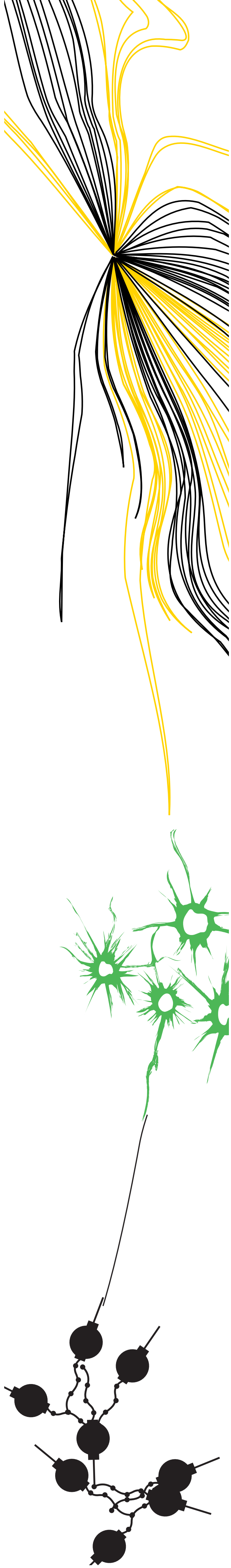


Faculty of Engineering Technology (ET)
Document number: 393

Computational modeling of interstitial fluid in the brain neuropil for an improved understanding of solute transport in the brain

Ruben Roozendaal
s1738526
M.Sc Thesis
6th January 2022

Supervisor:
Dr. K Jain
Engineering Fluid Dynamics Group
Faculty of Engineering Technology
University of Twente
P.O. Box 217
7500 AE Enschede
The netherlands

UNIVERSITY OF TWENTE.

Abstract

Alzheimer's disease is neurological disorder that gradually destroys the brain. It is the most common cause of dementia and it currently targets around 38 million people world wide. People with Alzheimer's disease suffer from memory loss, language difficulties, such as having problems coming up with words, movement difficulties, vision issues and more. The human brain has long been a mystery, which made finding the cause of Alzheimer's disease difficult. It is believed that accumulation of certain waste proteins produced in the brain is a cause of the neurological disorder. Proteins like amyloid β are transported away via the interstitial fluid (ISF) towards perivascular spaces where it is removed from the brain. The means of transportation of these solutes is debated. Conventional theories of the ISF flow state that solutes move towards the perivascular spaces purely due to diffusion. A more recent theory proposed a glymphatic system, that would serve as a lymphatic system, because the brain lacks one. It does that by the means of glial cells, hence the name glymphatic system. The pulsation of arteries in the brain would cause a bulk flow towards the periveinous spaces. This research studied the flow in the interstitial space, and evaluated whether diffusion is still the dominant transport phenomenon for solutes in the interstitial space when the glymphatic system is taken into consideration.

The interstitial flow is simulated using the Lattice Boltzmann Method. This is a CFD method based on a statistical approach of the collisions and movement of particles. The Lattice Boltzmann method lends itself for being used in complex geometries such as the interstitial space. The structured mesh is relatively easy to create and the calculations can be relatively easy parallelised. Parallelisation is in particular interesting in combinations with High Performance Computing, in which many processing cores are used for more computing power.
Simulations were done on meshes with around 385 million elements. The results were in agreement with flow values found in literature. Diffusion still seemed like the more dominant transport phenomenon for amyloid β, even with the glymphatic bulk flow. However, the results were not as convincing as in other studies, with Péclet numbers averaging around 0.1808 for amyloid β monomers, and 0.5213 for oligomers consisting of 24 monomers.

Keywords: Lattice Boltzmann Method; Computational Fluid Dynamics; Alzheimer's Disease; Amyloid β; Interstitial Flow; Glymphatic

Acknowledgements

While working on this thesis the past year, I have had a lot of help and support from people around me. While doing research for a thesis with a lot of theoretic computational work, you do not work closely with a lot of people. Especially with the pandemic going on, and a lot of working from home, there was a lot of working alone. All of this made the talks and discussions about my thesis I did have and all other social contact even more valuable.

To start, I would like to thank Kartik Jain, my daily supervisor for all his help. His kindness and patience past year have helped me a lot. His expertise on the brains fluid flows, and especially on the theory of the Lattice Boltzmann Method, combined with his ability to explain all this very clearly, helped a lot in my understanding of the necessary theory. In periods when simulations or the super computer were not cooperating and it felt like months were being wasted, his calm and patience put me at ease. Your broad knowledge about LBM and the simulation framework really helped to solve the problems we encountered. I feel like our way of thinking during meetings was quite similar, which made the meetings more fun and interesting and it gave me more confidence. Your supervision really helped during the past year, I want to thank you for both your knowledge and your kindness!

Next I would like to thank my housemates. During my whole study time they have always been good friends who made every day a bit more fun. But especially last year with COVID 19 going around and a lot of working from home, your company has really made the home office days a lot better. You were the best co-workers during working hours and the best group of friends outside working hours ;). I really could not have asked for a better group than you guys! Especially Lieke, supported me a lot. Thank you!

I would also like to thank my parents, brother and sisters. Thank you for your interest and support past year.

And last but not least I would like to thank my friends at the university. Thank you Jan for your help and discussions about the Lattice Boltzmann Method and my research. Your experience with a similar research has given me a lot of useful insights. Also a special thanks to Jeroen and Frank, my roommates at the university for the last few months. I liked our time there and it made working on my thesis a lot more fun. Almost the same goes for Maarten, Auke, Evert, Harold and Fausto. Working at our thesis at the same time has been valuable for me. I would also like to thank Sietse, Nina and Bram for their support past year.

Nomenclature

abbreviations

AD	Alzheimer's Disease
APP	Amyloid β Precursor Protein
CFD	Computational Fluid Dynamics
CSF	Cerebrospinal Fluid
EM	Electron Microscope
ISF	Interstitial Fluid
ISS	Interstitial space
LBM	Lattice Bolzmann method
LGM	Lattice Gas Models
NFT	Neurofibrillary Tangles
SP	Senile Plaques
WHO	World health organisation

Parameters

λ	tortuosity	[-]
μ	Dynamic viscosity	$[Pa \cdot s]$
ν	Kinematic viscosity	$[m^2/s]$
ν_L	Kinematic lattice viscosity	[-]
ρ	Density	kg/m^3
τ	Relaxation factor	[-]
ω	Relaxation rate	[-]
c	Lattice velocity	[-]
$c/\mathbf{u}$	velocity	[m/s]
c_s^2	Lattice speed of sound	[-]
D	Diffusion coefficient	$[m^2/s]$
f	Particle distribution function	[-]
f^{eq}	Equilibrium distribution function	[-]
L	Traveled path	$[m]$
L_BC	Length of the bounding cube	$[m]$
m	Mass	$[kg]$
P	Pressure	$[Pa]$
Pe	Péclet number	[-]
Re	Reynolds Number	[-]
T	Temperature	[K]
t	time	$[s]$

Contents

Chapter 1

Introduction

Currently more than 55 million people suffer from dementia world wide, and these numbers are increasing with nearly 10 million each year according to the World Health Organisation (WHO). Around 60-70 % of these cases are related to Alzheimer's disease (AD). AD has a big effect on the the life of its patients. Early stage symptom include forgetfulness and getting lost in familiar places, and these symptoms increase in severity in later stages to being unaware of time and place and not recognising the faces of friends and family to the point that it is impossible to take care of oneself. Also the personality of patients may change including becoming more aggressive. Next to some preventive advises, like enough physical and mental exercise, there is currently no cure for Alzheimer's disease other than fighting some symptoms to improve the live of people affected. The economic and social impact of AD are both significant. It is estimated that the total societal cost of dementia globally was around 1.3 trillion US dollar and is expected to increase to 2.8 trillion US dollar in 2030 due to both an increase in patients and medical costs. [1]

Research on brain tissue of AD patients showed an accumulation of toxic forms of the protein Amyloid β, which is a waste product of brain cells. This accumulation is the result of disturbances in the balance between the production and removal of this protein. [2] [3] [4]. In order to bring back this balance it is important to know how waste proteins like Amyloid β are removed from healthy brains. Increasing the clearance of these waste products would be a good step towards curing patients with Alzheimer's disease and even preventing the onset of this disease. For this purpose, research has been done looking into the flows of the fluids inside the brain and the driving force of the transport of solutes in these fluids. In the absence of blood and lymphatic fluid inside the brain, nutrition and waste products are transported via interstitial fluid (ISF) and cerebrospinal fluid (CSF).

Early research on mice concluded that the majority of excess Amyloid β is transported by diffusion and cleared through the blood brain barrier and only a fraction is removed by bulk flow of ISF. However, more recent studies with better imaging technologies suggest that the part that is cleared by bulk flow is actually larger than hypothesised before. Because of the complex geometry of the brain and the small scale, it is difficult to build an experimental setup that represents the interstitial space (ISS) in the brain well enough. With increasing computational power, computational fluid dynamics can give more new insights in the ISF flow and the transport of waste products. Some research has been done using common computational fluid dynamics methods based on solving the Navier-Stokes equation and the advection diffusion equation on geometries created based on the general porosity of the interstitial space, as well as on geometries retrieved from electron microscope imaging.

This thesis will use the Lattice Boltzmann Method on such a EM generated geometry. The Lattice Boltzmann Method is a CFD method that uses a statistical approach to solve the Boltzmann transport formula. Common CFD methods like Finite Element Method or Finite Volume Method solve the Navier-Stokes equations in each point. The derivatives in these equations are approached using neighbouring points. This makes it harder to run a simulation in parallel on multiple cores, since nodes on one core will always need information from another core. LBM can be run on large scale computers more easily. Furthermore, the uniformly spaced grid makes the meshing of complex structures easier. This project will use the Dutch supercomputers Cartesius and Snellius to simulate the interstitial flow in the brain geometry to find out whether the transport of solutes in the interstitial space of the brain is mainly diffusion or bulk flow driven.

1.1 Research Question

The main question of this thesis is: **"Is the transport of amyloid β in the interstitial space in the brain mainly diffusion or bulk flow driven?"**
In order to answer this question, several sub questions arise:

1. **Does the velocity profile of interstitial fluid represent the velocity of amyloid β molecules ?**
The simulation framework used in this thesis is not yet able to simulate solutes in a flow. In order to get results, the simulation is done with only interstitial fluid. It is important to know whether this has a significant effect on the results.

2. **How can the Peclèt number be derived from the LBM simulations?**
The ratio between advective and diffusive transport is given by the Peclèt number. In order to derive this ratio distinction needs to be made between the advective transport of amyloid β and the diffusive transport.

3. **Using what boundary conditions can the flow in the in the ISS be simulated realistically?**
Little is known about the flow properties inside the interstitial space, there are estimations on pressure drops and flow velocities. Based on these estimations, boundary conditions need to be formed. The influence of a change in these boundary conditions on the results needs to be checked.

4. **What effect do external periodical influences have on the results**
The estimations for the flow values mentioned above are mostly derived from averaging over time. However periodical influences like the breathing cycle and the cardiac cycle are thought to have an influence on the ISF flow.

1.2 Scope

In this research the nature of particle transport in the brain will be studied. More specifically, this research will focus on the protein amyloid β. The results of this research can be valuable for the understanding of Alzheimer's disease since amyloid β is a waste product produced in the brain that is associated with Alzheimer's disease. Understanding the driving force behind the disposal of this protein might help creating a preventive life style or even help finding a cure. While this would be a very positive outcome, the scope of this research is to find the driving force behind the particle transport. There will be no medical advice given in this thesis.

This in this thesis the Lattice Boltzmann Method is used. A general understanding is necessary, but it does not have to be used to make a code for the simulation. Packages from a framework called APES will be used for this purpose.

1.3 outline

This report will start looking into the current state of research regarding the understanding of Alzheimers disease. It will also take a look into the necessary knowledge about the brain physiology and the current state of research on the fluid flows in the brain.
After that, the theory of the Lattice Boltzmann Method is explained also more information is given about the APES simulation framework.
Then simulattions on a pipe flow will be done, both to verify the Lattice Bolzamnn Method and to preform a mesh convergence study. This is followed by the results of the flow simulations in the brain geometry.
Finally the report ends with the conclusions, points of discussion and some recommendations.

Chapter 2

Current State of Research

2.0.1 Pathology of Alzheimer's Disease

Alzheimer's disease is the most common form of dementia. It is a neurodegenerative disease targeting mostly the elderly population. The main two hypotheses for the cause of Alzheimer's are Senile Plaque (SP) made of Amyloidβ and neurofibrillary tangles (NFTs) formed by phosphorylated tau protein [5].
The amyloidβ theory was already proposed in 1991 [6]. Amyloidβ is produced by hydrolysis of the Aβ Precursor Protein (APP). This protein goes through a cell wall and can be hydrolysed in 3 different ways. The difference in these three ways is the location of the hydrolysis on the APP molecule. There are four different secretases (α-, β-, γ- and η-) which slice the APP on different positions.[7] See Figure 2.1.

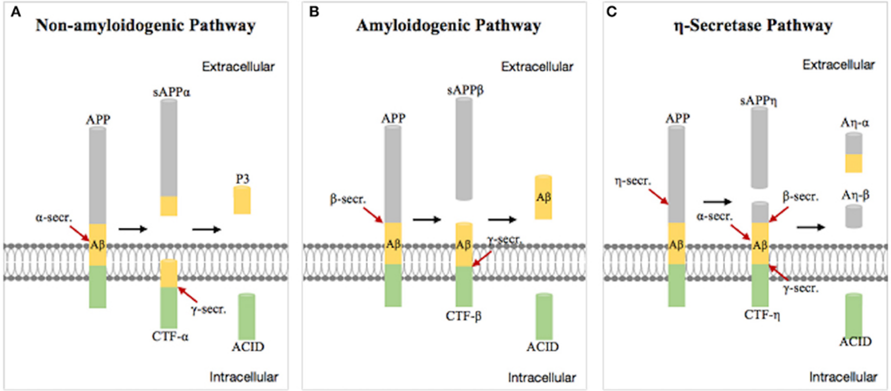


Figure 2.1: Different ways of APP hydrolysis [5]

In the case of option B, the amyloidogenic pathway, the amyloidβ part of the APP molecule stays intact. These amyloidβ monomers can come together and form beta sheets. The beta in beta sheets is unrelated to the beta in amyloidβ. A beta sheet is a common structure in polypeptides, where a long chain of peptides folds multiple times onto itself, forming a sheet with hydrogen bonds. These neurotoxic amyloid plaques "induce mitochondrial damage, unstable homeostasis, and synaptic dysfunction.. ..Eventually, neuronal dysfunction and apoptosis occur, leading to Alzheimer's disease"[5].[7]
The NFT hypothesis refers to the phosphorylation of the tau protein. Phosphorylation is the attachment of a phosphoryl group to a protein. The tau protein maintains the tubular form of the axons in neurons. When more and more phosphoryl groups attach to the tau protein problems arise. What happens exactly is a complex process depending on multiple factors, which is still not totally understood.The phosphorylation causes problems with the tubular shape of the axons, furthermore the tau proteins will interact with each other forming neurotoxic NFTs.[5]

In this thesis, the focus will be on the amyloidβ hypothesis. This problem is currently being approached from multiple perspectives. One way to to prevent the forming of the senile plaques is to stop amyloidβ from being made[8][9][10], other research focuses on the stopping the amyloidβ molecules from clustering together[11], while other research focuses on the removal of the protein in both the peripheral nervous system[12] and the central nervous system[13][14]. Here, the focus will be on the latter. For that purpose a closer look is taken into the flows of the fluids in the brain.

2.0.2 Brain physiology and its fluid flows

The brain is, as part of the central nervous system, a crucial organ in the human body. It is used for motor control, managing sensory impulses, regulation of autonomic functions and hormone control, memory, language, cognition and more. The brain cells, or neurons are connected to each other via axons and dendrites, these are branches of the cells that can send (axon) or receive (dendrite) electrical signals to other neurons. These connections between neurons make brain activity possible.[15]

Just like other cells, neurons need fuel to function properly. The brain consumes about 20% of the energy used in the entire body [16]. The supply of fuels like glucose or ketone bodies as well as the disposal of waste products happens through a system of fluids. The flows of these fluids are quite complex, and especially for the lymphatic drainage, the flow is poorly understood [17][18][19]. In Figure 2.2 an overview of these flows can be seen. Cerebrospinal fluid is the fluid in the large cavities in the brain, called ventricles. The CSF flows through these ventricles and then flow into the subarachnoid space where it surrounds the brain and spinal cord functioning as a protective cushion. Interstitial fluid enters the parenchyma (the functioning tissue of the brain) through arteries and flows either into the CSF or into a vein. Both CSF and ISF have the function of delivering nutrients to and removing waste products from the neurons.[18]

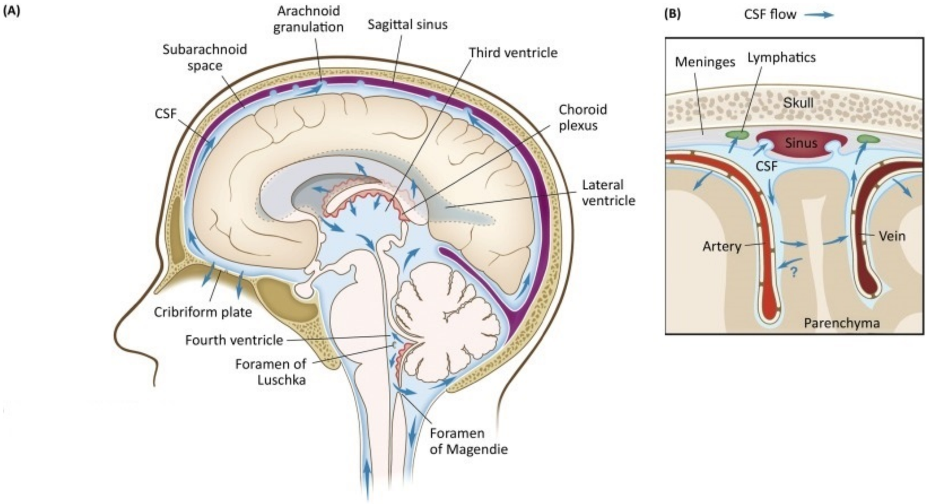


Figure 2.2: Overview of the fluid flows in the brain (A), and a focused section zoomed to see the ISF flow. [17]

Given the crucial role these fluids play in the well being of the brain, it is important to have an idea of how and where they are created, what route they take through the brain and where they end up. This will not only provide background information for the topics of this thesis, but it will also be used to retrieve the necessary flow conditions for the simulations later on in this report. In Figure 2.3 a schematic overview is given of the CSF and ISF flows in the brain.

