free parameters $\boldsymbol{\mu}$

Figure 4.13 shows the outline of the reference geometry (red) and CFD geometry (green). The velocity error plot for Case 3a is shown to help distinguish the influence of each RBF. The location for the RBFs are marked with arrows in Figure 4.13a as well.



(a) Initial configuration **(b)** Iteration 1 **(c)** Iteration 2

(d) Iteration 3 **(e)** Iteration 4 **(f)** Iteration 5

(g) Velocity magnitude error (m/s)

Figure 4.13: In this figure is plotted the velocity error for each voxel for the initial configuration and the first 5 optimization steps. The outline of the CFD geometry is plotted in green and outline for the target reference geometry is plotted in red.

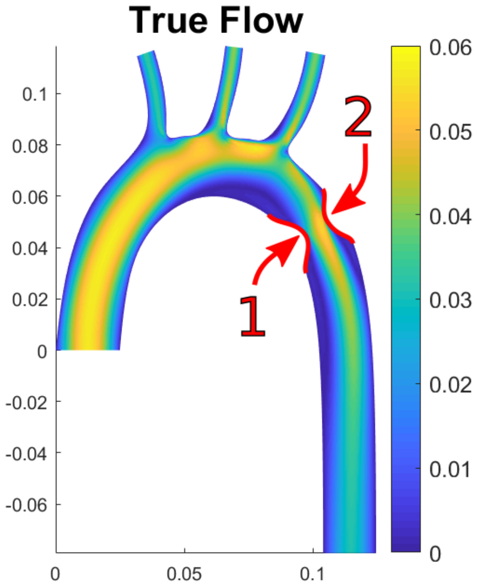
Figure 4.13a shows that RBF 1, 2 and 4 appears to have the largest affect on the error and Figure 4.13d shows that good approximations for these are found after only 3 iterations. On the other hand RBF 3 and 5 have less influence on the flow and are therefore harder to find. Especially RBF 3 is practically unchanged even after 5 iterations. Figure 4.11b shows that RBF 3 is located just after the flow detaches from the lower boundary. Because the optimizer only compares differences in flow velocity it is very difficult to find a good approximation for RBF 3.

4.4.2 Case 3b: Aortic Coarctation

For Case 3b the goal was to see if a similar setup could be used to find the degree of blockage from an aortic coarctation. In Figure 4.14 can be seen the general setup for Case 3b. Compared to Case 3a (see Figure 4.11b), Case 3b has fewer free parameters and the offset caused by each was set to a larger value (see Table 4.3 and Figure 4.14b).

(a) Configuration for Case 3b.

Case 3b	
Geometry	Aorta
$\boldsymbol{\mu}$	v_x^{inlet}, v_y^{inlet}, dx_1, dy_1, dx_2, dy_2



(b) Example of a reference flow for Case 3b with the bumps caused by the two radial basis functions marked.

Figure 4.14: General setup Case 3b with inlet and outlet velocity and two radial basis functions resulting in a total of 6 free parameters $\boldsymbol{\mu}$.

In in Table 4.3 can be seen the results from one run of Case 3b with no noise added to the MR-image and a Reynolds number of 500. Figure 4.15 shows the CFD solution for the true configuration, the initial guess and the final SBI flow for this run. This shows how the narrowing from the Aortic Coarctation forces the flow to accelerate. It also shows how the flow detaches from the lower wall.

To visualize the optimization process the velocity error plots for the initial configuration and subsequent 5 iterations are shown in Figure 4.16. The outline of the SBI geometry (in green) and the reference configuration (in red) are overlayed the velocity error plots.

