SENSITIVITY ANALYSIS OF SIMULATED BLOOD FLOW IN CEREBRAL ANEURYSMS

by

ØYVIND EVJU

THESIS
for the degree of
MASTER OF SCIENCE

(Master i Anvendt matematikk og mekanikk)

Faculty of Mathematics and Natural Sciences
University of Oslo

August 2011

Det matematisk- naturvitenskapelige fakultet
Universitetet i Oslo

Preface

This thesis was written between August 2010 and August 2011 as part of a masters degree in applied mathematics. It is a thesis stretching across several scientific disciplines, such as mathematics, medicine, fluid dynamics and programming. Quite a bit of source code have been developed, mainly in Python, along with a few code snippets in C++. The code is built using the Python interface of the finite element library FEniCS[1]. The complexity of the source code rarely exceeds that of *A FEniCS Tutorial*[2], but if the reader has no prior knowledge of FEniCS, this is recommended reading if one wants to thoroughly understand the source code.

In the big picture, the aim of this thesis is to improve the understanding of blood flow in parts of the brain, and the development of a vascular decease called aneurysms. This is a vast area of research, and thus, the approach to this problem is much more specific. The main objective of this thesis changed through the process. From what started out as a pure study of the non-Newtonian effects of blood flow, it has ended up with a quite comprehensive sensitivity analysis of different assumptions that has to be made when using computational models to simulate blood flow in and around aneurysms.

With this thesis a two year masters degree comes to an end. I would especially like to thank my supervisors Kent-Andre Mardal and Kristian Valen-Sendstad for invaluable input through this last year, and for material such as CT-images, meshes and code snippets needed to produce the results of this thesis. A big thanks also goes to Simula Research Laboratory for all the computing resources made available to me.

This also marks the end for six years of studies, and I would like to express my gratitude to my fellow students Christian, Magne, Gabriela and Eline (among others), for many good discussions and a great social environment through these years. A special thanks goes to Torunn for keeping my spirits up these last few months. Finally, a big thanks goes to my sister Ingrid for setting the bar, and to my parents, Astrid and Åge, for all their love and support through the years.

Øyvind Evju
August, 2011

Contents

1. Introduction

An aneurysm is an abnormal bulge of a blood vessel, which often occurs in arteries in the vicinity of the Circle of Willis, part of the brains blood supply. These cerebral aneurysms may grow and occasionally rupture, causing a serious condition called subarachnoid hemorrhage (stroke), i.e., bleeding in and around the brain. This condition often leads to serious brain damage or death[3,4]. Worldwide, there are about 10.5 cases per 100,000 person years of subarachnoid hemorrhage caused by aneurysm rupture[5].

In Section 2 the medical background of aneurysms and blood flow is outlined, and factors contributing to aneurysm development are presented. It is not known exactly what triggers aneurysms to initiate, grow and occasionally rupture. In an attempt to better our understanding of cerebral aneurysm development, numerical models may be used to simulate the blood flow and arteries in and around the Circle of Willis.

To model blood flow, one needs to solve the Navier-Stokes equations. These equations arise from the simple principles of conservation of energy, momentum and mass, and are truly magnificent in that they seem to model any fluid qualitatively correct. In Section 3, the Navier-Stokes equations are derived.

In this thesis, a Navier-Stokes solver has been modified from the project *nsbench*[1]. The implementation is done as described in Section 4, and the implementation is verified by comparing to exact solutions in Section 5. Throughout the text small code snippets are found to illustrate the implementation of problems discussed, and in Appendix A some larger sections of code can be found. All the source code, as well as animations of the simulations done can be found at `http://folk.uio.no/oyvinev/master`.

By using computer models to simulate the blood flow in arteries, much more information can be gathered than what physical experiments can provide. Both the spatial and temporal resolution of the simulations are far superior to any measurement methods currently available. In simulations, both the pressure and the velocity field is known at any point of the flow domain. These simulations can be done causing minimal disturbance to the patient, as only a digital 3D-image[2] of the area of interest is needed to perform simulations. This also minimizes both the geographical and time limitations of physical experiments, as the simulations may be performed anytime, anywhere. They may also be performed several times, using different physical parameters, e.g. to simulate different treatment methods.

However, a problem with the simulations is accuracy. Although the Navier-Stokes equations may be solved to a very high precision, these solutions are heavily dependent on the applied boundary conditions and physical parameters, as well as the computational domain. All these factors are highly patient specific, and patient specific data are often unavailable or inaccurate.

---

[1]`https://launchpad.net/nsbench/`

[2]The images are produced by using either computed tomography (CT) or magnetic resonance imaging (MRI).
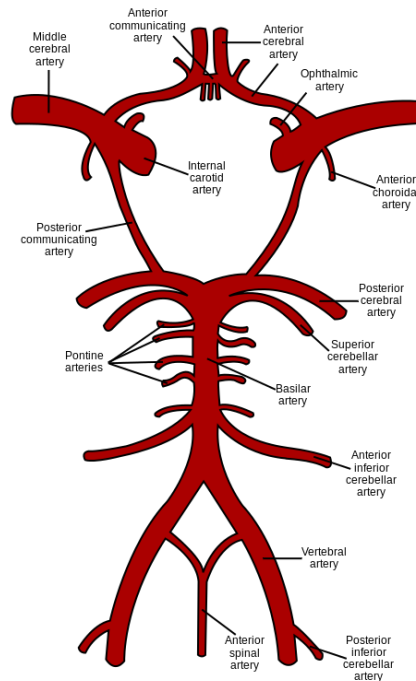
Figure 1.1: A classical textbook representation of the Circle of Willis and surrounding arteries.

The physical domain is usually specified by converting data from a digital 3D-image to a mathematical description over which the governing equations are to be solved. These images usually have a quite coarse resolution, making it difficult to get an accurate description of, in this case, the cerebral aneurysm and the surrounding vessels. This problem is addressed to some extent in Section 6.

To accurately describe blood flow, a model for the viscosity and density of blood must be prescribed. These parameters again depend on the composition of blood. It has been observed that the blood behaves as a non-Newtonian fluid, i.e. the viscosity is not constant. However, most of the studies done on cerebral aneurysm blood flow assume blood to behave as a Newtonian fluid, i.e. the viscosity is constant. The effects of this assumption is addressed in Section 6 and Section 7.

Boundary conditions are necessary to describe the nature of the flow flowing in and out of the domain. They often involve a prescribed inflow velocity or flux, and a prescribed pressure at the outlets. Often, no patient specific data is available, and thus, a common approach is to use average values measured in a different group of patients. This may differ much from what is the actual case, as the anatomy and flow patterns in the Circle of Willis are very individual. Effects of different boundary conditions are studied in Section 7. In the same section, uncertainty about the composition of blood is also addressed, in the context of a hypothesis regarding aneurysm rupture in women.

Even though many things are still unknown on what causes aneurysm growth and rupture, the

viscous force of wall shear stress (WSS) seems to be of special importance. Because of this, we will use this effect to benchmark the effects of the previously mentioned uncertainties.

2. Medical Background

2.1. Cerebral aneurysms

Cerebral aneurysms are most common in and around the Circle of Willis, a network of arteries at the base of the brain. With a risk of rupturing and causing a hemorrhage into and around the brain, they are considered a serious *cerebrovascular* disorder. The size and form of cerebral aneurysms vary greatly, from aneurysms that are barely distinguishable from the vessel, to aneurysms which may be more than 50 mm in diameter. Compared to the average diameters of some of the arteries in and around the Circle of Willis (see Table 2.1), it is not difficult to recognize the severity of this condition. The common shapes of cerebral aneurysms are shown in Figure 2.1. In Section 6 a single saccural aneurysm at the *middle cerebral artery* (MCA) bifurcation is segmented out and studied, and in Section 7 twelve different saccular MCA aneurysms are studied.

Artery	Average diameter[6][7]	Percentage of cerebral aneurysms[8]
Posterior Cerebral Artery	2.2mm	-
Posterior Communicating Artery	1.4mm	25%
Internal Carotid Artery	4.2mm	7.5%
Anterior Cerebral Artery	2.3mm	-
Anterior Communicating Artery	1.9mm	30%
Middle Cerebral Artery	2.5mm	20%

Table 2.1: Arteries in an around the Circle of Willis.

(a) Fuseform **(b)** Saccular/sidewall **(c)** Saccular/ bifurcation
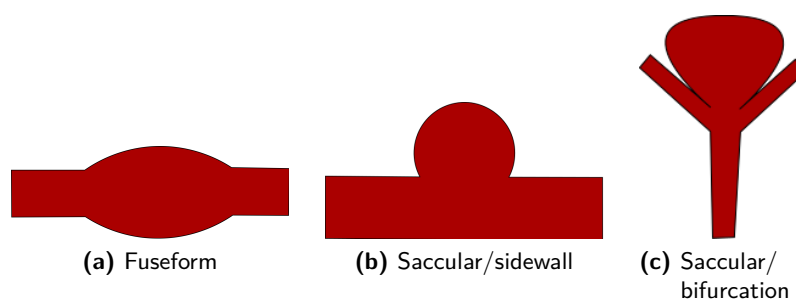
Figure 2.1: Classification of aneurysms.

The exact reasons for aneurysm initiation, development and rupture are largely unknown. However, some factors have been identified to contribute, e.g. environmental factors such as smoking, alcoholism and hypertension[3,9]. It is also known that people with an asymmetric or incomplete Circle of Willis are more prone to develop aneurysms[10]. There are also sex related differences, as the rupture risk for women is 1.6 times that of men[5].

Aneurysm rupture-risk is often seen in coherence with the aneurysms size and location, along with the patients age and medical history, among others[11]. However, hemodynamic forces such as pressure and WSS also seem to have considerable effects on aneurysms towards growth and rupture[12].

An aneurysm is a diseased vessel. Healthy arteries consist mainly of three layers; the intima, the media and the adventitia[13]. The endothelial cells of the intima reacts to external stimuli exerted by blood flow, by signalling the muscle cells in the media such that the artery flexes, and thus reduces the mechanical stresses. This process is known as *mechanotransduction*. In aneurysms, this response is damaged. In aneurysm walls, both the endothelial cells and muscle cells are depleted, and the wall thickness may be reduced to about 25% of that of a healthy artery. Thus, the walls are more brittle and less responsive to mechanical stimuli.

High WSS (above 40 Pa) is known to cause damage to the endothelial cells[14], but even too low WSS might cause inflammatory responses in the vessel walls[15]. This supports the hypothesis that WSS may be significant in predicting aneurysm development.

To avoid aneurysm rupture, different treatment methods may be applied, such as surgical clipping or endovascular coiling. These procedures are however costly[16], and there are associated risks with the different types of treatment[17]. Thus, one would not want to perform treatment unnecessary. It is therefore vital to gain a thorough understanding of the physical aspect of aneurysm rupture, to make sure the correct treatment is applied.

2.2. Blood properties

Blood does not behave like a Newtonian fluid such as e.g. water. Due to its composition of blood cells, platelets, proteins etc. it displays some noteworthy non-Newtonian properties. There are also sex related differences in the composition of blood which may affect the blood viscosity, and thus also the viscous forces that blood exhibits on the vessels.

The average adult human body (70 kg) has about 5 liters of blood[18]. Its density is close to water, approximately 1056 kg/m^3 [19], which means that blood amounts to about 7-8% weight of the average adult. The blood cells found in blood are red blood cells, white blood cells and platelets. These elements are suspended in a liquid called plasma, which is mainly water (about 92%) and proteins. Plasma accounts for 55% of whole blood by volume, and behaves like a Newtonian fluid.

Of the remaining 45%, as much as 95% is red blood cells, and thus, the effects of platelets and white blood cells on the rheology^3 of blood are very small. The non-Newtonian characteristics mainly arise from the interaction between red blood cells and some of their properties. The percentage of blood volume accounted for by red blood cells, is referred to as the hematocrit level. Typical hematocrit levels are 38-43 in normal blood, and values are typically lower for women than for men. The hematocrit level also changes with age, especially in women around menopausal age. A hypothesis made is that this change explains or partly explains

---

^{3}Rheology is the study of flow of materials.
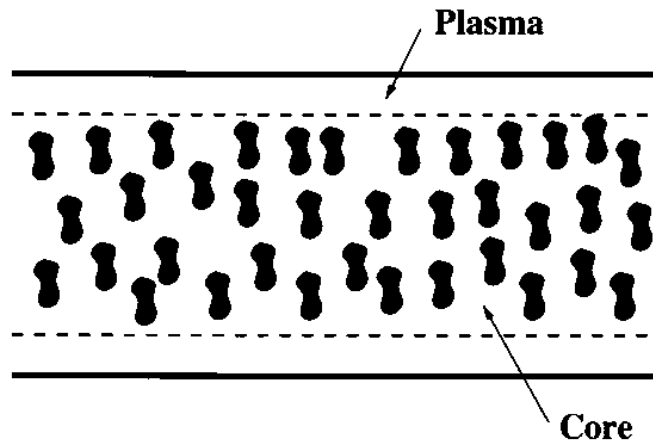
Figure 2.2: Illustration of a core layer formed by red blood cells and a boundary consisting of plasma.(Reprinted with permission from *Kundu, Cohen: Fluid Mechanics, 4th edition.*[20])

why women have a greater risk of aneurysm rupture. This hypothesis will be studied to some extent in Section 7.5.

The viscosity of blood is, due to its composition, not constant as in a Newtonian fluid. At low shear rates the red blood cells tend to connect and form microstructures called *rouleaux*. As the shear rate increases, these formations are ripped apart, and the viscosity decreases. This effect is a viscoelastic property of blood. The response to changes in shear rate are not instantaneous, and thus defines a *thixotropic* response.

In arteries in and around the Circle of Willis, the shear rates are usually so high that these formations no longer occur. However, the distribution of blood elements is not uniform. At higher shear rates the red blood cells tend to form a *core layer* in the center of the vessels, leaving a boundary layer of much less viscous plasma. This means that the viscosity increases further away from the vessel wall. Because the shear rate is higher towards the walls of the blood vessel, this can be seen as a shear thinning effect. It has been shown that this effect is the predominant one in blood flow similar to ours[21], and thus, it is this effect which will be the main focus in this thesis.

There are many different models for the viscosity of blood. Seven different non-Newtonian viscosity models have been implemented in this thesis for comparison and analysis, and they are all presented in Section 3.3.

2.2 Blood properties

3. Modeling blood

3.1. Mathematical Model

The Navier-Stokes equations are the mathematical description of all fluid motion. The equations are a system of partial differential equations which solves the pressure field p and the velocity field $\mathbf{u}$ for any given fluid density ρ and kinematic viscosity μ, along with boundary and initial conditions. They arise from the physical principles of conservation of mass and momentum. In this section, the Navier-Stokes equations for incompressible flow will be derived[20,22,23,24].

The main mathematical theorem needed to derive the Navier-Stokes equations is the following theorem:

Theorem (Reynold's transport theorem). *Let $\Omega(t) \subset \mathbb{R}^3$ be a material volume. Assume $f : \Omega(t) \times [0, T] \to \mathbb{R}$ and f differentiable. Then the following holds*

$$\frac{d}{dt} \int_{\Omega(t)} f(\mathbf{x}, t) \, d\Omega(t) = \int_{\Omega(t)} \frac{\partial f(\mathbf{x}, t)}{\partial t} \, d\Omega(t) + \int_{\partial\Omega(t)} f(\mathbf{x}, t)\mathbf{u} \cdot \mathbf{n} \, dS \qquad (3.1)$$

where $\mathbf{u}$ is the velocity of the material.

Conservation of mass

Now consider an arbitrary material volume $\Omega(t) \subset \mathbb{R}^3$ filled with fluid without sources or sinks. By the mass conservation principle we must have

$$\frac{d}{dt} \int_{\Omega(t)} \rho \, d\Omega(t) = 0$$

Applying Reynold's transport theorem, we get

$$\frac{d}{dt} \int_{\Omega(t)} \rho \, d\Omega(t) = \int_{\Omega(t)} \frac{\partial \rho}{\partial t} \, d\Omega(t) + \int_{\partial\Omega(t)} \rho\mathbf{u} \cdot \mathbf{n} \, dS = 0$$

By using Green's theorem to the surface integral in the above expression to get

$$\int_{\partial\Omega(t)} \rho\mathbf{u} \cdot \mathbf{n} \, dS = \int_{\Omega(t)} \rho\nabla \cdot \mathbf{u} \, d\Omega$$

we can reformulate the expression as a volume integral:

$$\int_{\Omega(t)} \frac{\partial \rho}{\partial t} + \rho\nabla \cdot \mathbf{u} \, d\Omega(t) = 0.$$

For this to be true for an arbitrary volume $\Omega(t)$, the expression inside the integral must be equal to zero. In the special case of incompressibility ρ is constant, and we are left with the continuity equation for an incompressible fluid,

$$\nabla \cdot \mathbf{u} = 0. \tag{3.2}$$

Conservation of momentum

Consider again the material volume $\Omega(t)$, and assume constant density. Newton's second law of motion states that the time derivative of the linear momentum of the system is equal to the sum of forces acting upon it:

$$\frac{d}{dt} \int_{\Omega(t)} \rho \mathbf{u} \, d\Omega(t) = \sum \mathbf{F}. \tag{3.3}$$

Applying Reynold's transport theorem elementwise to the left hand side, and then integration by parts to the surface integral yields

$$\frac{d}{dt} \int_{\Omega(t)} \rho \mathbf{u} \, d\Omega(t) = \int_{\Omega(t)} \frac{\partial(\rho \mathbf{u})}{\partial t} \, d\Omega(t) + \int_{\partial\Omega(t)} (\mathbf{u} \cdot \mathbf{n}) \rho \mathbf{u} \, dS$$

$$= \int_{\Omega(t)} \frac{\partial(\rho \mathbf{u})}{\partial t} \, d\Omega(t) + \int_{\Omega(t)} \rho \mathbf{u} \cdot \nabla \mathbf{u} + \rho(\nabla \cdot \mathbf{u})\mathbf{u} \, d\Omega(t)$$

$$= \int_{\Omega(t)} \rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} \, d\Omega(t)$$

where the continuity equation $\nabla \cdot \mathbf{u} = 0$ is used in the last step.

The forces acting on the fluid can be either *body forces* denoted as $\mathbf{f}$ (e.g. gravity) or *surface forces* (pressure, viscous forces), given as the Cauchy stress tensor, which is defined as

$$\sigma(\mathbf{u}, p) = 2\mu\epsilon(\mathbf{u}) - pI$$

where

$$\epsilon(\mathbf{u}) = \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right),$$

i.e., the symmetric gradient.

Following Newton's third law of motion, the surface force at a surface within our volume is counterweighted by an opposite surface force. Thus, the only contribution to the system as a whole we get from surface forces, are from the surface itself. We can then write the sum

of forces as

$$\sum \mathbf{F} = \int_{\Omega(t)} \mathbf{f} \, d\Omega(t) + \int_{\partial\Omega(t)} \sigma \cdot \mathbf{n} \, dS$$

$$= \int_{\Omega(t)} \mathbf{f} + \nabla \cdot 2\mu\epsilon(\mathbf{u}) - \nabla p \, d\Omega(t)$$

by using Green's theorem. Hence we have two volume integrals who should be equal for an arbitrary volume, meaning that the integrands must be exactly equal. We are thus left with

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = \mathbf{f} + \nabla \cdot 2\mu\epsilon(\mathbf{u}) - \nabla p \tag{3.4}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{3.5}$$

which are known as the Navier-Stokes equations for an incompressible flow.

3.2. Units

The units used for blood modeling differ through literature. Most commonly used is the standard SI units or the cgs (centimetre-gram-second) system. Some non-SI units are also common, in particular the *mmHg* unit for measuring blood pressure, which is commonly used in medicine. In this thesis, the length unit is chosen as millimetre to best get a grasp of the size of the arteries modelled. The mass unit is chosen as grams to keep a conversion factor (from SI units) of 1 for the Pascal unit used to measure pressure and shear stress.

Quantity	Unit	Unit symbol
Length	Millimetre	mm
Mass	Gram	g
Time	Second	s

(a) Base units.

Quantity	SI Unit	Used unit	Conversion factor
Velocity	$\frac{m}{s}$	$\frac{mm}{s}$	10^3
Pressure	$Pa = \frac{kg}{m \cdot s^2}$	$\frac{g}{mm \cdot s^2}$	1
Stress	$Pa = \frac{kg}{m \cdot s^2}$	$\frac{g}{mm \cdot s^2}$	1
Mass density	$\frac{kg}{m^3}$	$\frac{g}{mm^3}$	10^{-6}
Dynamic viscosity	$Pa \cdot s = \frac{kg}{m \cdot s}$	$\frac{g}{mm \cdot s}$	1
Kinematic viscosity	$\frac{m^2}{s}$	$\frac{mm^2}{s}$	10^6

(b) Derived units.

Table 3.1: Units used.

3.3. Viscosity Models

As mentioned in Section 2.2, the viscosity of blood displays some non-Newtonian effects, in particular that blood is a shear-thinning fluid. Thus, the viscosity depends on the shear rate, which is given by

$$\dot{\gamma} = \sqrt{2\|\epsilon(\mathbf{u})\|_F}$$

where $\|\cdot\|_F$ denotes the Fröbenius norm.

```python
from dolfin import *

class ViscosityModelBase:
    #Base class for all viscosity models
    def __init__(self,options,mesh):
        #Store options
        self.options = options

        #Common viscosity model parameters
        self.rho = self.options["rho"] #g/mm^3

        self.mesh=mesh
        self.scalar = FunctionSpace(mesh, "DG", 0)
        self.nu = Function(self.scalar)

    def gamma(self,u):
        # Returns the shear rate
        return pow(0.5*inner(grad(u)+transpose(grad(u)),
                            grad(u)+transpose(grad(u))), 0.5)
```

Listing 3.1: Implementation of the base class common for all viscosity models.

To simulate the properties of blood flow as well as possible, several models for the viscosity have been implemented. Seven different non-Newtonian models will be studied in this thesis, and a direct comparison between them can be seen in Figure 3.1, which shows the viscosity as a function of $\dot{\gamma}$.

Of the seven models, five use parameters collected from the same data set. These incorporate asymptotic viscosities for zero and infinite shear stress, and an overview of the models along with their respective material parameters are given in Table 3.2. The other two, a Casson type model and a power-law model, both incorporate the hematocrit level to predict the viscosity.

The power-law type model implemented, has a viscosity function given by

$$\mu = k\dot{\gamma}^{(n-1)}. \tag{3.6}$$

Model	$\dfrac{\mu(\dot{\gamma})-\mu_\infty}{\mu_0-\mu_\infty}$	Material constants
Powell-Eyring	$\dfrac{\sinh^{-1}(\lambda\dot{\gamma})}{\lambda\dot{\gamma}}$	$\lambda = 5.383\,s$
Cross	$\dfrac{1}{1+(\lambda\dot{\gamma})^m}$	$\lambda = 1.007\,s,\ m = 1.028$
Modified Cross	$\dfrac{1}{(1+(\lambda\dot{\gamma})^m)^a}$	$\lambda = 3.736\,s,\ m = 2.406,\ a = 0.254$
Carreau	$(1+(\lambda\dot{\gamma})^2)^{(n-1)/2}$	$\lambda = 3.313\,s,\ n = 0.3568$
Carreau-Yasuda	$(1+(\lambda\dot{\gamma})^a)^{(n-1)/a}$	$\lambda = 1.902\,s,\ n = 0.22,\ a = 1.25$

Table 3.2: Parameters for five of the seven viscosity models for blood, with $\mu_0 = 0.056\,Pa\,s$ and $\mu_\infty = 0.00345\,Pa\,s$. Material parameters have all been collected from [25].

An extension of this model to be used in blood modeling has been developed by Walburn and Schneck [26]. This incorporates the hematocrit level as a parameter by using

$$k = C_1 e^{C_2 H}$$
$$n = 1 - C_3 H$$

where H is the hematocrit fraction, and C_1, C_2 and C_3 are constants chosen from viscometer data[4] as

$$C_1 = 0.00148, \quad C_2 = 5.12, \quad C_3 = 0.499.$$

The main weakness of this model is that it does not incorporate an asymptotic viscosity when the shear rate tends to infinity. Thus, this model might predict an unphysical viscosity at high shear rates.

The final model implemented is a Casson model. The Casson equation for viscosity in a shear thinning fluid is given as [27,28]

$$\mu = \frac{\tau_y}{\dot{\gamma}} + \frac{2\sqrt{\mu_\infty}\sqrt{\tau_y}}{\sqrt{\dot{\gamma}}} + \mu_\infty \tag{3.7}$$

where τ_y is the yield stress of blood. The Casson model parameters can be modified to incorporate plasma viscosity, cell rigidity and hematocrit level in the following ways [29,30]:

$$\tau_y = 0.02687 H^3 \qquad \mu_\infty = \eta_0 T_k (1-H)^{-2.5}$$

where the plasma viscosity is chosen to be $\eta_0 = 0.00145\,Pa\,s$ [31] and the cell rigidity index number is chosen to be $T_k = 0.62$ [32,33]. Unless noted otherwise, the hematocrit (H) will be kept constant at 40%.

---

[4] A viscometer is an instrument to measure the viscosity of a fluid.

```python
from viscosityModelbase import *

class ViscosityModel(ViscosityModelBase):
    "Carreau viscosity model"

    def __init__(self,options,mesh):
        ViscosityModelBase.__init__(self,options,mesh)

        # Model-specific parameters
        self.l = 3.3135
        self.n = 0.3568

        # Asymptotic viscosities
        self.mu_0 = 0.056 #Pa*s
        self.mu_inf = 0.00345 #Pa s

    def __call__(self,u):
        gamma = self.gamma(u)

        mu = (1.0+(self.l*gamma)**2.0)**((self.n-1)/2.0)
                    *(self.mu_0-self.mu_inf)+self.mu_inf
        self.nu.assign(project(mu/self.rho, self.scalar))

        return self.nu
```

Listing 3.2: Implementation of the Carreau viscosity model.

Note from the implementations seen in Listing 3.1 and Listing 3.2 that the viscosity is projected onto a *Discontinuous Galerkin* (DG) function space of order 0. In other words, in the implementation, the viscosity is assumed to be *locally* Newtonian.