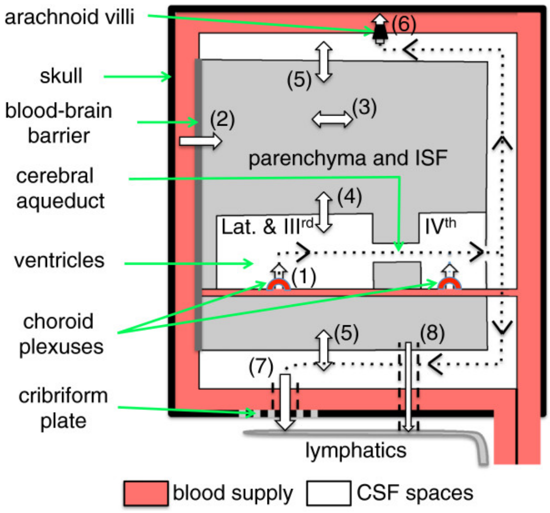


Figure 2.3: Schematic overview of the fluid flows in the brain [18]

1. CSF is produced in the the choroid plexusses. These are networks of nerves that make CSF from fluids and nutrients provided by the blood supply. They are located in the ventricles where the CSF flow starts.

2. Fluid flows through the blood brain barrier into the parenchyma, once this fluid is through the blood brain barrier it is called interstitial fluid.

3. The ISF moves through the parenchyma, where neurons exchange nutrients and waste products with it.

4. and 5. CSF and ISF can exchange water, nutrients and waste products in both the interfaces between the parenchyma and ventricles and the parenchyma and subarachnoid space.

6. Fluid from the CSF in the subarachnoid space flows into the vascular system through arachnoid granulations into venous sinuses. These venous sinuses have the same function as veins, however, their outside layers are made of different materials [20].

7. Fluid from CSF also flows from the subarachnoid space into the lymphatic system.

8. ISF can also leave the brain via the lymphatic system in the neck directly, without entering the CSF first. [18]

In this research a closer look will be taken into the ISF flow in the parenchyma. Accumulation of the protein Amyloidβ, can have negative effects on the health of the brain, like increasing the risk of Alzheimer's disease. This is elaborated further in subsection 2.0.1: **Pathology of Alzheimer's disease**.

The flow of ISF in the parenchyma is shown in Figure 2.4. Fluid leaves the arteries through the blood brain barrier, consisting of endothelial cells with tight junctions, astrocytes and pericytes. This barrier only lets certain molecules through using active transporter proteins and only small molecules can diffuse through. The blood brain barrier maintains homeostasis in the brain so the neurons are able to function properly even with fluctuations in blood values. [21]

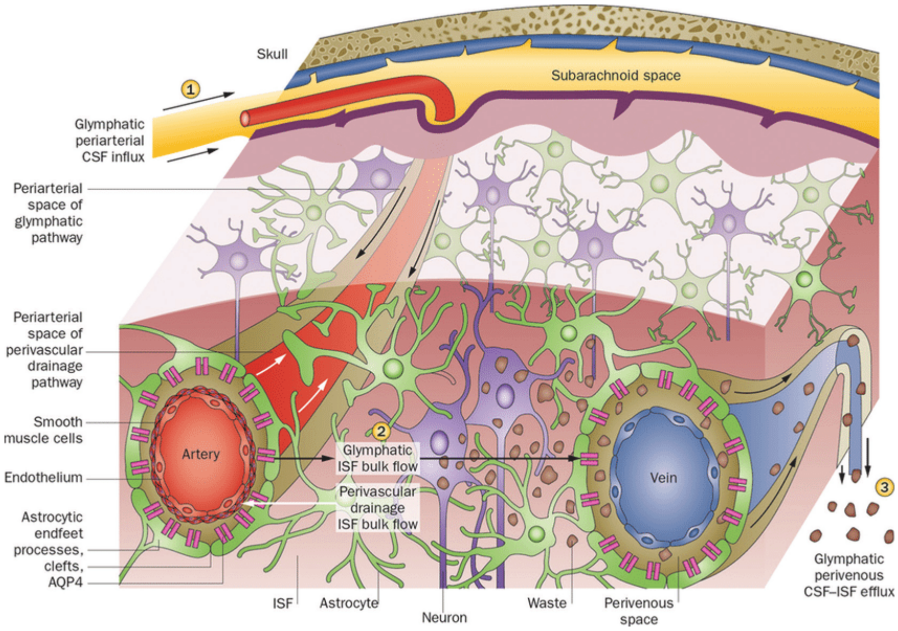

Skull
Subarachnoid space
1
Glymphatic periarterial CSF influx
Periarterial space of glymphatic pathway
Periarterial space of perivascular drainage pathway
Smooth muscle cells
Endothelium
Astrocytic endfeet processes, clefts, AQP4
Artery
2
Glymphatic ISF bulk flow
Perivascular drainage ISF bulk flow
Vein
3
Glymphatic perivenous CSF–ISF efflux
ISF
Astrocyte
Neuron
Waste
Perivenous space

Figure 2.4: Overview of ISF flow in parenchyma [22]

The blood vessels penetrating the brain are surrounded by CSF, This is called the perivascular space. AQP4 channels in the astrocytic cells around the perivascular space provide a passage for CSF to flow into the interstitial space. These AQP4 channels are aqua-porous proteins, meaning that they can let through water. [21] The ISF then flows through the interstitial space, clearing it of waste products. One of these waste products, Amyloidβ is thought to be a cause of Alzheimers disease. The ISF moves towards a vein where it enters the periveinous space together with the waste products. From there the waste products are now transported out of the brain via either the vascular system or the lymphatic system.

This flow in the parenchyma was long thought to be purely diffusion driven. Waste particles can both disperse into the periarterial space as well as into the periveinous space. since the solutes will be taken away via these spaces, a concentration gradient appears and solutes in the middle of the parenchyma start to diffuse to the perivascular spaces. A relatively recent study from 2012 proposed the "glymphatic (glial-lymphatic) hypothesis", which states that there is also a convective component in this flow[23]. The bulk flow is argued to originate from the pulsatile behaviour from the arteries which generates a directional flow towards the periveinous spaces[24]. This glymphatic system would explain the clearance of waste products in the brain in the absence of a lymphatic system. A schematic visualisation of the difference between the conventional view and the glymphatic hypothesis can be seen in Figure 2.5.

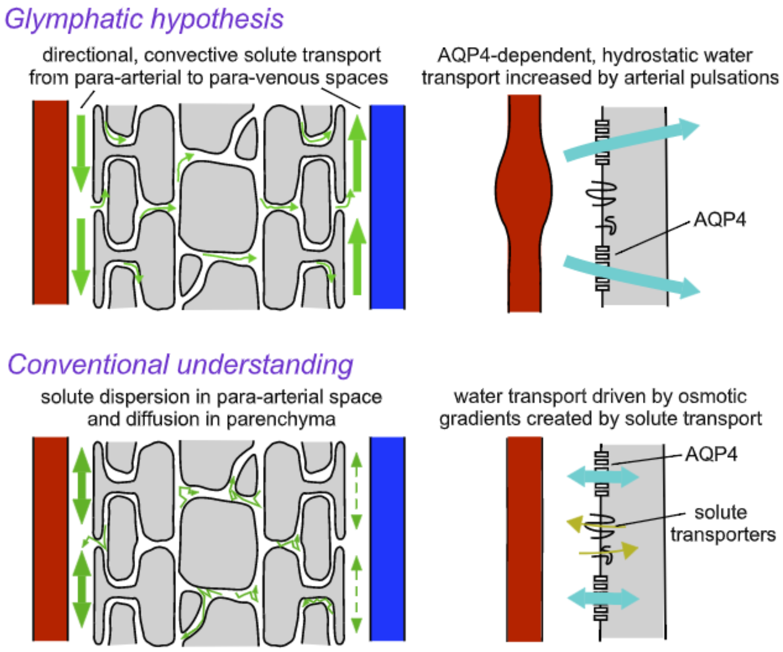

Glymphatic hypothesis
directional, convective solute transport from para-arterial to para-venous spaces
AQP4-dependent, hydrostatic water transport increased by arterial pulsations
AQP4
Conventional understanding
solute dispersion in para-arterial space and diffusion in parenchyma
water transport driven by osmotic gradients created by solute transport
AQP4
solute transporters

Figure 2.5: Glymphatic hypothesis [24]

The idea of this glymphatic system was welcomed in the scientific community, but it is not a proven concept[24]. When the glymphatic system is not considered to be correct, the answer on the question in this research would be very easy. In the absence of bulk flow the transport of solutes would be purely diffusion driven. In this report, the glymphatic hypothesis is assumed to be correct and the results of this research will tell if the nature of particle transport changes significantly.

The necessary flow properties to run simulations can be found in literature. For the derivation of the flow velocity, an ionic substance was applied to mouse and rat brains and the concentration over time was monitored. Then, simulations with different bulk flow velocities were preformed to see at which velocity the concentration curve matched the experimental data. Keeping inter-animal tissue variation in mind, the average velocity of the ISF flowing towards the veins ranges from 5 to 15 $\mu m/s$[25][26][27] and averages around 10 $\mu m/s$ [28][29]. These values are based on an asleep state. When awake, brain cells are more active and take up more space. This results in a decrease of void volume, the volume outside the cells. When awake, the velocity of the ISF can decrease by up to 93%, resulting in a velocity of 0.7 $\mu m/s$.

The pressure drop over the interstitial space is still a topic of debate, but it can be estimated using the hydrostatic pressure gradient in the brain and the arterial pressure. For the former, an upper limit of 1 mmHg/mm is commonly used [30][28]. For the arterial pressure gradient, the pressure drop over the arterioles, capillaries and venules in the brain is of importance. This pressure drop ranges from 65-100 mmHg. This pulsation translates through the elastic walls of the arterioles, according to literature, only a small percentage of this pressure translates into a pulsation in the interstitial space [31]. Pressure differences in the interstitial space range from

0-10 mmHg.[28][31][32] This pressure difference divided over the vascular separation of 238 μm [30] (the distance between arterioles and venules) will give a hydrodynamic pressure gradient of 0-42 mmHg/mm.

On the geometry, the following can be found. The interstitial space can be grouped into two types of volumes: tunnels and sheets. Sheets are the flat volumes between two cell walls, they are 10-40 nm wide. tunnels are the regions at the intersection of sheets, these tunnels are wider, about 40-80 nm.[30] The interstitial space has a tortuosity of about 1.7 [33]. This means that the distance a particle has to travel to get from A to B, divided by the linear distance is on average 1.7. This number helps with finding the diffusion coefficient of Amyloidβ. The diffusion coefficient of Amyloidβ monomers in aqueous solutions at 37°C is approximately $1.8 \cdot 10^{-6} cm^2/s$. The effective diffusion coefficient in brain tissue is then given by:

$$D_{eff} = \frac{D}{\lambda^2} = 0.623 \cdot 10^{-6} cm^2/s.$$

Where λ is the tortuosity.[33]. The D_{eff} of oligomers consisting of 24 momomers is also given in the same paper. This is $0.216 \cdot 10^{-6} cm^2/s$. This method of finding the effective diffusion coefficient does not take into account hindered diffusion near walls and due to the fact that the diffusing solutes have dimensions in the same order of magnitude as the gap widths in the interstitial space. The coefficients are however verified in in vivo experiments and they seem to be correct.[33]

This research thesis will focus on the clearance of amyloidβ in the central nervous system. Knowing whether the clearance of waste products happens mostly because of diffusion or advection, will help in understanding how amyloidβ is cleared from the brain. Flow simulations using the Lattice Boltzmann Method (LBM) will be used in an electron microscope scan of the neuropil of the hippocampus. How the LBM works will be explained in chapter 3: **Lattice Boltzmann Method**.

Chapter 3

Lattice Boltzmann Method

The Lattice Boltzmann Method is a relatively upcoming CFD technique. It has some advantages over common CFD techniques mainly because of the fact that it is very suitable to run on parallel processing cores. That makes it easy to run large simulations on super computers, which get their large computing power by using a lot of cores. Common CFD methods approximate Differential equations, like the Continuity equation, the Navier-Stokes equation and the Total energy equation. This approximation is done by discretising the grid into nodes and calculating the derivatives in each node based on neighbouring nodes. Using given values of either the solution itself, the derivative of the solution of a combination of both on the boundaries, a solution for the entire volume is found. The fact that information on all the nodes is necessary to calculate the solution in a node makes it difficult to run a simulation on multiple processing cores. Since they would have to exchange a lot of information. In the Lattice Boltzmann Method, this is less of a problem, since a big part of the calculations can be done for each element individually.

The conservation equations, and CFD methods relying on these equations, are based on macroscopic values like pressure, density velocity and temperature. They rely on the continuum hypothesis, the volume in the simulation is seen as a continuous blob of fluid and not as a collection of individual molecules. Since the macroscopic values of the fluid are a consequence of averaging properties of individual molecules or collisions between molecules, there should be enough molecules in each node for this averaging to make any sense. One could imagine that this works fine as long as the volume and the spatial step is significantly larger than the size of the molecules of the fluid. This can be represented by the Knudsen number:

$$Kn = \frac{l_{mfp}}{L} \tag{3.1}$$

Where L is a relevant length scale, e.g. the channel width, and l_{mfp} is the mean free path of a molecule, the mean distance traveled between collisions. In liquid this number does not say a lot, since the molecules are loosely bound. The mean free path can be estimated by taking the distance between the molecules [34]. For liquid water, a mean free path of 0.31 nm is often used[35]. With channels sometimes as narrow as 10 nm, the Knudsen number in the neuropil will reach values of 0.031. With such high Kn values, the volume cannot be seen as a continuum anymore and a statistical model is needed.[35]

The Lattice Boltzmann Method uses statistical mechanics and kinetic theory of gasses to solve the Boltzmann transport equation. Historically, LBM originates from the Lattice Gas Automata (LGA), which uses the same principles but has its drawbacks due to its relevant scale. Lattice gas methods look at the individual particles that make up the fluid. In these models, gas particles exist on a lattice where they can stream to neighbouring nodes and collide with other particles, obeying the conservation of mass and momentum. Since individual particles are considered, the spatial scale of the simulation can be much smaller than in conventional CFD methods. See Figure 3.1[36]. This would solve the problem of the too high Kn number for conventional CFD methods. However, this method also has its drawbacks. One of the drawbacks can be seen in Figure 3.1. The time scale of these small scale simulations is very small. This is quite intuitive, since one time step would cover at most one collision of each particle. Considering the amount of collisions a particle can make in a second, it becomes obvious that this time step cannot be very large. Typically a time step in these molecular models are of the order of magnitude of $10^{-15}s$[37], meaning you would need one thousand million million time steps to simulate one second. Another downside of this method is the computational power that is required to simulate every single particle in a volume. Also, in lattice gas models particles in a position can only stream in one direction, while there could be more options that obey the mass and momentum equilibrium. Therefor, they require suitable averaging to make up for the multiple possible streaming steps.

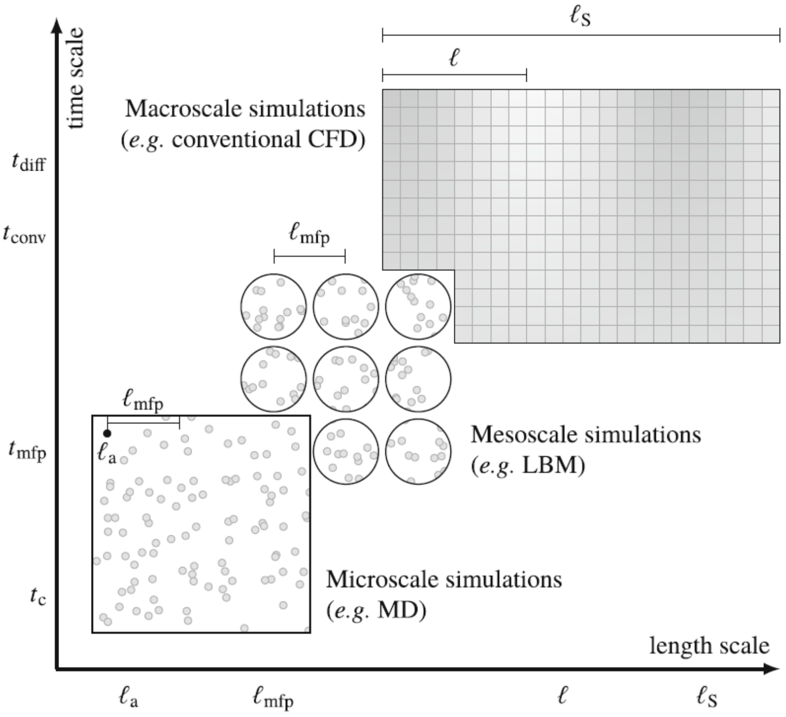


Figure 3.1: Relevant scales [36]

A way to solve these problems is to look at a scale between the miscroscale and the macroscale; a mesoscale. This is what the Lattice Boltzmann Method does. On each node, there is now a cluster of particles instead of a Boolean expression of either one or zero particle. Now each streaming step will not be solely in one direction, but it will be a distribution of streams in each possible direction.

The Lattice Boltzmann Method derives these distributions by using statistical mechanics on the Boltzmann transport equation. This chapter will start looking at the derivation of the Boltzmann equation, which forms the basis of LBM. This equation will be discretised and applied on a microscale, to derive a lattice gas model. From here, the step will be made towards a Lattice Boltzmann model, using clusters of particles. This chapter ends with explaining the application of this equation on the lattice, and some extensions and improvements on the theory.

3.1 The Boltzmann Equation

The Boltzmann equation can be derived by taking a look at the possible state of particles in a system. Molecules in a fluid are constantly moving and interacting with each other. On a macro scale the fluid can be described with properties like density, pressure and temperature, but these properties are all consequences of molecular behaviour. Density could be described by taking the amount of particles in a certain volume, the pressure is a consequence of an amount of particles hitting an area with a certain velocity, and temperature can be derived from the kinetic energy of the particles in the fluid. So on a molecular level, the state of a fluid at a certain moment can be described by knowing the position and velocity of the particles. Knowing this state of every particle in the fluid would be impossible. It is much more doable to instead look at the probability of a certain amount of particles to be in a certain position with certain velocities. This is called the Particle Distribution Function $f(\vec{x}, \vec{\xi}, t)$, and it is the fundamental variable in kinetic theory. f not only represents the mass density in a three dimensional physical space, but also in a three dimensional velocity space. So it would give the mass of the particles in a certain moment in time t, in a certain position $\vec{x}$, with a certain velocity $\vec{\xi}$.