Table 4.3: *Configuration for Case 3b. with inlet and outlet velocity and two radial basis functions as the free parameters $\boldsymbol{\mu}$, the true value $\boldsymbol{\mu}^{ref}$, and initial guess $\boldsymbol{\mu}^0$.*

Case 3b						
General	0% noise, Re 500, 3:rd order mesh					
$\boldsymbol{\mu}$	v_x^{inlet}	v_y^{inlet}	dx_1	dy_1	dx_2	dy_2
Unit	cm/s	cm/s	mm	mm	mm	mm
$\boldsymbol{\mu}^{ref}$	0	5.66	4.9	2.8	-4.9	-2.8
$\boldsymbol{\mu}^{0}$	0	2.83	1.5	0.84	-1.5	-0.84
$\boldsymbol{\mu}^{*}$	-0.008	5.66	4.52	3.62	-5.07	-2.47
Error (abs)	-0.008	0.0015	0.376	-0.822	0.169	-0.331

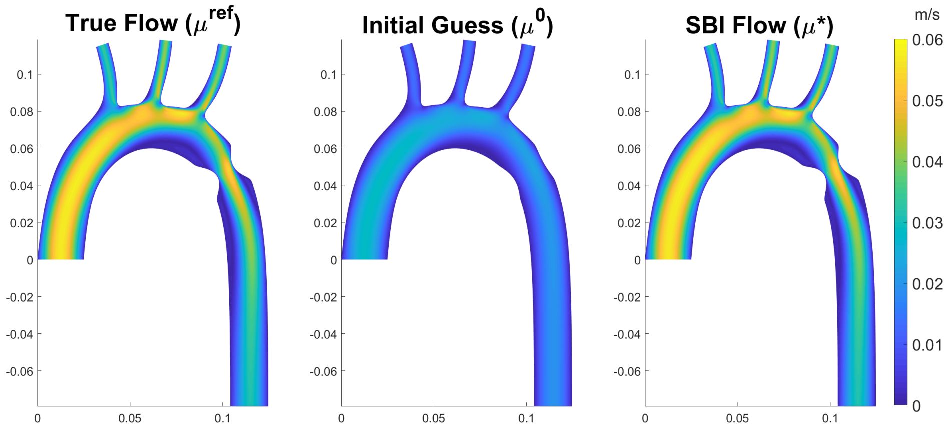
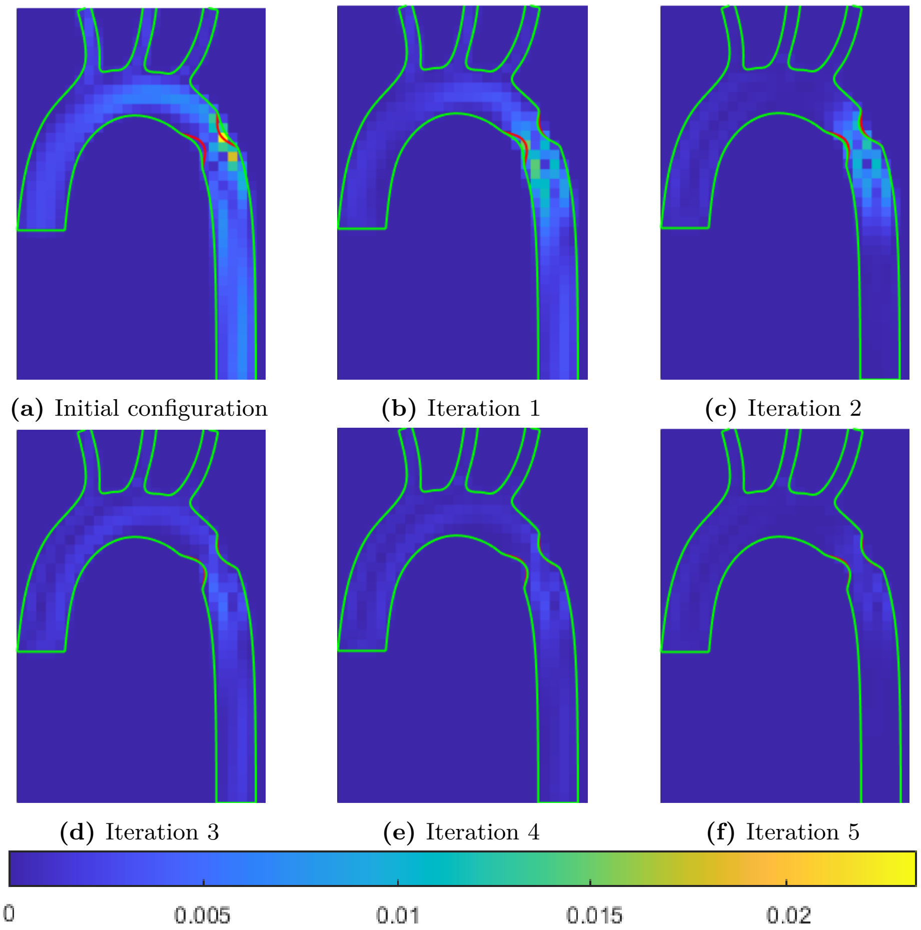