3.4. Initial Conditions

In order to solve the Navier-Stokes equations, we need to prescribe initial conditions at a time $t = t_0$ for both the velocity and the pressure. How these initial conditions are chosen is not very significant, as the solutions will tend to "forget" the initial state of the fluid as simulations run. However, one should at least use initial conditions which satisfy the Navier-Stokes equations. One possibility is to choose a steady-state solution as initial conditions, but for simplicity the initial conditions used in this thesis are simply $\mathbf{u} = \mathbf{0}$ and $p = 0$, and then allowing the simulations to run long enough to "forget" the initial state.
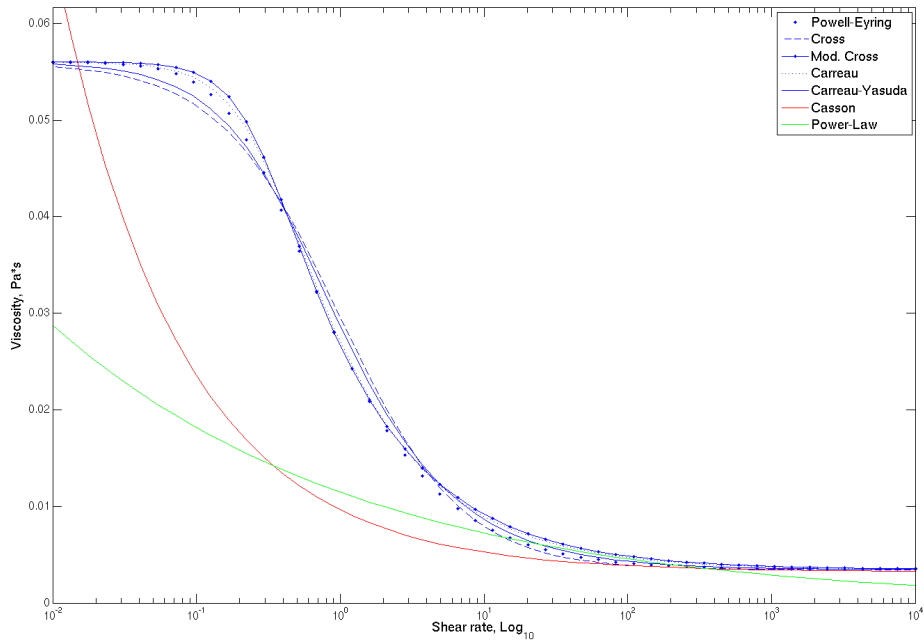
Figure 3.1: Plots of viscosity versus shear rate for the different viscosity models.

For the simulations where a pulsatile flow is studied, the simulations are run for 3 cardiac cycles to reach a periodic flow, and results are reported from the fourth cycle.

3.5. Boundary Conditions

At the vessel walls the no-slip condition is imposed. That is, because of molecular adherence between the fluid and the wall, the velocity relative to the wall will be equal to zero. Throughout this thesis, we will only work with rigid walls, meaning that the no-slip condition reduces to $\mathbf{u} = \mathbf{0}$ at the wall. The assumption of rigid walls is clearly a simplification, as blood vessels respond to the blood flow by expanding and contracting. However, the walls in cerebral aneurysms are much stiffer than their surrounding arteries[34], which to some extent validates the assumption. Using elastic walls would also be much more computationally demanding, and there is little data quantifying the responsiveness of the vessel walls in cerebral aneurysms.

At the outlets, homogeneous Neumann conditions are prescribed for the velocity ($\nabla\mathbf{u}\cdot\mathbf{n} = \mathbf{0}$). The pressure is prescribed by using a *resistance* boundary condition, taking into account the physiology of arteries downstream . The resistance boundary condition is given as

$$p = \hat{p} + C \int_{\Omega} (\mathbf{u} \cdot n)\, dS$$

```python
def boundary_conditions(self, V, Q, t):
    #Create boundary conditions.
    bv0 = DirichletBC(V, Constant((0.0,0.0,0.0)), self.bc_markers,1)
    bv1 = DirichletBC(V, self.inflow_function, self.bc_markers,2)
    bp0 = DirichletBC(Q, Constant(0), self.bc_markers,3)

    bcu = [bv0,bv1]
    bcp = [bp0]
    return bcu, bcp
```

Listing 3.3: Implementation of the boundary conditions on simple geometries with a single outlet.

Artery	C $(Pa \cdot s \cdot mm^{-3})$	Radius (mm)
Thoracic artery	0.18	9.99
External Carotid Artery	5.43	1.50
Middle Cerebral Artery	5.97	1.43
Anterior Communicating Artery	8.48	1.20
Posterior Communicating Artery	11.08	1.05

Table 3.3: Resistance boundary coefficients.

where $\hat{p}$ is the mean intracranial pressure. This is set equal to 0 because the incompressible Navier-Stokes equations only depend on the pressure gradient, and not on the pressure itself. The value of the coefficient C for arteries of different sizes have been gathered from[35] and is given in Table 3.3. Note how this constant increases with decreasing radius. For the simple models with only one outlet, homogeneous Dirichlet conditions are prescribed for the pressure.[5]

As an inflow condition both a prescribed pressure and a prescribed velocity has been considered. Prescribing a pressure at the inlet would depend on the viscosity model, because the pressure drop will depend on the internal forces, which is a function of the viscosity. Also, since there have been better access to velocity measurements and velocity profiles, a velocity inlet condition was chosen. As a fully developed laminar flow velocity profile is parabolic (see Section 5.1), a parabolic inlet profile has been imposed on all simulations throughout this thesis. For modeling the blood flow in the Circle of Willis, this simplification is shown to be a decent approximation, given a sufficient entrance length[36]. In simulations on aneurysm geometries, a time-dependent pulsatile inflow condition was used to scale the parabolic velocity profile.

In Section 7, we investigate how some of the assumptions made on the boundary conditions may affect the solution.

---

[5]Because of a scaling error in the implementation discovered late in the process, the resistance boundary condition intended in Section 6 was effectively a homogeneous Dirichlet condition instead.

```python
def OutflowBoundaryValue(self, i):
    n = FacetNormal(self.mesh)
    flux = dot(self.u,n)*ds(i)
    Q = assemble(flux,mesh=self.mesh,
            exterior_facet_domains=self.bc_markers)
    C = 5.97
    p0 = 0
    R = C*Q+p0
    return R
```

Listing 3.4: Implementation of the resistance boundary condition.
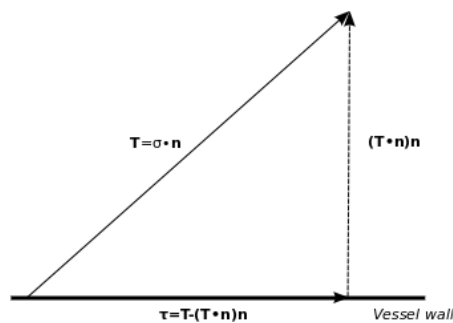
3.6. Stress Calculation

The WSS is calculated from the traction vector T, which in turn is defined from the stress tensor σ as

$$T = \sigma \mathbf{n}.$$

The WSS is the viscous force working parallel to the vessel wall. That is, it is the tangential component of the traction vector. Written in mathematical notation, the WSS-vector is given as

$$\tau_{\mathbf{w}} = T - (T \cdot \mathbf{n})\mathbf{n}. \tag{3.8}$$

Figure 3.2: Illustration of WSS.

The pressure is a force working perpendicular to the surface, and thus plays no role in wall shear stress distribution. Hence we can neglect this term from σ in our computations to increase efficiency. The implementation is seen in Listing 3.5.

In this thesis we only work with the *magnitude* of the WSS, and thus we from now on denote the WSS as $\tau_w := \|\tau_{\mathbf{w}}\|$.

```python
def compute_stress(self):
        if self.stress is None:
                DG = FunctionSpace(self.mesh,"DG",0)
                self.tau = Function(DG)
                m = TestFunction(DG)

                scaling = 1/FacetArea(self.mesh)
                n = FacetNormal(self.mesh)

                #Stress calculation
                sigma = self.rho*self.nu*(grad(self.u)
                                    + grad(self.u).T)
                T = -sigma*n
                Tn = inner(T,n)        #Scalar
                Tt = T-Tn*n            #Vector

                self.stress = scaling*m*sqrt(inner(Tt,Tt))*ds

        assemble(self.stress, tensor = self.tau.vector())

        return self.tau
```

Listing 3.5: Implementation of the WSS calculation.

3.7. Final Model

Assumptions made on the final model are

- Body forces such as gravity are negligible

- Rigid and no-penetration walls

- Incompressibility

- The thixotropy of blood is negligible, i.e. the viscosity is not time-dependent

- A locally Newtonian viscosity

- Isothermal flow

The final model we end up with is thus

$$
\left.
\begin{aligned}
\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} &= \nabla \cdot 2\nu\epsilon(\mathbf{u}) - \frac{1}{\rho}\nabla p \\
\nabla \cdot \mathbf{u} &= 0 \\
\mathbf{u}(\mathbf{x}, 0) &= \mathbf{0} \\
p(\mathbf{x}, 0) &= 0
\end{aligned}
\right\} \quad \text{for } \mathbf{x} \in \Omega
$$

$$
\begin{aligned}
\mathbf{u} &= \mathbf{0} && \text{for } \mathbf{x} \in \Gamma_w \\
\mathbf{u} &= \mathbf{u_0} && \text{for } \mathbf{x} \in \Gamma_I \\
p &= p_0 && \text{for } \mathbf{x} \in \Gamma_O
\end{aligned}
$$

where $\nu = \frac{\mu}{\rho}$ is the kinematic viscosity, Ω is our domain and Γ_w, Γ_I, Γ_O are the vessel wall, the inlet and the outlet(s), respectively, whose union is $\partial\Omega$ and intersection is empty.

3.7   Final Model

4. Numerical methods

4.1. Incremental Pressure Correction Scheme (IPCS)

Reconsider the Navier-Stokes equations for incompressible fluid flow:

$$\left.\begin{array}{rl} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} & = \nabla \cdot 2\nu\epsilon(\mathbf{u}) - \frac{1}{\rho}\nabla p \\ \nabla \cdot \mathbf{u} & = 0 \end{array}\right\} \text{ for } \mathbf{x} \in \Omega. \tag{4.1}$$

The Navier-Stokes equations are difficult to solve due to their nonlinear nature, the combination of a hyperbolic and a parabolic term, constraints on the solution, and the fact that there are two unknowns present. There are several algorithms designed to deal with these issues, and their performance in terms of speed and accuracy depend very much on the nature of the problem. However, according to tests on a variety of problems, the Incremental Pressure Correction Scheme (IPCS), has proved to be a relatively good approach [37].

The IPCS is an operator-splitting algorithm. This is done by using a known approximation to the pressure in (4.1), and thus removing one unknown from the equation. We start by discretizing in time. The time discretization is done using the backward Euler difference scheme, $\dot{u} \approx \frac{u^n - u^{n-1}}{\Delta t}$. The non-linear terms can be linearized in several ways. By evaluating the convection semi-implicit as $\mathbf{u}^{n-1} \cdot \nabla \mathbf{u}^n$ instead of fully explicit, the CFL-constraint[6] is much less restrictive on the choice of time stepping in terms of stability. Evaluating the viscosity as $\nu := \nu(\mathbf{u}^{n-1})$, we are left with a linearized set of equations to be solved at each time step, known as the Oseen equations:

$$\mathbf{u}^n + \Delta t \mathbf{u}^{n-1} \cdot \nabla \mathbf{u}^n - \Delta t \nabla \cdot 2\nu\epsilon(\mathbf{u}^n) + \frac{\Delta t}{\rho}\nabla p^n = \mathbf{u}^{n-1}$$
$$\nabla \cdot \mathbf{u}^n = 0. \tag{4.2}$$

However, p^n is still unknown here. We wish to find an approximated value for the velocity field, $\mathbf{u}^* \approx \mathbf{u}^n$, by replacing p^n with the known p^{n-1} in the equation above. We are thus left with the simpler equation

$$\mathbf{u}^* + \Delta t \mathbf{u}^{n-1} \cdot \nabla \mathbf{u}^* - \Delta t \nabla \cdot 2\nu\epsilon(\mathbf{u}^*) + \frac{\Delta t}{\rho}\nabla p^{n-1} = \mathbf{u}^{n-1}. \tag{4.3}$$

The divergence-free condition on $\mathbf{u}^*$ is not needed to solve this equation, but we are still interested in finding a solution $\mathbf{u}^n$ that is divergence-free. By defining a function for the velocity correction as $\mathbf{u}^c = \mathbf{u}^n - \mathbf{u}^*$, it is clear that by subtracting (4.3) from (4.2), and noting that $\nabla \cdot \mathbf{u}^n = 0 \Rightarrow \nabla \cdot \mathbf{u}^c = -\nabla \cdot \mathbf{u}^*$, we get a new set of equations for $\mathbf{u}^c$:

$$\mathbf{u}^c + \Delta t \mathbf{u}^{n-1} \cdot \nabla \mathbf{u}^c - \Delta t \nabla \cdot 2\nu\epsilon(\mathbf{u}^c) + \frac{\Delta t}{\rho}\nabla \Phi^n = 0$$
$$\nabla \cdot \mathbf{u}^c = -\nabla \cdot \mathbf{u}^* \tag{4.4}$$

---

[6]The Courant-Friedrichs-Lewy (CFL) condition describes the limitations on the time stepping based on the flow velocity and the spatial discretization.

where $\Phi^n = p^n - p^{n-1}$. However, the operator splitting here is a first order approximation, $O(\Delta t)$. Thus, without decreasing the order of the approximation, we can simplify the above equations to the following:

$$\mathbf{u}^c + \frac{\Delta t}{\rho}\nabla\Phi^n = 0$$
$$\nabla \cdot \mathbf{u}^c = -\nabla \cdot \mathbf{u}^* \qquad (4.5)$$

which is reducible to a Poisson problem,

$$\Delta\Phi^n = \frac{\rho}{\Delta t}\nabla \cdot \mathbf{u}^*. \qquad (4.6)$$

The boundary conditions used to solve this equations depend on the kind of boundary conditions applied to the original problem. Solving this, the corrected pressure and velocity field are easily calculated through $p^n = \Phi^n + p^{n-1}$ and $u^n = u^* - \frac{\Delta t}{\rho}\nabla\Phi^n$.

Quickly summarized, the IPCS is done in four steps:

1. Calculate the tentative velocity u^* by replacing the unknown pressure p^n with the known approximation p^{n-1}.

2. Solve the Poisson equation $\Delta\Phi^n = \frac{\rho}{\Delta t}\nabla \cdot u^*$, where $\Phi^n = p^n - p^{n-1}$.

3. Calculate the corrected pressure $p^n = \Phi^n + p^{n-1}$.

4. Calculate the corrected velocity $u^n = u^* - \frac{\Delta t}{\rho}\nabla\Phi^n$.

4.2. The Finite Element Method (FEM)

In many areas of science, partial differential equations (PDEs) needs to be solved. An analytical approach to solving these may be very difficult, time demanding and in many cases exact solutions are non-existent. Numerical methods are used to approximate a solution to a given PDE. Where exact solutions often are required to lie in infinite dimensional function spaces, a numerical solution is found in a function space with finitely many degrees of freedom.

The Finite Element Method (FEM) is a method for solving differential equations. Compared to the more common finite difference method, it may seem more difficult to grasp, but it has its clear advantages. Its main advantage is how it can be adapted to complicated domains with varying mesh size. Using FEniCS to implement this method also provides a very simple way to vary the geometry over which the equations are to be solved. For the applications needed in this thesis, this property comes in very handy, as the domains studied range from simple two-dimensional channels to complex three-dimensional models of blood vessels.

Given a PDE, the Finite Element Method follows a few steps:

1. Define the weak form of the equation.

2. Discretize the domain and define a finite dimensional function space in which to search for an approximate solution.

3. Define the discretized weak form of the equation.

4. Reduce the problem to a system of algebraic equations.

5. Solve or approximate a solution of the algebraic equations.

A general partial differential equation can be expressed as

Find $u \in V$ such that
$$Lu = f \tag{4.7}$$

where V is a function space, $L : V \to V'$ is a spatial differential operator and $f \in V'$. The weak form is then found by introducing a suitable test space $\hat{V}$ and defining a bilinear and linear form

$$a(u, v) := [Lu](v)$$
$$(f, v) := f(v)$$

and writing the problem as

Find $u \in V$ such that
$$a(u, v) = (f, v) \quad \forall v \in \hat{V} \tag{4.8}$$

If u solves this equation, we say that u is a *weak solution* to the original equation. The weak solution is (generally) found in an infinite dimensional function space (e.g. a Sobolev space). Thus, for the numerical approach, we need to find a *finite dimensional* subspace in which to search for an approximate solution. That is, we want to move from the infinite dimensional problem (4.8) to the finite dimensional problem

Find $u \in V_h$ such that
$$a(u, v) = (f, v) \quad \forall v \in \hat{V}_h \tag{4.9}$$
where $V_h \subset V$, $dim \hat{V}_h < \infty$ and $dim(V_h) < \infty$.

The process of finding these finite dimensional subspaces are where the *finite elements* are introduced. A commonly used definition is the one introduced by Ciarlet[38], and given here as in Brenner, Scott[39], along with a definition for the *nodal basis*:

Definition (Finite element). *Let*

(i) $K \subseteq \mathbb{R}^n$ *be a bounded closed set with nonempty interior and piecewise smooth boundary (the **element domain**),*

(ii) $\mathcal{P}_K$ *be a finite-dimensional space of functions on K (the space of **shape functions**) and*

(iii) $\mathcal{N}_K = \{N_1, N_2, \ldots, N_k\}$ *be a basis for $\mathcal{P}'_K$ (the set of **nodal variables**)*

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

Definition (Nodal basis). *Let $(K, \mathcal{P}, \mathcal{N})$ be a finite element. The basis $\{\phi_1, \phi_2, \ldots, \phi_k\}$ of $\mathcal{P}$ dual to $\mathcal{N}$ (i.e. $N_i(\phi_j) = \delta_{ij}$)is called the nodal basis of $\mathcal{P}$.*

By using these two definitions, we can discretize our domain Ω. We discretize by choosing $\tau = \{(K, \mathcal{P}, \mathcal{N})_1, (K, \mathcal{P}, \mathcal{N})_2, \ldots, (K, \mathcal{P}, \mathcal{N})_k\}$ such that

$$\bigcup_{i=1,\ldots,n} K_i = \Omega$$

and

$$int(K_i) \bigcup int(K_j) = \emptyset \text{ for } i \neq j.$$

The set of $\mathcal{P}_K$'s is then a finite dimensional function space over Ω, denoted by V_h, in which we can search for an approximate solution. By inserting an anzats, $u_h = \sum U_i \phi_i$, for the formulation of our approximated solution and replacing the test function with an arbitrary basis function ϕ_j we can write out the problem as a variant of (4.9)

Find $u_h \in V_h$ such that

$$a(u_h, \phi_j) = (f, \phi_j), \quad j = 1, \ldots, n \qquad (4.10)$$

where $V_h \subset V$, $dim\hat{V}_h < \infty$ and $dim(V_h) < \infty$.

This equation can then be reduced into a finite number of algebraic equations, which can then be solved to find our approximated solution.

4.3. FEM applied to IPCS

Let $\mathbf{u}, \mathbf{v}$ be vectors, A, B be matrices and $\mathbf{n}$ be the outward pointing normal on the domain Ω. For this section, we introduce the following notation:

Notation	Meaning
$\langle \mathbf{u}, \mathbf{v} \rangle$	$\int_\Omega \mathbf{u} \cdot \mathbf{v} \, d\mathbf{x}$
$\langle A\mathbf{n}, \mathbf{v} \rangle_{\partial\Omega}$	$\int_{\partial\Omega} A\mathbf{n} \cdot \mathbf{v} \, dS$
$\langle A, B \rangle$	$\int_\Omega tr(AB^T) \, d\mathbf{x}$

Weak formulation

To apply the finite element method to IPCS, we need to define the weak formulation of equation (4.3) and (4.6) to obtain our linear and bilinear forms. For simplicity, we write $\mathbf{u}$ instead of $\mathbf{u}^*$. We start by rewriting (4.3) as

$$\frac{1}{\Delta t}\mathbf{u} + \mathbf{u}^{n-1} \cdot \nabla\mathbf{u} - \nabla \cdot 2\nu\epsilon(\mathbf{u}) = \frac{1}{\Delta t}\mathbf{u}^{n-1} - \frac{1}{\rho}\nabla p^{n-1} \qquad (4.11)$$

To find the weak formulation for this equation, we multiply with a test function $\mathbf{v}$ and integrate over the domain. For simplicity, the weak formulation of the left hand side and right hand side are done separately.

LHS of (4.11):

$$
\begin{aligned}
a_1(\mathbf{u}, \mathbf{v}) &= \int_\Omega \left(\frac{1}{\Delta t}\mathbf{u} + \mathbf{u}^{n-1} \cdot \nabla\mathbf{u} - \nabla \cdot 2\nu\epsilon(\mathbf{u}) \right) \cdot \mathbf{v} \, d\mathbf{x} \\
&= \frac{1}{\Delta t}\int_\Omega \mathbf{u} \cdot \mathbf{v} \, d\mathbf{x} + \int_\Omega (\mathbf{u}^{n-1} \cdot \nabla\mathbf{u}) \cdot \mathbf{v} \, d\mathbf{x} - \int_\Omega \nabla \cdot 2\nu\epsilon(\mathbf{u}) \cdot \mathbf{v} \, d\mathbf{x} \\
&= \frac{1}{\Delta t}\langle \mathbf{u}, \mathbf{v} \rangle + \left\langle \mathbf{u}^{n-1} \cdot \nabla\mathbf{u}, \mathbf{v} \right\rangle + \int_\Omega tr\left(2\nu\epsilon(\mathbf{u})\nabla\mathbf{v} \right) \, d\mathbf{x} - \int_{\partial\Omega} \nu\nabla\mathbf{u}\mathbf{n} \cdot \mathbf{v} \, dS \\
&\quad - \int_{\partial\Omega} \nu(\nabla\mathbf{u})^T\mathbf{n} \cdot \mathbf{v} \, dS \\
&= \frac{1}{\Delta t}\langle \mathbf{u}, \mathbf{v} \rangle + \left\langle \mathbf{u}^{n-1} \cdot \nabla\mathbf{u}, \mathbf{v} \right\rangle + \langle 2\nu\epsilon(\mathbf{u}), \nabla\mathbf{v} \rangle - \left\langle \nu(\nabla\mathbf{u})^T\mathbf{n}, \mathbf{v} \right\rangle_{\partial\Omega}.
\end{aligned}
$$

Note that the term $\int_{\partial\Omega} \nu\nabla\mathbf{u}\mathbf{n} \cdot \mathbf{v} \, dS$ vanishes. This is because the test function $\mathbf{v}$ is defined to be identically equal to zero on the part of the boundary where Dirichlet conditions are imposed, and homogeneous Neumann conditions are applied to the velocity on the remaining parts of the boundary.

RHS of (4.11):

$$L_1(\mathbf{v}) = \int_\Omega \left(\frac{1}{\Delta t} \mathbf{u}^{n-1} - \nabla p^{n-1} \right) \cdot \mathbf{v} \, d\mathbf{x}$$

$$= \int_\Omega \frac{1}{\Delta t} \mathbf{u}^{n-1} \cdot \mathbf{v} \, d\mathbf{x} - \int_\Omega \nabla p^{n-1} \cdot \mathbf{v} \, d\mathbf{x}$$

$$= \frac{1}{\Delta t} \left\langle \mathbf{u}^{n-1}, \mathbf{v} \right\rangle + \int_\Omega p^{n-1} \nabla \cdot \mathbf{v} \, d\mathbf{x} - \int_{\partial\Omega} p^{n-1} \mathbf{v} \cdot \mathbf{n} \, dS$$

$$= \frac{1}{\Delta t} \left\langle \mathbf{u}^{n-1}, \mathbf{v} \right\rangle + \left\langle p^{n-1}, \nabla \cdot \mathbf{v} \right\rangle - \left\langle \mathbf{n}, p^{n-1}\mathbf{v} \right\rangle_{\partial\Omega}$$

Equation (4.6) written can be written in weak form by multiplying with a scalar test function q and integrating over the domain:

$$a_2(\Phi^n, q) = \int_\Omega q \Delta \Phi^n \, d\mathbf{x}$$

$$= -\int_\Omega \nabla \Phi^n \cdot \nabla q \, d\mathbf{x} + \int_{\partial\Omega} q \nabla \Phi \cdot \mathbf{n} \, dS$$

$$= -\int_\Omega \nabla \Phi^n \cdot \nabla q \, d\mathbf{x} - \int_{\Gamma_I} q \frac{\rho}{\Delta t} \mathbf{u}^c \cdot \mathbf{n} \, dS - \int_{\Gamma_W} q \frac{\rho}{\Delta t} \mathbf{u}^c \cdot \mathbf{n} \, dS + \int_{\Gamma_O} q \nabla \Phi \cdot \mathbf{n} \, dS$$

$$= -\int_\Omega \nabla \Phi^n \cdot \nabla q \, d\mathbf{x}$$

$$L_2(q) = \int_\Omega q \frac{\rho}{\Delta t} \nabla \cdot \mathbf{u}^* \, d\mathbf{x}$$

The surface terms vanishes from a_2 because $\mathbf{u}^c|_{\Gamma_W} \equiv \mathbf{0}$, $\mathbf{u}^c|_{\Gamma_I} \equiv \mathbf{0}$ and the Dirichlet boundary condition prescribed for the pressure on the outlet, results in $q|_{\Gamma_O} \equiv 0$.