The macroscopic variables total mass density, total momentum density and total energy density can be derived from this variable. The density can be derived by taking the integral of f over all the velocities $\vec{\xi}$:

$$\rho(\vec{x}, t) = \int f(\vec{x}, \vec{\xi}, t) d^3\xi \tag{3.2}$$

The momentum density can be calculated by multiplying the mass density by the velocity:

$$\rho(\vec{x}, t)\vec{u}(\vec{x}, t) = \int \vec{\xi} f(\vec{x}, \vec{\xi}, t) d^3\xi \tag{3.3}$$

The energy density is calculated by integrating the kinetic energy densities of all the particles:

$$\rho(\vec{x}, t)E(\vec{x}, t) = \frac{1}{2} \int \vec{\xi}^2 f(\vec{x}, \vec{\xi}, t) d^3\xi \tag{3.4}$$

This shows that, although we are looking at particles, or bundles of particles, the connection to the macroscopic scale can still be made quite easily. Of course integrating over all possible particle velocities might be a lot of work, but discretising the velocity space will help a lot as we will see later on in section 3.2.

The state of the total system can of course change over time, due to collisions for example, the velocity of particles in a certain region can change and the total state will then not be the same. This change of the total state can be denoted by the material derivative of $f(\vec{x}, \vec{\xi}, t)$ over time $\frac{Df}{Dt}$. Since $f(\vec{x}, \vec{\xi}, t)$ depends on multiple variables, there are multiple factors that can cause this change of the total state. Since $f(\vec{x}, \vec{\xi}, t)$ can be described as the mass of the particles with a certain velocity in a certain volume, both the number of particles in the certain volume and the velocity of the particles can change over time. The chain rule can be used to write out the material derivative.

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x_i}\left(\frac{dx_i}{dt}\right) + \frac{\partial f}{\partial \xi_i}\left(\frac{d\xi_i}{dt}\right) \tag{3.5}$$

Some simplifications are possible here, the change of location of a particle over time is the particle velocity ξ_i. And the change of the velocity of a particle over time is the particle acceleration, which according to Newtons second law of motion can also be written as $\frac{F_i}{\rho}$, where F_i is the body force per unit volume[38]. The change of the total state can now be described as:

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + \xi_i \frac{\partial f}{\partial x_i} + \frac{F_i}{\rho}\frac{\partial f}{\partial \xi_i} \tag{3.6}$$

Without external factors, a system will move towards an equilibrium state. In this equilibrium state, the equation above will be zero since the total system will not change anymore. The distribution of f in this equilibrium state is called Maxwell-Boltzmann distribution[36]. In Figure 3.2 the change of this distribution can be seen for when the temperature changes. What is also interesting is that this distribution can be described as the amount of particles with a certain velocity. $\frac{Df}{dt}$ is equal to zero when the distribution of f equals the equilibrium state. When this is not the case, there must be particles with a velocity that does not agree with the equilibrium state. The only way for a particle to change its velocity is to interact with another particle (collisions), there is now way the particles can slow down due to friction or something, because on this scale that would again be an interaction with a different particle. Since the change of the state of a system is always due to collisions, the material derivative $\frac{Df}{Dt}$ could be set equal to a certain collision operator as a funtion of the current state $\Omega(f)$. This gives the Boltzmann equation:

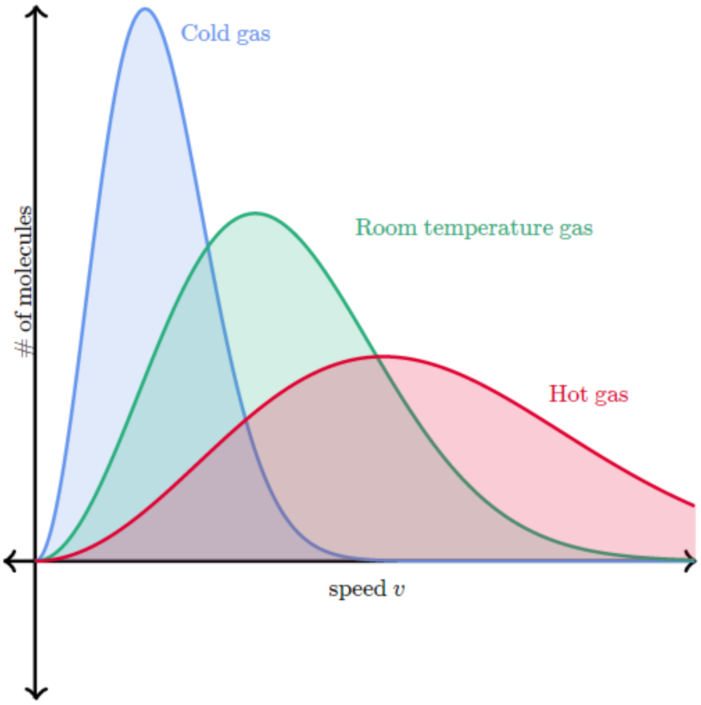


Figure 3.2: Maxwell-Boltzmann distributions for different temperatures [39]

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + \xi_i \frac{\partial f}{\partial x_i} + \frac{F_i}{\rho}\frac{\partial f}{\partial \xi_i} = \Omega(f) \tag{3.7}$$

The exact form of this operator is not set in stone, but it has to obey certain rules. There should be a conservation of mass, momentum and energy in the system. In order to acchieve that, Equation 3.2 to 3.4 should not change in time. Applying the same integrals to the left hand side of Equation 3.7 gives the time derivatives of the total density, momentum and energy. These derivatives should be equal to zero. Consequently, the following has to hold for $\Omega(f)$:

$$\begin{aligned} \int \Omega(f) d^3\xi = 0 \\ \int \vec{\xi}\, \Omega(f) d^3\xi = 0 \\ \int \vec{\xi}^2\, \Omega(f) d^3\xi = 0 \end{aligned} \tag{3.8}$$

"Boltzmann's original collision operator is of the form of a complicated and cumbersome double integral over velocity space. It considers all the possible outcomes of two-particle collisions for any choice of intermolecular forces."[36] As you can imagine, solving this complicated integral can take up a lot of time and computational power, there are more simple operators that also work. The simplest version of $\Omega(f)$ is the BGK operator, named after Bhatnagar, Gross and Krook. it is given by:

$$\Omega(f) = \frac{f_{eq} - f}{\tau} \tag{3.9}$$

It replaces the integral with an operator that only depends on one variable, τ: the relaxation time. This relaxation time determines the rate at which the current state f moves toward the equilibrium state f_{eq}. It is clear that this does not exactly depict realty, since this operator only allows the state to move towards equilibrium, while in reality particles move more randomly and it would be very well possible to move away from the equilibrium state. Moving towards equilibrium is only a result of the fact that the equilibrium state is a more favourable state to be in, so statistically a collision would push the state more towards equilibrium. That in combination with the vast amount of particles that collide every time step, it would be statistically sound to say that the BGK operator is a usable simplification of reality. There are more possible collision operators, some of which will be elaborated later on in section 3.4.

Before moving on to applying the Boltzmann equation on a discretised grid, there is still one more thing worth too take a closer look at: The physical meaning of this collision operator. Most of the macroscopic properties of a flow could already be derived from $f(\vec{x}, \vec{\xi}, t)$, so what does this new term in the equation represent? $\Omega(f)$ determines the rate at which the system moves towards equilibrium, and it is influenced by the interactions between the particles within the flow. A low value for $\Omega(f)$ would mean that interactions between the particles slow down the relaxation. This describes viscosity. Fluids with a high viscosity would have a longer relaxation time because of the resistance it gets from itself, resulting in a low value for $\Omega(f)$.

3.2 Discretisation

In this section, the Boltzmann equation will be used to describe the state of a system on a discretised grid. First, a short look is taken into a lattice gas model on microscale, since the concept of collisions is more straightforward there. After that the translation will be made to the mesoscale where the Lattice Boltzmann method is explained.

3.2.1 Lattice Gas Models

The first lattice gas model that could be used for simulations was made in 1986 [36]. Consider a 2D plane discretised in an evenly spaced triangular grid. Each node on this grid now has six neighbouring nodes. Each node can either contain a particle or not, indicated by a Boolean. Additionally, next to only discretising the physical space, the velocity space is also discretised. In one time step, a particle can only travel to one of the six neighbouring nodes, resulting in six possible values for the velocity, indicated by c_i. The state variable $f(\vec{x}, \vec{\xi}, t)$ from the continuous Boltzmann equation has now been reduced to the occupation number $n_i(\vec{x}, t)$ and its value does not represent the mass of the particles with a certain velocity $\vec{\xi}$ in a certain volume, but whether a particle with a certain discrete velocity c_i is present on a certain position in the grid. It has either a value of 1 or 0. The mass density and momentum density can again be calculated in a similar fashion as in Equation 3.2 to 3.4, but now, the integrals are replaced with summations, since there is a finite amount of possible velocities.

$$\rho(\vec{x}, t) = \frac{m}{v_0} \sum_i n_i(\vec{x}, t), \quad \rho\vec{u}(\vec{x}, t) = \frac{m}{v_0} \sum_i c_i n_i(\vec{x}, t) \tag{3.10}$$

Where m is the particle mass and v_0 is the volume of a node. The Boltzmann equation can also be discretised in the following way:

$$\frac{Df}{Dt} = \Omega(f) \;\rightarrow\; \frac{\Delta n}{\Delta t} = \Omega_i \;\rightarrow\; \frac{n(\vec{x}, t + \Delta t) - n(\vec{x}, t)}{\Delta t} = \Omega_i(\vec{x}, t) \tag{3.11}$$

Now the collision operator Ω is the difference between the states before and after the collision step. So this new state after collision can be calculated by:

$$n_i^\star(\vec{x}, t) = n_i(\vec{x}, t) + \Omega_i(\vec{x}, t)\Delta t \tag{3.12}$$

Since n_i and n_i^* can only have a value of 1 or 0, Ω_i can have a value of 1, 0 or -1. This collision operator again has the same constraints as in the continuous Boltzmann equation as in that it has to obey the conservation equations.

Equation 3.12 is one of the two core steps in each time step, called the collision step. Figure 3.3 shows possible states before and after collision. After the collision step, each node has a new state n_i^* where each particle has a new velocity. That is when the next core step starts: the streaming step. In this step, the particles travel with their new velocities to the next node. So an element on location $\vec{x}$ with velocity c_i after collision causes $n^*(\vec{x}, t)$ to have a value of one. After streaming a neighboring node on location $\vec{x} + c_i\Delta t$ will have that same state. This can be notated as:

$$n_i(\vec{x} + c_i\Delta t, t + \Delta t) = n_i^\star(\vec{x}, t) = n_i(\vec{x}, t) + \Omega_i(\vec{x}, t)\Delta t \tag{3.13}$$

The next time step, this process of collision and streaming is repeated. Each time step a translation is made to the macroscopic values density ρ and velocity $\vec{u}$, to calculate the equilibrium state. Based on the equilibrium, the collision operator is derived, this will be elaborated further

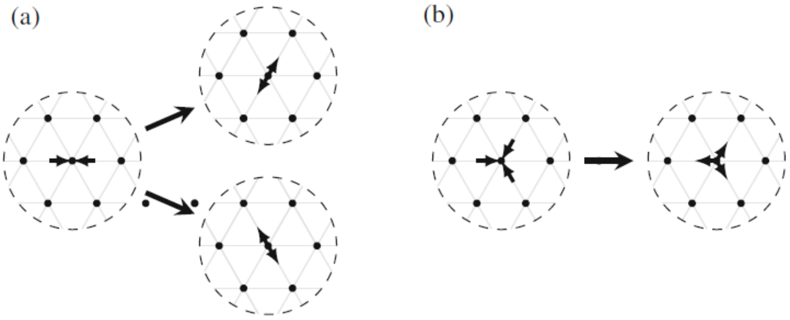


Figure 3.3: Examples of possible collisions [36]

in section 3.3 **Lattice Boltzmann Method**. After that the collision and streaming step can be done as described above. These kind of models get their charm from their simplicity, however there are some big downsides. The biggest downside of this method is the statistical noise it produces. The control volume is quite small and molecules can move in and out randomly. Causing macroscopic values like e.g. density to fluctuate, even when the fluid is in equilibrium. Also, for these models As you can see in Figure 3.3, multiple collisions are possible in order to obey the mass and momentum equation, but only one is chosen. Averaging over either time or multiple similar volumes could give a solution, but that will not completely solve the problem and it will cost more computing power. Another solution will bring us to the core part of this chapter. Instead of looking at the occupation number as a Boolean for whether there is a particle at a certain location with a certain velocity, we now look at what percentage of the particles in the node will probably have that velocity. This means the scale moves from micro to meso, since it is impossible to take a percentage of a single particle other than 0% or 100%. Each element will now contain a number of elements with all different velocities.

3.3 Lattice Boltzmann Method

3.3.1 Discretised Lattice

Both the Lattice Gas Model and the Lattice Boltzmann Method make use of an evenly spaced grid. Before the LBM is explained, more attention is payed to this titular lattice. Where the Lattice gas Models use a triangular grid, LBM uses a square grid. While the discretisation in space for this square grid is quite simple, there are multiple stencils possible for the velocity sets. These stencils can be described by the format DxQy, where x indicates the number of dimensions (1, 2 or 3) and y indicates the number of neighboring nodes a node is in contact with. This can differ based on the fact whether certain diagonal connections are taken into account. Common lattice configurations are D2Q9, D3Q15 and D3Q19, these are illustrated in Figure B.7. In this section, the focus will be on the D2Q9 stencil because of its simplicity compared to the 3D stencils, but the concept is the same for all of them.

As mentioned before, the velocity space is discretised and the different velocities are indicated by c_i. In the case of D2Q9 i ranges from 0 to 8, where c_0 is the velocity for the particles that do not leave the current cell. c_1 to c_4 are the velocities in the direction of the orthogonally adjacent cells, these vectors have a length of 1. c_5 to c_8 are the velocities in the direction of the diagonally

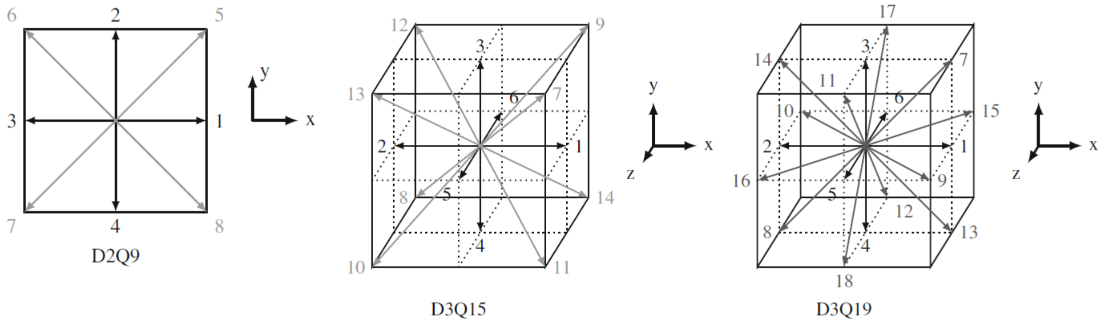


Figure 3.4: Common lattice configurations [36]

i	0	1	2	3	4	5	6	7	8
w_i	$\frac{4}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$
c_{ix}	0	+1	0	−1	0	+1	−1	−1	+1
c_{iy}	0	0	+1	0	−1	+1	+1	−1	−1

Table 3.1: Weight factors for the D2Q9 stencil

adjacent cells, they have a length of $\sqrt{2}$. The magnitude of each velocity is such that particles with that velocity exactly reach the next element in one time step. Because of the different length of each vector, this difference influences the probability of particles having that velocity. This problem is solved by giving each velocity a different weight. The weights for the D2Q9 stencil are given in Table 3.1 These weights are not chosen arbitrarily. They originate from a 4D stencil that is evenly spaced. In this stencil all the velocities are a permutation of (±1, ±1,0,0). Since these velocities all have the same magnitude, their weight factors are all the same. These velocity vectors can be mapped onto a 3D space, and these can again be mapped onto a 2D space. This discretised velocity space with their weight factors are used in the derivation of the equilibrium distribution, which is a crucial part in the Lattice Boltzmann Theory.

In this research, the D3Q15 stencil is used, the weight factors of this stencil is given in Table 3.2

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
w_i	$\frac{1}{3}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$
c_{ix}	0	+1	−1	0	0	0	0	+1	−1	+1	−1	0	0	+1	−1	+1	−1	0	0
c_{iy}	0	0	0	+1	−1	0	0	+1	−1	0	0	+1	−1	−1	+1	0	0	+1	−1
c_{iz}	0	0	0	0	0	+1	−1	0	0	+1	−1	+1	−1	0	0	−1	+1	−1	+1

Table 3.2: Weight factors for the D3Q19 stencil

3.3.2 LBM procedure

Compared to the Lattice Gas models, in the Lattice Boltzmann method, the space and velocity space are discretised as described above. However, instead of the occupation number $n_i(\vec{x},t)$, there is now a discretised particle distribution function $f_i(\vec{x},t)$, where the subscript i indicates the possible velocities c_i. Just like the $f(\vec{x},\vec{\xi},t)$, it represents the mass of the particles with a certain velocity in a certain location. But now both the velocities and location are discretised. And in a very similar way, the mass density and momentum density can be derived.

$$\rho(\vec{x},t) = \sum_i f_i(\vec{x},t), \quad \rho\vec{u}(\vec{x},t) = \sum_i c_i f_i(\vec{x},t) \tag{3.14}$$

The rest of the procedure is also very similar to the Lattice Gas model described above, see Figure 3.5. Once the macroscopic values are known, the equilibrium state can be computed. Based on the current state and the equilibrium distribution, the collision operator $\Omega(f)$ is derived, and the collision step is executed. Now the difference between the Lattice Gas Model becomes apparent. In the LGM each possible velocity in a node could either contain a particle, or not. In LBM each possible velocity in a node contains a percentage of the particles in the node. This percentage moves in one time step to the next node, this is called the streaming step. In the next time step this whole procedure repeats itself.