Figure 4.15: Comparison between the CFD solution for the True flow, initial guess and final SBI flow. Displayed is the velocity magnitude in m/s. In these simulation a finer mesh was used for the Truth simulation than the SBI simulation.



(a) Initial configuration
(b) Iteration 1
(c) Iteration 2
(d) Iteration 3
(e) Iteration 4
(f) Iteration 5
(g) Velocity magnitude error (m/s)

Figure 4.16: Alternative 2/3 The initial configuration and progression of first 5 optimization steps for a shape optimization using 2 RBFs. Green outline is the CFD geometry and red is the target *Truth* geometry.

Figure 4.16a shows the initial configuration with an error caused by the initial underestimation in inlet velocity. It also shows the velocity difference caused by the coarctation. Another observation is that the optimizer finds a good approximation for RBF 2 already in iteration 1 while it is slower to do so for RBF 1. In Case 3a we saw how the framework had a hard time finding the correct offset for RBFs that was in slow or stationary flow compared to the max velocity (see Figure 4.13) and this appears to be the case in Case 3b as well.

Table 4.3 shows the final error for each parameter $\boldsymbol{\mu}$. To get a better idea of how good these results are the Wall Shear Stress along the lower boundary was plotted for the initial guess and final SBI flow together with the reference solution in Figure 4.17.

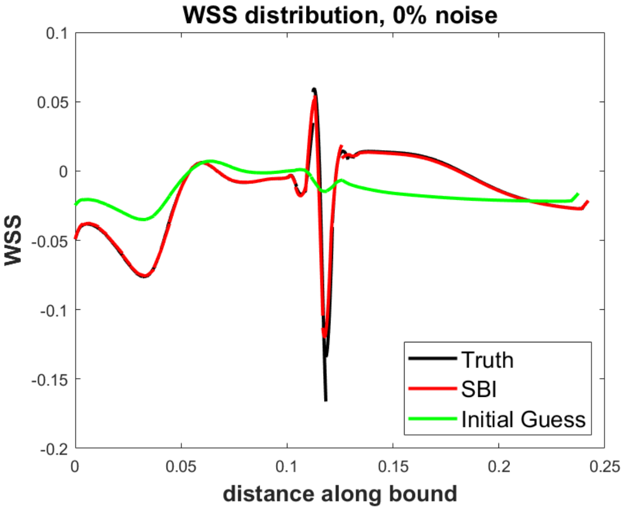
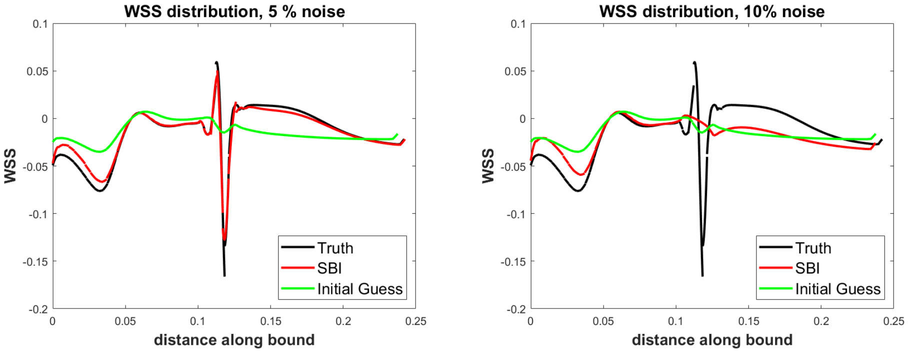