Both weak forms are now in the form formalized in (4.8). The uniqueness of a solution follows from the Lax-Milgram theorem. Notice how the degree of the derivative reduces from the original problem. Hence, the notation *weak* formulation. For Φ and $\mathbf{u}$ to solve these equations, they must lie in the Hilbert spaces

$$H^1(\Omega) = \left\{ v : \Omega \to \mathbb{R} \; \middle| \; \int_\Omega \|v\|^2 + \|\nabla v\| \, d\mathbf{x} < \infty \right\} \tag{4.12}$$

$$\mathbf{H^1}(\Omega) = \left\{ \mathbf{v} : \Omega \to \mathbb{R}^d \; \middle| \; v_i \in H^1(\Omega) \text{ for } i = 1, \dots, d \right\}. \tag{4.13}$$

Finite element formulation

We have unlimited choices of finite dimensional function spaces in which to search for our solution, but we will focus only on two. The partitioning τ of the domain is done by

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

# Test and trial functions
v = TestFunction(V)
q = TestFunction(Q)
u = TrialFunction(V)
p = TrialFunction(Q)

#Tentative velocity
a11 = (1/k)*inner(v,u)*dx
a12 = 2.0*inner(epsilon(v), nu*epsilon(u))*dx \
        -inner(v, nu*grad(u).T*n)*ds
a13 = inner(v, grad(u)*u0)*dx

L1 = (1/k)*inner(v,u0)*dx \
        +1/rho*inner(epsilon(v), p0*Identity(u.cell().d))*dx \
        -1/rho*inner(v, p0*n)*ds

# Pressure correction
a2 = inner(grad(q), grad(p))*dx
L2 = inner(grad(q), grad(p0))*dx - rho*(1/k)*q*div(u1)*dx

# Velocity correction
a3 = inner(v, u)*dx
L3 = inner(v, u1)*dx - k/rho*inner(v, grad(p1 - p0))*dx
```

Listing 4.1: Implementation of the weak formulation.

simplexes K_i, i.e. triangles or tetrahedra, dependent on the dimension of the domain. Over each simplex, one defines polynomials of degree one or two, P_1 or P_2. These are known as Lagrange elements, and are illustrated in Figure 4.1. From this, we can define finite dimensional function subspaces of the Hilbert spaces introduced in the previous section, namely

$$CG_q = \left\{ v : \Omega \to \mathbb{R} \ \middle| \ v \in H^1(\Omega) \text{ and } v|_{K_i} \in P_q \text{ for } K_i \in \tau \right\} \tag{4.14}$$

$$\mathbf{CG}_q = \left\{ \mathbf{v} : \Omega \to \mathbb{R} \ \middle| \ \mathbf{v} \in \mathbf{H^1}(\Omega) \text{ and } \mathbf{v}|_{K_i} \in [P_q]^d \text{ for } K_i \in \tau \right\}. \tag{4.15}$$

CG denotes *Continuous Galerkin*. The condition that any function of CG_q or $\mathbf{CG}_q$ lies in a Hilbert space is what makes them continuous.
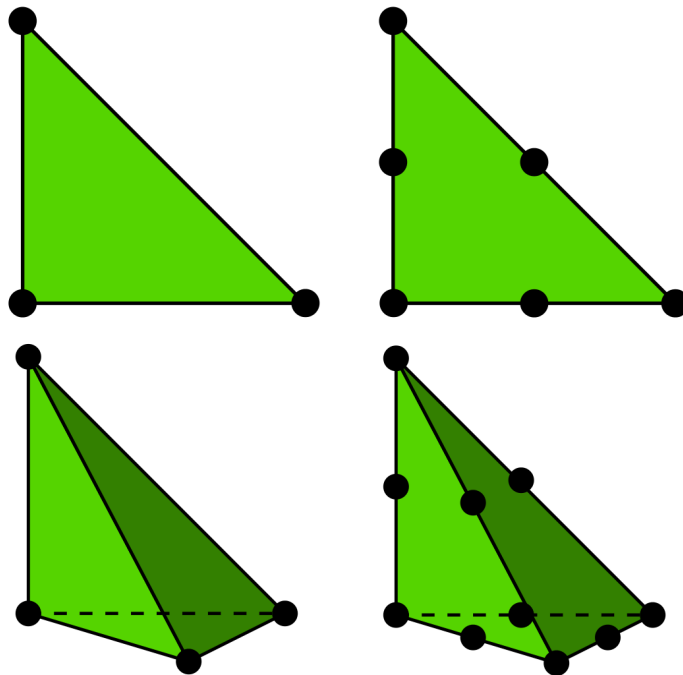
Figure 4.1: First and second order Lagrange elements for 2 and 3 dimensions (taken from [40]).

The polynomials are evaluated at *nodes* at each simplex. Each node has a corresponding basis function, which is defined to equal 1 at the node, and zero at all other nodes. Because of these basis functions, we can make an anzats of our solution to be linear combinations of these, i.e.

$$\mathbf{u}_h = \sum_{i=1}^{N} \mathbf{U}_i \phi_i$$

$$\Phi_h = \sum_{i=1}^{M} \Phi_i \zeta_i,$$

where $\mathbf{U}_i$ and Φ_i are unknown. Since our solution must satisfy our weak form for all $\mathbf{v}$ and q, they must in particular satisfy the equation for arbitrary test functions ϕ_j and ζ_j. Thus, we can write our finite dimensional problems as

$$a_1(\mathbf{u}_h, \phi_j) = \sum_{i=1}^{N} \mathbf{U}_i a_1(\phi_i, \phi_j) = L_1(\phi_j)$$

$$a_2(\Phi_h, \zeta_j) = \sum_{i=1}^{N} \Phi_i a_1(\zeta_i, \zeta_j) = L_2(\zeta_j)$$

This is actually a system of linear algebraic equations. By writing $A_1^{ij} = a_1(\phi_i, \phi_j)$, $A_2^{ij} = a_2(\zeta_i, \zeta_j)$, $L_1^j = L_1(\phi_j)$ and $L_2^j = L_2(\zeta_j)$ we can rewrite our problem as the matrix-vector equations

$$A_1 \mathbf{U} = L_1$$
$$A_2 \mathbf{\Phi} = L_2$$

where $\mathbf{U}$ and Φ are our unknown vectors.

The full implementation to solve the system in a time loop is seen in Listing 4.2. The *assemble* calls create the matrices and vectors. Note that much of the assembly is done outside the time loop. This is done to increase efficiency, and can be done to the parts of the system that are not time-dependent. The full systems are solved using a generalized minimal residual method along with an incompressible LU (ILU) or algebraic multigrid (AMG) preconditioner.

```
#Assemble matrices
A11 = assemble(a11)
A12 = assemble(a12)
A13 = assemble(a13)
A2 = assemble(a2)
A3 = assemble(a3)
A1 = A11+A12+A13
b = assemble(L1)

# Time loop
self.start_timing()
for t in t_range:
    # Get boundary conditions
    bcu, bcp = problem.boundary_conditions(V, Q, t)

    # Update viscosity if needed, and reassemble
    if(str(viscositymodel) is not "Constant"):
        nu.assign(viscositymodel(u0))
        assemble(a12, tensor=A12)
    assemble(a13, tensor=A13)
    A1.assign(A11+A12+A13)

    # Compute tentative velocity step
    assemble(L1, tensor=b)
    [bc.apply(A1, b) for bc in bcu]
    solve(A1, u1.vector(), b, "gmres", "ilu")
    # Pressure correction
    assemble(L2,tensor=b)
    [bc.apply(A2, b) for bc in bcp]
    solve(A2, p1.vector(), b, 'gmres', 'amg_hypre')
    # Velocity correction
    assemble(L3,tensor=b)
    [bc.apply(A3, b) for bc in bcu]
    solve(A3, u1.vector(), b, "gmres", "ilu")