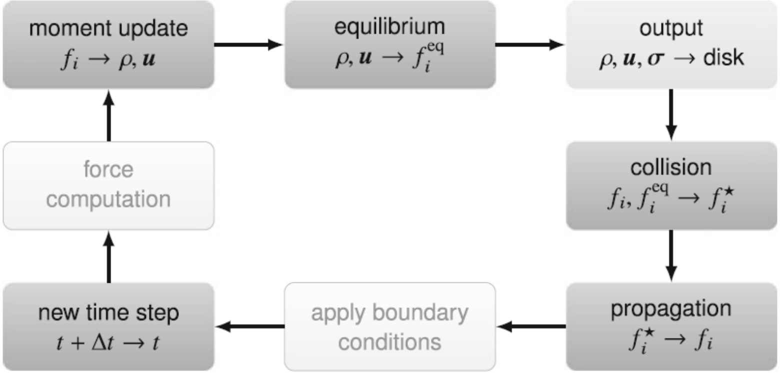

moment update
$f_i \rightarrow \rho, \boldsymbol{u}$
equilibrium
$\rho, \boldsymbol{u} \rightarrow f_i^{\text{eq}}$
output
$\rho, \boldsymbol{u}, \boldsymbol{\sigma} \rightarrow$ disk
force computation
collision
$f_i, f_i^{\text{eq}} \rightarrow f_i^{\star}$
new time step
$t + \Delta t \rightarrow t$
apply boundary conditions
propagation
$f_i^{\star} \rightarrow f_i$

Figure 3.5: Steps in the LBM procedure. Dark grey are necessary steps, light grey are possible output steps [36]

The unknown factor in this story is still the the derivation of the equilibrium distribution f^{eq}. This equilibrium distribution is derived from the Maxwell-Boltzmann distribution, and it is given by[36]:

$$f_i^{\text{eq}}(\vec{x},t) = w_i\rho\left(1 + \frac{\vec{u}\cdot c_i}{c_s^2} + \frac{(\vec{u}\cdot c_i)^2}{2c_s^4} - \frac{\vec{u}\cdot\vec{u}}{2c_s^2}\right) \tag{3.15}$$

Where c_s is the speed of sound, which for all the velocity stencils mentioned in subsection 3.3.1 is equal to $\frac{1}{\sqrt{3}}$. c_i indicates the discretised velocity in direction i, with w_i as the corresponding weight factor. The equilibrium distributions also depend on the local quantities ρ and $\vec{u}$, which are derived using Equation 3.14. Now the derivation of f^{eq} is known, it can be used to express a collision operator.

3.4 Collision operators

As mentioned in section 3.1 **Boltzmann Equation**, the simplest and most common collision operator in the LBM theory is the BGK operator, given by:

$$\Omega(f) = \frac{f_{eq} - f}{\tau} \tag{3.16}$$

Where the current state f moves towards the equilibrium distribution f^{eq} at a rate depending on τ, the relaxation time. A low value for τ means a quick relaxation, which is in line with a low viscosity, and a high value for τ means a slow relaxation which is in line with a high viscosity. $\Omega(f)$ gives the change in f from one time step to the next. This can be written as:

$$f_i\left(x + c_i\Delta t, t + \Delta t\right) = f_i(x,t) + \Omega(f)\Delta t = f_i(x,t) + \frac{\Delta t}{\tau}\left(f_i^{\mathrm{eq}}(x,t) - f_i(x,t)\right) \tag{3.17}$$

Up until now τ is a totally free to choose degree of freedom, however, there are some restrictions for the value of τ. It becomes apparent that different values for $\frac{\Delta t}{\tau}$ give different results. If τ is exactly as large as the time step, f_i relaxes directly to f^{eq}, this is called *full relaxation.* If τ is bigger than the time step, it will take multiple time steps for the state to converge to the equilibrium distribution, this is called *under-relaxation.* If τ is smaller than one time step, f_i will move to the equilibrium distribution faster than one time step this is called *over-relaxation.* This will cause an overshoot with oscillating peaks, which for some cases will eventually move towards f^{eq}. For other cases the peaks will diverge, and the simulation will become unstable. The latter is the case when τ is more than twice as small as Δt. In this case, the collision operator "moves" f_i initially in the direction of the equilibrium distribution. But after less than half a time step, it would have already reached the right value. But the time step does not stop there and f_i moves further, away from f_{eq} for more than half a time step, thus ending up further away from the right solution than before the time step. For the values of τ that are less than twice as small, the amplitude of the peaks decrease, and the simulation is stable.

For accuracy reasons it is wise to keep the value for τ as small as possible [36], while a higher value of τ makes for a faster convergence to the equilibrium state. Sometimes it is desirable to locally tweak the the values of τ to get the best of both worlds. However, the BGK operator does not allow that. In those cases, different operators can be used like the Two Relaxation Time operator (TRT) or the Multi Relaxation Time operator (MRT). In this research, the MRT operator is used.

MRT operators use velocity moments in order to be able to vary the relaxation rate $\omega = \frac{1}{\tau}$. When looking at the BGK operator, it is known that it has to satisfy the following conservation equations.

$$\begin{aligned} 0 &= \sum_i \Omega_i = -\sum_i \omega\left(f_i - f_i^{\mathrm{eq}}\right) = -\omega\left(\rho - \rho^{\mathrm{eq}}\right) \\ 0 &= \sum_i \Omega_i c_i = -\sum \omega\left(f_i c_i - f_i^{\mathrm{eq}} c_i\right) = -\omega\left(j - j^{\mathrm{eq}}\right) \end{aligned} \tag{3.18}$$

Now the density ρ and the momentum j are the zeroth and first moments of velocity. BGK relaxes population distributions all with the same rate. The idea of the MRT operator is that these moments are relaxed all at individual rates. These moments are derived by means of a certain transformation matrix $\mathbf{M}$. For a velocity stencil DxQy, the moment $m_k$can be described as[36]:

$$m_k = \sum_{i=0}^{y-1} M_{ki} f_i \quad \text{for } k = 0, \ldots, y-1 \tag{3.19}$$

It can be written in matrix vector notation:

$$\vec{m} = \mathbf{M} \ \vec{f} \tag{3.20}$$

It is clear that the matrix **M** transforms the population vector $\vec{f}$ into the moment vector $\vec{m}$. The latter will now be relaxed at different rates. The equations with which this is done can be derived from the eqution using the BGK operator, where it is possible to multiply one side of the equation with the identity matrix **I**.

$$\begin{aligned} f\left(x + c_i \Delta t, t + \Delta t\right) - f(x,t) &= M^{-1} M \omega \left[f^{\text{eq}}(x,t) - f(x,t)\right] \Delta t \\ &= M^{-1} \omega \left[M f^{\text{eq}}(x,t) - M f(x,t)\right] \Delta t \\ &= M^{-1} \omega I \left[m^{\text{eq}}(x,t) - m(x,t)\right] \Delta t \\ &= M^{-1} S \left[m^{\text{eq}}(x,t) - m(x,t)\right] \Delta t \end{aligned} \tag{3.21}$$

Where $m^{eq} = M f^{eq}$ and **S** is a diagonal matrix containing all the values of ω. Now the term $\mathbf{S}(m^{eq} - m)$ is very similar to $\omega(f^{eq} - f)$ in the equation with the BGK operator. This expression is the relaxation of the moments. In the case of the BGK operator, all the values for ω in **S** are the same. It is possible to use different rates of relaxation for different moments, this is the idea of the MRT operator. As you can seethis relaxation in the moment space is converted back into population space with the use of M^{-1}.

The procedure for the LBM is still mainly the same, however there are now some extra steps.

1. The macroscopic values are calculated in the usual way.
2. The population distribution f is transformed to moment space:

$$\vec{m} = \mathbf{M}\vec{f} \tag{3.22}$$

3. The moment equilibrium is calculated, this can be done in two ways. The most straightforward way is to calculate the equilibrium distribution and transform that using **M**. A different more efficient way is to calculate m^{eq} from the macroscopic values. This way you do not have to transform the equilibrium distribution each time step which decreases the amount of calculations necessary.
4. Once m^{eq} is known, the collision can take place according to Equation 3.21.
5. After collision, the moments are transformed back to population distributions using:

$$\vec{f^*} = \mathbf{M}^{-1} \vec{m^*} \tag{3.23}$$

6. After that, the streaming step can take place in the usual way.

3.5 Boundary conditions

So far the Lattice Boltzmann Method as explained above, will work as long as each element has neighboring elements in all the directions. Without introducing boundaries, this would mean that the simulated fluid would have to go on until infinity. This would have no practical use, therefor boundaries with their corresponding boundary conditions are introduced. In this section, the two types of boundary conditions used in this research will be elaborated. These boundary conditions are the bounce-back boundary condition and the Dirichlet boundary condition.

3.5.1 Bounce-Back Boundary Condition

The most common boundaries are walls and in-/outlets. Walls make up the structure the flow is going through and the in-/outlets are the places the flow enters or leaves. For a wall boundary condition, there are some requirements. The flow must not be able to go through the wall and there should be a no slip condition. The bounce-back condition obeys these requirements and because it is very simple, it is used very often.

The idea of this boundary condition is quite self explanatory, when particles are about to move towards a node behind a wall, they bounce back to the node they came from. The next time step the velocity of these particles are in the opposite direction.

In Figure 3.6 the multiple steps are elaborated in a bit more detail. Figure 3.6 shows nodes near a wall. From (A) to (B), the streaming step happens as it would at any other position. The particles travel to neighboring cells, but now these cells are part of the wall. After applying the bounce-back condition in image (C), the velocities of these particles are reversed and now point the other way. In the streaming step at (D) the particles stream back to the node they came from. They now have the opposite velocity compared to the situation in (A).

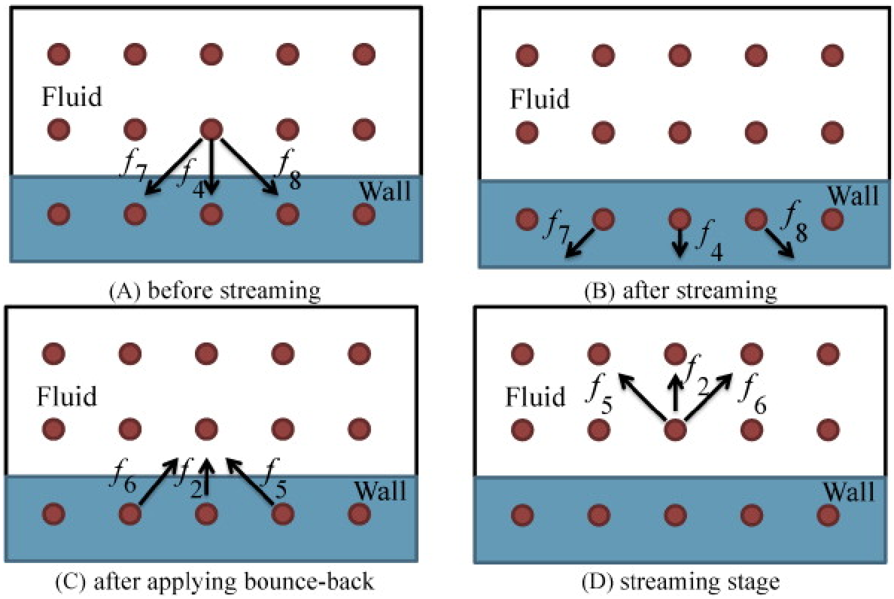


Figure 3.6: Velocity bounce-back boundary condition [40]

3.5.2 Dirichlet Boundary Conditions

The common boundary condition for walls has now been explained. That leaves the boundary condition at the in-/outlets. In these boundaries the fluid enters or leaves the system, so a bounce-back will not be useful there. Instead it is desirable to assign a macroscopic value to these boundaries like a velocity or a pressure, which can be taken from literature in order to make your simulation match reality. The implementation of this boundary condition will be explained using a D2Q9 stencil, but the same principle holds for three-dimensional stencils. Consider a cell like in Figure 3.7 at a lower boundary of a system. The boundary goes through the middle of the cell with directions 2, 5 and 6 pointing into the system, and 4, 7 and 8 pointing out of the system.

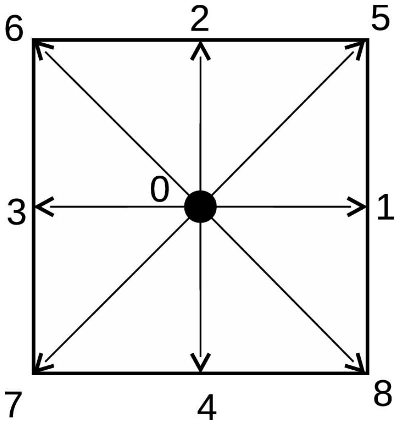


Figure 3.7: A D2Q9 cell

For a given velocity, the unknowns are the particle distributions pointing into the system, f_2 f_5 and f_6, and the macroscopic density ρ. The equation for the density and the momentum in y-direction (the direction in which 2 is pointing) are given below.

$$\rho = f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8 \tag{3.24}$$

$$\rho v = (f_5 + f_2 + f_6 - f_7 - f_4 - f_8)\, c_y \tag{3.25}$$

Where c_y is the y-component of the velocity.
These two equations can be combined by multiplying the first equation with c_y and adding them together.

$$\begin{aligned} \rho c_y + \rho v &= c\,(f_0 + f_1 + f_3 + 2\,(f_2 + f_5 + f_6)) \\ \rightarrow \ \rho &= \frac{c_y}{c_y + v}\,(f_0 + f_1 + f_3 + 2\,(f_2 + f_5 + f_6)) \end{aligned} \tag{3.26}$$

There is an equilibrium over the boundary, since a Dirichlet boundary condition gives a constant value to a macroscopic property at that location. So whatever goes out of the node through direction 2, should come in through direction 4. The same goes for direction 6 and 8, and direction 5 and 7. This gives the following extra conditions:

$$\begin{aligned} f_4 - f_4^{eq} &= f_2 - f_2^{eq} \\ f_7 - f_7^{eq} &= f_5 - f_5^{eq} \\ f_8 - f_8^{eq} &= f_6 - f_6^{eq} \end{aligned} \tag{3.27}$$

Using the known expressions for the equilibrium equations we get:

$$\begin{aligned} f_2 &= f_4 + \frac{2}{3}\rho v \\ f_5 &= f_7 - \frac{1}{2}(f_1 - f_3) + \frac{1}{6}\rho v + \frac{1}{2}\rho u \\ f_6 &= f_8 + \frac{1}{2}(f_1 - f_3) + \frac{1}{6}\rho v - \frac{1}{2}\rho u \end{aligned} \tag{3.28}$$

The expressions in Equation 3.28 together with Equation 3.26 give us 4 equations to constrain the 4 unknowns for a given velocity. Considering the fact that the pressure is given by $p = c_s^2\rho$, the same approach can be used to define a pressure at a Dirichlet boundary condition.

3.6 Why is LBM used in this research?

The Lattice Boltzmann method is, just like conventional CFD methods, a way to simulate fluid flows through geometries. Conventional CFD methods are already widely known and have proven themselves to be applicable in many situations. Furthermore, flow through the exact geometry used in this research has already been done using common CFD methods [30]. One may wonder why the LBM is used on this geometry as well.

One reason has already been mentioned. The Kn number, relating to the ratio of the mean free path to the channel width, in this geometry is high for assuming the fluid to be a continuum. A flow can be assumed a continuum for the order of magnitude $Kn < 0.01$[41] and in the neuropil, the Knudsen number reaches values of 0.031 which might still be in the order of magnitude of 0.01, put it would be the very upper limit.

Another thing the same article mentioned were computational limits. They managed to run a simulation with 84 million elements, but that required some heavy computing power. One big advantage of LBM is how easy it is to parallelise the simulation. The heavy calculating power is mostly needed for the collision step, and that step does not require information from neighboring elements. So for most of the calculations processing cores do not have to share information. This allows for simulations with even more elements spread over many cores on super computers. Further in this research we will see that simulations with hundreds of millions of element are still possible.

Another advantage of LBM is the relative ease of creating the mesh. Since all the nodes lay in a perfect lattice, there is not variation the mesh can have. Especially in complex geometries like the neuropil, meshing can take up a lot of time when using conventional CFD methods. In the Lattice Boltzmann method a mesh in such a geometry can be created in within an hour.

Chapter 4

Simulation Framework

In the previous chapter the idea and calculations behind the Lattice Boltzmann method are elaborated. Although knowledge of this method is very important to understand what happens inside the simulations, the calculations in this research are done by software from a software package called APES. It is a bundle of tools to simulate flow build around an octree mesh infrastructure [42]. These are tools capable of running simulations on large parallel computing facilities or supercomputers. In Figure 4.1, all the available software from this package can be seen. In this research, the meshing tool Seeder will be used to create the mesh. The Lattice Boltzmann simulation package called Musubi will be used to apply the flow calculations on this mesh. Harvester will be used to make the data available for post-processing. This post-processing will be done using Paraview and Matlab. Seeder creates a evenly spaced mesh, in order to obtain sufficient accuracy, the grid size should be small on important places. This causes the mesh to be fine over the entire volume. For such a mesh and the simulation that is applied to it, a lot of computing power is required. The big advantage of LBM is the great applicability of parallel computing, which makes it possible to divide the required computational power over many processing cores. In this thesis, the Dutch national supercomputers Cartesius and Snellius were used. All the packages mentioned above will be further elaborated in this chapter.