Figure 4.17: Wall Shear Stress Comparison Here is plotted the Comparison of the Wall Shear Stress distribution along the lower boundary before and after optimization.

To get a quick idea of how the setup in Case 3b handle noise the wall shear stress for the same setup but with 5 and 10 % noise respectively added to the MR-image can be seen in Figure 4.18a and 4.18b.



(a) **(b)**

Figure 4.18: Same setup as in Figure 4.17 but with 5 and 10 % noise.

Figure 4.18a and 4.18b shows how the match in wall shear stress between the SBI flow and the reference flow is still good for for 5% noise but for 10% noise no good approximation for RBF 1 was found from the optimization.

Looking at Figure 4.15 it appears that just like for RBF 3 in Case 3a this RBF is located in an area of relatively slow flow as the faster flowing fluid tends towards the outside of the bend. If the initial guess for RBF 1 is too conservative it has a very low impact on the flow and when the initial MRI is very noisy these small variations are not enough to find a good approximation for the RBF.

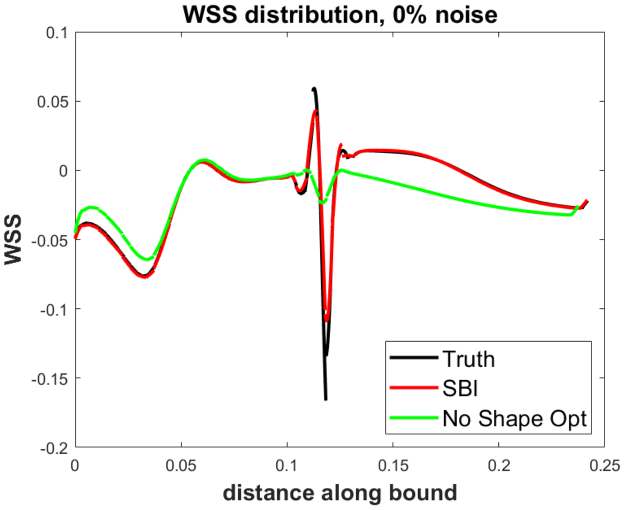


Figure 4.19: Wall Shear Stress Optimization Comparison Here is plotted the Comparison of the Wall Shear Stress distribution along the lower wall before and after optimization. In this figure is plotted a comparison between optimizing over only inlet velocity and optimizing over coarctation as well.

Finally, to better quantify the potential improvement from optimizing the coarctation and velocity simultaneously Figure 4.19 shows a comparison of optimization with and without optimization of the coarctation using the same initial guess as seen in Figure 4.15. Figure 4.15 shows how the test without shape optimization differs from the reference both directly at the inlet and after the coarctation while the test using shape optimization matches the reference solution much better. This illustrates how localized errors in the geometry can have both upstream and downstream effects on the final simulation.

Chapter 5

Conclusion

The purpose of this thesis was to explore if it is possible to correct for different errors in the geometry used for the Simulation Based Imaging. We have done this in three main cases investigating errors in the diameter of a channel, rotation and translation, as well as a more granular approach using Radial Basis Functions for deforming the domain. These cases has then been tested for robustness based on the initial magnitude of the error as well as the amount of noise in the initial MR-image.