    # Update
    self.update(problem, t, u1, p1,nu)
    u0.assign(u1)
    p0.assign(p1)
```

Listing 4.2: Solving the linear systems at each time step in a time loop.

5. Verification of implementation and numerical experiments

To check the implementation for convergence and to check different properties of the approximations, we look at Newtonian laminar flow in straight tube and channel with radius a. To these problems one can derive exact solutions of the Navier-Stokes equations, given a set of boundary conditions. Since the focus of this thesis is the effects of WSS, the exact solution for WSS is the benchmark of convergence.

In our test case, we use a simple model of the female middle cerebral artery (MCA) with physical parameters gathered from [6,41], a commonly used Newtonian blood viscosity and a high average velocity to stress test the stability of our model,

$$V = 750 \ mm/s$$
$$a = 1.21 \ mm$$
$$L = 20 \ mm$$
$$\mu = 0.00345 \ Pa \cdot s$$

where a is the radius and L is the length of the channel/tube. The corresponding Reynold's number[7] is

$$\text{Re} = \frac{2\rho V a}{\mu} \approx 555.$$

The 2D-mesh is created from the FEniCS built-in mesh *Rectangle*. To get a 3D-mesh of a cylinder where the refinement of the mesh could easily be changed, a modifiable mesh was created from code using the FEniCS module *MeshEditor*. The source code of this can be found in Appendix A.

5.1. Exact solutions

We start by deriving an exact solution for a straight, cylindrical tube with radius a. To take advantage of the symmetry of the tube, the Navier-Stokes equations for a Newtonian, incompressible fluid are rewritten in cylindrical coordinates [20] where x is the direction of the tube

$$\frac{\partial u_r}{\partial t} + (\mathbf{u} \cdot \nabla)u_r - \frac{u_\theta^2}{r} = -\frac{1}{\rho}\frac{\partial p}{\partial r} + \nu \left(\nabla^2 u_r - \frac{u_r}{r^2} - \frac{2}{r^2}\frac{\partial u_\theta}{\partial \theta} \right)$$
$$\frac{\partial u_\theta}{\partial t} + (\mathbf{u} \cdot \nabla)u_\theta + \frac{u_r u_\theta}{r} = -\frac{1}{\rho r}\frac{\partial p}{\partial \theta} + \nu \left(\nabla^2 u_\theta + \frac{2}{r^2}\frac{\partial u_r}{\partial \theta} - \frac{u_\theta}{r^2} \right) \quad (5.1)$$
$$\frac{\partial u_x}{\partial t} + (\mathbf{u} \cdot \nabla)u_x = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu \nabla^2 u_x.$$

In a fully developed, laminar flow in a straight tube, the flow is only in the direction of the tube, and none of the variables depend on θ. We therefore denote the x-component of the velocity simply as u from now on.

---

[7]The Reynold's number is a dimensionless number which is the ratio of the inertial to the viscous forces. Traditionally, a Reynold's number of less than 2100 indicates laminar flow (as opposed to turbulent flow).
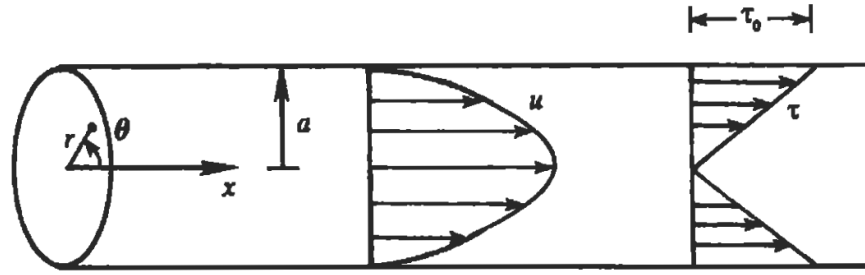
Figure 5.1: Laminar flow in a straight channel/cylinder.(Reprinted with permission from *Kundu, Cohen: Fluid Mechanics, 4th edition*[20]).

The first momentum equation above reduces to

$$-\frac{\partial p}{\partial r} = 0.$$

The pressure is thus only dependent of x. The third equation reduces to

$$-\frac{dp}{dx} + \frac{\mu}{r}\frac{d}{dr}\left(r\frac{du}{dr}\right) = 0$$

The first term can only be a function of x and the second term can only be a function of r (since the flow is assumed fully developed). Thus, both terms must be constant. Integrating twice gives

$$u = \frac{r^2}{4\mu}\frac{dp}{dx} + A\ln r + B.$$

The velocity must be bounded, hence, A=0. The no-slip condition at the wall $(r = a)$ gives $B = \frac{a^2}{4\mu}\frac{dp}{dx}$. Thus, we are left with a parabolic velocity distribution across the tube of the form

$$u = \frac{r^2 - a^2}{4\mu}\frac{dp}{dx} \tag{5.2}$$

The shear stress in cylindrical coordinates may be written as

$$\tau = \mu\left(\frac{\partial u_r}{\partial x} + \frac{\partial u}{\partial r}\right).$$

In our special case, the shear stress distribution reduces to

$$\tau = \mu\frac{du}{dr} = \frac{r}{2}\frac{dp}{dx}.$$

Which means that the magnitude of the WSS is exactly equal

$$\tau_w = \left|\frac{a}{2}\frac{dp}{dx}\right| \tag{5.3}$$

A two dimensional straight channel

The derivation above also applies for the 2D case of a straight channel. We can find $\frac{dp}{dx}$ by our average flow velocity, V, which is known through our inlet boundary condition, and can be expressed as

$$V = \frac{1}{2a} \int_{-a}^{a} u \, dy = -\frac{a^2}{6\mu} \frac{dp}{dx}$$

Meaning that an exact solution of the WSS can be written as

$$\tau_w = \frac{3\mu V}{a}$$

Using our parameters, the exact WSS to an accuracy of 4 decimals is

$$\tau_w = 6.4153 \; Pa. \tag{5.4}$$

A three dimensional cylindrical tube

The average velocity in our fully developed flow can be written as

$$V = \frac{1}{\pi a^2} \int_{0}^{a} \int_{0}^{2\pi} u \, d\theta \, dr = -\frac{a^2}{8\mu} \frac{dp}{dx}.$$

Meaning that in the three dimensional case, the WSS can be written as

$$\tau_w = \frac{4\mu V}{a}$$

which, in our test case, to an accuracy of 4 decimals, amount to

$$\tau_w = 8.5537 \; Pa. \tag{5.5}$$

5.2. 2D results

P2-P1 vs P1-P1

Two different sets of elements will be compared. *P2-P1 elements* denote a quadratic approximation of the velocity fields ($\mathbf{CG}_2$) and linear approximation of the pressure field (CG_1), and *P1-P1* denote a linear approximation for both the velocity field and the pressure field. To compare P2-P1 against P1-P1 elements, we study the solution where the flow is fully developed and has reached steady state. To ensure that the solution is fully developed, the velocity inlet condition on the velocity is set to be parabolic. It is not known exact how long it will take the solution to converge to a steady state solution (this depends on the fluids

dt(s)\h(mm)	5.00e-1	2.50e-1	1.25e-1	6.25e-2
8.00e-2	7.37e-1	6.49e-2	2.34e0	6.89e0
4.00e-2	1.00e-3	2.02e-2	6.69e-2	1.56e-1
2.00e-2	2.00e-6	1.30e-5	3.40e-5	5.30e-5
1.00e-2	2.60e-5	1.90e-5	1.20e-5	2.50e-5
5.00e-3	1.30e-5	1.10e-5	1.20e-5	1.70e-5
2.50e-3	9.00e-6	8.00e-6	1.10e-5	1.70e-5
1.25e-3	1.40e-5	9.00e-6	1.00e-5	1.60e-5

Table 5.1: 2D-results for the relative error in WSS. P2-P1 elements.

dt(s)\h(mm)	2.50e-1	1.25e-1	6.25e-2	3.13e-2
8.00e-2	6.84e-1	5.03e-1	1.14e0	4.55e0
4.00e-2	8.76e-2	2.63e-2	2.47e-2	1.01e-1
2.00e-2	9.10e-2	4.18e-2	2.08e-2	1.02e-2
1.00e-2	9.10e-2	4.18e-2	2.07e-2	1.02e-2
5.00e-3	9.10e-2	4.18e-2	2.07e-2	1.01e-2
2.50e-3	9.10e-2	4.18e-2	2.07e-2	1.01e-2
1.25e-3	9.10e-2	4.18e-2	2.07e-2	1.01e-2

Table 5.2: 2D-results for the relative error in WSS. P1-P1 elements.

initial state), so a reference value time of T=0.265s was chosen by running the model at a very fine time stepping.

The error reported is the relative L^2-error of the WSS over a small area of the wall in the middle of the channel. The cell size, h, is the smallest circumradius of a cell in the mesh, and all meshes are practically uniform. Results of the P2-P1 and P1-P1 simulations can be seen in Table 5.1 and Table 5.2 respectively. The meshes used in the P2-P1 simulations are coarser, due to the fact that there are more unknowns per cell in P2-elements.

The results for P2-P1 elements are very close to the exact solution, and the deviation can probably be explained by the numerical solution not being completely fully developed and steady state. The P1-P1 elements show a nice, approximately linear convergence with h, however, it does not compare to P2-P1 elements in terms of accuracy.

The time stepping seems to have relatively little influence on the convergence. We can only see some signs of convergence in Δt on the finest meshes on P2-P1 elements. This is hardly a surprise, seeing as we only study a steady state solution. However the solution does need to converge to this steady state, which it seems to do at a fairly large time step.

Model	WSS (Pa)	Abs. deviation (Pa)	Rel. deviation (%)
Newtonian	6.4153	0.0	0.0
Carreau	6.8690	0.4537	7.1
Carreau-Yasuda	6.6408	0.2255	3.5
Cross	6.4802	0.0649	1.0
Modified Cross	6.9653	0.5500	8.6
Powell-Eyring	6.5535	0.1382	2.2
Casson	6.2413	-0.1740	-2.7
Power-law	5.0620	-1.3533	-21.1

Table 5.3: Predicted WSS by the different non-Newtonian models.

	WSS (Pa)	Abs. deviation (Pa)	Rel. deviation (%)
Newtonian	6.4153	0.0000	0.0
Diam. -15%	7.5474	1.1321	17.6
Diam. -10%	7.1281	0.7128	11.6
Diam. -5%	6.7529	0.3376	5.5
Diam. +5%	6.1098	-0.3055	-4.8
Diam. +10%	5.8321	-0.5832	-9.1
Diam. +15%	5.5785	-0.8368	-13.0

Table 5.4: Effects of geometric uncertainties.

Non-Newtonian models

To get an idea of the difference in WSS the different viscosity models will predict, all the different models are implemented using the same parameters as for the Newtonian results in the previous section. In these simulations, the convection term was set equal to zero, as this ensured a fully developed flow profile in the relatively short channel. The results are seen in Table 5.3.

As expected from Figure 3.1 the results deviate quite a bit from the Newtonian result. Only the Casson and power-law models predict a lower shear stress than the Newtonian model. Figure 5.2 shows the different fully developed velocity profiles for the Modified Cross and the power-law model, as these differ most from the Newtonian result, and the Casson model. As shown, the velocity profile of the non-Newtonian viscosity models are somewhat flattened. Note how even though the difference between the velocity profiles may seem very small, the difference in WSS may still be significant.
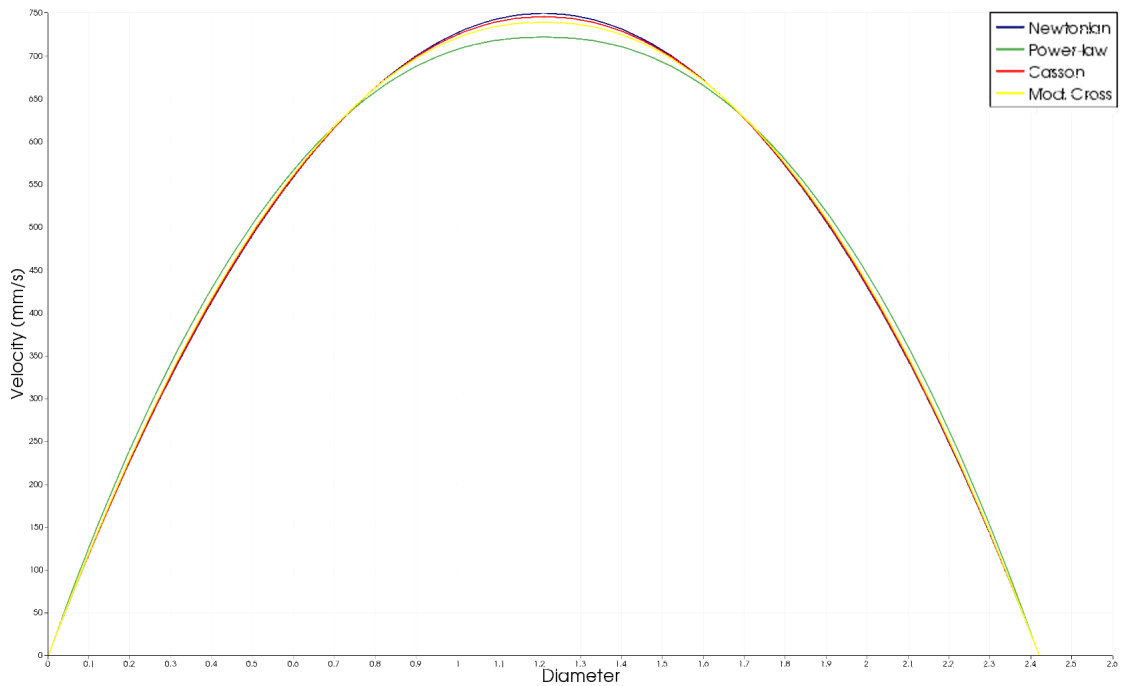
Figure 5.2: Comparison of fully developed velocity profiles.

Geometric sensitivity

There is a lot of uncertainty related to the physical parameters when using patient specific data. The resolution of the images taken of the brain may be very poor compared to the actual size of the vessels of interest, and thus a source of error arise. With a resolution of 2 pixels/mm, it is not unreasonable to assume at least a measurement error that in our test case would amount to 5-15% difference in the diameter of the vessel. Thus, we wish to see how much effect on the WSS such geometrical uncertainty will have. The results can be seen in Table 5.4. These results are directly comparable to the 3D-case, as can be seen from the derivation of the exact solutions in Section 5.1.

We see that the geometric insecurity seems significant, and compared to the non-Newtonian models this seems to dominate the significance of the non-Newtonian effects. However, these are only preliminary results, and more care will be taken in addressing the matter in Section 6.

dt(s)\h(mm)	1.10e0	6.60e-1	4.00e-1	2.80e-1
8.00e-2	-	-	1.96e-1	2.24e-1
4.00e-2	3.59e-1	-	1.04e-1	6.80e-2
2.00e-2	-	-	7.35e-2	4.51e-2
1.00e-2	-	2.05e-1	8.06e-2	3.57e-2
5.00e-3	-	1.99e-1	9.05e-2	5.07e-2
2.50e-3	3.69e-1	2.03e-1	9.08e-2	5.20e-2
1.25e-3	3.70e-1	2.04e-1	9.08e-2	5.20e-2

Table 5.5: 3D-results for the relative error in WSS. P2-P1 elements.

dt(s)\h(mm)	4.00e-1	2.00e-1	1.50e-1	1.00e-1
8.00e-2	9.47e-2	1.51e-2	4.69e-2	1.39e-1
4.00e-2	1.04e-1	5.18e-2	3.91e-2	2.44e-2
2.00e-2	1.14e-1	6.43e-2	5.25e-2	3.87e-2
1.00e-2	1.10e-1	5.86e-2	4.80e-2	3.53e-2
5.00e-3	1.09e-1	5.58e-2	4.38e-2	2.87e-2
2.50e-3	1.07e-1	5.51e-2	4.30e-2	2.83e-2
1.25e-3	1.07e-1	5.52e-2	4.29e-2	2.83e-2

Table 5.6: 3D-results for the relative error in WSS. P1-P1 elements.

5.3. 3D results

P2-P1 vs P1-P1

The results for the three dimensional cylinder have been obtained in a similar fashion as those for the two dimensional channel.

The results differ quite a bit from the 2D-results. This is because in this 3D-model, the curvature of the cylinder wall needs many cells to be accurately described. Since this is the case, the P1-P1 elements have a major advantage in that they are by far more computationally efficient than the P2-P1 elements, and thus, more cells can be used. The effect this has on the accuracy can clearly be seen in Table 5.5 and Table 5.6.

The convergence rate for the P1-P1 elements seems to be almost linear, whereas the P2-P1 elements seems to demonstrate a better convergence rate. Some instability were seen on the lower resolution meshes for P2-P1 elements.
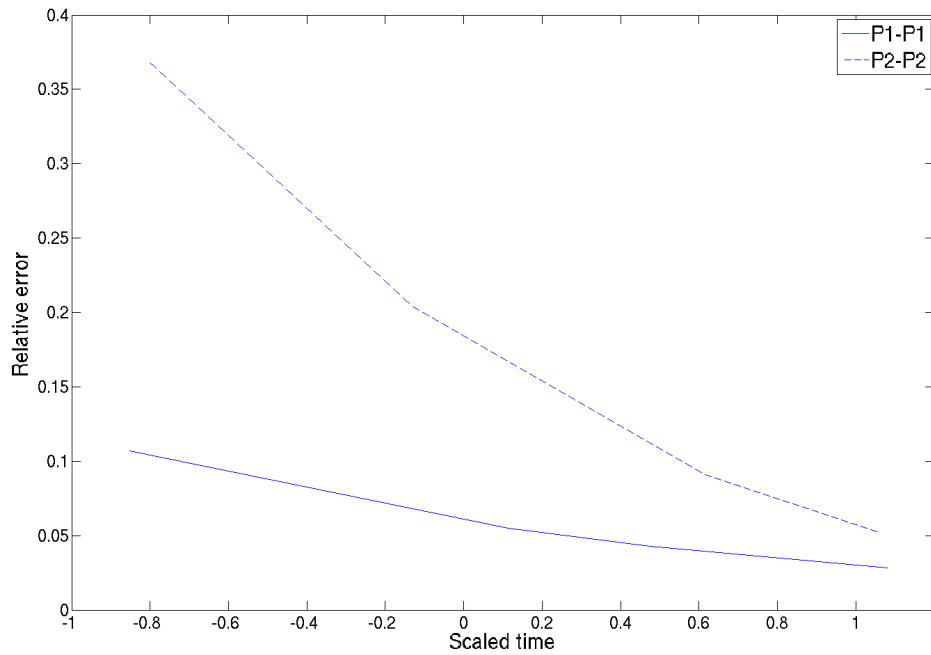
Figure 5.3: Computational needs for three dimensional P2-P1- and P1-P1 elements.

Efficiency

Whether P2-P1 elements or P1-P1 elements are to be preferred depend on the efficiency. The selected refinements for Table 5.5 and Table 5.6 are chosen such that their computational needs are similar, and that they are within range of what is computationally possible with larger models, i.e. aneurysm models. A comparison of the computational needs of the different elements can be seen in Figure 5.3.

5.4. Conclusions

For the computational power available here, the P1-P1 elements are clearly favorable.

The time step needed seems quite large, but there are several things not taken into account in these simulations. Both the operator splitting and linearization of the viscosity term introduce errors of order of order $O(\Delta t)$. Secondly, these simulations are run with a constant inflow condition, rather than a pulsatile inflow, which will be used in our aneurysm simulations. Therefore, a time step of $\Delta t = 1.25e - 3$ is chosen for all simulations in the following chapters.

6. Analysis of simulations on a single aneurysm

Many different factors might contribute to inaccuracies in our simulations. Both geometric sensitivity and non-Newtonian effects have been mentioned and treated to some extent in Section 5 but will be addressed again in this section. The effects of varying the hematocrit level will also be studied in this section. The selected geometry on which to perform these studies, is a saccular aneurysm located at the bifurcation of the middle cerebral artery, in a female patient of age 70. It is a medium sized aneurysm with a volume of approximately 350mm^3 and a diameter of 10mm. The aneurysm is known to have ruptured at a later time.

The aneurysm has been segmented from CT-images in three different ways, using VMTK[42]. This particular set of CT-images had a voxel size[8] of 0.39mm x 0.39mm x 0.7mm.

This is a qualitative study in order to get an idea of what kind of differences arise between different assumptions on geometry and viscosity. A quantitative approach to assess the effects of the assumption of Newtonian vs non-Newtonian flow is done in Section 7.4. Similarly, a quantitative approach to assessing the effect of a change in hematocrit is done in Section 7.5.

(a) Cross section   (b) Isosurface   (c) Zoom-in on aneurysm
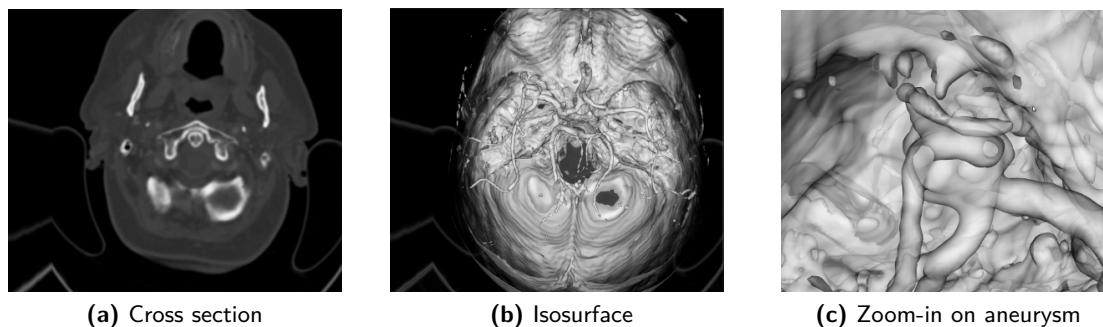
Figure 6.1: The CT images which the aneurysm has been segmented from.

6.1. Models

Assumptions on flow

The aneurysm geometry studied differs from a typical "textbook" aneurysm geometry. As CT images do not depict the flow velocity or direction of the flow, but merely an intensity, the images provide an idea of where there is blood, but some assumptions need to be made on where there is *blood flow*. The aneurysm is depicted from different angles with labeled inflow and possible outflow in Figure 6.2.

The assumption made is that all the vessels around the aneurysm are in fact arteries, and that the blood flows out through both outlets A,B and C, and that the arteries from B and

---

[8]Voxel size is the three dimensional equivalent to pixel size.
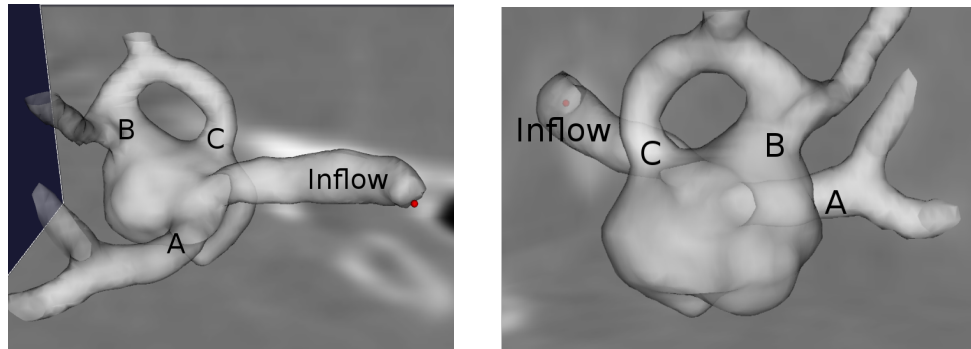
Figure 6.2: The aneurysm seen from two different angles

C reconnect downstream. Other possible interpretations are that blood flows out through only A and C or only A and B, leaving the remaining arteries without a direct connection with the aneurysm. This clearly demonstrates the uncertainty related to medical images of a coarse resolution.

Generating the models

The CT image is a greyscale image, with high intensity levels where the blood concentration is high, i.e., in arteries and veins. However, exactly where the *limit* for what is the inside of a vessel and what is not is difficult to say. Thus, three different levels (100, 120, 140) have been used as a lower threshold for what is to be interpreted as arteries when segmenting the image. The upper and lower threshold is chosen at the extremes, such that it just captures the basic geometry of the vessels. The middle value is chosen as the mean value of the two extrema. All three thresholds seem to capture the basic geometry well.

(a) Low threshold   **(b)** Medium threshold   **(c)** High threshold
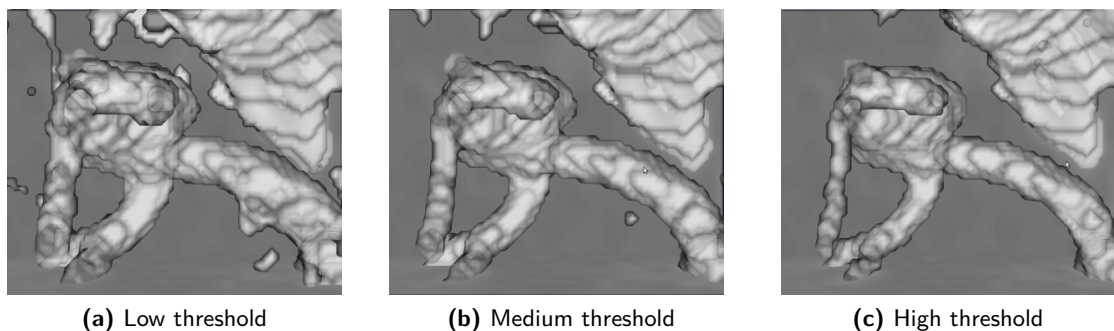
Figure 6.3: Outlines of the geometry for the different thresholds

To smoothen the surface, different VMTK[42] scripts were used. In the script *vmtklevelsetsegmentation*, the input parameters were chosen to preserve the characteristics of the geometry

(a) Low threshold   (b) Medium threshold   (c) High threshold
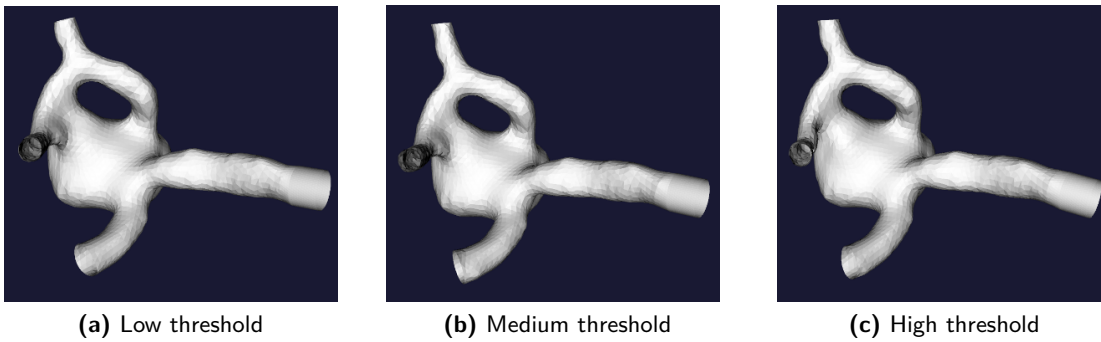
Figure 6.4: Final models

as

- NumberOfIterations: 300

- PropagationScaling: 0.15

- CurvaturScaling: 0.0

- AdvectionScaling: 1.0.

In the script *vmtksurfacesmoothing* the smoothing parameters were chosen as recommended at the VMTK website, namely

- Passband: 0.1

- Iterations: 30.

The input parameters were the same for all models. A short cylindrical extension of approximately one diameter was added to the inlets to enable a completely parabolic inlet profile. The final models are depicted in Figure 6.4. Notice how the smoothing seem to have affected the geometry by comparing Figure 6.3 to Figure 6.4. The final smoothed models seem much more similar than the simple threshold segmentation, which might indicate that the smoothing parameters are quite significant for the simulations. This matter will not be addressed here.

The volumes of the low, medium and high threshold were 560, 571 and 558mm^3 respectively. Even with a possibility of some differences arising from clipping the surface at different places, this clearly indicates that the three different thresholds have produced very similar geometries.

Meshes

The meshes created from the models were also created using VMTK. Efforts were made to refine the resolution where most needed. That is, the aneurysm itself have a finer resolution
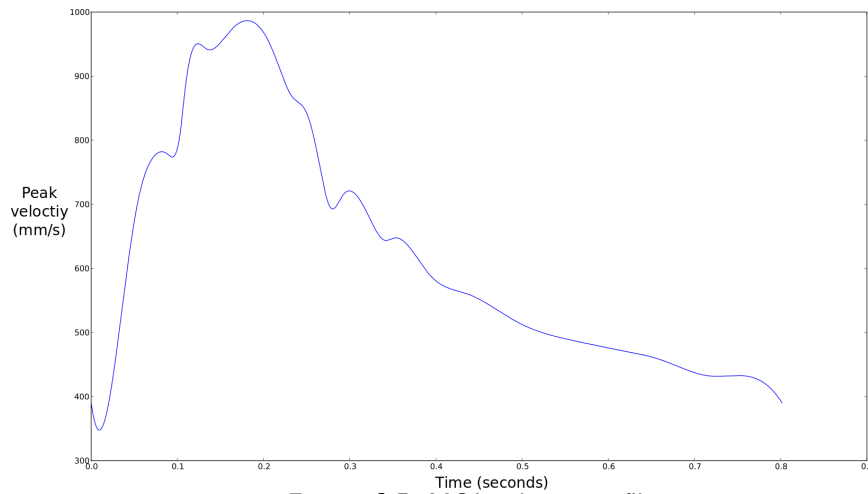
Figure 6.5: MCA velocity profile

than the inlet and outlets. This was done using the user-contributed script *vmtkdijkstradistancetopoints*. A boundary layer was also added, as any errors occurring close to the wall may give rise to significant errors in the calculation of WSS. The finale meshes were created using the script *vmtkmeshgenerator*.

Inlets and outlets were marked using FEniCS Meshbuilder.

All three meshes have a total cell count of approximately 1,300,000 cells.

6.2. Simulations

The inlet velocity is set to be parabolic (see Section 3.5). The spatial peak velocity is determined by using a pulsatile flow profile gathered from a similar-aged female patient undergoing cerebrovascular treatment, at a heart rate of 75 bpm. It has been scaled to a timed average of 575mm/s, which has been found to be an average flow velocity in the MCA, in women over 60[43]. The flow profile is depicted in Figure 6.5.

The WSS was of special interest. It was measured over the full aneurysm surface, which was manually selected using Meshbuilder (see Figure 6.6). Efforts were made to ensure that the equivalent areas were marked in all three segmented geometries.

In the simulations where a Newtonian viscosity model is used, the viscosity is assumed to be $\mu = 0.00345 Pa \cdot s$.
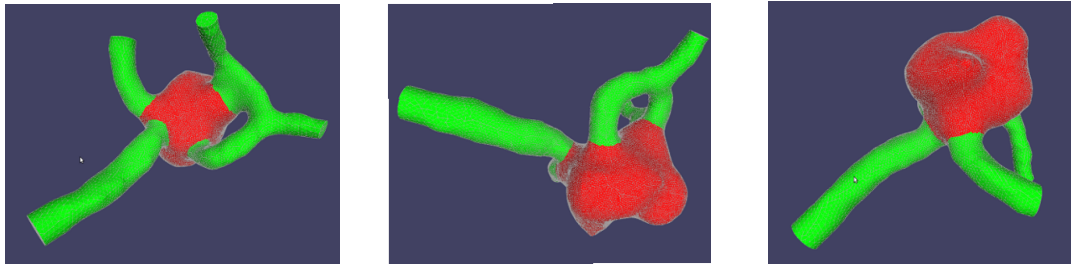
Figure 6.6: Aneurysm marking seen from different angles.

6.3. Visualization of results

Paraview[44] was chosen to visualize the results. In the following sections both the velocity changes and WSS changes are compared with respect to geometry, hematocrit levels and non-Newtonian effects. Two different methods were chosen to compare the velocity. First, a single point inside the aneurysm was chosen, and the velocity magnitude was measured there for comparison through a full cycle. This point was assumed to be representative of the aneurysm interior. Secondly, a slice in the middle of the aneurysm was extracted (see Figure 6.7), where the velocity magnitude was compared at both systole and diastole. The direction of the flow has not been compared in this chapter.
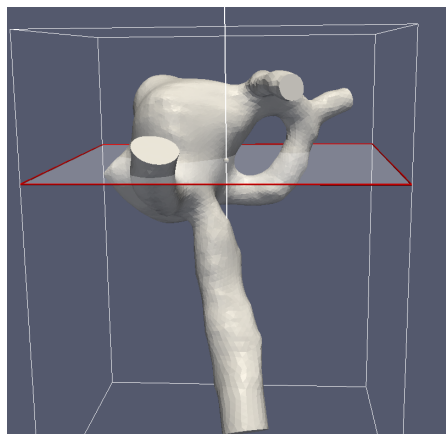
Figure 6.7: Slice through the middle of the aneurysm to compare the velocity magnitudes.

To compare the WSS, three different viewing angles were chosen to best view the difference over the whole aneurysm dome.

In most visualizations, both the velocity magnitude and the WSS was compared *relatively*

to reference values $(\mathbf{u}_R, \tau_R)$. These relative values was calculated as follows

$$D_{\mathbf{u}_R}(\mathbf{u}(\mathbf{x}), \mathbf{x}) = \frac{\|\mathbf{u}(\mathbf{x})\| - \|\mathbf{u}_R(\mathbf{x})\|}{\|\mathbf{u}_R(\mathbf{x})\|}$$

$$D_{\tau_R}(\tau, \mathbf{x}) = \frac{\tau(\mathbf{x}) - \tau_R(\mathbf{x})}{\tau_R(\mathbf{x})}$$

6.4. Geometric effects

To assess the geometric sensitivity, simulations were run using all of the three different segmentations, with a Newtonian viscosity model. The reference values were chosen as the results from simulations on the medium threshold geometry.

Velocity
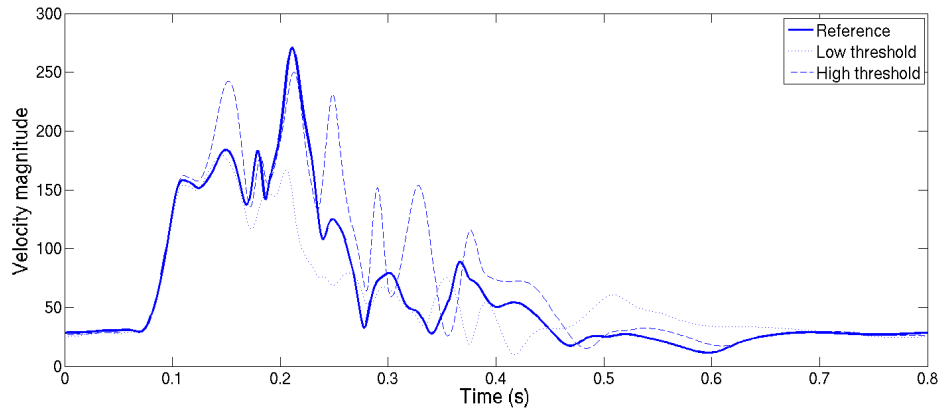
Figure 6.8: The figure shows velocity magnitude measured at a point in the middle of the aneurysm, for all three different geometries, over one full cycle.

The results seen in Figure 6.8 show surprisingly large deviations from the reference value, particularly at systole. At end systole, the velocity magnitude for the high threshold are, for a short period of time, over three times as large as the reference. One would expect the high threshold geometry to predict higher velocities inside the aneurysm as the volumes of the arteries and the aneurysm would be smaller. Similarly, one would expect lower velocities for the low threshold geometry. However, the way the inlet condition has been set, this effect is dampened by the reduction in cross sectional area at the inlet which affects the inlet flux of the about 5% from the low threshold geometry to the high threshold geometry. It is however possible to see a slight tendency of these velocity differences.

(a) Low threshold **(b)** Medium threshold **(c)** High threshold
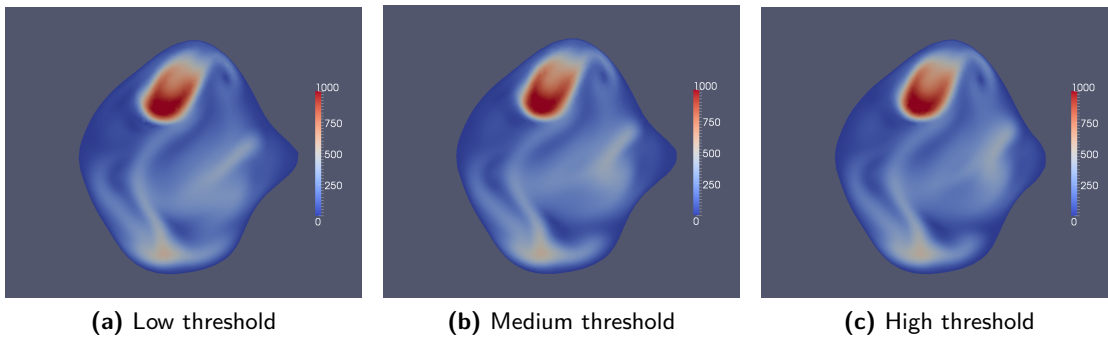
Figure 6.9: The figure shows systolic velocity magnitude in a slice through the aneurysm for the different geometries. The velocity magnitude ranges from 0 to 1000 mm/s.


(a) Low threshold **(b)** Medium threshold **(c)** High threshold
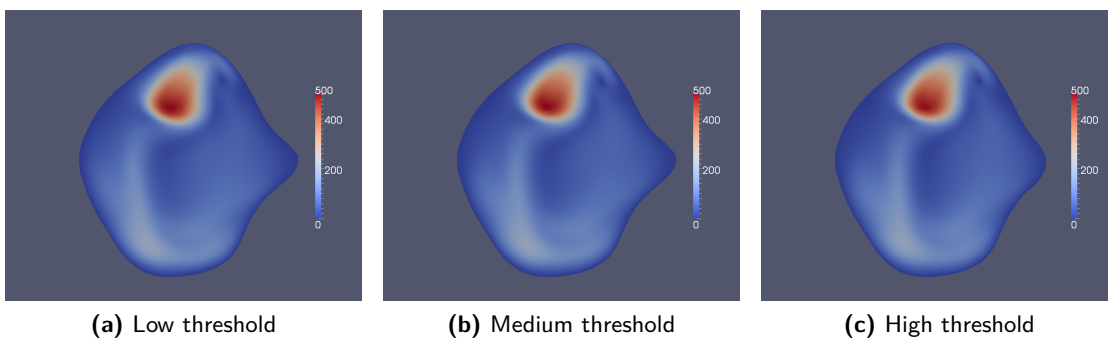
Figure 6.10: The figure shows diastolic velocity magnitude in a slice through the aneurysm for the different geometries. The velocity magnitude ranges from 0 to 500 mm/s.

We see in Figure 6.9 and Figure 6.10 that the differences seen in Figure 6.8 is difficult to see in this slice. There is very little difference overall, but particularly at diastole, the flow patterns seem almost equal. Thus, even though there can be locally quite large deviations, the general properties of the flow do not seem to change dramatically.

Wall shear stress