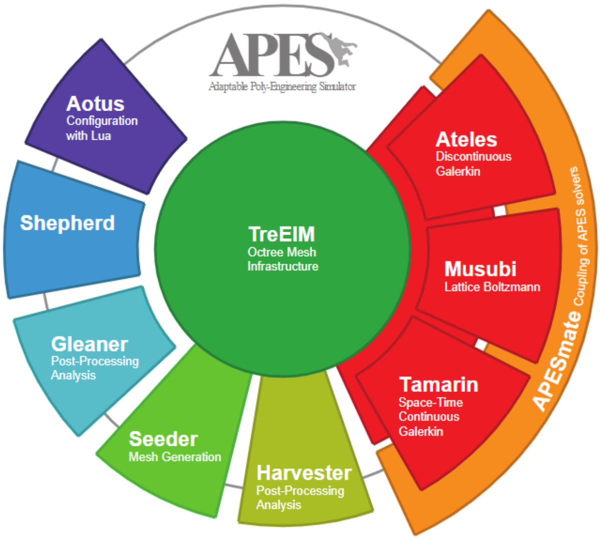


Figure 4.1: Tools provided by APES

4.1 Seeder

Seeder is the program that is used to create the mesh. As mentioned before, the mesh it creates is an evenly spaced octree mesh. Seeder starts with a bounding cube with length L_{BC}, on the zeroth level, the mesh consist of just this cube. This mesh can be refined by splitting this cube into eight smaller cubes, all with length $L_{BC}/2$, hence the name octree. This is the first refinement level. These refining steps are preformed until the desired element size is achieved. The amount of times the cube is split in eight smaller cubes is called the refinement level. For each refinement level the amount of elements in the bounding cube is multiplied by 2^3.

The result of this refinement is just a bounding cube containing a lot of smaller cubes. This is not yet the mesh. In order to create the mesh of the geometry, the bounding cube is placed around the geometry. It is important that the geometry fits entirely inside the bounding cube. Therefor L_{BC} should be chosen such that it is larger than the largest length of the geometry. Also in order to make the smaller cubes exactly as large as your desired Δx, L_{BC} should be the desired Δx times a power of 2. Now that the bonding cube covers the entire geometry, all the elements that are entirely inside the geometry will be "flooded", meaning that they will be part of the mesh. The elements that are outside or partly outside of the geometry will not be part of the mesh. On flat surfaces, the square elements can cover the geometry very well. However, curved surfaces are herd to approach especially when the elements are quite large. Therefor, it is important that the elements are sufficiently small. This is a drawback of this type of mesh, it is not possible to only have a fine mesh on certain location. If a fine mesh is necessary on one location, the entire mesh will be fine.

The input parameters Seeder requires are the length of the bounding cube and the element size dx, from that the refinement level can be calculated. It also needs the shapes and sizes and locations of the boundaries. Simple shapes like planes can be defined in Seeder itself, but it is also possible to load an STL file for more complex geometries. Furthermore it requires the seed origin, that is the location where the flooding starts. This way, Seeder knows what is the inside of the geometry and what is the outside.

4.2 Musubi

When the mesh is created the LBM simulations can be done on it. In this thesis this is done by Musubi. Musubi follows the LBM theory described in chapter 3. In this section some important options will be discussed as well as the necessary input parameters and their influence on the results.

First of all Musubi requires the physical flow properties, like the density and viscosity as well as the boundary conditions e.g. an inlet and outlet velocity or pressure. From the density and viscosity, the kinematic viscosity can be derived, as shown in Equation 4.1. These are properties determined by the actual flow you want to simulate, so generally speaking there is not much to change about these values in order to tweak the simulation.

$$\nu = \frac{\mu}{\rho} \tag{4.1}$$

As mentioned before, viscosity on a micro regime is not really a thing. It is a consequence of particles interacting and it is closely related to the relaxation rate ω and to the time step Δt and the spatial step Δx. This relation is shown in Equation 4.2 and it ensures that the results of the LBM simulations follow the macroscopic physics described by the Navier-Stokes equation.

$$\nu = \frac{\frac{1}{\omega} - \frac{1}{2}}{3} \frac{\Delta x^2}{\Delta t} \tag{4.2}$$

Since Δx is known from the mesh and the viscosity is a physical parameter, the time step follows from this equation. There is still a possibility to tweak it a little bit by changing the relaxation rate. The relaxation rate can not get to big. Due to stability reasons mentioned in section 3.4, it must stay below 2. This also becomes clear when looking at Equation 4.2, if ω is larger than 2, the viscosity will have a negative value. On the other hand, for accuracy reasons, it is preferable to keep omega as high as possible[36]. That is why ω is often chosen to be close to 2. This high relaxation rate gives the impression that the viscosity is very low, however, the physical viscosity did not change. The apparent viscosity, or lattice viscosity changed. A clear distinction must be made between these physical properties and their lattice variant. Lattice properties are in terms of "amount of time steps" and "amount of spatial steps" instead of seconds and meters. The relevant lattice properties are the lattice viscosity and the lattice velocity of the flow. The relations for these properties with their physical counterpart is given in

$$\nu_L = \nu \frac{\Delta t}{\Delta x^2} = \frac{\frac{1}{\omega} - \frac{1}{2}}{3}, \quad u_L = u \frac{\Delta t}{\Delta x} \tag{4.3}$$

The lattice viscosity is relevant, because it indicates the time needed for relaxation. Just like the relaxation rate which is logical, since ω is the only variable in it. The lattice velocity must not be too high. For the assumption of compressible flow, the Mach number should not be too high. The macroscopic velocities of the interstitial fluid are very low, so on this scale it is not a problem. However, in the mesh the lattice speed of sound is important, the mesh and velocity stencil in this research have $c_s^2 \frac{1}{\sqrt{3}}$. Incompressibility is defined as a maximum density change of 5%, which happens at a Mach number of 0.3. Therefor the lattice velocity should not be bigger than 0.1.

Next to the parameters mentioned above, Musubi requires some more information, like the amount of iterations it should run, whether the flow is incompressible, the collision operator, the type of velocity set and the boundary conditions. This is enough to run the simulation, but it does not store any information yet. It is possible in Musubi to track properties in points, lines planes and volumes to analyse the data.

Chapter 5

Methodology and Results

In this chapter the approach of this research will be elaborated, encountered problems as well as the solutions for these problems will be explained. The results of the simulations will be shown in this chapter.

5.1 Pipe Flow Simulations

This research starts with a preliminary study on the flow in a pipe. This has multiple reasons. First of all, simulations on a simple geometry will help getting familiar with the software and the effect of changes in the parameters. Additionally, a mesh convergence study on the neuropil geometry would be difficult, since it is not possible to compare the results to experimental results. It is partially possible, when looking if the pressure drop of average velocity are comparable to values found in literature. However, this does not give a clear view when the results converge to the actual value on each location. Mesh convergence on a pipe flow also does not give a definitive answer to when the mesh is converged in the neuropil. However, looking at the convergence when increasing the amount of elements in radial direction would give a compelling argument as to when a flow in the neuropil would be converged. Finally the pipe flow is used test a theory that is used for increasing the time step, this is further elaborated in section 5.2 **The Viscosity Problem**.

The flow through a cylindrical pipe has been studied for many years. Analytical solutions for the velocity profile, pressure and shear stresses are available for fully developed, laminar flow can be found in literature. So for a convergence study, the error can be calculated exactly. The terms "laminar" and "fully developed" are key aspects in this test. In order for the flow to be laminar, the Reynolds number should be below 2300. The Reynolds number is given by Equation 5.1.

$$Re = \frac{Du}{\nu} \tag{5.1}$$

Where D is the diameter of the pipe, u is the mean velocity of the flow, and ν is the kinematic viscosity. Substitution of the values for this test case give a Reynolds number of 100. At the inlet of the pipe, a constant velocity is defined as the boundary condition. This velocity profile has to develop over a certain length. This length is called the entrance length and it is given by Equation 5.2[43]

$$L_{h,\text{ laminar}} = 0.0575\,\mathrm{Re}_D\,D \tag{5.2}$$

With a Reynolds number of 100, and a diameter of 1 cm, the entrance length is 5.75 cm. From that point on, the flow can be considered as fully developed.

In this developed region, certain flow properties can be calculated analytically. The pressure drop is given by

$$\Delta p = \frac{8\mu L Q}{\pi R^4} = \frac{8\mu L u_{mean}\pi R^2}{\pi R^4} = \frac{8\mu L u_{mean}}{R^2} \tag{5.3}$$

In the cylindrical coordinates x, r and θ, the velocity profile is independent of x and θ, because the flow is developed and does not change in x direction anymore, and there is a rotational symmetry. Furthermore, the flow is steady, so it is also independent of time. Finally it is assumed that the velocity in r and θ direction are equal to 0. With these assumptions the axial momentum equation simplifies to Equation 5.4.

$$\frac{1}{\mu}\frac{\partial p}{\partial x} = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) \tag{5.4}$$

Since the left side only depends on x and the right side only depends on y, both sides must be constants. When looking at the pressure drop over a length L, the left hand side of the equation becomes $\frac{\Delta p}{\mu L}$. This boils down to:

$$u = \frac{\Delta p r^2}{4\mu L} + c_1 \ln r + c_2 \tag{5.5}$$

Using the fact that u is finite for $r = 0$ gives $c_1 = 0$. The second constant can be solved using $u = 0$ at $r = R$. this gives the expression for the velocity profile.

$$u = \frac{\Delta p(R^2 - r^2)}{4\mu L} \tag{5.6}$$

The maximum velocity can be found by substituting Equation 5.3 into Equation 5.6 on $r = 0$:

$$u_{max} = 2u_{mean} \tag{5.7}$$

For the test case the following parameters are used: The properties in the table are approximately the properties of water. It is not necessary for the test case to mimic a specific fluid, but by using the approximate values for water, it is certain that the test case will at least have realistic values. At the entire entrance of the pipe, the velocity is $10 mm/s$. This is the chosen boundary condition for the inlet which has as a result that the mean velocity in the pipe is also 10 mm/s. At the outlet, the pressure is set to 0, meaning that the pressure drop from any point in the pipe until the end of the pipe would be equal to the pressure at that location. At the walls the bounce-back boundary condition is applied, and the relaxation rate ω that is used is 1.94.

L	100 mm	D	10 mm
u_{mean}	10 mm/s	μ	$1.0 \cdot 10^{-6}$ $kg/mm/s$
ρ	$1.0 \cdot 10^{-6}$ kg/m^3	ν	1 mm^2/s
Re	100	p_{outlet}	0 Pa

Table 5.1: Flow properties of the pipe flow test case

A tracker is placed in the middle of the pipe near the end at 90 mm from the start. This is further than the entrance length, so the flow will be developed there. Since it is in the middle of the pipe, u_{max} is evaluated, the analytical solution for u_{max} is 20 mm/s. For the pressure drop, the analytical result is calculated by taking $L = 10$ mm because the tracker is 10 mm from the end. This results in a pressure drop of $3.2\dot{1}0^{-5}$ $kg/mm/s^2$.

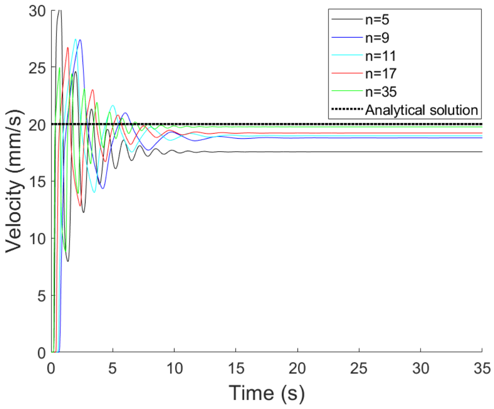


Figure 5.1: Velocity in x direction for various grid sizes

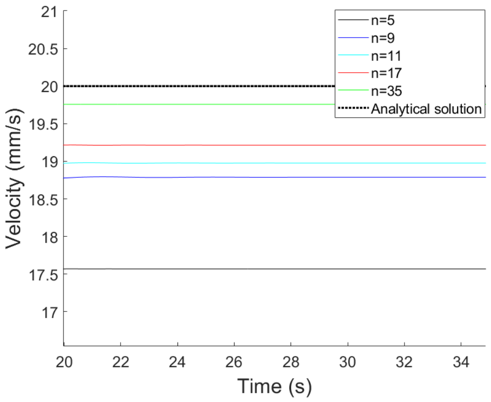


Figure 5.2: Zoomed in on relaxed section

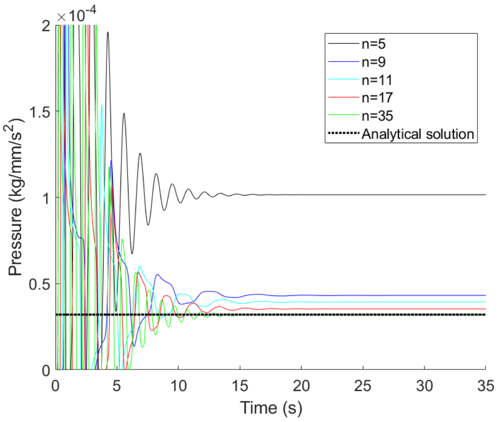


Figure 5.3: Pressure for various grid sizes

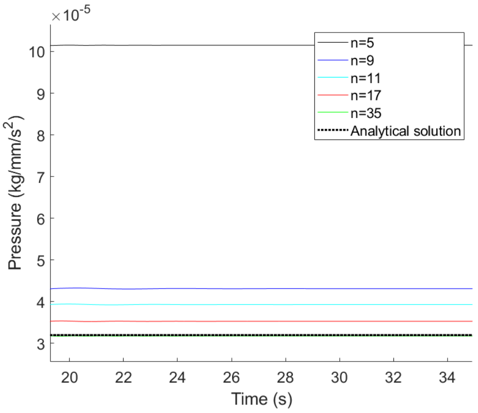


Figure 5.4: Zoomed in on relaxed section

A grid refinement study will show when a sufficiently fine grid is reached. The element size would be an interesting parameter to look at since that tells something about how fine the grid is. However, on different scales, different element sizes are needed for the same accuracy. Since the neuropil is on the scale of nanometers, not the element size is of interest here, but the amount of elements in radial direction. In Figure 5.1 to 5.4, the velocities and pressures at the location of the tracker can be seen for various grid sizes. The amount of elements in radial direction is given by the number n. The question now arises ”how accurate is sufficiently accurate?”. In order to answer that question, the type of results are of interest. The main question of this research is whether the transport of amyloid β is mainly bulk flow driven, or diffusion driven. For that, the

Péclet number of the flow is evaluated. The Péclet number gives the ratio between advective transport rate and diffusive transport rate. Or when put differently, it gives the ratio between the diffusion time and the advection time. This is the time it takes a solute particle to travel a certain distance L purely driven by either diffusion or advection respectively.

$$t_{Diff} = \frac{L^2}{D}, \quad t_{Adv} = \frac{L}{u}, \quad \rightarrow Pe = \frac{t_{Diff}}{t_{Adv}} = \frac{Lu}{D} \tag{5.8}$$

When $Pe << 1$, the flow is diffusion driven, and when $Pe >> 1$ the flow is advection driven. Since Both the diffusion coefficient and the distance are no results from the simulation, but taken from literature, only the u is influenced by the simulation results. One could say that even the simulation with 5 elements is already accurate enough, since the relative error is around 0.125. This relative error would not significantly change the Péclet number. However, the error in the pressure drop is significant, which may lead to higher errors in different geometries. Furthermore, since the inlet boundary condition is a velocity, the flow rate through the pipe, even with a few elements, is expected to match the solution more closely. The pressure drop is a result of the calculations and is a better measure of how well the simulation went. Therefore, 5 elements in radial direction is not enough for a sufficient simulation. The error quickly decreases from there. In this research, keeping in mind that the simulations will be very computationally heavy, a minimum of 10 elements is taken as sufficient. More elements is preferable as long as computational limits allow it.

5.2 The Time Step Problem

In the Lattice Boltzmann Method, the spatial step, time step and viscosity are linked together with the following relation:

$$\nu_{lattice} = \nu_{physical} \frac{\Delta t}{\Delta x^2} \tag{5.9}$$

The small spatial step necessary to work in such a microscopic geometry as the neuropil, causes the time stem to be even smaller. If Δx is decreased by a factor x, then Δt decreases with a factor x^2. In the case of the neuropil, Δx is in the order of nanometers, when keeping the viscosity the same, Δt then becomes of the order of magnitude $10^{-14} s$. This would mean that a hundred thousand billion time steps are necessary to simulate about one second. This would make it impossible to let the simulation relax to the correct solution. A possible workaround would be to change the viscosity by a factor of 10^{-8}. This changes the time step to a much more useful value, it is then in the order of magnitude 10^{-6}, meaning only a million time steps are needed to simulate one second.

The velocity profile is believed to be the same. The flow would stay laminar, since the Reynolds numbers in the flow of the brain are extremely low. They lay between $2 \cdot 10^{-9} - 1.6 \cdot 10^{-8}$, so increasing it by 10^8 would still leave the Reynolds number to be far below the critical value of 2300.

The interstitial fluid is often assumed to be a Newtonian fluid [29], which is defined by:

$$\tau = \mu \frac{du}{dy} \tag{5.10}$$

τ varies linearly over the width of a channel, y, so the the velocity profile is parabolic in the y direction. Since there is also a no slip condition at the boundaries, the velocity profile is believed to stay the same since the same amount of fluid will go through (due to the boundary condition of a constant volume rate).

This velocity profile could already be used for obtaining the Péclet number. In order to check whether this workaround is actually valid, the idea is to let the flow develop with a lower viscosity, then use the velocity profile of this simulation as an initial condition for a simulation with the right viscosity. If the velocity profile stays the same after some time this method is assumed to be valid.

In Musubi it is possible to write a restart file at the end of a simulation. With such a restart file it is possible to start a simulation where the previous simulation ended. The restart file stores the particle distribution functions in each element, so the next time step, it can start directly with step 1 from the procedure described in section 3.4. This, however lead to an unexpected result. Figure 5.5 and 5.6 show the x velocity measured by a tracker at a certain location in the neuropil before and right after the viscosity change.