Case 1 showed that geometry correction based on MR-flow data was possible for a very simple case. Case 2 demonstrated the possibility of correcting for a rotation and translation offset of the geometry compared to the MR-image. Optimizing over rotation and translation successfully removed any errors caused by the offset and the accuracy of the final SBI was only dependent on the noise level.

Case 3a and 3b was the most complex of the cases and showed both promise as well as identifying some potential issues and problems to the approach. While for both Case 3a and 3b we managed to demonstrate that the framework could find the initial errors to the outer walls. They also showed that moving the boundaries that was in stagnating or slower moving flow conditions was more difficult. Any deformations located within these flow conditions will be difficult to find as they have a very low impact on the flow and therefore as the MR-flow image only contains information about the flow velocity. As seen in Case 3b the introduction of noise makes this issue even harder as the noise can hide the small effects the deformed bound still has.

5.1 Future Work

There are several interesting future steps and investigations to be done to evaluate the framework. For example this thesis put a lot of emphasis on how the noise of the MR-flow image affected the different geometry corrections but not how different MR-flow voxel resolutions would affect the performance and accuracy. This could easily be conducted without any need to further expand the framework and could provide further insight into the possibilities and limitations of shape optimization.

As mentioned the current framework does not support calculating the gradient for the mesh motions using the adjoint method. To improve performance and enable using more parameters to control the boundary implementing this support would be very beneficial.

The ultimate goal is to evaluate the methods in both 3D and with time dependency. Due to limitations in the efficiency of the current code adding time dependency would probably be more feasible than 3D in the near future and could also provide more interesting results for the shape optimization.

Bibliography

[1] Gregory A Roth, Degu Abate, Kalkidan Hassen Abate, Solomon M Abay, Cristiana Abbafati, Nooshin Abbasi, Hedayat Abbastabar, Foad Abd-Allah, Jemal Abdela, Ahmed Abdelalim, et al. Global, regional, and national age-sex-specific mortality for 282 causes of death in 195 countries and territories, 1980–2017: a systematic analysis for the global burden of disease study 2017. *The Lancet*, 392(10159):1736–1788, 2018.

[2] Daniel J Holland, Dmitry M Malioutov, Andrew Blake, Andrew J Sederman, and LF Gladden. Reducing data acquisition times in phase-encoded velocity imaging using compressed sensing. *Journal of magnetic resonance*, 203(2):236–246, 2010.

[3] Naoyuki Kimura, Masanori Nakamura, Kenji Komiya, Satoshi Nishi, Atsushi Yamaguchi, Osamu Tanaka, Yoshio Misawa, Hideo Adachi, and Koji Kawahito. Patient-specific assessment of hemodynamics by computational fluid dynamics in patients with bicuspid aortopathy. *The Journal of Thoracic and Cardiovascular Surgery*, 153(4):S52–S62, 2017.

[4] Jonas Lantz, Johan Renner, Toste Länne, and Matts Karlsson. Is aortic wall shear stress affected by aging? an image-based numerical study with two age groups. *Medical Engineering & Physics*, 37(3):265–271, 2015.

[5] Töger J, Zahr M, Aristokleous N, Markenroth Bloch K, Carlsson M, Persson P. Blood flow imaging by optimal matching of computational fluid dynamics to 4D-flow data. *Magnetic Resonance in Medicine*, 84(October):2231–2245, 2020.

[6] Simon Wolfgang Funke, Magne Nordaas, Øyvind Evju, Martin Sandve Alnæs, and Kent Andre Mardal. Variational data assimilation for transient blood flow simulations: Cerebral aneurysms as an illustrative example. *International journal for numerical methods in biomedical engineering*, 35(1):e3152, 2019.

[7] William Harvey et al. Exercitatio anatomica de motu cordis et sanguinis in animalibus. *Frankfurt am Main*, 1628, 1928.

[8] Nationalencyklopedin. blodomlopp. `http://www.ne.se/uppslagsverk/encyklopedi/lång/blodomlopp`. Retrieved 2021-02-14.