Figure 6.11 shows that the average WSS is hardly affected at all by the different geometries. In fact, the largest deviation from the reference value at the medium threshold geometry is below 5% over the full cycle. However, there is still a question whether the distribution has changed. When considering the small changes seen in the velocity distribution, one would not expect any major differences in WSS distribution either. Looking at the systolic distribution of WSS in Figure 6.12, it is clear that this is in fact the case. Similar results can be seen at diastole (Figure 6.13), hardly any change is seen in the distribution of WSS.

51
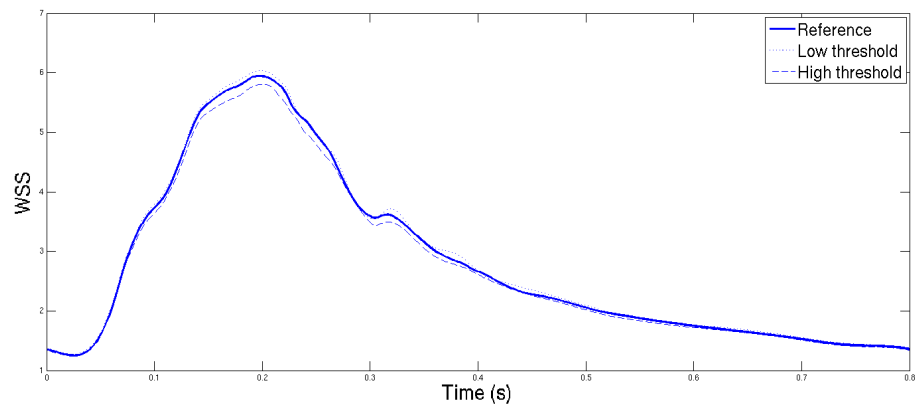
Figure 6.11: The figure shows average WSS measured over the aneurysm surface, for all three different geometries, over a full cycle.

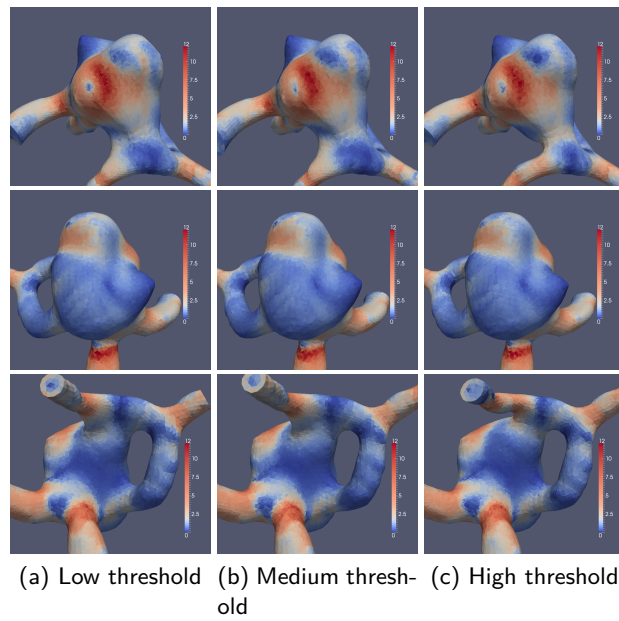(a) Low threshold   (b) Medium thresh-   (c) High threshold
old
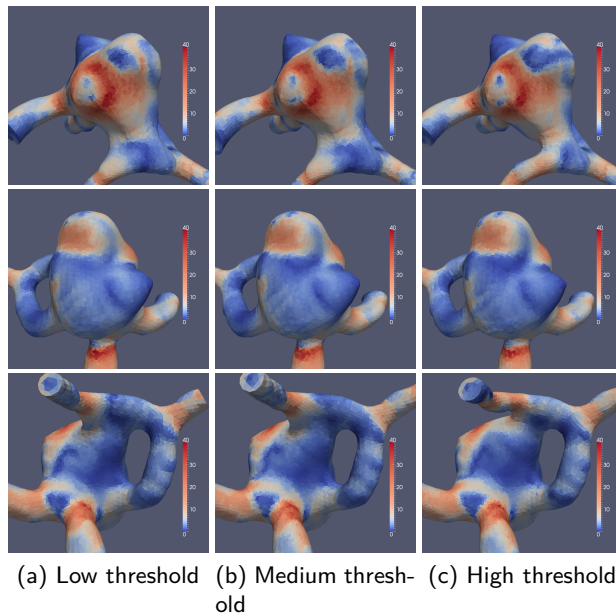
Figure 6.12: The figure shows systolic WSS seen from different angles, for all three different geometries. The range is from 0 to 40 Pa.

6.5. Non-Newtonian effects

To evaluate the non-Newtonian effects four different viscosity models were implemented. A Newtonian viscosity model was chosen as a reference. The Casson model was chosen because it incorporates hematocrit and a yield stress, and the Modified Cross and Power-law models were chosen on basis of the results in Section 5, as these two showed the largest deviation from the Newtonian WSS values (see Table 5.3).

(a) Low threshold   (b) Medium thresh-   (c) High threshold
                        old

Figure 6.13: The figure shows diastolic WSS seen from different angles, for all three different geometries. The range is from 0 to 12 Pa.
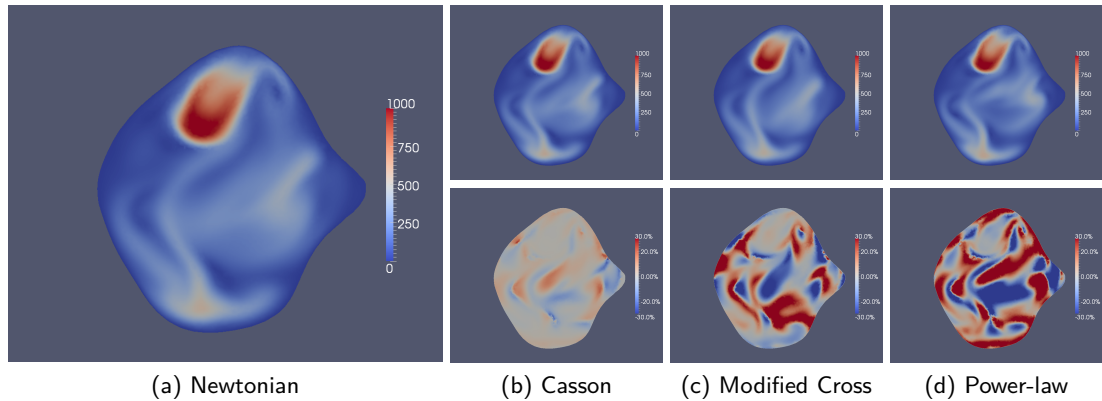
Velocity
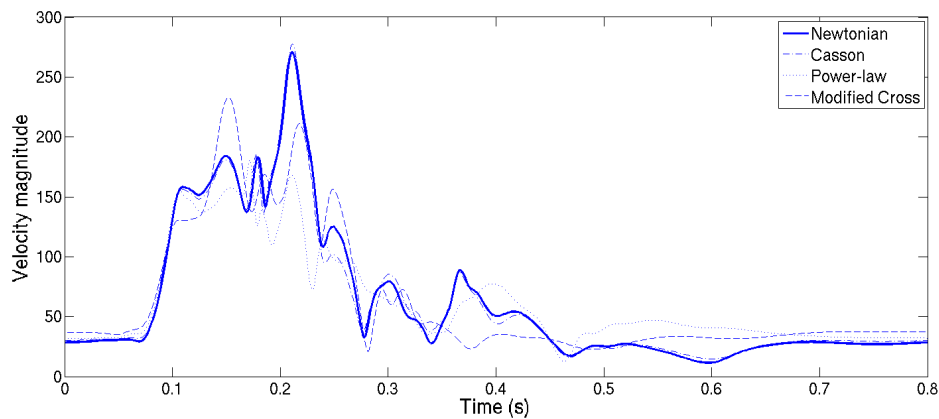
Figure 6.14: The figure shows velocity magnitude measured at a point in the middle of the aneurysm, for all four different viscosity models, over a full cycle.

Looking at the velocity in the interior point in Figure 6.14, we see a generally nice correspondence to the velocity profile applied to the inlet. However, there seem to be more high-frequent changes at this point than at the inlet, which might indicate some instability in the flow. One might expect that the shear-thinning behaviour of the implemented non-Newtonian models would cancel out some of the high frequencies, but any such effects (if

(a) Newtonian   (b) Casson   (c) Modified Cross   (d) Power-law

Figure 6.15: The figure shows velocity magnitude at systole. (a) Shows the reference velocity magnitude for the Newtonian model used as reference. Top row of (b)-(d) shows the velocity magnitude for the non-Newtonian models, and the bottom row of (b)-(d) shows the relative magnitude difference for the non-Newtonian models, compared against the Newtonian model. The velocity is ranging from 0 to 1000 mm/s, and the relative magnitude is ranging from -30% to 30%.

present) are very hard to see.

From Figure 6.15 and Figure 6.16 we see that the local non-Newtonian effects might be much more significant than what Figure 6.14 predicts. The relative differences between the different models seem very large in parts of the aneurysm. Note that the large relative differences occur mainly where the flow velocity is low. In the areas of high velocity, the non-Newtonian effects are barely noticeable at all. Note also that there seem to be much more significant differences in flow pattern at systole than at diastole. This corresponds well with Figure 6.14, where there are clearly more high frequency difference at systole than at diastole.

Wall shear stress

Figure 6.17 shows that the Casson and Modified Cross models both predict very similar average WSS compared to the Newtonian model. The power-law model predicts a significantly lower WSS, and this is likely related to the lack of an asymptotic viscosity as the shear rate increases. As the shear rates in these calculations reach as high as $20000 s^{-1}$, it is no surprise that it predicts quite different results compared to the other models.

When looking at a more detailed view, as in Figure 6.18 and Figure 6.19, we see that the local differences might be very significant. The Casson-model predicts a very similar WSS distribution to that of the Newtonian, with only a few smaller areas that are clearly different from the Newtonian model. The Modified Cross seem to predict quite different result in the areas of low WSS, and it seems to generally produce slightly higher WSS overall. The power-law model predicts significantly lower WSS in areas of high WSS, and higher in areas

(a) Newtonian   (b) Casson   (c) Modified Cross   (d) Power-law
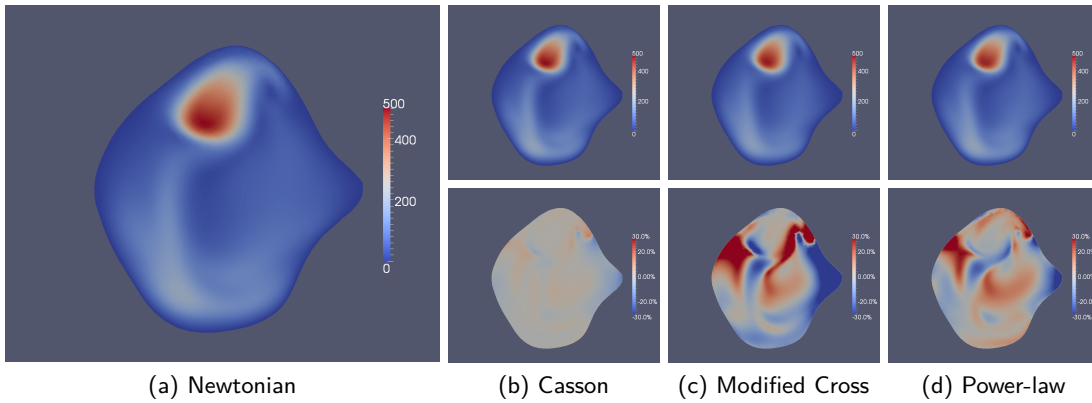
Figure 6.16: The figure shows velocity magnitude at diastole. (a) Shows the reference velocity magnitude for the Newtonian model used as reference. Top row of (b)-(d) shows the velocity magnitude for the non-Newtonian models, and the bottom row of (b)-(d) shows the relative magnitude difference for the non-Newtonian models, compared against the Newtonian model. The velocity is ranging from 0 to 500 mm/s, and the relative magnitude is ranging from -30% to 30%.
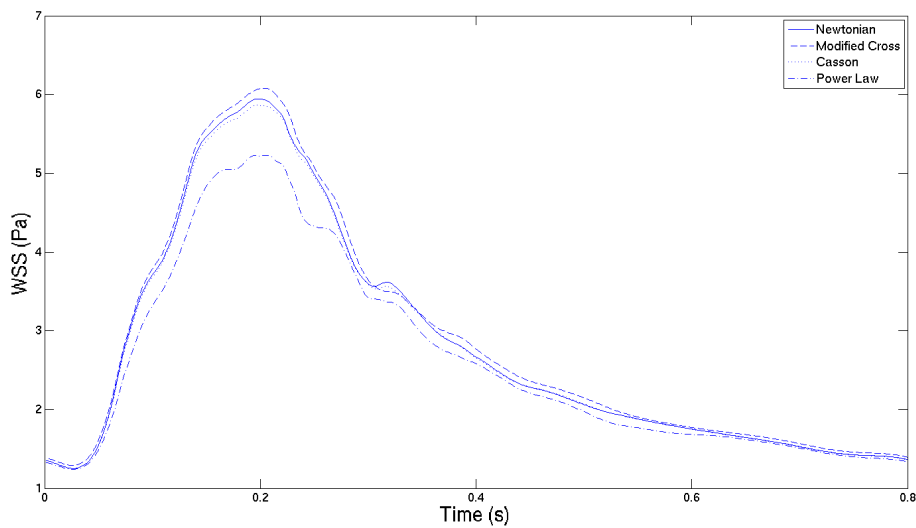
Figure 6.17: The figure shows average WSS measured over the aneurysm surface, for all four viscosity models, over a full cycle.

of low WSS.

Comparing systole to diastole, the models seem to agree much more at diastole than at systole, as expected from the results regarding the velocity differences.

(a) Newtonian   (b) Rel. Casson   (c) Rel. Mod. Cross   (d) Rel. Power-law
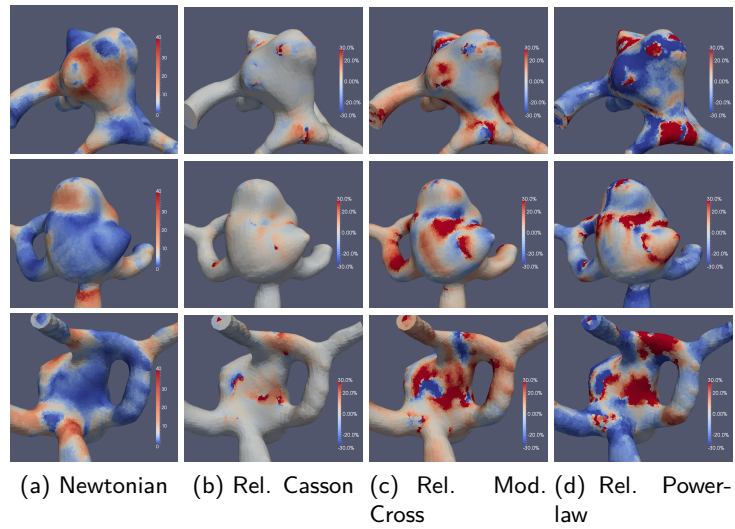
Figure 6.18: The figure shows systolic relative differences in WSS, seen from different angles. Column (a) shows the Newtonian reference value. Columns (b)-(d) shows the differences measure for the different viscosity models, relative to the Newtonian value.. The WSS range from 0 to 40 Pa, and the relative differences range from -30% to 30%.


(a) Newtonian   (b) Rel. Casson   (c) Rel. Mod. Cross   (d) Rel. Power-law
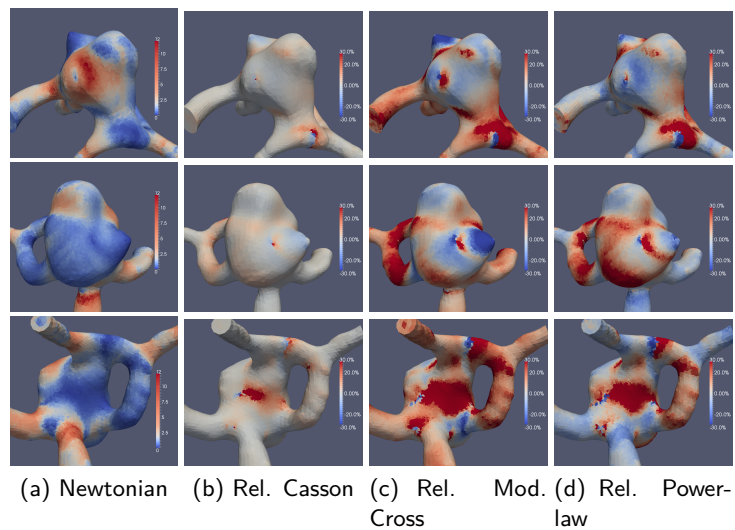
Figure 6.19: The figure shows diastolic relative differences in WSS, seen from different angles. Column (a) shows the Newtonian reference value. Columns (b)-(d) shows the differences measure for the different viscosity models, relative to the Newtonian value. The WSS range from 0 to 12 Pa, and the relative differences range from -30% to 30%.

6.6. Effects of different hematocrit levels

To assess the effects of different hematocrit levels, the Casson model was implemented with three different levels of hematocrit, 37.5%, 40% and 42.5%. The Casson model incorporates the hematocrit level in both the yield stress and the asymptotic viscosity as the shear rate increases. The results of the simulations run with a hematocrit value of 40% are selected as the reference values, and the low and high hematocrit values are compared to these.
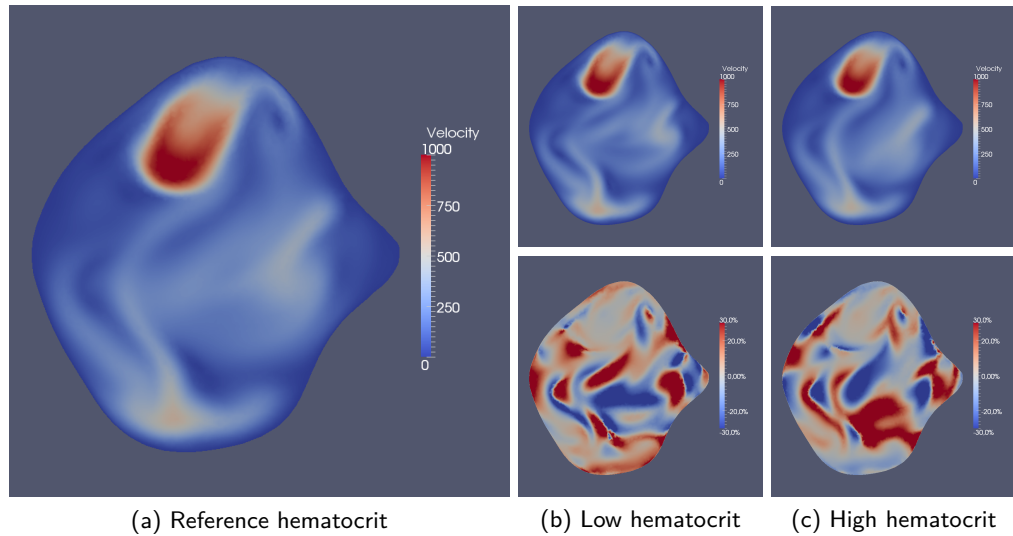
Velocity
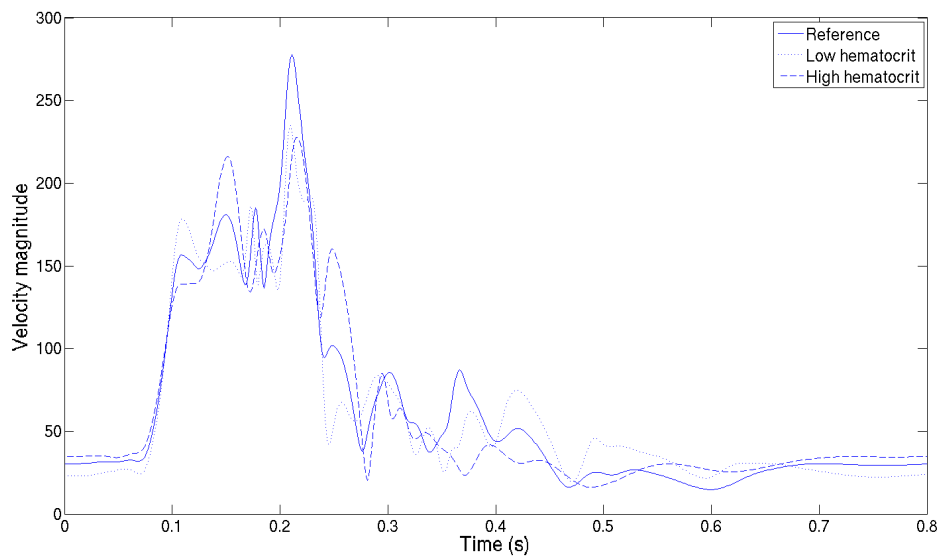
Figure 6.20: Velocity magnitude measure at a point in the middle of the aneurysm for the different hematocrit levels.

From Figure 6.20, it is difficult to see any clear tendencies in the blood velocity inside the aneurysm. However, at late diastole and early systole, it seems as though the higher hematocrit predicts higher velocities. This is in one way surprising, as the higher hematocrit yields higher viscosity, and would thus intuitively produce slower blood flow. However, in our simulations, we do not prescribe any pressure at the inlet, and thus, the simulations might compensate for a higher viscosity by predicting a higher pressure drop.

Studying the velocity magnitude in Figure 6.21 and Figure 6.22, we see again that there are locally large differences, especially at diastole. We note that at the higher hematocrit the blood flow is seemingly less chaotic, i.e., the velocity gradient seems smaller. At diastole, we see results contradicting the results seen in Figure 6.20, as the blood velocity seems slightly higher for the low hematocrit flow, than the high hematocrit flow. This is evidence that studying local changes might not tell the full story.

(a) Reference hematocrit   (b) Low hematocrit   (c) High hematocrit

Figure 6.21: The figure shows velocity magnitude at systole, ranging from 0 to 1000 mm/s. On the left is the reference velocity magnitude with a hematocrit level of 40%. The top row shows velocity magnitude at low and high hematocrit levels. The bottom row shows the relative magnitude difference compared to the reference velocity, ranging from -30% to 30%.


(a) Reference hematocrit   (b) Low hematocrit   (c) High hematocrit
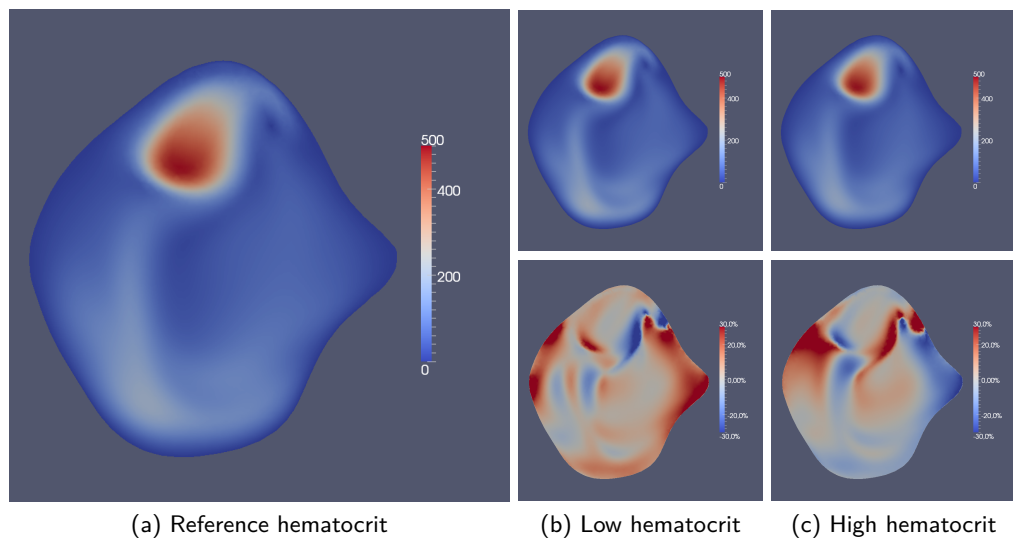
Figure 6.22: The figure shows velocity magnitude at diastole, ranging from 0 to 500 mm/s. On the left is the reference velocity magnitude with a hematocrit level of 40%. The top row shows velocity magnitude at low and high hematocrit levels. The bottom row shows the relative magnitude difference compared to the reference velocity, ranging from -30% to 30%.
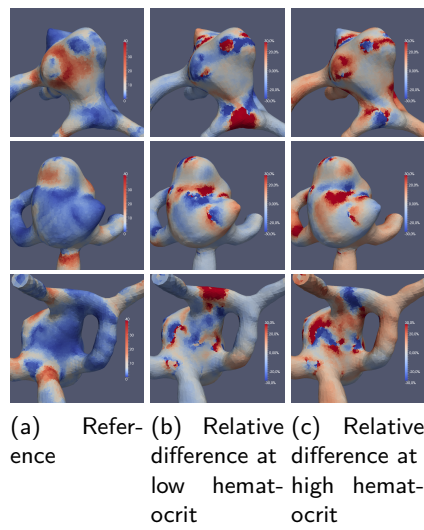
Wall shear stress

Looking at the average WSS through a full cycle in Figure 6.23, we see a clear tendency of higher WSS at higher hematocrits. This is expected, as the higher hematocrit will predict higher viscosity, and the WSS is expected to be proportional to the viscosity (see Section 5). However, Figure 6.24 and Figure 6.25 clearly shows that locally, the effect may work opposite. As noted on velocity, the higher hematocrit seems to produce a lower velocity gradient, and thus might be responsible for lowering the WSS. This seems to be true in areas of low WSS, but it does not seem to weigh up for the effects of heightened viscosity.

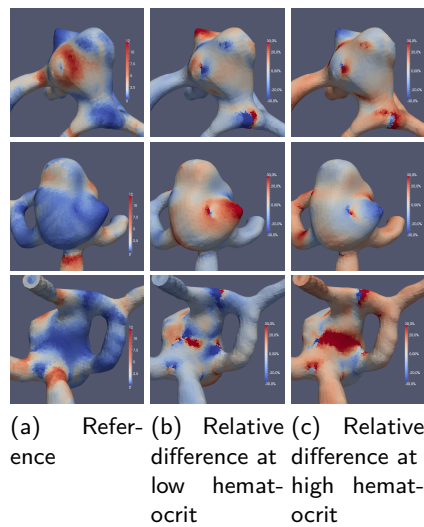At the areas of high WSS, an increased hematocrit level predicts heightened WSS.
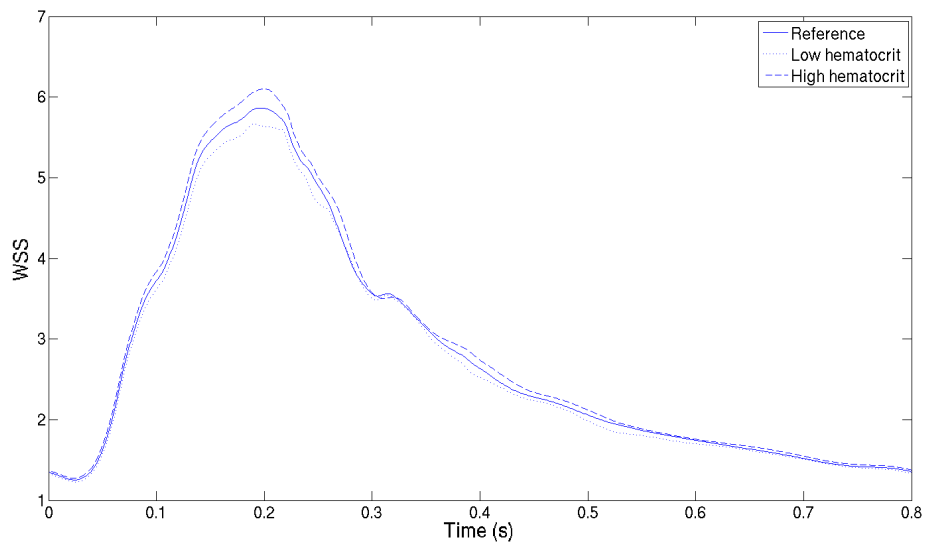
Figure 6.23: Average WSS measured over the aneurysm for the different hematocrit levels.

(a)   Refer- (b) Relative (c) Relative
ence   difference at   difference at
               low   hemat-   high hemat-
               ocrit   ocrit

Figure 6.24: The figure shows velocity magnitude at systole. (a) Shows the reference velocity mag-
nitude for the model using 40% hematocrit as reference. Top row of (b)-(c) shows
the velocity magnitude for the low and high hematocrit models, and the bottom row
of (b)-(c) shows the relative magnitude difference for the low and high hematocrits,
compared against the medium hematocrit model. The velocity is ranging from 0 to
1000 mm/s, and the relative magnitude is ranging from -30% to 30%.

(a)   Refer- (b) Relative (c) Relative
ence   difference at   difference at
               low   hemat-   high hemat-
               ocrit   ocrit

Figure 6.25: The figure shows velocity magnitude at diastole. (a) Shows the reference velocity
magnitude for the model using 40% hematocrit as reference. Top row of (b)-(c) shows
the velocity magnitude for the low and high hematocrit models, and the bottom row
of (b)-(c) shows the relative magnitude difference for the low and high hematocrits,
compared against the medium hematocrit model. The velocity is ranging from 0 to
500 mm/s, and the relative magnitude is ranging from -30% to 30%.

7. Quantitative analysis

In this section four different effects are studied:

- Neglecting the shear thinning (non-Newtonian) behaviour of blood (Section 7.4).

- A heightened hematocrit level (Section 7.5).

- Increased flow velocity/flow rate at the inflow (Section 7.6).

- Using resistance boundary conditions, instead of constant pressure boundary conditions at the outlets(Section 7.7).

A quantitative approach has been made, as 12 different saccular MCA aneurysms of different types, sizes and shapes were used as test geometries. Methods for measuring the effects have been derived, and the effects have been studied through one cardiac cycle on each aneurysm.

The studies done on increased hematocrit and increased inflow is motivated from a medical point of view, and although the main focus here will be the errors this might introduce into the numerical simulations, the results can also be regarded as preliminary research on how these changes might affect aneurysm development.

All the results can be found in Section 7.4-Section 7.7. The perhaps most interesting results are those produced by increasing the inlet flow (Section 7.6), as these were the results that most consistently and significantly contradicted expectations by showing a much larger increase in WSS than expected from known theory.

7.1. The aneurysms and meshes

All the 12 aneurysms studied have been segmented from CT images using VMTK, where a user-contributed script (*vmtkmeshclipcenterlines*) was used to automate isolation of the aneurysm dome, such that no error would arise from human discretion.[9]. An example of this isolation is seen in Figure 7.1. The classification of the aneurysms, along with some key properties associated with them, can be seen in Table 7.1.

All meshes have between 800,000 to 1,200,000 cells, so that the computational time required would be similar. However, there are large density differences, as the average cell volumes range from 1.27e-4 to 1.30e-3mm^3.

7.2. Simulations

The simulations were run using the same inlet flow profile as depicted in Figure 6.5. Using data from [43], we find that a suitable value for a time-averaged peak velocity is 695 mm/s, and the inlet flow profile is scaled accordingly.

---

[9]This work had already been done in connection with a previous research project at Simula Research Laboratory.

Patient	Classification	Volume (mm^3)	Surface area (mm^2)	Ruptured
M1	Sidewall	212	193	No
M2	Bifurcation	33	61	No
M3	Bifurcation	54	80	Yes
M5	Sidewall	43	63	No
M8	Bifurcation	13	26	No
M9	Bifurcation	18	31	Yes
M11	Bifurcation	509	357	Yes
M12	Bifurcation	278	208	Yes
M15	Sidewall	33	59	No
M16	Bifurcation	265	236	Yes
M18	Bifurcation	24	50	No
M20	Bifurcation	251	197	No

Table 7.1: The table describes some important properties of the aneurysms studied.

(a) Full mesh   **(b)** Segmented aneurysm
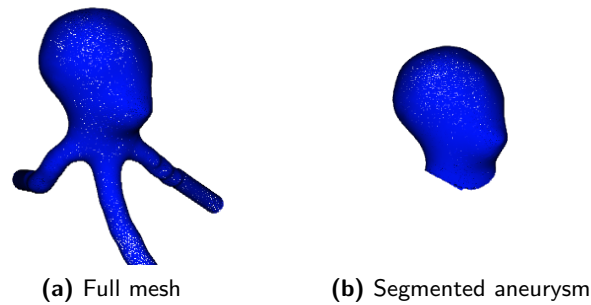
Figure 7.1: The figure shows how an aneurysm has been segmented from the mesh.

In all comparisons two different sets of solutions has been produced, one for each set of input parameters. One is used as a reference to measure the effect in question.

As the simulations were run, the velocity field, pressure field and the WSS were saved in XML-files at at every 5th time step, producing vector and scalar fields for further analysis over one cardiac cycle. A script was later used to load these solutions, and calculate functionals for the differences in WSS and velocity at each time step. The source code of this script can be found in Appendix A. The isolation of the aneurysms is taken advantage of to mainly study the solution and the WSS inside the aneurysm dome and on the aneurysm surface.

7.3. Methods for measuring difference between solutions

To measure the differences between the two sets of solutions produced, two metrics and norms are introduced. One for the differences in the velocity field within the aneurysm

dome, and one for the differences in WSS over the aneurysm surface. Both metrics require that the WSS and the velocity field in one of the two sets of solutions is used as a reference, denoted as $(\tau_R, \mathbf{u}_R)$. The WSS and velocity field of the solution to be compared against the reference is denoted as $(\tau_C, \mathbf{u}_C)$.

Velocity metric

The velocity metric is defined as

$$S_{\mathbf{u}_R}(\mathbf{u}_C) = 1 - \beta e^{-a} - \gamma e^{-m}$$

where

$$a = \frac{1}{\pi} \cos^{-1}\left(\frac{\mathbf{u}_R \cdot \mathbf{u}_C}{\|\mathbf{u}_R\|\|\mathbf{u}_C\|} \right) \qquad\qquad 0 \leq a \leq 1$$

$$m = \frac{\|\mathbf{u}_R - \mathbf{u}_C\|}{\|\mathbf{u}_R\|} \qquad\qquad 0 \leq m$$

and $\beta + \gamma = 1$ are weights, which here was chosen as $\beta = \gamma = \frac{1}{2}$. It is easy to see that a large value corresponds to a large difference in the velocity field. This metric was calculated at each vertex inside the aneurysm.

To get a measure for the difference in velocity field over the entire aneurysm, we introduce the mean L^1-norm of this metric,

$$\begin{aligned} D_{\mathbf{u}} &= \frac{1}{V_A} \int_{\Omega_A} S_{\mathbf{u}_R}(\mathbf{u}_C) \, d\mathbf{x}, \\ &= \frac{1}{V_A} \|S_{\mathbf{u}_R}(\mathbf{u}_C)\|_{L^1(\Omega_A)} \end{aligned}$$

where Ω_A is the aneurysm dome with corresponding volume V_A.

Wall shear stress metric

The WSS metric was calculated at each facet as

$$T_{\tau_R}(\tau_C) = 1 - e^{-t}$$

where

$$t = \left| \frac{\tau_R - \tau_C}{\tau_R} \right| \qquad\qquad 0 \leq t$$

7.3 Methods for measuring difference between solutions

To get a measure for the difference in WSS, the mean L^1-norm over the aneurysm surface of this metric is applied,

$$D_\tau = \frac{1}{S_A} \int_{\partial\Omega_A} T_{\tau_N}(\tau_C) \, dS,$$

$$= \frac{1}{S_A} \|T_{\tau_R}(\tau_C)\|_{L^1\left(\partial\Omega_A\right)}$$

where $\partial\Omega_A$ is the aneurysm surface with corresponding surface area S_A.

Notes on the metrics

The two metrics introduced can be compared to a simpler approach, such as e.g. $\hat{S} = \beta a + \gamma m$ or $\hat{T} = t$. Where these measures are unbounded, the metrics S and T are bounded. The WSS metric is bounded by $0 \leq T_{\tau_R}(\tau_C) < 1$ and the velocity metric used is bounded by $0 \leq S_{\mathbf{u}_R}(\mathbf{u}_C) < 1 - \frac{1}{2}e^{-1} \approx 0.816$. It is also clear that $S < \hat{S}$ and $T < \hat{T}$. This is a desired effect, as any differences in areas of low velocity or low WSS might be given unjustifiable focus using the metrics $\hat{S}$ or $\hat{T}$ (small absolute difference, large relative difference).

Other

In the following sections, some simple statistics has been used to compare solutions, and determine the significance of the results. This has been done by using the MatLab function *ttest*[10]. All assumptions made and explanation of values used is found in the documentation of this function. A significance level of 5% has been used.

In some sections the pressure drop over the aneurysm has been reported. This has been calculated from "probe locations" one vessel diameter before and after the aneurysm, within the artery. The average pressure drop is calculated as such:

$$\Delta p = \frac{1}{N} \sum_{j=1}^{N}(p_i - p_j),$$

where p_i is the pressure at inlet probe location, p_j is the pressure at outlet j and N is the number of outlets after the aneurysm.

---

[10]http://www.mathworks.com/help/toolbox/stats/ttest.html

7.4. Non-Newtonian effects

Only two different viscosity models have been considered, as an attempt has been made to quantify the non-Newtonian effects. A solution using a Newtonian model has been used as reference, and is compared to a solutions using the Casson model. To isolate the non-Newtonian differences, it was important that the Casson model converged to the Newtonian viscosity as the shear rate approached infinity. That way, the only differences occurring were in fact due to the shear thinning and yield stress terms of the Casson model. Hence, by using the Casson model as described in Section 3.3, we set the Newtonian viscosity to be $\mu = \mu_\infty$, where μ_∞ is the asymptotic viscosity of the Casson model ($\approx 0.00322 Pa\,s$).

Results

Figure 7.3 shows the resulting difference measures, plotted against time. First of all, note that $D_\mathbf{u}$ and D_τ are quite different; in some of the aneurysms, they are as large as 0.2 for both D_τ and $D_\mathbf{u}$. This suggests that the non-Newtonian effects might be of some significance in some of the aneurysms, most notably in aneurysms M3, M5, M12 and M16. However, in most of the aneurysms, the values of the norms are below or around 0.1, indicating much less significant effects of the non-Newtonian properties.

The differences seem to be at their highest at diastole and/or early systole, where the corresponding inlet velocity is at its lowest. The differences seem to be at their lowest at peak systole. This might indicate that lower velocities increase the importance of the non-Newtonian effects, but it is worth noting that not all the simulations show the same effect, e.g., aneurysm M2 seem to have the highest differences at end systole.

Seeing as the differences between the aneurysms is quite large, efforts have been made to find out why. A natural starting point was to measure the average shear rate in all aneurysms, as this is how the viscosity models differ. Comparisons of the average shear rate to time-averaged $D_\mathbf{u}$ and D_τ, can be seen in Figure 7.2, and they are quite compelling. As expected from the shear thinning nature of the Casson model, the differences are clearly largest in the aneurysms which predicts the lower shear rate. There are some deviations to this tendency, e.g. the small differences measured at a shear rate of approximately $140s^{-1}$. It seems as though the non-Newtonian effects are negligible when the average shear rate exceeds $500s^{-1}$, but since our simulations only predicted three geometries with these kinds of shear rates, there is not enough data to draw any certain conclusions. However, there is a very strong indication that the non-Newtonian effects decrease as the shear rate increase.

To get a better understanding of what causes an increase of the non-Newtonian effects, four aneurysms have been selected for a more thorough analysis, displayed in Figure 7.4-Figure 7.7. Aneurysms M3 and M12, which by far show the most significant differences of the four, have very few similarities. In aneurysm M3, there is very little blood flow within the aneurysm dome, and the velocities are close to zero. In M12, there is much blood flow, and the flow pattern seems much more disturbed than in M18 and M20. When studying the metric S we

see a clear tendency of much higher values where the blood flow is practically non-existent. The same applies for the WSS. In the areas of very low WSS, the non-Newtonian effects are by far more significant than in areas of high WSS.

The actual values of the predicted WSS is summarized in in Table 7.2. The maximum deviation in average WSS is -4.42%, and the maximum deviation in maximum WSS is -2.83%. In Section 5 we showed a linear relation between viscosity and WSS. Thus, we would expect the Casson model to predict higher WSS, due to a higher viscosity. This might be the case for average WSS, but this is not measured with statistical significance (mean=0.41%, P=0.22). However, the Casson model actually predicts *lower* maximum WSS (mean=-2.31%, P=0.0091) than the Newtonian model.

No correlation is seen between the type of the aneurysm and the non-Newtonian effects. There are sidewall aneurysms that both predict quite high non-Newtonian significance (M5, M15), and another one that predicts very low non-Newtonian significance (M1). The same applies for the bifurcation aneurysms.

The size of the aneurysm does not seem to be of importance.
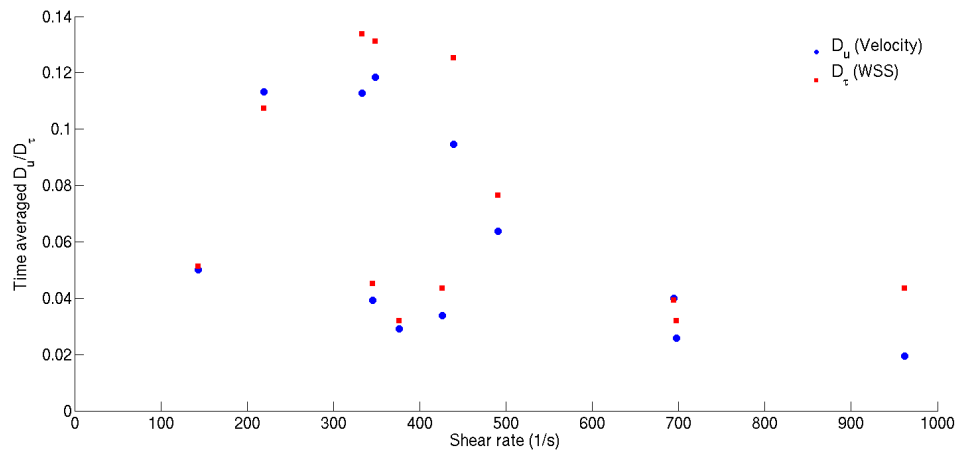
Figure 7.2: The figure shows the time averaged $D_{\mathbf{u}}$ and D_{τ} plotted against time- and space-averaged shear rate, measured from the Newtonian results over a full cycle.
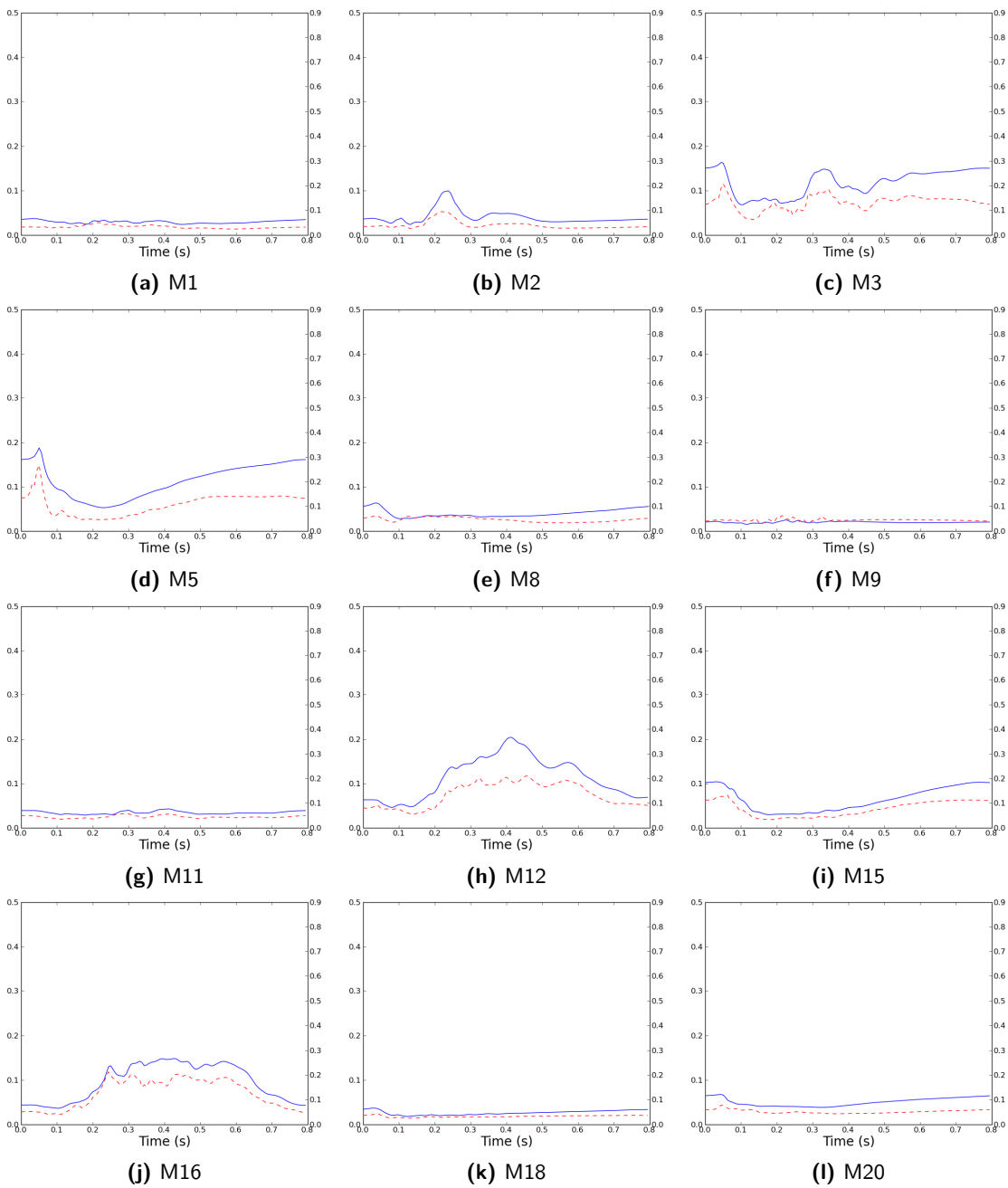