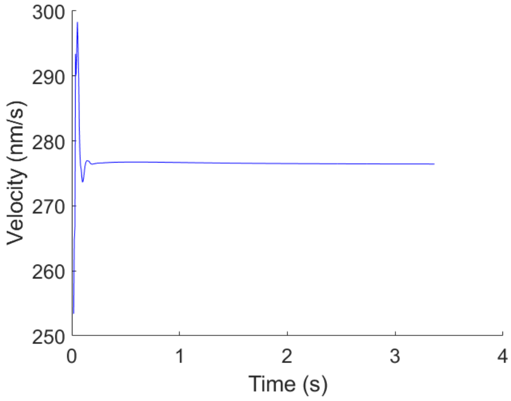


Figure 5.5: Velocity before viscosity change

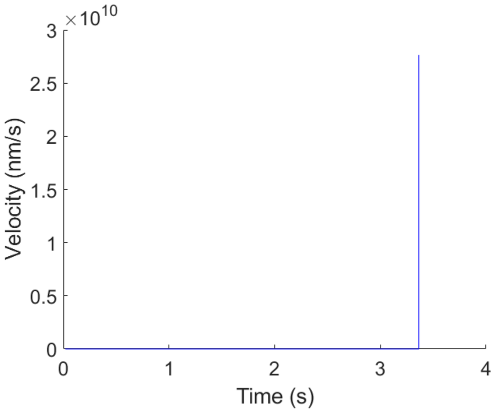
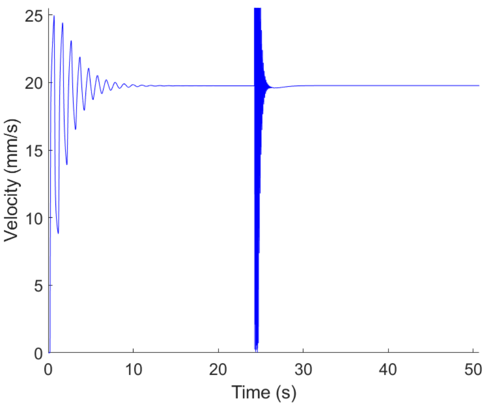


Figure 5.6: Velocity after viscosity change

The velocity increased exactly with a factor 10^8. This can be explained by taking a look at how the velocity sets are defined in LBM. The magnitude of each velocity c_i is such that particles with that velocity exactly reache the next element in one time step. Furthermore, the time step is implicitly determined using the spatial step and the viscosity. This time step just decreased with a factor 10^8. This means that the velocities c_i increased with the same factor, because the same distance has to be travelled in a much smaller time.
By applying this sudden viscosity change, the fluid is accelerated. This results in the fact that the flow now has to relax to the right solution again, but now the particle distribution is even further away from the equilibrium distribution. In this research the velocity profile from the simulations with the wrong viscosity are assumed to be representative for the situation with the right viscosity. This cannot be validated because of the reasons mentioned above.

It would still be interesting to know if the simulation framework is able to deal with such a sudden viscosity change. The initial change is known and can be explained, but whether the solution will relax to the correct answer with the new viscosity is interesting to test. For this purpose the pipe flow from before is considered again. The case with 35 elements in radial direction is used since for this simple geometry, computational power is not an issue. After the relaxation, the viscosity is increased with a factor 10. After this increase, the simulation continues until the solution relaxes to a certain value. The results can be seen in Figure 5.7 and 5.8.



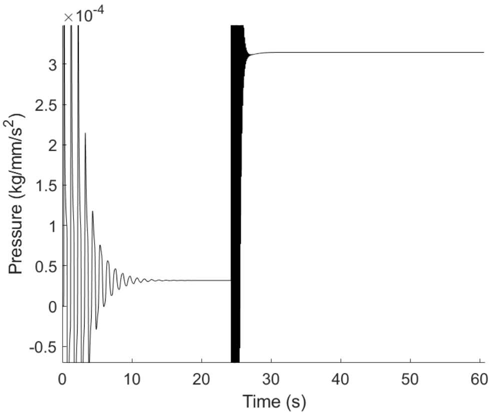


Figure 5.7: The effect of viscosity change in LBM on velocity

Figure 5.8: The effect of viscosity change in LBM on pressure

Since the inlet boundary condition is a flow rate, it is not surprising that the velocity is the same as before the viscosity change. The pressure initially increased with a factor 100. This is because both the viscosity and the velocity increased with a factor 10. But when the velocity decreased to the correct value, the pressure ended up being a factor 10 higher as before. Looking at Equation 5.3, this is in line with the theory.

To recapitulate, this chapter looks into whether it is possible to let the flow develop with a lower viscosity, and then, change the viscosity to the correct value. The extremely small time step with the right viscosity makes it impossible for the flow to develop in an acceptable time. On this pipe flow study it was tested whether using this method would speed up the relaxation of the flow. However, as it turned out the flow has to relax again after changing the viscosity, so this workaround is not useful for this research. It might however be useful to know that changing the viscosity half way through a simulation is supported by the APES framework.

5.3 Neuropil flow simulations

Now that the Lattice Boltzmann Method is tested on a simple geometry, the minimum amount of cross sectional elements in a channel is determined and a solution for the small time step has been found, the simulations on the neuropil can be preformed. The neuropil is brain tissue from the brains parenchyma. It is a densely packed tissue with brain cells surrounded by interstitial fluid. The geometry used in this research is the same as the geometry used in a comparable study [30]. This study used a publicly available reconstruction of the neuropil based on electron microscope scans of the neuropil in the hippocampus of a rat. The result of this reconstruction can be seen inFigure 5.9. Views from different angles can be seen in Appendix A.

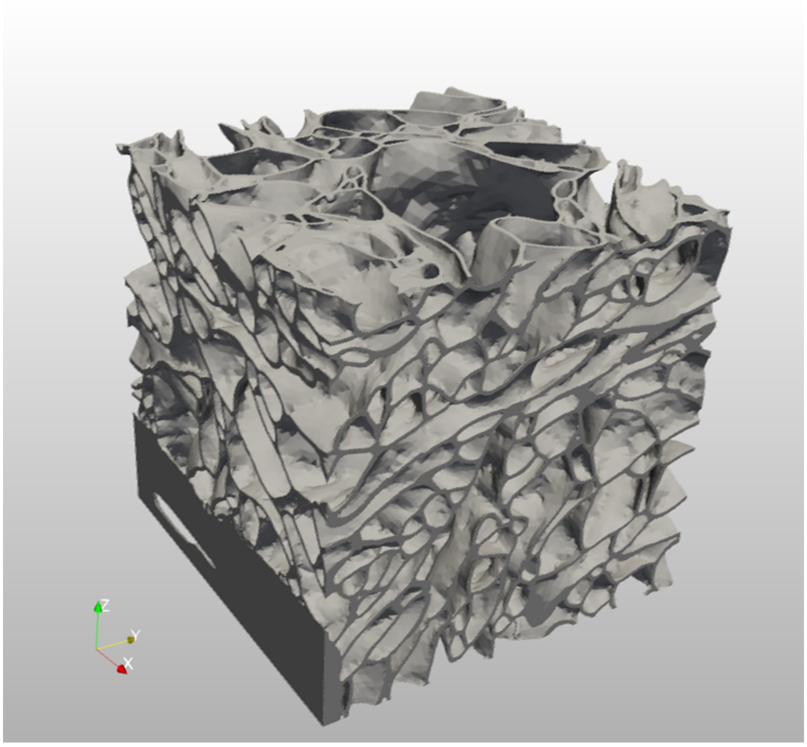


Figure 5.9: EM scan of a part of the neuropil

The geometry has dimensions of around $4x4x4$ μm. It is centered on a large apical dendrite of a CA1 neuron [44], which is located in the hippocampus. In this research the focus will be on the flow of ISF around the brain cells. The hollow parts that are visible in Figure 5.9 are the brain cells, so the flow will go through the seemingly solid parts in the pictures. The volume fraction of this scan, i.e. the volume of the space around the cells divided by the total volume, is 20%[30]. This is in line with an asleep state[29].

As can be seen, the geometry has a solid slab of material on the left bottom. This will create a wide tunnel in the mesh through which the fluid can flow much more easily than through the rest of the geometry. This is probably a mistake made during the electron microscope scan or the post processing thereof. This is a problem since it will influence the results. To solve this, a slice including this slab will be cut off.

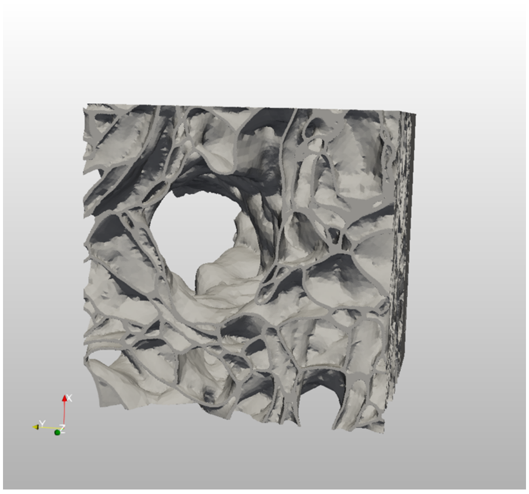


Figure 5.10: Hole in the neuropil geometry

Another noteworthy aspect of this geometry is a large hole going through the entire cube. This hole is about 2 μm in diameter and it is probably the result of an axon going through the tissue[44]. An axon is a long part of a neuron cell that sends away information. They have a typical diameter of around 1 μm, but a diameter of 2 μm is not uncommon[45]. There is no way to remove this hole without also removing half the geometry. While this axon is naturally there, unlike the slab, it does influence the results, and this should be kept in mind. Because the flow has to go around the axon, and there is no option for the flow to divert to the space next to the cube, all the fluid is pushed through a narrower space than normal. This will probably locally cause higher velocities and pressure drops. This has to be kept in mind when examining the results.

In this research point trackers are placed to track flow properties over time, Also 9 planes are evaluated at the end of the simulation, three in each direction. This is because evaluating the entire geometry would require too much memory. Figure 5.11 shows the plane on which the trackers are located. Figure 5.12 shows the locations of the trackers indicated by a red dot. The side of the flow inlet is indicated with the red line, the fluid flows in positive x-direction (to the right). Figure 5.13 shows the planes on which the flow is visualised in red.

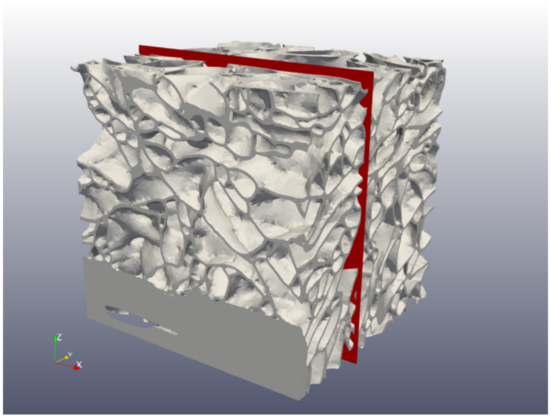


Figure 5.11: The trackers are located on the red plane

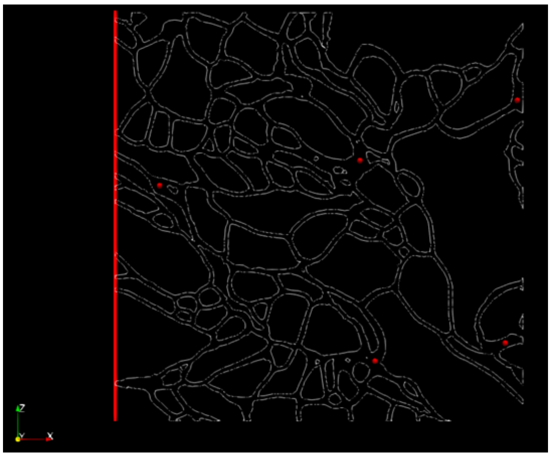


Figure 5.12: The locations of the trackers are indicated with a red dot. The side of the flow inlet is indicated with a red line.

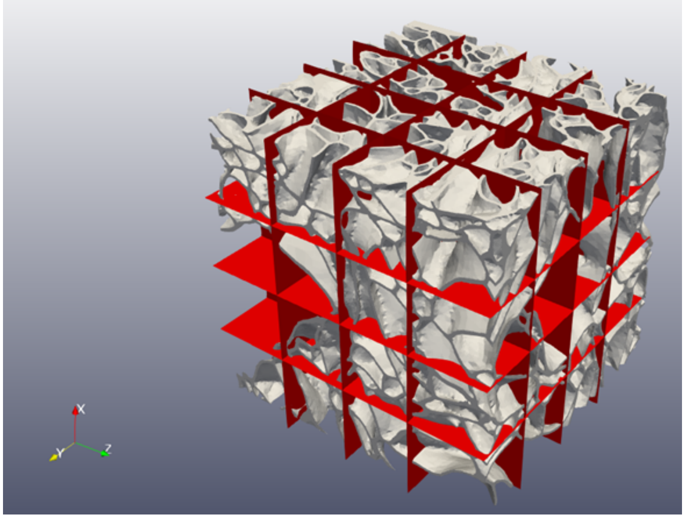


Figure 5.13: An overview of the location of the planes that the results are based on

5.3.1 Mesh Convergence

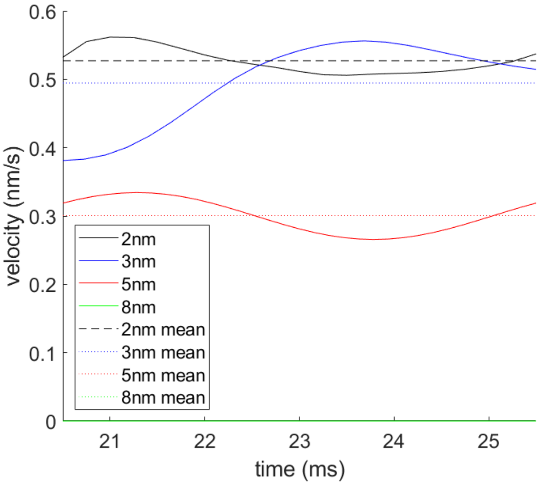


Figure 5.14: Velocities for different mesh sizes

In order to save time and computational power, the mesh convergence study is done on a smaller cube with the dimensions of $1x1x1\mu m$. For this convergence study, a constant flow is sent through this smaller cube. In the pipe flow study it was possible to choose a point that would not shift when the mesh size changed. For an odd number of elements in radial direction, the middle element would always have its center in the middle. For this geometry it is not possible to find such a point. Therefore, a cube of $10x10x10nm$ is placed in the geometry. An average over all the elements in this cube is taken to make up for the shifting of elements for different mesh sizes. The velocity and pressure for different mesh sizes can be seen in Figure 5.14 and 5.15. Table 5.2 shows more information on the simulations done for the convergence.

$N_{elements}$	dx $[nm]$	dt $[s]$	compute time $[s]$ (including meshing)	number of cores used	u_{mean} $[nm/s]$	p_{mean} $[GPa]$
± 330,000	8	$4.1 \cdot 10^{-5}$	10.93 (26.54)	1200	0	0
± 1.5 million	5	$1.6 \cdot 10^{-5}$	14.43 (87.62)	1200	0.301	$0.60 \cdot 10^{-5}$
± 8.5 million	3	$5.8 \cdot 10^{-6}$	18.76 (140.78)	1200	0.495	$1.05 \cdot 10^{-5}$
± 27 million	2	$2.6 \cdot 10^{-6}$	21.45 (443.83)	1200	0.527	$1.12 \cdot 10^{-5}$

Table 5.2: Convergence values on the smaller cube

It is directly visible that a mesh with an element size of 8 *nm* does not give good results. For the other three element sizes, the solutions still seems to be relaxing towards their final value. However, the values oscillate around a certain average value. It can be seen that the values for the 2 *nm* mesh and the 3 *nm* mesh are quite close together. The value for the 5 *nm* mesh differs by approximately a factor 2. The first two meshes are both considered as possible mesh sized based on this test. A visual inspection of the mesh showed that on narrow channels the 3 *nm* mesh had around 9-11 elements and the 2 *nm* mesh had around 15-18 elements. When following the guidelines from the mesh convergence on the pipe flow, both would be sufficient, but of course the finer mesh would be preferable, if it is computationally feasible.

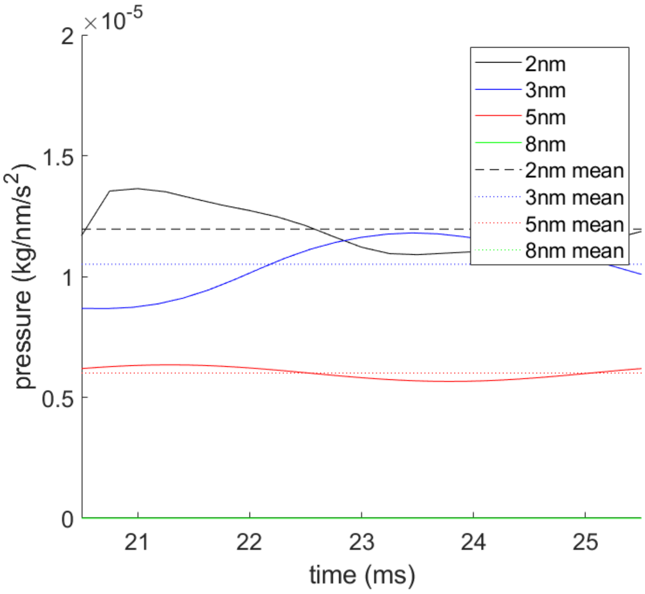


Figure 5.15: Pressures for different mesh sizes

To check the feasibility, the total amount of elements is of interest. The 2 *nm* mesh has around 1.3 billion elements. Which is not impossible, but it is very costly, Both in time and in CPU hours. The 3 *nm* mesh has around 385 million elements which is much more feasible. When going to a coarser mesh in such a complex structure with narrow channels, it is possible that parts of the geometry will not be flooded anymore. These numbers are reassuring, when refining with a factor of 1.5, the number of elements will increase with roughly a factor $1.5^3 = 3.375$. When multiplying 385 million with this factor, you get 1.299 billion. From this it can be concluded that no parts of the mesh are omitted in this coarser mesh. To check whether the geometry is filled well, visual inspections were done on several slices of the mesh. One of these slices can be seen in Figure 5.16. At the bottom an example can be seen of what it would look like if a part is not filled. There would be white lines from the original STL file of the EM scan without the blue of the mesh. The empty part at the bottom is due to the cutting of the slab mentioned before. These inspections were done on nine planes, three in each direction. All of these slices were filled correctly. One other thing mentioned before, is that in curved geometries, the mesh should be fine enough to represent the curves. When looking at an especially curvy part of Figure 5.16, shown in Figure 5.17, it can be seen that the curves are filled well and that individual elements are hardly visible.

These observations, together with the conclusion that the 3 nm mesh scored adequately regarding its results lead to the conclusion that this mesh will be used for the bulk of the simulations. If time and CPU budget allows for it, the 2 nm mesh will be used for a simulation to compare the results.