[9] Elles J Dijkema, Tim Leiner, and Heynric B Grotenhuis. Diagnosis, imaging and clinical management of aortic coarctation. *Heart*, 103(15):1148–1155, 2017.

[10] Michael D Hope, Alison K Meadows, Thomas A Hope, Karen G Ordovas, David Saloner, Gautham P Reddy, Marcus T Alley, and Charles B Higgins. Clinical evaluation of aortic coarctation with 4d flow mr imaging. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 31(3):711–718, 2010.

[11] Norrving B Mendis S, Puska P. Global atlas on cardiovascular disease prevention and control. *World Health Organization in collaboration with the World Heart Federation and the World Stroke Organization*, page 3, 2011.

[12] Yoram Richter and Elazer R Edelman. Cardiology is flow, 2006.

[13] Adel M Malek, Seth L Alper, and Seigo Izumo. Hemodynamic shear stress and its role in atherosclerosis. *Jama*, 282(21):2035–2042, 1999.

[14] Connor V Cunnane, Eoghan M Cunnane, and Michael T Walsh. A review of the hemodynamic factors believed to contribute to vascular access dysfunction. *Cardiovascular Engineering and Technology*, 8(3):280–294, 2017.

[15] Johannes Töger. *Measurement and analysis of intracardiac blood flow and vortex ring formation*. Lund University, 2014.

[16] EL Hahn. Detection of sea-water motion by nuclear precession. *Journal of geophysical research*, 65(2):776–777, 1960.

[17] GL Nayler, DN Firmin, DB Longmore, et al. Blood flow imaging by cine magnetic resonance. *J Comput Assist Tomogr*, 10(5):715–722, 1986.

[18] Dyverfeldt P, Bissell M, Barker A, Bolger A, Carlhäll C, Ebbers T, Francios C, Frydrychowicz A, Geiger J, Giese D, Hope M, Kilner P, Kozerke S, Myerson S, Neubauer S, Wieben O, Markl M. 4D flow cardiovascular magnetic resonance consensus statement. *Journal of Cardiovascular Magnetic Resonance*, 17(1):1–19, 2015.

[19] W. Malalasekera H.K. Versteeg. *An Introduction to Computational Fluid Dynamics*. Pearson Educational Limited, 2007. [Lund].

[20] Johan Renner. Towards subject specific aortic wall shear stress. Technical report, Citeseer, 2011.

[21] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009.

[22] Vinicius C Rispoli, Jon F Nielsen, Krishna S Nayak, and Joao LA Carvalho. Computational fluid dynamics simulations of blood flow regularized by 3d phase contrast mri. *Biomedical engineering online*, 14(1):1–23, 2015.

[23] Samar A Mahrous, Nor Azwadi Che Sidik, and Khalid M Saqr. Newtonian and non-newtonian cfd models of intracranial aneurysm: a review. *CFD Letters*, 12(1):62–86, 2020.

[24] Julio Garcia, Roel LF Van Der Palen, Emilie Bollache, Kelly Jarvis, Michael J Rose, Alex J Barker, Jeremy D Collins, James C Carr, Joshua Robinson, Cynthia K Rigsby, et al. Distribution of blood flow velocity in the normal aorta: effect of age and gender. *Journal of Magnetic Resonance Imaging*, 47(2):487–498, 2018.

[25] M.J. Zahr. Finite element course: Chapter 7. University of Notre Dame, Dept Aerospace & Mechanical Engineering.

[26] Charles George Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, 1970.

[27] Matthworks help center, fminunc, find minimum of unconstrained multivariable function. `https://se.mathworks.com/help/optim/ug/fminunc.html#bqbnbyu-1`. Hämtad 2021-03-31.

[28] Matthew Joseph Zahr. *Adaptive model reduction to accelerate optimization problems governed by partial differential equations*. PhD thesis, Stanford University, 2016.