Figure 7.3: The figure shows the non-Newtonian effects for each aneurysm over a full cycle. The blue line is $D_{\mathbf{u}}$ and is scaled by the left axis. The dotted, red line is D_τ and is scaled by the right axis. All plots use the same axes.

Aneurysm		WSS Functional	Newtonian	Casson	Rel. diff
M1		Time averaged	3.638	3.665	0.75%
		Max	44.176	43.912	-0.60%
M2		Time averaged	5.168	5.195	0.54%
		Max	46.341	44.378	-4.24%
M3		Time averaged	3.149	3.145	-0.11%
		Max	76.393	70.94	-7.14%
M5		Time averaged	1.491	1.457	-2.29%
		Max	36.211	35.0	-3.34%
M8		Time averaged	2.341	2.378	1.57%
		Max	57.434	56.729	-1.23%
M9		Time averaged	6.844	6.998	2.24%
		Max	66.566	62.173	-6.60%
M11		Time averaged	4.751	4.756	0.10%
		Max	40.065	40.16	0.24%
M12		Time averaged	2.947	2.968	0.73%
		Max	44.927	44.41	-1.15%
M15		Time averaged	2.903	2.902	-0.05%
		Max	101.288	101.23	-0.06%
M16		Time averaged	3.419	3.449	0.87%
		Max	46.888	46.689	-0.42%
M18		Time averaged	4.517	4.516	-0.03%
		Max	44.383	44.131	-0.57%
M20		Time averaged	1.289	1.297	0.61%
		Max	59.512	57.994	-2.55%

Table 7.2: The table shows WSS functionals for the different aneurysms, and compare the Newtonian functionals to the Casson functionals. The *Time averaged* is the space averaged WSS over the full cycle, and the *Max* value is the extreme value for the full cycle. All WSS values are in Pa. The difference reported is in %, relative to the Newtonian result.
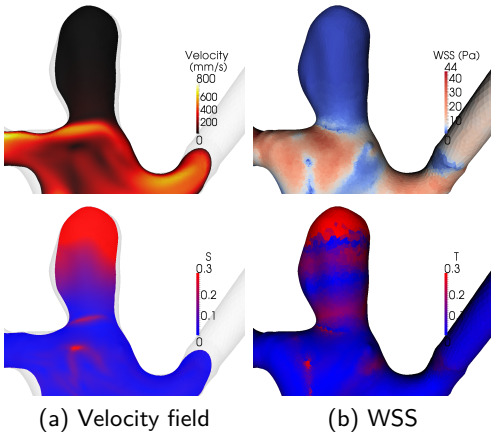
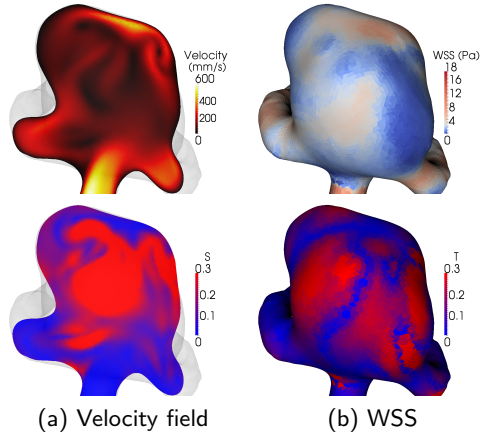(a) Velocity field   (b) WSS

Figure 7.4: M3

(a) Velocity field   (b) WSS

Figure 7.5: M12

(a) Velocity field   (b) WSS

Figure 7.6: M18

(a) Velocity field   (b) WSS
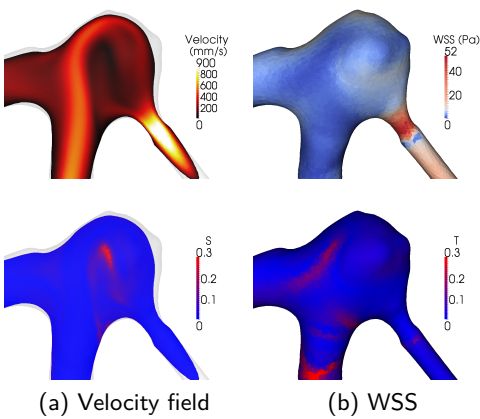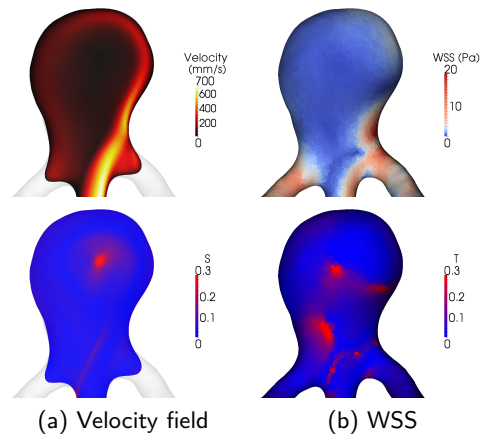
Figure 7.7: M20

Figure 7.4-**Figure 7.7:** The figures show selected aneurysms for comparison, taken at the time step where M3 and M12 showed the largest non-Newtonian effects. M18 and M20 were both evaluated at early diastole. Columns (a) show the velocity, taken at a slice through the aneurysm. The top row shows the velocity magnitude, and the bottom row shows the metric S. Columns (b) show the WSS from a selected angle. The top row shows the magnitude of the WSS, and the bottom row show the metric T.

7.5. Effects of increased hematocrit

The average age of aneurysm rupture is 52 years[45]. As already mentioned in Section 2.1, women are more likely than men to suffer from aneurysm rupture. In their late forties and early fifties, women go through menopause. The average age has been reported to be 51.7[46]. This is very close to the average rupture age, and thus triggers the hypothesis of correlation.

During menopause, the physiology of women change. Of particular interest here, is the change in blood properties. In the Tromsø Study[47] performed at the University Hospital of North Norway, hematocrit levels were studied in a large group of patients. This revealed an increased hematocrit level of about two percentage points from pre-menopausal to post-menopausal age for women, a change not seen in men in the same age span (see Figure 7.8). This figure also shows a higher hematocrit level for smokers than non-smokers, in particular for women. Smoking is a known factor in aneurysm rupture risk, and perhaps this can be explained or partially explained by the increase in hematocrit.
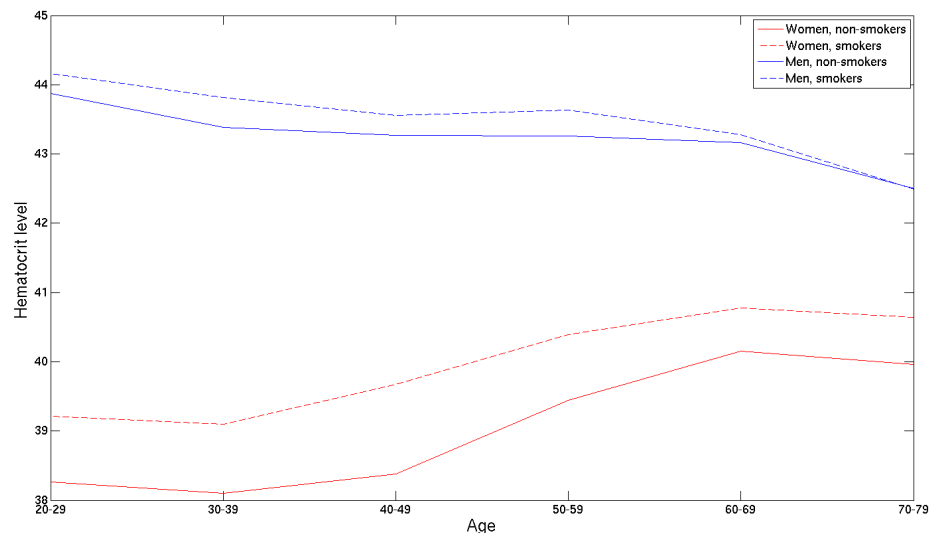
Figure 7.8: The figure shows the average hematocrit for women and men, categorized as smokers and non-smokers, stratified in 10-year intervals.

The analysis has been done using the Casson model, which incorporates the hematocrit level. Simulations were run with hematocrit levels of 38% and 40%. The effects of increased hematocrit were then measured, using the solution produced with hematocrit level of 38% as reference.

Results

As seen in Figure 7.9, the resulting metrics show almost identical results to what was found when assessing the non-Newtonian effects (Figure 7.3). Thus, discussion regarding the metrics can be found in Section 7.4.

Looking closer at the differences in pressure drop in Table 7.3, the increase in hematocrit seem to indicate a slightly higher pressure drop, but the data does not show this tendency with significance (mean=1.57%, P=0.087). The deviations may be quite significant in some aneurysms, as the change in pressure drop are as high as 7.1% in aneurysm M8.

The WSS changes is summarized in Table 7.4. The only significant change found was in average WSS. This increases with increased hematocrit (mean=1.56%, P=0.026). No significant changes is found regarding maximum WSS (mean=-1.14%, P=0.54). The changes vary quite a bit, as the changes in maximum WSS range from -12.7% (M3) to 5.7% (M1). The average WSS range from -3.2% (M5) to 5.2% (M9), however only one aneurysm indicates a decrease in average WSS when increasing the hematocrit level.

Aneurysm	Low hematocrit	High hematocrit	Rel. diff
M1	68.7	72.1	4.9%
M2	108.6	112.9	4.0%
M3	217.7	213.8	-1.8%
M5	51.4	53.1	3.3%
M8	29.7	31.8	7.1%
M9	155.1	159.5	2.8%
M11	135.3	134.2	-0.8%
M12	131.8	133.7	1.4%
M15	22.6	22.1	-2.2%
M16	117.4	118.5	0.9%
M18	65.3	64.8	-0.8%
M20	66.6	66.6	0.0%

Table 7.3: The table shows the average pressure drop in Pa over the aneurysms.
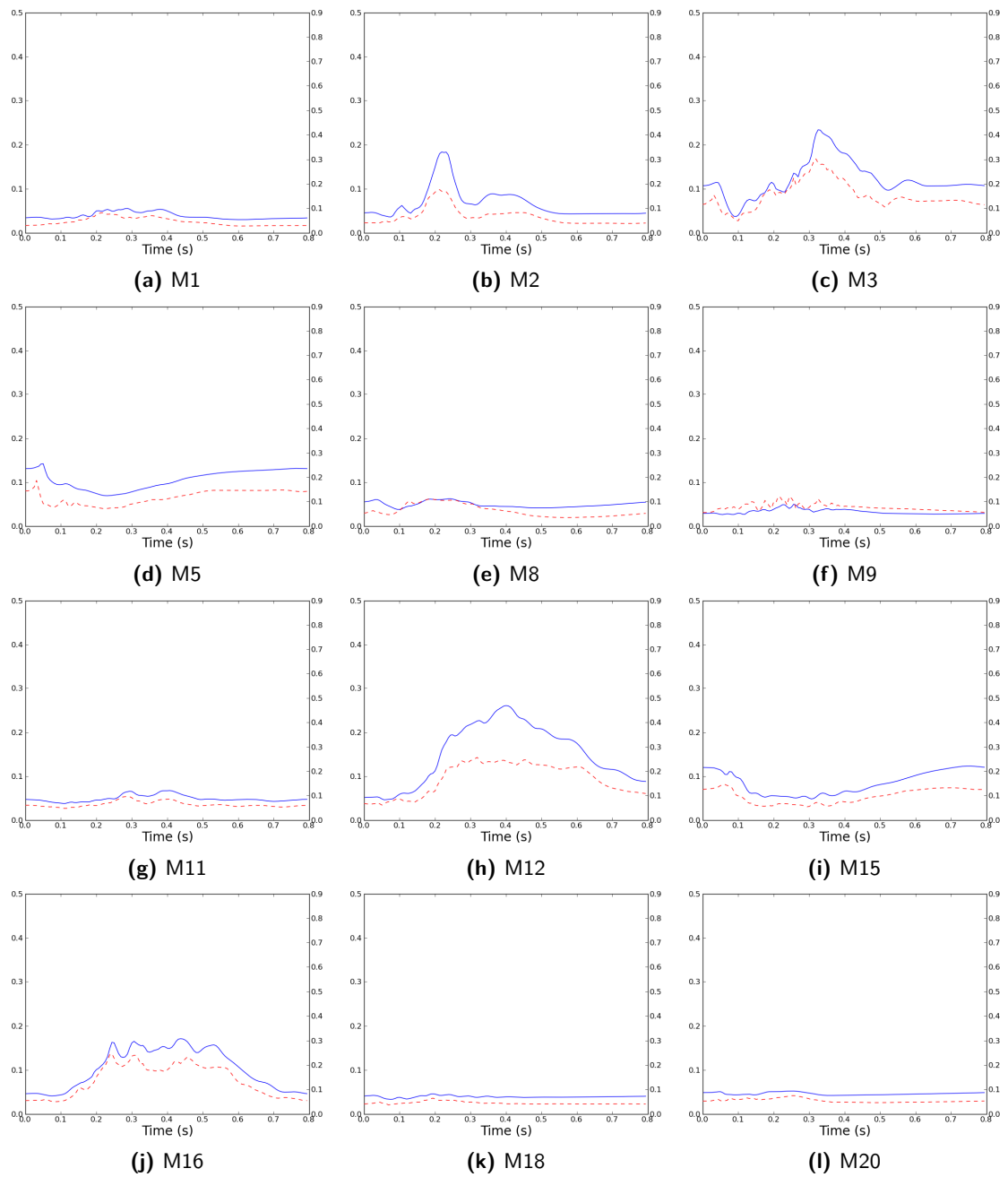
Figure 7.9: The figure shows the effects of increased hematocrit level for each aneurysm over a full cycle. The blue line is $D_{\mathbf{u}}$ and is scaled by the left axis. The dotted, red line is D_τ and is scaled by the right axis. All plots use the same axes.

Aneurysm		WSS Functional	Low hematocrit	High hematocrit	Rel. diff
M1		Time averaged	3.576	3.665	2.49%
		Max	41.553	43.912	5.68%
M2		Time averaged	5.079	5.195	2.29%
		Max	47.643	44.378	-6.85%
M3		Time averaged	3.145	3.145	0.01%
		Max	81.298	70.940	-12.74%
M5		Time averaged	1.505	1.457	-3.19%
		Max	36.551	35.000	-4.24%
M8		Time averaged	2.283	2.378	4.14%
		Max	55.039	56.729	3.07%
M9		Time averaged	6.650	6.998	5.24%
		Max	69.555	62.173	-10.61%
M11		Time averaged	4.699	4.756	1.20%
		Max	38.571	40.160	4.12%
M12		Time averaged	2.917	2.968	1.76%
		Max	44.657	44.410	-0.55%
M15		Time averaged	2.879	2.902	0.81%
		Max	96.926	101.230	4.44%
M16		Time averaged	3.380	3.449	2.04%
		Max	45.165	46.689	3.37%
M18		Time averaged	4.488	4.516	0.63%
		Max	43.396	44.131	1.69%
M20		Time averaged	1.281	1.297	1.24%
		Max	58.586	57.994	-1.01%

Table 7.4: The table shows WSS functionals for the different aneurysms, and compare the low hematocrit functionals to the high hematocrit functionals. The *Time averaged* is the space averaged WSS over the full cycle, and the *Max* value is the extreme value for the full cycle. All WSS values are in Pa. The difference reported is in %, relative to the low hematocrit result.

7.6. Inlet boundary condition

In this section the differences made by increasing the inlet velocity and thus the inlet flux has been studied. The previously used time-averaged peak velocity of 695mm/s has been compared to a lower time-averaged peak velocity of 535mm/s. The low velocity solution was used as reference, as we thus also could get an idea of how increased blood flow affect the WSS in the aneurysm. This corresponds to an increase in inlet flux by 33%.

Results

Figure 7.10 shows the measures for the different inlet velocities. These are all very high. $D_{\mathbf{u}}$ almost never drops below 0.2 for any aneurysm, and it reaches as high as 0.45. D_τ is rarely below 0.5, and reach as high as 0.85. From this it seems as though the effects on WSS is far greater than the increase in blood flow would suggest.

It seems as though the effects are largest at early systole and through diastole, where the blood flow is at the lowest. This is as expected, as this is where the relative increase of blood flow is largest. However, this effect is not visible in all aneurysms, as M12 and M16 most notably show large difference at systole as well. This might indicate a change in flow patterns.

Aneurysm	Low inflow	High inflow	Rel. diff
M1	45.0	73.5	63.3%
M2	73.7	114.9	55.9%
M3	105.3	212.6	101.9%
M5	32.9	53.7	63.2%
M8	21.6	32.4	50.0%
M9	101.2	161.4	59.5%
M11	77.6	134.7	73.6%
M12	80.7	135.4	67.8%
M15	12.1	21.9	81.0%
M16	71.0	119.9	68.9%
M18	37.6	64.7	72.1%
M20	38.3	66.8	74.4%

Table 7.5: The table shows the average pressure drop over the aneurysms.

In Table 7.5 we see the pressure drop over each aneurysm. As with WSS, the pressure drop also show an increase much greater than the increase applied to the inlet, with an average increase of 69.3%.

In Table 7.6 we see a massive increase in both average and maximum WSS. The time averaged WSS is as much as doubled (M5) with the high inflow condition, and is on average 72.8% greater than for the low inflow condition. The maximum WSS show a very similar

behaviour, with an average increase of 73.6%. Both the average and maximum WSS show a significant higher increase than the 33% increase in inlet flow, with a P-value of less than 10^{-4}.

The differences in WSS is also clearly more prominent in the aneurysm part of the mesh, as seen in Table 7.7. This might indicate that the aneurysm itself is particularly sensitive to changes in inflow, compared to the surrounding arteries.

(a) M1

(b) M2

(c) M3

(d) M5

(e) M8

(f) M9

(g) M11

(h) M12

(i) M15

(j) M16

(k) M18

(l) M20
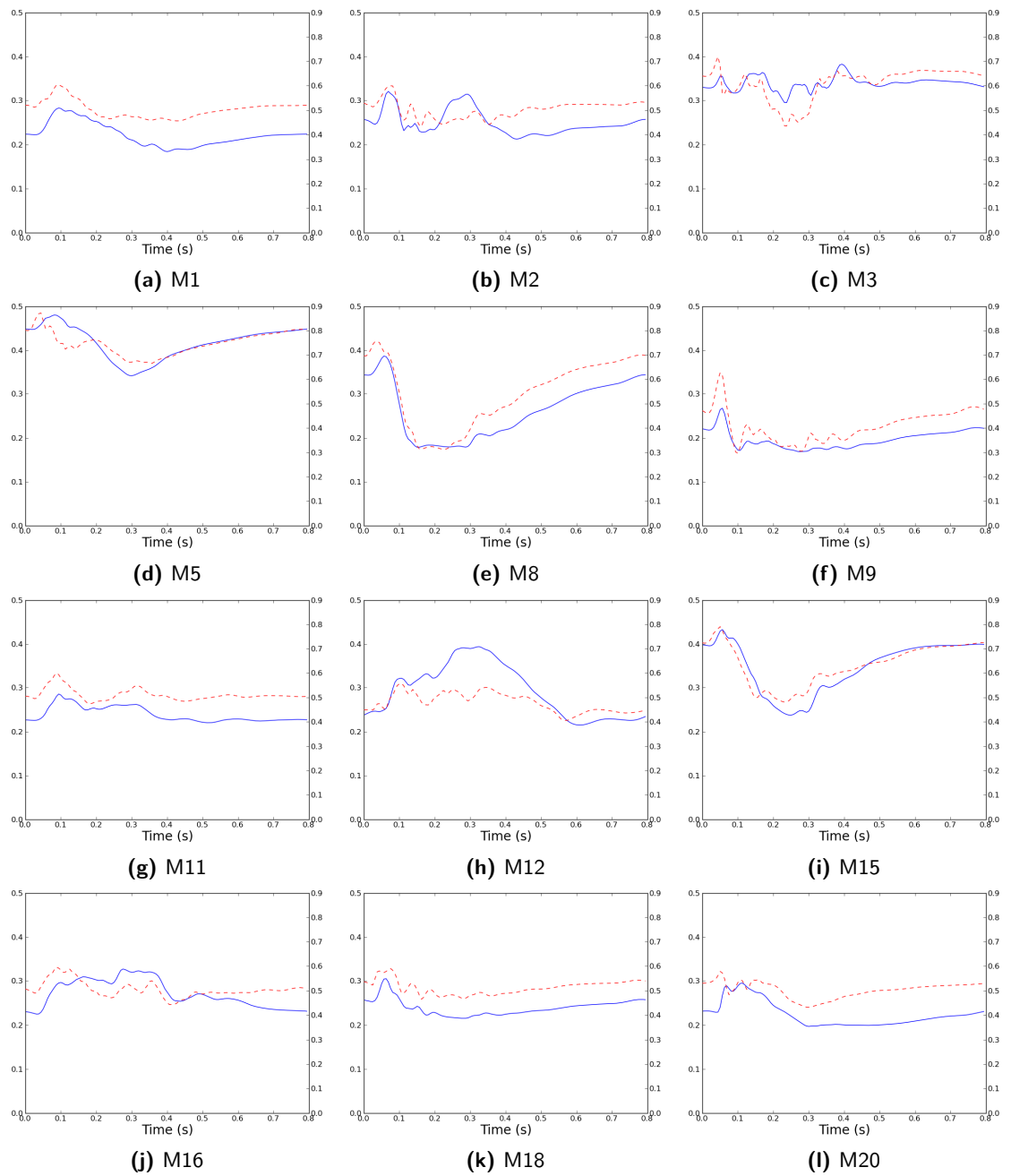
Figure 7.10: The figure shows the effects of increased flow for each aneurysm over a full cycle. The blue line is $D_\mathbf{u}$ and is scaled by the left axis. The dotted, red line is D_τ and is scaled by the right axis. All plots use the same axes.

Aneurysm		WSS Functional	Low inflow	High inflow	Rel. diff
M1		Time averaged	2.192	3.709	69.20%
		Max	25.362	44.505	75.48%
M2		Time averaged	3.150	5.261	67.05%
		Max	24.784	43.948	77.32%
M3		Time averaged	1.896	3.163	66.81%
		Max	36.439	70.672	93.95%
M5		Time averaged	0.686	1.450	111.31%
		Max	17.335	35.160	102.83%
M8		Time averaged	1.427	2.415	69.16%
		Max	30.761	58.681	90.76%
M9		Time averaged	4.762	7.139	49.92%
		Max	40.334	61.652	52.85%
M11		Time averaged	2.833	4.799	69.38%
		Max	26.622	41.396	55.50%
M12		Time averaged	1.754	2.987	70.33%
		Max	29.163	45.060	54.51%
M15		Time averaged	1.622	2.919	79.97%
		Max	65.009	105.215	61.85%
M16		Time averaged	2.088	3.477	66.49%
		Max	29.743	48.027	61.47%
M18		Time averaged	2.557	4.537	77.45%
		Max	24.935	44.916	80.14%
M20		Time averaged	0.738	1.299	76.12%
		Max	31.802	58.539	84.08%

Table 7.6: The table shows WSS functionals for the different aneurysms, and compare the low inflow functionals to the high inflow functionals. The *Time averaged* is the space averaged WSS over the full cycle, and the *Max* value is the extreme value for the full cycle. All WSS values are in Pa. The difference reported is in %, relative to the low inflow result.

Section	Average change in WSS	95% confidence interval
Full mesh	46.1%	[44.0%, 48.3%]
Aneurysm	72.8%	[63.7%, 81.8%]
Mesh without aneurysm	42.3%	[40.3%, 44.4%]

Table 7.7: The table shows the difference in average WSS in selected parts of the mesh.

7.7. Effects of a resistance boundary conditions

There are several different outlet pressure boundary conditions which may be reasonable to assume when modelling blood flow in and around the Circle of Willis. The simplest boundary condition is the zero pressure boundary condition which is enforced by setting $p = 0$ at all outlets, and this is commonly used. In this thesis we have used a resistance boundary condition which takes into account the downstream vasculature. In this section the tools introduced at the start of this chapter, has been used to study the differences these two boundary conditions make on the blood flow and WSS in our 12 aneurysms.

The solutions produced with a resistance boundary conditions is used as reference.

Results

In Figure 7.11, we see the corresponding difference metrics of the different outlet boundary conditions. Note that there are large differences between the aneurysms. Also note how some of the aneurysms display larger differences through parts of the cycle. This might come as a result of a changed flow pattern, whereas a constant difference metric might indicate a strengthening or weakening of the original flow pattern.

Figure 7.12-Figure 7.13 show some of the changes in flow patterns. Note that the patterns might change quite significantly within the aneurysm and toward the outlets, because of the change in pressure gradient induced by the outlet boundary conditions. Also note that the flow around the inlets show almost no difference at all, indicating that the outlet conditions does not have any effect on the flow prior to the bifurcation. Looking again at Figure 7.11, we see that this explains why the difference metrics in aneurysm M15 are so small, as this aneurysm lies prior to the bifurcation.

In Table 7.8 the changes in outlet flux are summarized. It would be natural to predict that large changes in flux would correspond to large changes in the difference metrics. Comparing the flux changes to the difference metrics in Figure 7.11, we see that this is clearly not the case. E.g., M3 show hardly any difference in outlet flux, but quite significant changes are recorded by the difference metrics. Also, seeing as the difference metrics of e.g. aneurysms M5 and M8 are very similar, one might expect similar changes in outlet flux, however, this is not the case, as M5 predict a small change, whereas M8 predict a much higher change in outlet flux.

Looking at the changes in WSS given in Table 7.9, we see that the changes in some aneurysms are very large, whereas in others they are clearly negligible.

In general it seems very hard to predict which aneurysms will be more affected by the choices made regarding outlet boundary conditions, but as seen, the changes might be highly significant.
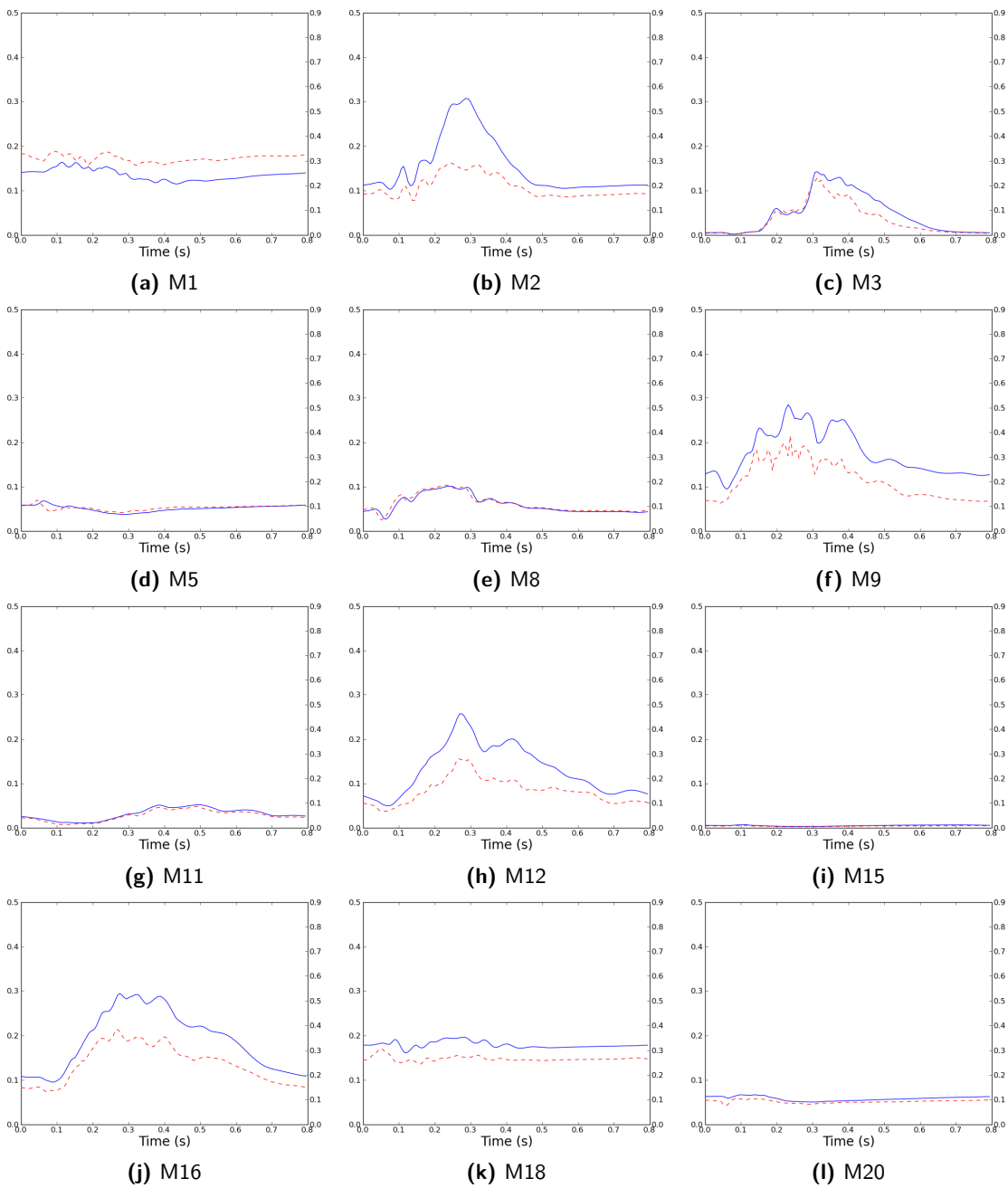
Figure 7.11: The figure shows the effects of a different outlet boundary condition for each aneurysm over a full cycle. The blue line is $D_{\mathbf{u}}$ and is scaled by the left axis. The dotted, red line is D_τ and is scaled by the right axis. All plots use the same axes.

(a) Resistance **(b)** Zero pressure **(c)** S
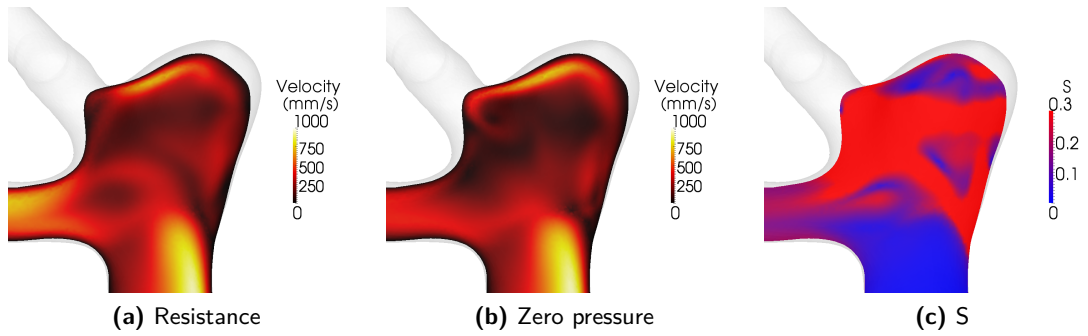
Figure 7.12: The figure shows the changed flow pattern in aneurysm M2, with the corresponding difference metric S.

(a) Resistance **(b)** Zero pressure **(c)** S
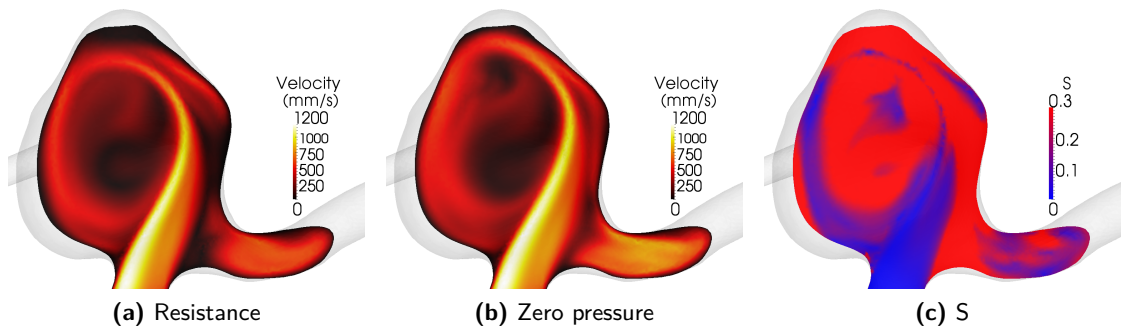
Figure 7.13: The figure shows the changed flow pattern in aneurysm M16, with the corresponding difference metric S.

Aneurysm	Traction free BC	Resistance BC	Rel. diff.
M1	1336	1658	24.1%
M2	482	737	53.0%
M3	760	763	0.4%
M5	558	594	6.5%
M8	219	320	46.2%
M9	597	883	47.8%
M11	1729	1793	-3.7%
M12	727	803	10.5%
M15	606	697	15.0%
M16	688	978	42.1%
M18	76	252	231.6%
M20	365	403	10.4%

Table 7.8: The table shows the time averaged flux at a chosen outlet in each aneurysm, and the difference from the traction free to the resistance boundary condition. The outlet chosen is the outlet which showed the largest relative difference.

Aneurysm		WSS Functional	Resistance	Traction free	Rel. diff
M1		Time averaged	3.709	5.050	36.15%
		Max	44.505	63.466	42.60%
M2		Time averaged	5.261	5.544	5.38%
		Max	43.948	46.011	4.69%
M3		Time averaged	3.163	3.186	0.73%
		Max	70.672	70.781	0.16%
M5		Time averaged	1.450	1.351	-6.83%
		Max	35.160	33.897	-3.59%
M8		Time averaged	2.415	2.714	12.42%
		Max	58.681	61.671	5.10%
M9		Time averaged	7.139	7.247	1.51%
		Max	61.652	96.742	56.92%
M11		Time averaged	4.799	4.751	-1.01%
		Max	41.396	41.405	0.02%
M12		Time averaged	2.987	2.997	0.35%
		Max	45.060	45.997	2.08%
M15		Time averaged	2.919	2.924	0.16%
		Max	105.215	105.320	0.10%
M16		Time averaged	3.477	3.428	-1.40%
		Max	48.027	49.594	3.26%
M18		Time averaged	4.537	4.621	1.86%
		Max	44.916	43.097	-4.05%
M20		Time averaged	1.299	1.244	-4.27%
		Max	58.539	61.801	5.57%

Table 7.9: The table shows WSS functionals for the different aneurysms, and compare the resistance BC functionals to the zero pressure BC functionals. The *Time averaged* is the space averaged WSS over the full cycle, and the *Max* value is the extreme value for the full cycle. All WSS values are in Pa. The difference reported is in %, relative to the resistance BC result.

8. Discussion

8.1. Qualitative results

After analyzing the geometric effects, non-Newtonian effects and effects of different hematocrit levels presented in Section 6, it is difficult to say anything about which of these effects is most significant. However, what is clear is that both the blood flow and the WSS distribution are qualitatively the same; the flow does not seem to change direction, and the areas of high WSS does not seem to change. That being said, the local changes might be relatively large, especially in areas of low flow velocity and low WSS.

The geometric effects were difficult to compare relative to each other, as the meshes used in the simulations are different, and thus also the vector and scalar fields produced. However, determining the threshold value when segmenting the CT images does not seem to be crucial in achieving correct blood flow. However, a recently published article by Gamburato et al.[48] showed that the surface smoothing of the vascular geometry is significant for the WSS distribution.

As for the non-Newtonian effects we saw the clearest differences where the velocity and WSS was low. There were large differences between the models, and the power-law model stood out in particular, and predicted significantly lower WSS. Because not all of these viscosity models had the same asymptotic value, they are not suitable for a direct comparison with the Newtonian model to determine the non-Newtonian effects such as i.e. shear thinning.

The different hematocrit levels showed a somewhat strange behaviour when looking at the velocity magnitude. The effects of changed hematocrit are very difficult to conclude anything from after these investigations, but this subject was studied more closely later.

Overall, we saw that the differences seemed more important at systole than at diastole, with a possible exception for the non-Newtonian models, where the diastolic differences in WSS seemed to be just as significant as at systole.

8.2. Quantitative results

The results of Section 7 is quite compelling, and a summary can be seen in Table 8.1. How boundary conditions are set, seem to have very large effect on the forces created by blood flow in cerebral aneurysms. Parameters related to blood viscosity seem to be of much less significance, however, several interesting conclusions may still be drawn from the results.

Neglecting the non-Newtonian effects will cause a significant change in the predicted maximum WSS. One should definitely be aware of this when simulating this kind of blood flow, but it does not defend the approximately 25% additional computational time required. Gamburato et al.[48] concluded that non-Newtonian effects should not be neglected when using a patient specific geometry, however, the results of this thesis shows that the non-Newtonian effects may well be neglected if the patient specific boundary data is unknown or uncertain.

Description	Effects on	Mean (%)	SD (%)	P-value
Taking non-Newtonian effects into account.	Average WSS	0.411	1.097	0.221
	Max WSS	**-2.305**	2.526	**0.009**
	Pressure drop	0.353	1.484	0.428
Increasing hematocrit from 38% to 40%.	Average WSS	**1.555**	2.100	**0.026**
	Max WSS	-1.136	6.166	0.537
	Pressure drop	1.567	2.889	0.087
Increasing peak inflow velocity by 30% from 535mm/s to 695mm/s.	Average WSS	**72.766**	14.280	0.000
	Max WSS	**73.565**	16.617	0.000
	Pressure drop	**69.300**	13.402	0.000
Using zero-pressure BCs on outlets as opposed to resistance BCs.	Average WSS	3.754	**11.262**	-
	Max WSS	9.405	**19.345**	-
	Pressure drop	-1.206	**25.665**	-

Table 8.1: Summary of the results of Section 7. The *Mean* column describes the average change made on the 12 aneurysms, from the reference solution to the solution with the respective parameter change. The *SD* column describes the sample standard deviation. The *P-value* describes the probability that the hypothesis of zero change is true. Values of special interest highlighted in bold. No P-values are calculated for the different outlet boundary conditions, as no valid assumption on distribution can be made.

The effects of the physiologically motivated increase in hematocrit level was also deemed significant, with an average increase in WSS of 1.555%. As it is natural to assume that this effect is, in general, not dependent of the choice of boundary conditions, this result is definitely interesting. Without taking into account other potential physiological effects of menopause, the increase in hematocrit might help explain help explain why the average age of cerebral aneurysm rupture are so similar to the average age of menopause.

Perhaps the most surprising results were produced by increasing the inflow flux. As the calculations in Section 5 show, one would expect a similar proportional relationship between the inflow flux and the WSS. However, this is clearly not the case. The effects on the forces, both WSS and pressure, are much greater than what a proportional relationship would suggest.

The different outlet boundary conditions in some cases yielded surprisingly similar solutions. However, in others, there were major differences. Which aneurysms showed the greatest differences were difficult to predict, thus, this is a very possible source of error when simulating blood flow in cerebral aneurysms.

8.3. Conclusion and further research

It seems clear from the results produced in this thesis, that from the different tests that has been done, the applied boundary conditions show the greatest sensitivity to change when the WSS is used as a measure. This would suggest than in future studies, more in vivo, patient specific measurements would be of great use, to more accurately prescribe realistic boundary conditions.

However, compelling results may still be made without these data. E.g., the hematocrit increase studied in this thesis showed an increase in average WSS. This result should by no mean be disregarded simply because we do not have patient specific data available, as it is natural to assume that this effect would be present with any set of realistic boundary conditions.

In this thesis, some sets of parameters have been tested for sensitivity, but there are many other possible sources of inaccuracies. E.g. the assumption of rigid walls or the effects of changing heart rate. These are both, to some extent, unanswered questions.

Results from this thesis might also in itself inspire further work. The results of increasing the inflow flux for example, showed an unexpected relationship between flux and WSS. An hypothesis to consider is that this relationship is perhaps quadratic; the inflow flux was increased by a factor of 1.333, whereas the average WSS increased by a factor of $1.728(=(1.315)^2)$. To test this hypothesis the inflow flux would have to be varied, and the corresponding WSS changes noted.

Much work is thus still left to be done in this area of research, but with time our understanding improves, measurement instruments are getting more accurate and computational power increase. We may thus have good hopes that simulations of this kind can one day be used in medicine as a diagnostic tool, and as an instrument to determine necessary treatment.

8.3 Conclusion and further research

A. Source code

A.1. Cylinder mesh