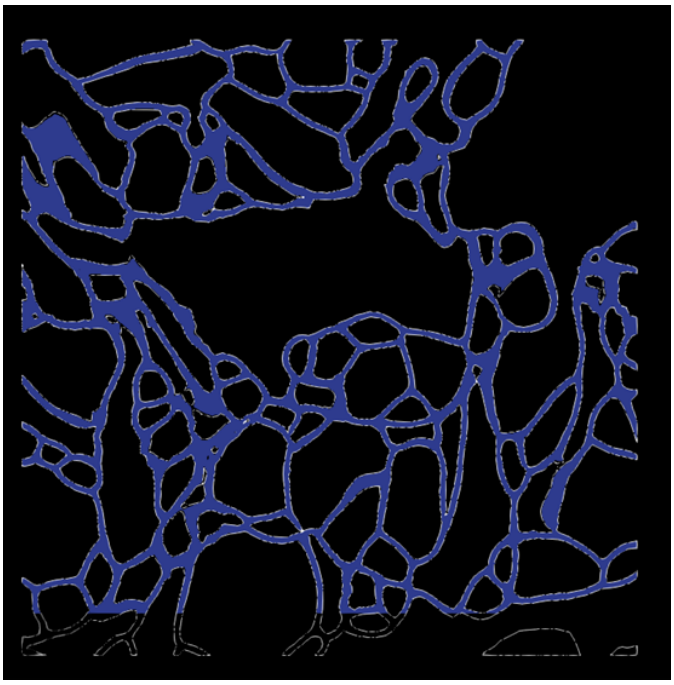


Figure 5.16: Blue color shows the volume mesh with fluid cells whereas the white lines mark the boundaries of the geometries that are filled with aforementioned fluid cells.

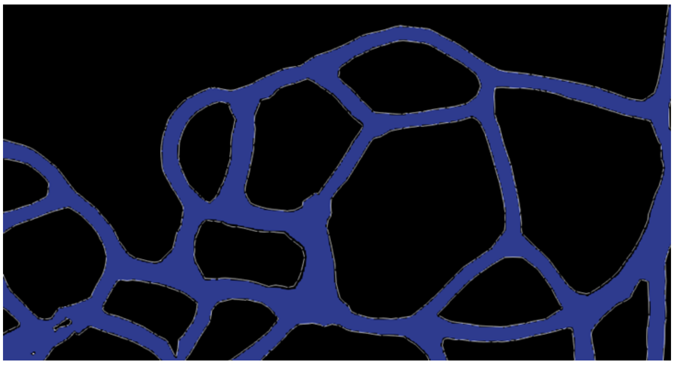


Figure 5.17: Curved part of the slice shown above

5.3.2 Simulation 1: Steady flow with average ISF velocity for a sleeping person

Now that the mesh is complete, the flow simulations can be preformed. The cube will be treated as a neuropil cube somewhere in the middle of an artery-vein pair, meaning that the bulk flow will go from one side to the opposite side. The side of the inlet will have a velocity boundary condition and the outlet will have pressure of 0 Pa. The rest of the sides, as well as the walls of the neuropil will have the bounce-back boundary condition.

In the first simulation, a steady flow will be considered. In reality, the flow in the interstitial space has a pulsatile nature, however, the exact shape of this pulse is not known. The steady state will give an averaged view of the situation. Later on a simple sine is sent through the neuropil to see the effect of a pulsatile flow.

Inlet velocity	166 nm/s	Outlet pressure	0 Pa
Element size	3 nm	Number of elements	$385 \cdot 10^6$
Dynamic viscosity μ	$8 \cdot 10^{-12}$ $Pa \cdot s$	Density ρ	1000 kg/m^3
Kinematic viscosity ν	$8 \cdot 10^{-15}$ m^2/s	Relaxation rate ω	1.94
Time step	$5.8 \cdot 10^{-6} s$		

Table 5.3: Caption

The flow values for the first simulation can be found in Table 5.3. The inlet velocity is based on the average flow velocity of a sleeping person of 10 $\mu m/min$[25]. The dynamic viscosity is taken from the previous research that worked with the same geometry [30]. However, this viscosity has been decreased with a factor 10^{-8} to get a workable time step. If this was not done, the time step would have been $5.8 \cdot 10^{-14} s$.

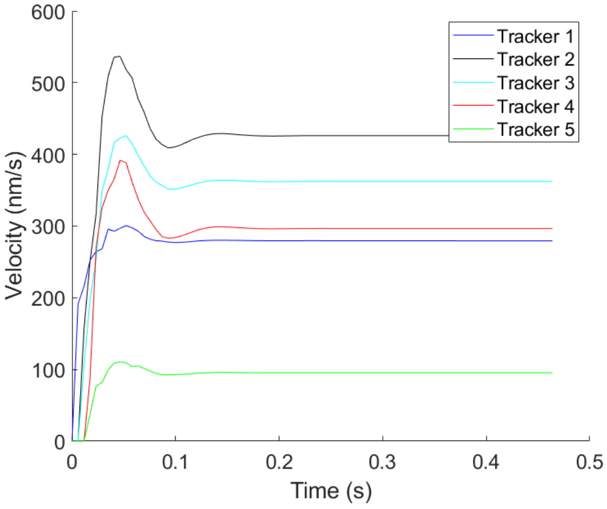


Figure 5.18: Velocities of 5 different trackers

The simulation must be run until a steady state is reached. Five trackers were placed on different locations in the neuropil, ranging from close to the inlet to close to the outlet, to make sure the solution has reached a constant value in time over the entire geometry. The location of these trackers can be seen in Figure 5.12. As can be seen in Figure 5.18, the solution has converged to a constant value for each velocity after about 0.25 s, which is after about 43.000 time steps. The fact that the solution has been relaxed means that the values for velocity and pressure at the end of this simulation can be used for the results.

For the visualisation of the results, it was not possible to inspect the entire mesh. The memory required to do that was too much. Therefore, again 3 slices in each direction are considered. The velocity profile and the pressure in one of these slices can be seen in Figure 5.19 and 5.20 respectively. The velocities are given in nm/s and the pressures are given in $kg/nm/s^2$ or GPa. The velocity profile looks logical. In the narrow channels and in the channels in the z-direction, the velocity is lower than in the wider sections. The average velocity over all the slices is 160.8 nm/s which is close to the average flow velocity 166 nm/s. A minor difference is to be expected, because the cross-section is not everywhere the same as at the entrance. A closer inspection on all the slices showed that the flow is laminar on every point. So the viscosity change in order to increase the time step is therefore considered to be justified.

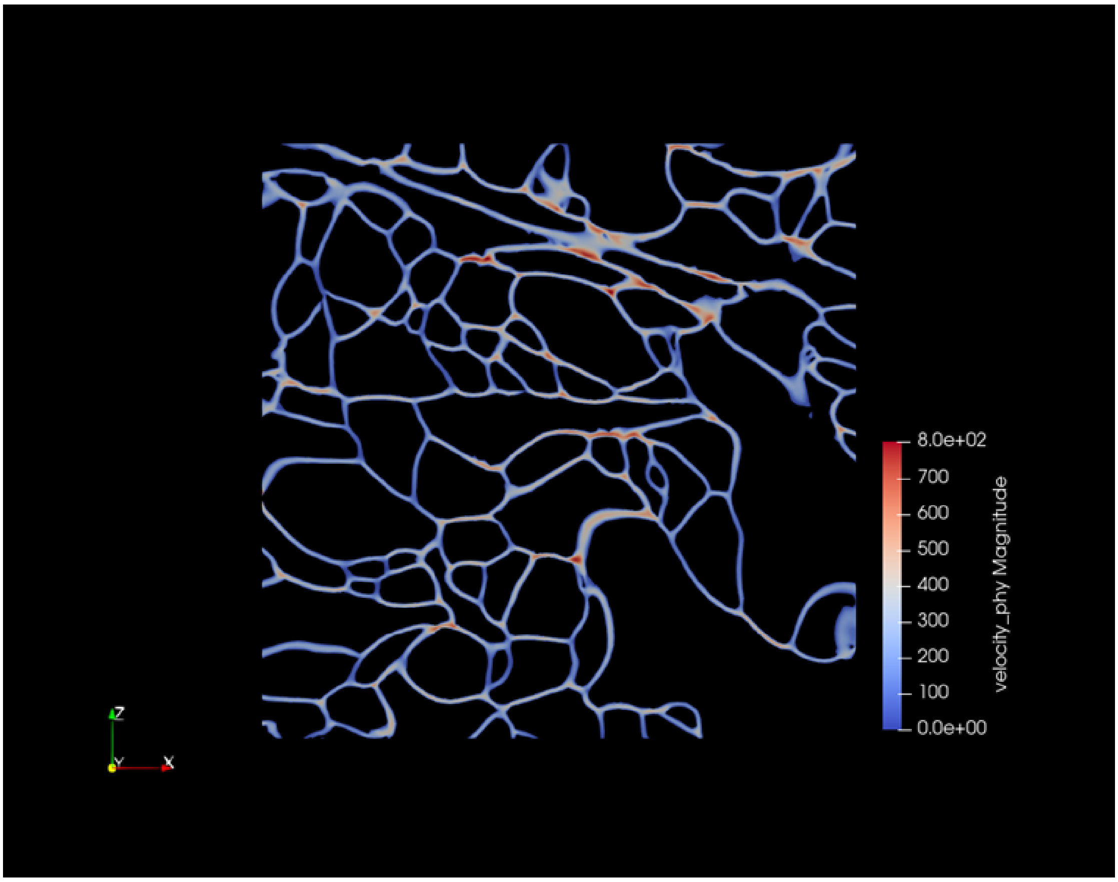


Figure 5.19: Velocity profile in one of the neuropil slices. Velocities are given in nm/s

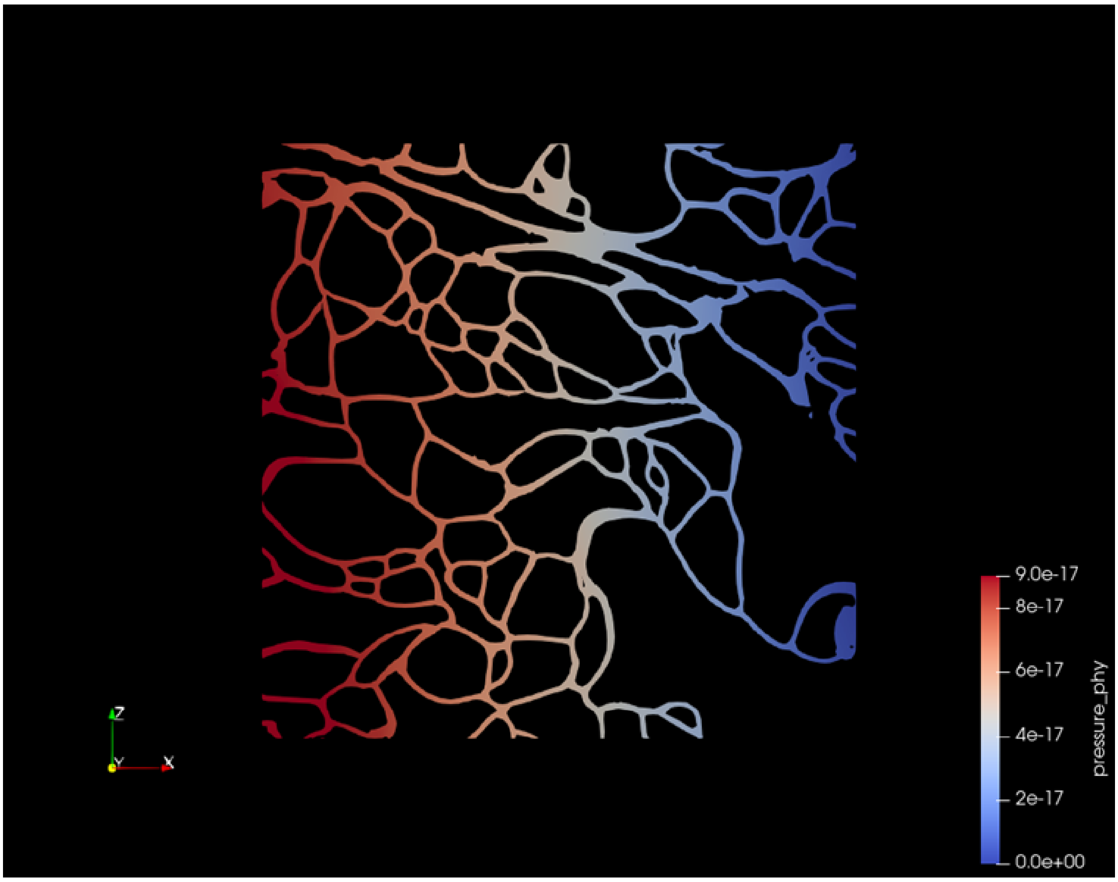


Figure 5.20: Pressure in one of the neuropil slices. Pressures are given in GPa

The pressure drop over the geometry is $9.2 \cdot 10^{-17}$ GPa, which gives a pressure gradient of $1.725 \cdot 10^{-7}$ $mmHg/mm$. However, this was still with the wrong viscosity. As explained before, there is a linear relation assumed between the viscosity and the pressure drop. So when the viscosity is increased by a factor 10^8, the pressure gradient becomes 17.25 $mmHg/mm$. Because of the pulsations of the arteries in the brain, the pressure gradient fluctuates between $0 - 45$ $mmHg/mm$, according to the glymphatic hypothesis. This is quite a range, but since the pressure drop is of the same order of magnitude as the values in the range, it is considered to be realistic. Especially when keeping in mind that this simulation is based on a steady flow, and the flow in reality is pulsatile, which would explain the higher range.

Since the flow is relaxed, the velocity profile looks logical, with no turbulence, and the pressure gradient falls in the acceptable range, the simulation is considered to have realistic values. Now the Péclet number should be calculated in order to see what transport mechanism is more dominant. Recall Equation 5.8:

$$t_{Diff} = \frac{L^2}{D}, \quad t_{Adv} = \frac{L}{u}, \quad \rightarrow Pe = \frac{t_{Diff}}{t_{Adv}} = \frac{Lu}{D} \tag{5.11}$$

The Péclet number depends on a diffusion constant D. In this research, two different values for D are used, one for monomers of the protein amyloid β ($0.623 \cdot 10^{-6}$ cm^2/s), and one for oligomers consisting of 24 monomers ($0.216 \cdot 10^{-6}\ cm^2/s$). These values have already been corrected for the tortuosity of the neuropil, so the distance L for the diffusion term refers to the linear distance. For the advection term, this is not the case, so this still has to be done. For a tortuosity of 1.7, this gives the following expression for the Péclet number in this specific case.

$$Pe = \frac{Lu}{1.7D} \tag{5.12}$$

In this case, the the chosen value for L represents the linear distance the solutes have to travel. The vascular separation in the human brain is 238 μm[30], this is the average distance between a artery-vein pair. In this research, the distance L is taken in the middle of this average distance, so at 119 μm. This number is taken as a constant value since does not change significantly over the geometry. The geometry is $4\mu m$ wide, so it would range from $117 - 121\mu m$. The post-processing in Paraview is significantly less cumbersome if this number is taken as a constant. In Figure 5.21 the local Péclet numbers for amyloid β monomers can be seen for one of the slices, and Figure 5.22 shows the same for oligomers. In the red regions, the Péclet number is smaller than 0.33, meaning that diffusion is at least three times more dominant than advection. In the green regions, advection is at least three times more dominant than diffusion. In the yellow regions, the Péclet number lays between 0.33 and 3.