```python
def create_mesh(self):
    #Create mesh
    N = self.N
    nz = int(N*self.z1)

    if(nz%2 is not 0):
        nz = nz+1

    mesh = Mesh()
    editor = MeshEditor()
    editor.open(mesh,3,3)

    #Create vertices
    no_of_vertices = nz*((N+1)*(N+1) + N**2)
    editor.init_vertices(no_of_vertices)
    no_of_cells = (nz-1)*12*N**2
    editor.init_cells(no_of_cells)

    vertex=0
    for iz in range(nz):
        z = self.z0+iz*(self.z1-self.z0)/(nz-1)
        for iy in range(N+1):
            y = -self.r+2.0*self.r*iy/N
            for ix in range(N+1):
                x = -self.r+2.0*self.r*ix/N
                if(abs(x) < DOLFIN_EPS and abs(y) < DOLFIN_EPS):
                    x_trans = x
                    y_trans = y
                else:
                    xx = 0.5*(x+y)
                    yy = 0.5*(-x+y)
                    x_trans = xx*(abs(xx)+abs(yy))/sqrt(xx*xx+yy*yy)
                    y_trans = yy*(abs(xx)+abs(yy))/sqrt(xx*xx+yy*yy)

                editor.add_vertex(vertex,x_trans,y_trans,z)
                vertex = vertex+1
        for iy in range(N):
            y = -self.r + 2.0*self.r*(iy+0.5)/N
            for ix in range(N):
                x = -self.r + 2.0*self.r*(ix+0.5)/N

                if(abs(x) < DOLFIN_EPS and abs(y) < DOLFIN_EPS):
                    x_trans = x
                    y_trans = y
                else:
                    xx = 0.5*(x+y)
                    yy = 0.5*(-x+y)
                    x_trans = xx*(abs(xx)+abs(yy))/sqrt(xx*xx+yy*yy)
                    y_trans = yy*(abs(xx)+abs(yy))/sqrt(xx*xx+yy*yy)

                editor.add_vertex(vertex,x_trans,y_trans,z)
                vertex = vertex+1

    #Create cells
    cell = 0

    n = no_of_vertices/nz
    for iz in range(nz-1):
        for iy in range(N):
            for ix in range(N):
                v0 = iz*(N+1)**2 +iy*(N+1)+ix+iz*N**2
                v1 = v0+1
                v2 = v0+(N+1)
                v3 = v1+(N+1)
                v4 = v0 + (N+1)**2-iy

                v5 = v0+n
                v6 = v5+1
                v7 = v5+(N+1)
                v8 = v6+(N+1)
                v9 = v5+(N+1)**2-iy

                editor.add_cell(cell,v4,v9,v5,v6)
                cell = cell+1
                editor.add_cell(cell,v4,v0,v1,v6)
                cell = cell+1
```

```
                    editor.add_cell(cell,v4,v0,v5,v6)
                    cell = cell+1

                    editor.add_cell(cell,v4,v6,v8,v9)
                    cell = cell+1
                    editor.add_cell(cell,v1,v3,v4,v8)
                    cell = cell+1
                    editor.add_cell(cell,v1,v4,v6,v8)
                    cell = cell+1

                    editor.add_cell(cell,v4,v9,v7,v8)
                    cell = cell+1
                    editor.add_cell(cell,v4,v2,v3,v8)
                    cell = cell+1
                    editor.add_cell(cell,v4,v2,v7,v8)
                    cell = cell+1

                    editor.add_cell(cell,v4,v5,v7,v9)
                    cell = cell+1
                    editor.add_cell(cell,v0,v2,v4,v7)
                    cell = cell+1
                    editor.add_cell(cell,v0,v4,v5,v7)
                    cell = cell+1

        editor.close()

        return mesh
```

A.2. Inlet flow profile

```
class InflowExpression(Expression):
    def __init__(self):
        Expression.__init__(self)

    def eval_data(self, value, data):
        value[0] = -data.normal().x()
        value[1] = -data.normal().y()
        value[2] = -data.normal().z()

    def dim(self):
        return 3
```

```
    def initiate_inflow(self):
        MCAtime = numpy.array([    0.,    27.,    42.,    58.,    69.,    88.,   110.,   130.,
                    136.,   168.,   201.,   254.,   274.,   290.,   312.,   325.,
                    347.,   365.,   402.,   425.,   440.,   491.,   546.,   618.,
                    703.,   758.,   828.,   897.,  1002.])/(75/60.0)/1000


        scale = 695
        #Create interpolated mean velocity in time
        y1 = numpy.array([ 390.        ,  398.76132931,  512.65861027,  642.32628399,
                    710.66465257,  770.24169184,  779.00302115,  817.55287009,
                    877.12990937,  941.96374622,  970.        ,  961.2386707 ,
                    910.42296073,  870.12084592,  843.83685801,  794.7734139 ,
                    694.89425982,  714.16918429,  682.62839879,  644.07854985,
                    647.58308157,  589.75830816,  559.96978852,  516.16314199,
                    486.37462236,  474.10876133,  456.58610272,  432.05438066,  390.        ])/574.211239628*scale


        # Define the spline for enough heart beats
        k = int((self.T+1)/max(MCAtime))
        n = len(y1)

        y = numpy.zeros((n-1)*k)
        time = numpy.zeros((n-1)*k)

        for i in range(k):
                y[i*(n-1):(i+1)*(n-1)] = y1[:-1]
                time[i*(n-1):(i+1)*(n-1)] = MCAtime[:-1]+i*max(MCAtime)

        spline = intpol.UnivariateSpline(time,y,k=3, s=0)


        # Create parabolic profile
        V = VectorFunctionSpace(self.mesh, "CG",1)
        A = assemble(Constant(1)*ds(2), mesh=self.mesh, exterior_facet_domains=self.bc_markers)
        p = project(Expression(("x[0]", "x[1]", "x[2]")), V)
```

```
        # Find radius and center of inlet
        r = sqrt(A/DOLFIN_PI)
        center = []
        for i in range(3):
                center.append(assemble(p[i]*ds(2), mesh=self.mesh, exterior_facet_domains=self.bc_markers)/A)

        # Create a function of inward pointing vectors at the inlet
        m = TrialFunction(V)
        inflow_profile = Function(V)
        assemble(inner(InflowExpression(),m)*ds(2), tensor=inflow_profile.vector(), exterior_facet_domains=self.bc_markers)

        # Scale the inflow function to a parabolic profile with max magnitude of 1
        N = len(inflow_profile.vector())
        for i in range(N/3):
                n1 = inflow_profile.vector()[i]
                n2 = inflow_profile.vector()[N/3+i]
                n3 = inflow_profile.vector()[2*N/3+i]
                norm = sqrt(n1**2+n2**2+n3**2)

                if(norm > 0):
                        x = p.vector()[i]
                        y = p.vector()[N/3+i]
                        z = p.vector()[2*N/3+i]
                        d_c = sqrt((center[0]-x)**2+(center[1]-y)**2+(center[2]-z)**2)/r

                        inflow_profile.vector()[i]       = n1/norm*(1-d_c**2)
                        inflow_profile.vector()[N/3+i]   = n2/norm*(1-d_c**2)
                        inflow_profile.vector()[2*N/3+i] = n3/norm*(1-d_c**2)


        return inflow_profile, spline
```

A.3. Script to load solutions

The following Python script is used to load the solutions of Section 7, and calculate the metrics of Section 7.3 along with some other quantities. This is quite computationally demanding, and thus, we have used the *Instant* module to do the most demanding calculation in inline C++. This dramatically improves performance. All the results are written to file, so that they can easily be accessed.

```
#!/usr/bin/env python

import sys, time
from dolfin import *
import subprocess
import os
import instant
import matplotlib
matplotlib.use('Agg') # Must be before importing matplotlib.pyplot or pylab!
import matplotlib.pyplot as plt
from numpy import arccos, mean, max, min, nonzero, arange, array

cpp_vel_diff = """
void dabla(dolfin::GenericVector& S, dolfin::GenericVector& un, dolfin::GenericVector& uc,
                        double beta, double gamma, dolfin::GenericVector& aneurysm) {
        uint N = S.size();

        double m = 0;
        double a = 0;
        double s = 0;

        double un_mag = 0;
        double uc_mag = 0;
        double unuc = 0;
        double diff_mag = 0;

        for (unsigned int i=0; i < N; i++) {
                un_mag = sqrt(pow(un[i],2)+pow(un[N+i],2)+pow(un[2*N+i],2));
                uc_mag = sqrt(pow(uc[i],2)+pow(uc[N+i],2)+pow(uc[2*N+i],2));
                unuc = uc[i]*un[i]+uc[N+i]*un[N+i]+uc[2*N+i]*un[2*N+i];
                diff_mag = sqrt(pow(un[i]-uc[i],2)+pow(un[N+i]-uc[N+i],2)+pow(un[2*N+i]-uc[2*N+i],2));

                m = diff_mag/(un_mag+DOLFIN_EPS);
```

```
                a = 0;
                if(un_mag > 0) {
                        double val = unuc/(un_mag*uc_mag+DOLFIN_EPS);
                        if(val > 1.0) {
                                a = 0.0;
                        } else if(val < 0.0) {
                                a = 1.0;
                        } else {
                                a = 1/DOLFIN_PI*acos(val);
                        }
                }

                s = (1-beta*exp(-m)-gamma*exp(-a))*aneurysm[i];
                S.setitem(i,s);
        }
}
"""

cpp_tau_diff = """
void dabla(dolfin::GenericVector& T, dolfin::GenericVector& taun,
                        dolfin::GenericVector& tauc, dolfin::GenericVector& aneurysm) {
        uint N = T.size();

        for (unsigned int i=0; i < N; i++) {
                double t = 0;
                double diff = taun[i]-tauc[i];
                if(diff < 0) {
                        t = -diff/(taun[i]+DOLFIN_EPS);
                } else {
                        t = diff/(taun[i]+DOLFIN_EPS);
                }
                t = t*aneurysm[i];

                t = 1-exp(-t);
                T.setitem(i,t);
        }
}
"""

cpp_vect_magnitude = """
void dabla(dolfin::GenericVector& un_mag, dolfin::GenericVector& un, dolfin::GenericVector& aneurysm) {
    uint N = un_mag.size();
    double mag = 0;

        for (unsigned int i=0; i < N; i++) {
                mag = sqrt(pow(un[i],2)+pow(un[N+i],2)+pow(un[2*N+i],2));
                mag = mag*aneurysm[i];
                un_mag.setitem(i,mag);
        }
}
"""

include_dirs, flags, libs, libdirs = instant.header_and_libs_from_pkgconfig("dolfin")
headers= ["dolfin.h"]
vel_diff = instant.inline(cpp_vel_diff, system_headers=headers,
                                include_dirs=include_dirs, libraries = libs, library_dirs = libdirs)
tau_diff = instant.inline(cpp_tau_diff, system_headers=headers,
                                include_dirs=include_dirs, libraries = libs, library_dirs = libdirs)
vect_magnitude = instant.inline(cpp_vect_magnitude, system_headers=headers,
                                    include_dirs=include_dirs, libraries = libs, library_dirs = libdirs)

#Return the shear rate
def gamma(u):
        return pow(0.5*inner(grad(u)+transpose(grad(u)), grad(u)+transpose(grad(u))), 0.5)

#Calculate the pressure drop over the aneurysm
def pdrop(p, patient):
        if patient == "m1":
                x0 = array((93.70 ,80.68 ,69.97 ))
                x1 = array((100.10 ,80.59 ,65.58 ))
                x2 = x1

        if patient == "m2":
                x0 = array((119.22 ,105.01 ,82.23 ))
                x1 = array((124.58 ,108.93 ,79.27 ))
                x2 = array((121.42 ,107.06 ,78.46 ))

        if patient == "m3":
                x0 = array((104.54 ,98.74 , 64.17 ))
                x1 = array((104.85 ,103.44 ,64.56  ))
                x2 = array((107.72 ,96.74 , 59.59 ))

        if patient == "m5":
                x0 = array((105.46,102.47, 58.69 ))
```

```python
            x1 = array((108.97, 105.29, 60.48  ))
            x2 = x1

        if patient == "m8":
            x0 = array((62.33 ,125.52 ,69.15  ))
            x1 = array((59.68, 128.57, 70.63 ))
            x2 = x1

        if patient == "m9":
            x0 = array((120.09, 133.15, 65.04 ))
            x1 = array((125.66, 132.06, 65.85  ))
            x2 = array((121.56, 137.90, 64.47 ))

        if patient == "m11":
            x0 = array((103.25, 95.62, 47.98 ))
            x1 = array((105.42, 96.58, 39.32 ))
            x2 = array((105.27, 99.94, 52.73 ))

        if patient == "m12":
            x0 = array((111.26, 111.71, 57.20 ))
            x1 = array((112.15, 116.90, 58.78 ))
            x2 = array((115.41, 110.99, 52.80 ))

        if patient == "m15":
            x0 = array((101.82, 81.56, 62.05 ))
            x1 = array((97.10, 79.78, 63.80 ))
            x2 = x1

        if patient == "m16":
            x0 = array((78.09, 130.46, 59.64 ))
            x1 = array((74.98, 127.69, 57.28 ))
            x2 = array((69.83, 134.24, 63.42 ))

        if patient == "m18":
            x0 = array((133.34, 124.65, 37.80))
            x1 = array((136.47, 123.87, 37.59))
            x2 = array((133.57, 127.64, 37.73 ))

        if patient == "m20":
            x0 = array((32.09, 95.58, 107.17))
            x1 = array((32.00, 92.28, 106.77))
            x2 = array((29.65, 96.51, 109.07 ))

        values0 = array((0.0, 0.0, 0.0))
        values1 = array((0.0, 0.0, 0.0))
        values2 = array((0.0, 0.0, 0.0))
        p.eval(values0, x0)
        p.eval(values1, x1)
        p.eval(values2, x2)

        pressuredrop = values0[0] - (values1[0] + values2[0])/2.0

        return pressuredrop

def main(args):
        logging(False)
        dolfin.parameters["allow_extrapolation"] = True

        #Load command line arguments
        if(len(args)>=3):
                data_set = args[0]
                refinement = args[1]
                patients = args[2:]
        elif(len(args)==2):
                data_set = args[0]
                refinement = args[1]
                patients = ["m1", "m2", "m3", "m5", "m8", "m9", "m11", "m12", "m15", "m16", "m18", "m20"]
        else:
                print "Wrong number of arguments!"

        for patient in patients:
                #Set the correct folders where the saved solutions are found, based on the data set
                if data_set == "nonnewtonian":
                        reference_folder = "nonnewtonian/results/quantitative/"+str(refinement)+"/"+str(patient)+"/constant/"
                        comparison_folder = "nonnewtonian/results/quantitative/"+str(refinement)+"/"+str(patient)+"/casson/"
                        save_folder = "readoutput/nonnewtonian/"
                elif data_set == "hematocrit":
                        reference_folder = "hematocrit/results/quantitative/"+str(refinement)+"/"+str(patient)+"/casson/0.38/"
                        comparison_folder = "nonnewtonian/results/quantitative/"+str(refinement)+"/"+str(patient)+"/casson/"
                        save_folder = "readoutput/hematocrit/"
                elif data_set == "pressurebc":
                        reference_folder = "reference/results/quantitative/"+str(refinement)+"/"+str(patient)+"/constant/"
                        comparison_folder = "pressurebc/results/quantitative/"+str(refinement)+"/"+str(patient)+"/constant/"
                        save_folder = "readoutput/pressurebc/"
```

```python
elif data_set == "inletbc":
        reference_folder = "inletbc/results/quantitative/"+str(refinement)+"/"+str(patient)+"/constant/"
        comparison_folder = "reference/results/quantitative/"+str(refinement)+"/"+str(patient)+"/constant/"
        save_folder = "readoutput/inletbc/"
else:
        print "Did not recognize argument: ", data_set
        exit(1)

#Load mesh and aneurysm mesh
mesh = Mesh("reference/data/Aneurysms/meshes/"+str(refinement)+"/"+str(patient)+"_"+str(refinement)+".xml.gz")
aneurysm_mesh = Mesh("reference/data/Aneurysms/meshes/aneurysms/aneurysm_"+str(patient)+".xml.gz")

#Define function spaces
DG = FunctionSpace(mesh, "DG", 0)
V = VectorFunctionSpace(mesh, "CG", 1)
Q = FunctionSpace(mesh, "CG", 1)
q = TestFunction(Q)

#Create a DG0 function representing the aneurysm
aneurysm_DG = Function(DG)
av_DG = aneurysm_DG.vector()

#Compute the aneurysm part of the mesh (aneurysm=1, not aneurysm=0)
print 'Computing aneurysm surface/volume vector'
for c in cells(mesh):
        p = Point(c.midpoint().x(), c.midpoint().y(), c.midpoint().z())
        if aneurysm_mesh.any_intersected_entity(p)>0:
                av_DG[c.index()] = 1.0

#Create a CG1 function representing the aneurysm
aneurysm_Q = project(aneurysm_DG, Q)
av_Q = aneurysm_Q.vector()

for i in range(len(av_Q)):
        if(av_Q[i] < 0.5):
                av_Q[i] = 0.0
        else:
                av_Q[i] = 1.0

#Compute volume and surface area of the aneurysm
volume = assemble(aneurysm_Q*dx, mesh=mesh)
surface = assemble(aneurysm_DG*ds, mesh=mesh)
print "Aneurysm volume: ", volume
print "Aneurysm surface area: ", surface

#Define lists
time = []
Du = []
Dt = []
um = []
tmr = []
tmc = []
gammas = []
pdrops_r = []
pdrops_c = []
min_wss_reference = []
min_wss_comparison = []
max_wss_reference = []
max_wss_comparison = []

#Iterate through the saved solutions, one for each fifth timestep
for i in range(1,129):
        print "Timestep: %4d Time: %4f" %(i, 0.8*(i-1)/128)
        time.append((0.8*(i-1)/128))

        #Load velocity vectors
        u_r = Function(V, reference_folder+"u_"+str(i)+".xml")
        u_c = Function(V, comparison_folder+"u_"+str(i)+".xml")

        #Calculate the velocity differences
        S = Function(Q)
        beta = 0.5
        g = 0.5
        vel_diff(S.vector(), u_r.vector(), u_c.vector(), beta, g,  aneurysm_Q.vector())
        s = 1/volume*assemble(S*dx, mesh=mesh)
        Du.append(s)

        #Calculate (reference) velocity magnitude
        u_mag = Function(Q)
        vect_magnitude(u_mag.vector(), u_r.vector(), aneurysm_Q.vector())
        um.append(1/volume*assemble(u_mag*dx, mesh=mesh))


        #Load WSS
```

```
        tau_r = Function(DG, reference_folder+"tau_"+str(i)+".xml")
        tau_c = Function(DG, comparison_folder+"tau_"+str(i)+".xml")

        #Calculate WSS differences
        T = Function(DG)
        tau_diff(T.vector(), tau_r.vector(), tau_c.vector(), aneurysm_DG.vector())
        t = 1/surface*assemble(T*ds, mesh=mesh)
        Dt.append(t)

        #Calculate average WSS
        tau_av_n = 1/surface*assemble(tau_r*aneurysm_DG*ds, mesh=mesh)
        tau_av_c = 1/surface*assemble(tau_c*aneurysm_DG*ds, mesh=mesh)
        tmr.append(tau_av_n)
        tmc.append(tau_av_c)

        #Calculate max and min WSS
        tau_r_aneurysm = tau_r.vector().array()*aneurysm_DG.vector().array()
        tau_c_aneurysm = tau_c.vector().array()*aneurysm_DG.vector().array()
        max_wss_reference.append(max(tau_r_aneurysm))
        max_wss_comparison.append(max(tau_c_aneurysm))
        min_wss_reference.append(min(tau_r_aneurysm[tau_r_aneurysm>0]))
        min_wss_comparison.append(min(tau_c_aneurysm[tau_c_aneurysm>0]))

        #Calculate the pressure drops
        p_r = Function(Q, reference_folder+"p_"+str(i)+".xml")
        p_c = Function(Q, comparison_folder+"p_"+str(i)+".xml")
        pdrops_r.append(pdrop(p_r, patient))
        pdrops_c.append(pdrop(p_c, patient))

        #Calculate (reference) shear rate
        gamma_n = gamma(u_r)
        gamma_n = 1/volume*assemble(gamma_n*aneurysm_DG*dx, mesh=mesh)
        gammas.append(gamma_n)

#Plot Du and Dt
ax1 = plt.subplot(111)
plt.plot(time, Du, 'b-')
plt.xlabel("Time (s)",fontsize=20)
ax2 = plt.twinx()
plt.plot(time, Dt, 'r--')
filename = save_folder+str(patient)+".png"
plt.savefig(filename, bbox_inches='tight', pad_inches=0.1)
plt.clf()
plt.cla()

#Print all functionals
f = open(save_folder+str(patient)+".out", 'w')
f.write("***********ERRORS***********\n")
f.write("Average D_u: %s \n" %str(sum(Du)/len(Du)))
f.write("Average D_u: %4f \n" %(sum(Du)/len(Du)))
f.write("Average D_t: %s \n" %str(sum(Dt)/len(Dt)))
f.write("***********FLOW***********\n")
f.write("Average velocity: %s \n" %str(sum(um)/len(um)))
f.write("Average shear rate: %s \n" %str(sum(gammas)/len(gammas)))
f.write("***********WSS***********\n")
av_r = sum(tmr)/len(tmr)
av_c = sum(tmc)/len(tmc)
f.write("Average WSS Newtonian: %s \n" %str(av_r))
f.write("Average WSS Casson: %s \n" %str(av_c))
f.write("Average WSS diff: %s \n" %str((av_c-av_r)/av_r*100))
f.write("-----------\n")
f.write("Max WSS Newtonian: %s \n" %str(max(max_wss_reference)))
f.write("Max WSS Casson: %s \n" %str(max(max_wss_comparison)))
f.write("Max WSS diff: %s \n" %str((max(max_wss_comparison)-max(max_wss_reference))/max(max_wss_reference)*100))
f.write("-----------\n")
f.write("Min WSS Newtonian: %s \n" %str(min(min_wss_reference)))
f.write("Min WSS Casson: %s \n" %str(min(min_wss_comparison)))
f.write("Min WSS diff: %s \n" %str((min(min_wss_comparison)-min(min_wss_reference))/min(min_wss_reference)*100))
f.write("***********PRESSURE***********\n")
avp_r = sum(pdrops_r)/len(pdrops_r)
avp_c = sum(pdrops_c)/len(pdrops_c)
f.write("Average pdrop Newtonian: %s \n" %str(avp_r))
f.write("Average pdrop Casson: %s \n" %str(avp_c))
f.write("Average pdrop diff: %s \n" %str((avp_c-avp_r)/avp_r*100))
f.write("-----------\n")
f.write("Max pdrop Newtonian: %s \n" %str(max(pdrops_r)))
f.write("Max pdrop Casson: %s \n" %str(max(pdrops_c)))
f.write("Max pdrop diff: %s \n" %str((max(pdrops_c)-max(pdrops_r))/max(pdrops_r)*100))
f.write("***********NORMS***********\n")
f.write(str(Du))
f.write("\n-----------\n")
f.write(str(Dt))
f.close()
```

A.3 Script to load solutions

```python
if __name__ == "__main__":
    sys.exit(main(sys.argv[1:]))
```

References

[1] FEniCS Project. `http://www.fenicsproject.org/`.

[2] Hans Petter Langtangen. A FEniCS Tutorial. `www.fenicsproject.org/_downloads/fenics-tutorial-python.pdf`.

[3] J. van Gijn and G.J. Rinkel. Subarachnoid haemorrhage: diagnosis, causes and management. *Brain*, 124:249–78, 2001.

[4] S.A. Mayer, K.T. Kreiter, D. Copeland, et al. Global and domain-specific cognitive impairment and outcome after subarachnoid hemorrhage. *Neurology*, 59:1750–8, Dec 2002.

[5] F.H.H. Linn, G.J. Rinkel, A. Algra, and J. van Gijn. Incidence of subarachnoid hemorrhage: role of region, year, and rate of computed tomography: a meta-analysis. *Stroke*, 27:625–9, 1996.

[6] Haakon M. Lindekleiv, Kristian Valen-Sendstad, Michael K. Morgan, Kent-Andre Mardal, et al. Sex differences in intracranial arterial bifurcations. *Gender Medicine*, 7(2):149–155, 2010.

[7] Sylvia Kamath. Observations on the length and diameter of vessels forming the circle of willis. *Journal of Anatomy*, 133:419–423, 1981.

[8] Jonathan L. Brisman, Joon K. Song, and David W. Newell. Cerebral aneurysms. *The New England Journal of Medicine*, 355:928–39, 2006.

[9] Adnan Qureshi et al. Risk factors for multiple intracranial aneurysms. *Neurosurgery*, 43:22–26, July 1998.

[10] K.N. Kayembe, M. Sasahara, and F. Hazama. Cerebral aneurysms and variations in the circle of willis. *Stroke*, 15:846–850, 1984.

[11] David O Wiebers. Unruptured intracranial aneurysms: natural history, clinical outcome, and risks of surgical and endovascular treatment. *The Lancet*, 362(9378):103–110, 2003.

[12] George N. Foutrakis, Howard Yonas, and Robert J. Sclabassi. Saccular aneurysm formation in curved and bifurcating arteries. *AJNR Am J Neuroradiol*, 20(8):1309–1317, August 1999.

[13] YC Fung. *Biomechanics : mechanical properties of living tissues*. Springer Verlag, 1993.

[14] Donald L. Fry. Acute vascular endothelial changes associated with increased blood velocity gradients. *Circulation Research*, 22:165–197, 1968.

[15] Yiannis S. Chatzizisis, Ahmet Umit Coskun, Michael Jonas, Elazer R. Edelman, Charles L. Feldman, and Peter H. Stone. Role of endothelial shear stress in the natural

history of coronary atherosclerosis and vascular remodeling: Molecular, cellular, and vascular behavior. *Journal of the American College of Cardiology*, 49:2379–2393, 2007.

[16] Y.B. Roos, M.G. Dijkgraaf, K.W. Albrecht, et al. Direct costs of modern treatment of aneurysmal subarachnoid hemorrhage in the first year after diagnosis. *Stroke*, 33:1595–9, Jun 2002.

[17] Johnston SC, Wilson CB, Halbach VV, Higashida RT, Dowd CF, , et al. Endovascular and surgical treatment of unruptured cerebral aneurysms: comparison of risks. *Annals of Neurology*, 48(1):11–19, 2000.

[18] Ralph Taggart and Cecie Starr. *Biology: The Unity and Diversity of Life*. Wadsworth, 1989.

[19] Y.I. Cho and K.R. Kensey. Effects of the non-newtonian viscosity of blood on flows in a diseased arterial vessel. part 1: Steady flows. *Biorheology*, 28(3-4):241–262, 1991.

[20] Pijush K. Kundu and Ira M. Cohen. *Fluid Mechanics*. Academic Press, 4 edition, 2008.

[21] F.J.H. Gijsen, F.N. van de Vosse, and J.D. Janssen. The influence of the non-newtonian properties of blood on the flow in large arteries: steady flow in a carotid bifurcation model. *Journal of Biomechanics*, 32, 1999.

[22] Michael Griebel, Thomas Dornseifer, and Tilman Neunhoeffer. *Numerical Simulations in Fluid Dynamics: A Practical Introduction*. Society for Industrial and Applied Mathematics, 1997.

[23] Arne Jørgen Arnesen. Comparison of Finite Element Methods for the Navier-Stokes Equations. Master's thesis, University of Oslo, 2010. `http://www.duo.uio.no/sok/work.html?WORKID=104087&fid=59876`.

[24] Martin Sandve-Alnæs. Finite Element simulations of blood flow in the Circle of Willis. Master's thesis, University of Oslo, 2006. `http://folk.uio.no/martinal/master/master_thesis_martin_alnes.pdf`.

[25] Anne M. Robertson, Adélia Sequeira, and Robert G. Owense. *Cardiovascular Mathematics. Modeling and simulation of the circulatory system*, chapter 6. Springer Verlag Italia, 2009.

[26] F.J. Walburn and D.J. Schneck. A constitutive equation for whole human blood. *Biorheology*, 13(3):201–210, Jun 1976.

[27] Carolyn Fisher and Jenn Stroud Rossmann. Effect of non-newtonian behavior on hemodynamics of cerebral aneurysms. *Journal of Biomechanics*, 131(9), September 2009.

[28] K. Perktold, R.O. Peter, M. Resch, and G. Langs. Pulsatile non-newtonian blood flow in three-dimensional carotid bifurcation models: a numerical study of flow phenomena under different bifurcation angles. *Journal of Biomedical Engineering*, 13(6):507–515, nov 1991.

[29] Y. Leong Yeow et al. Model-independent relationships between hematocrit, blood viscosity, and yield stress derived from couette viscometry data. *Biotechnology Progress*, 18(5):1068–1075, 2002.

[30] L. Dintenfass. Red cell rigidity, tk, and filtration. *Clinical Hemorheology and Microcirculation*, 5:241–244, 1985.

[31] Daniel R. Gustafson. *Physics: Health and the Human Body*. Wadsworth, 1980.

[32] Jean-Frederic Brun, Ikram Aloulou, and Emmanuelle Varlet-Marie. Hemorheological aspects of the metabolic syndrome: markers of insulin resistance, obesity or hyperinsulinemia? *Clinical Hemorheology and Microcirculation*, 30(3-4):203–209, 2004.

[33] A. El Bouhmadi et al. Aggregability and disaggregability of erythrocytes in women suffering from ovarian cancer: evidence for an increased disaggregation threshold. *Clinical Hemorheology and Microcirculation*, 22(2):91–97, 2000.

[34] Maria Toth, György L. Nadasy, Istvan Nyary, et al. Sterically inhomogenous viscoelastic behavior of human saccular cerebral aneurysms. *Journal of vascular research*, 35:345–355, 1998.

[35] J. Alastruey, K. Parker, J. Peiro, S. Byrd, and S. Sherwin. Modelling the circle of willis to assess the effects of anatomical variations and occlusions on cerebral flows. *Journal of Biomechanics*, 40:1794–1805, 2007.

[36] Keri R. Moyle, Luca Antiga, and David Steinman. Inlet conditions for image-based cfd models of the carotid bifurcation: is it reasonable to assume fully developed flow? *Journal of Biomechanics*, 128(3):371–379, 2006.

[37] Kristian Valen-Sendstad, Kent-Andre Mardal, Harish Narayanan, and Mikael Mortensen. A comparison of some common finite element schemes for the incompressible navier-stokes equations. `http://folk.uio.no/kvs/paperI.pdf`, 2008.

[38] Philippe G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland Publishing Company, 1 edition, 1978.

[39] Susanne C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods*. Springer, 3 edition, 2008.

[40] Anders Logg, Garth N. Wells, Kent-Andre Mardal, et al. Automated scientific computing. FEniCS book project (`https://launchpad.net/fenics-book`).

[41] S Balaji Pai, RG Varma, and RN Kulkarni. Microsurgical anatomy of the middle cerebral artery. *Neurology India*, 53(2):186–190, 2005.

[42] Vmtk. `http://www.vmtk.org/`.

[43] Jaroslaw Krejza, Piotr Szydlik, David S. Liebeskind, Jan Kochanowicz, et al. Age and sex variability and normal reference values for the vmca/vica index. *American Journal of Neuroradiology*, 26:730–735, apr 2005.

[44] Paraview. `www.paraview.org`.

[45] Jay Dowell Humphrey. *Cardiovascular solid mechanics: cells, tissues, and organs*. Springer, 2001.

[46] K.A. Do, S.A. Treloar, N. Pandeya, et al. Predictive factors of age at menopause in a large australian twin study. *Human Biology*, 70(6):1073–1091, 1998.

[47] Bjarne K. Jacobsen, Anne Elise Eggen, Ellisiv B. Mathiesen, Tom Wilsgaard, and Inger Njølstad. Cohort profile: The Tromsø Study. *International Journal of Epidemiology*, 2011. (Electronic publication ahead of print: `http://ije.oxfordjournals.org/content/early/2011/03/20/ije.dyr049.short`).

[48] Alberto M. Gamburato, Joao Janela, Alexandra Moura, and Adelia Sequeira. Sensitivity of hemodynamic in a patient specific cerebral aneurysm to vascular geometry and blood rheology. *Mathematical Biosciences and Engineering*, 8(2), April 2011.