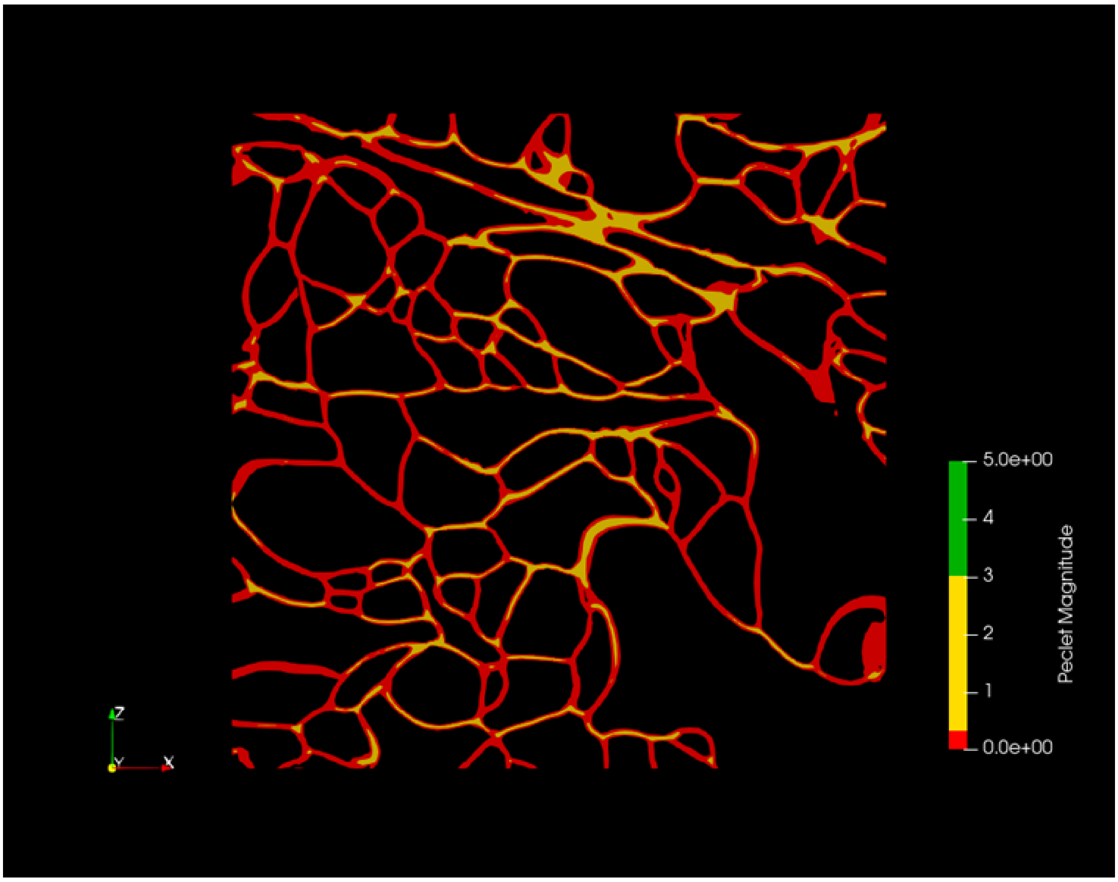


Figure 5.21: Péclet number for monomers in one of the neuropil slices

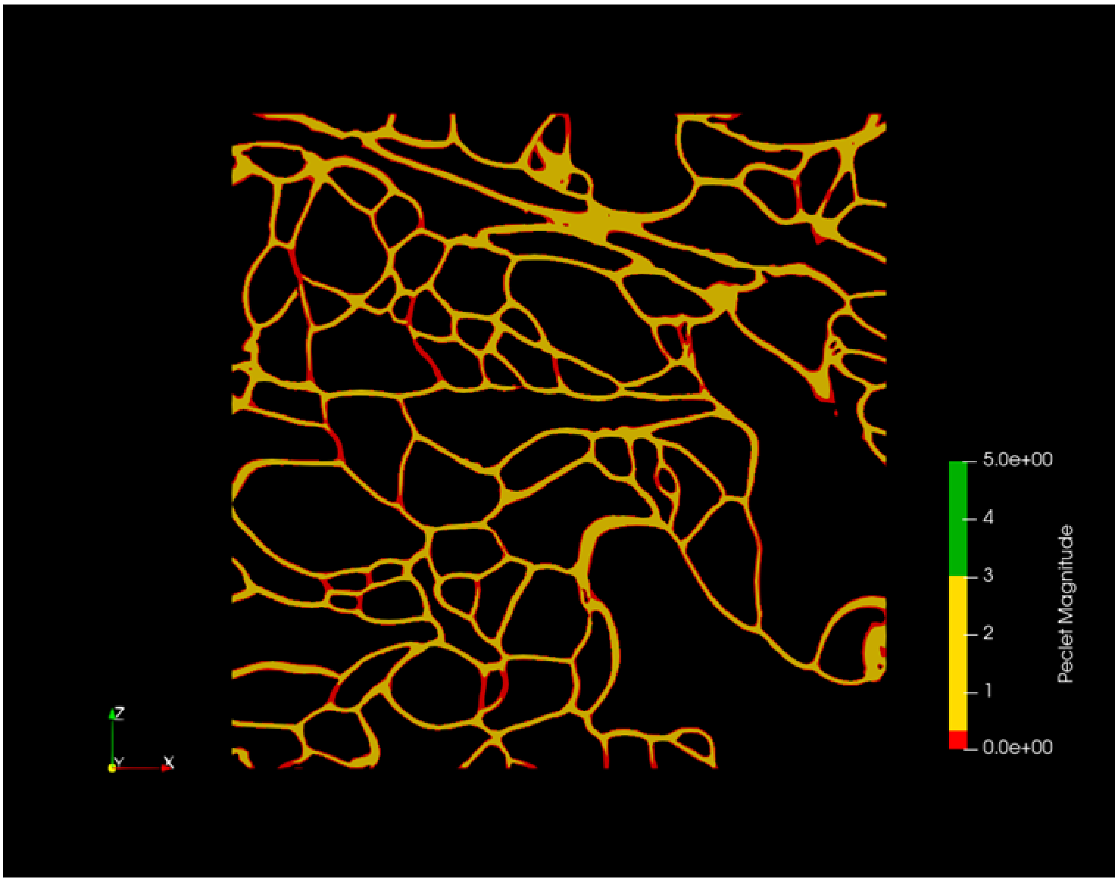


Figure 5.22: Péclet number for oligomers in one of the neuropil slices

The meaning of this local Péclet number may need some clarification. For this number, the local velocity is taken, but the characteristic length is still half the average vascular separation. So if a solute is in a green area, it would mean that it would have that Péclet number, only if those solutes would keep that velocity for the entire traveled path. The figures above mainly give an overall indication rather than values to specific locations. The Péclet number should be calculated with the average velocity over the entire path. The average Péclet numbers over all slices can be found in Table 5.4. The monomers are quite clearly diffusion driven, the oligomers tend to be diffusion driven but not clearly. An important side note is that for both the monomers and oligomers the same velocity profile is taken, while larger molecules might follow the flow less accurately. A big argument for why this difference is probably not that big is the extremely low Reynolds numbers in this flow. The inertia terms are negligible compared to the viscous terms, so accelerations in the flow would not have a lot of delay for the molecules in the flow.

Aβ Monomers	0.1808
Aβ Oligomers	0.5213

Table 5.4: Average Péclet number over all slices

5.3.3 Simulation 2 and 3: Steady flow with higher ISF velocity for a sleeping person, and a much lower ISF velocity for an awake person

To test the effect of the bulk flow velocity on the results, two simulations are preformed with different velocities. The first simulation was done with a velocity of 10 $\mu m/min$ which is 166 nm/s. The second simulation will have a velocity of 15 $\mu m/min$ or 250 nm/s which is at the upper limit of the plausible velocity range. The other simulation is for an awake state. In this state, the velocity can decrease by 93 %. It will have a velocity of 0.7 $\mu m/s$ or 11.7nm/s. This much lower value is due to the fact that brain cells grow a bit when they become more active, this causes the extra cellular space to shrink, and lower the velocity significantly.

High velocity

The results of these simulations are quite as expected. In the situation of a higher velocity, the pressure drop becomes higher and the means of transport shifts more towards advection. In the case of the lower velocity, exactly the opposite happens. The results of the simulation with the higher velocity can be found in Figure 5.23 to 5.26. In order to compare the results of this simulation with the results of the previous simulation, the scales in these figures are kept the same.

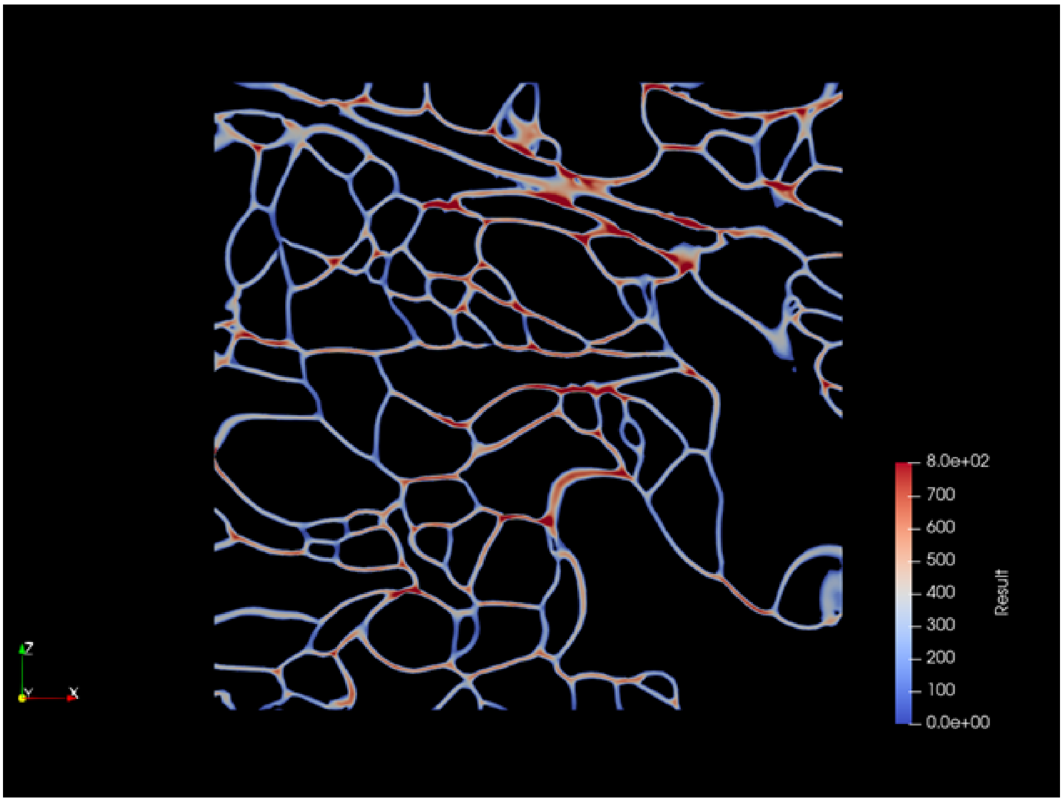


Figure 5.23: Velocity profile in one of the neuropil slices with a higher velocity. Velocities are given in nm/s

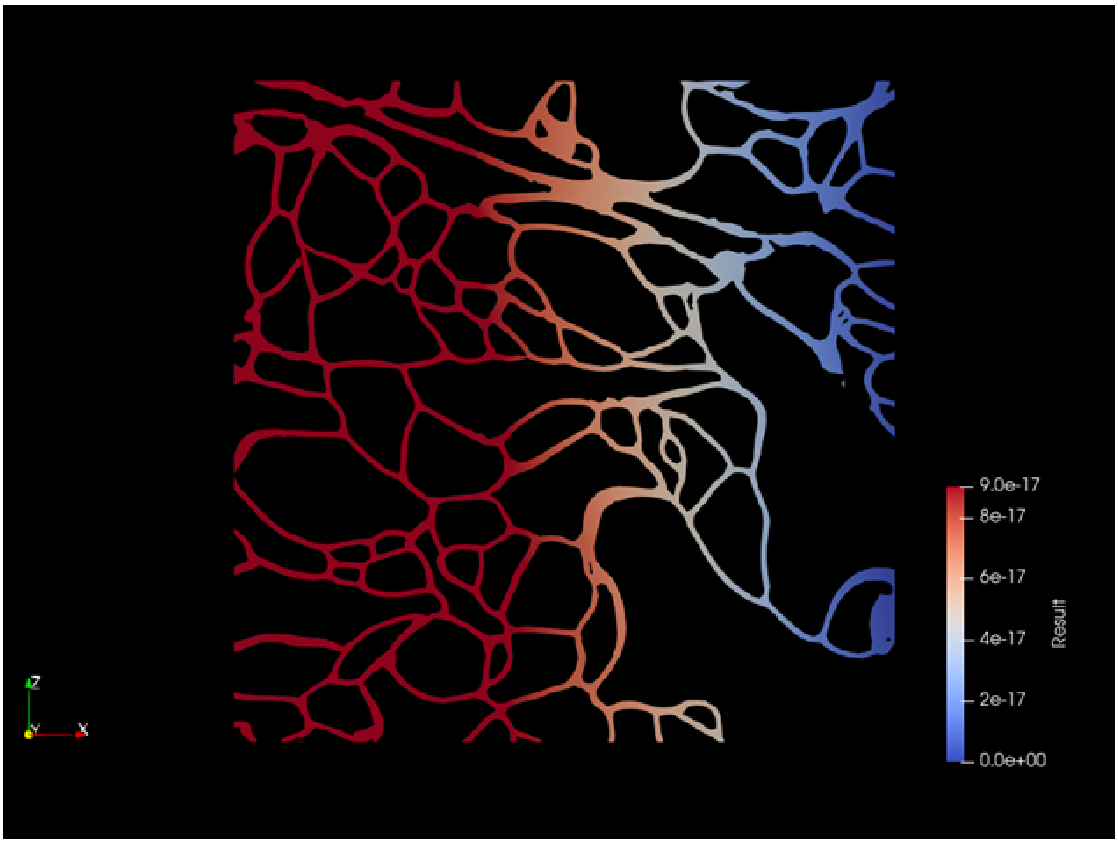


Figure 5.24: Pressure in one of the neuropil slices with a higher velocity. Pressures are given in GPa

In the velocity profile it can be seen that the velocity is indeed higher than the previous simulation, which is not surprising. The same goes for the pressure drop. What is interesting however, is that the average velocity and the pressure drop increased by almost exactly a factor 1.5. This is reassuring, since it is to be expected. The input velocity is 1.5 times as high, so the average velocity over all the slices should be 1.5 times as high as well. Since the velocity increased but the velocity at the walls is still zero due to the no slip boundary conditions, the velocity gradient and thus the pressure drop increased as well.

In the plots for the Péclet number for monomers, the yellow regions have grown. In the plot for the oligomers, it is visible that even some green regions have appeared. Again, these local values do not mean all that much, but is does show that the Péclet number increased, which was to be expected as well.

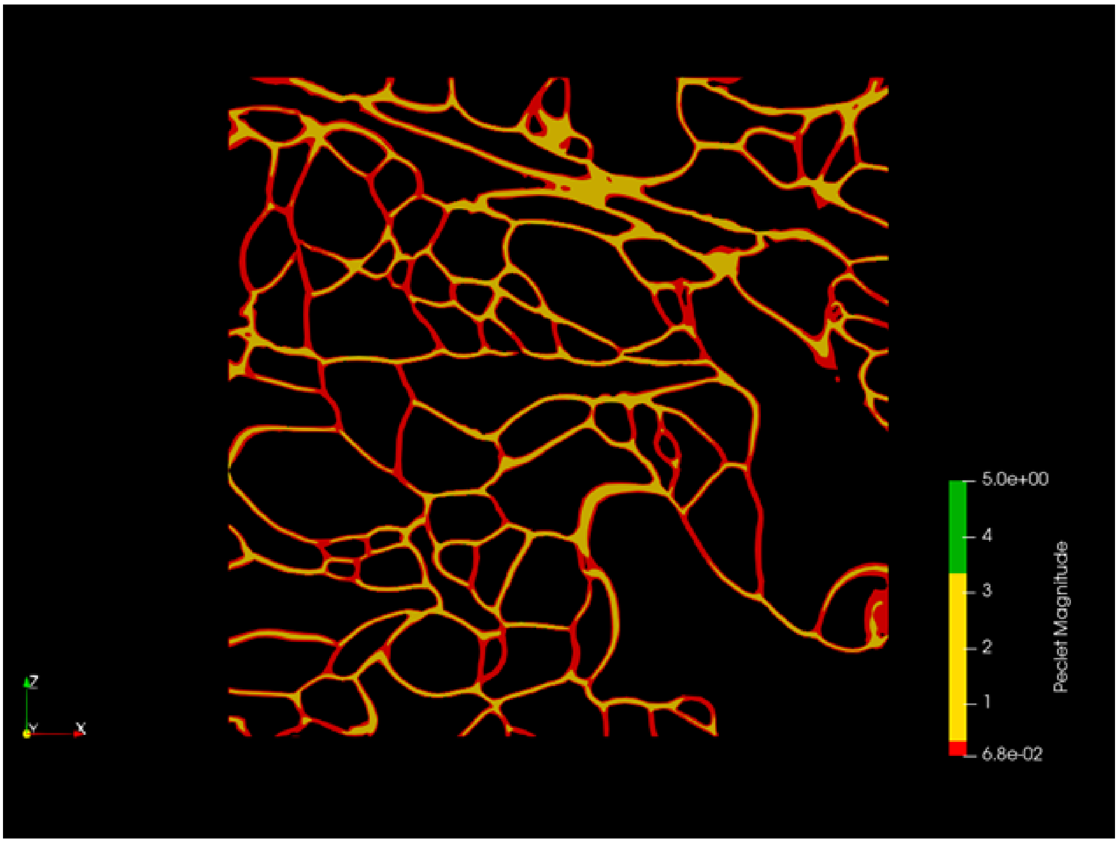


Figure 5.25: Péclet number for monomers in one of the neuropil slices with a higher velocity

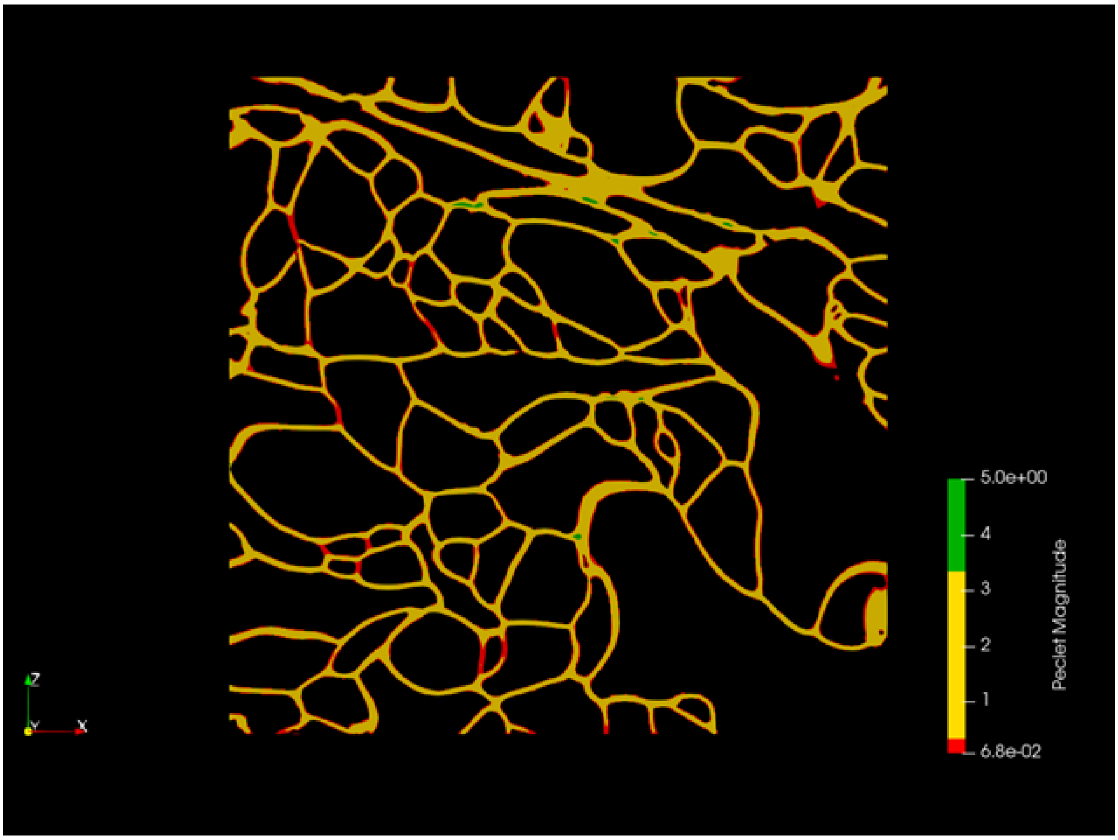


Figure 5.26: Péclet number for oligomers in one of the neuropil slices with a higher velocity

Low velocity

The results of the simulation with the much lower velocity can be seen in Figure 5.27 and 5.28. For these results, the scales have been readjusted to a lower scale. Otherwise, the plots would have just shown single colours. Again, the velocity and pressure drop changed as expected. The input velocity decreased by 93%, and so did the average velocity and pressure drop. The plots for the velocity profile and pressure drop are left out, because they do not really show any new information. The threshold for the red regions in the results for the Péclet number were decreased by 93% as well, otherwise the plot would have been entirely red.

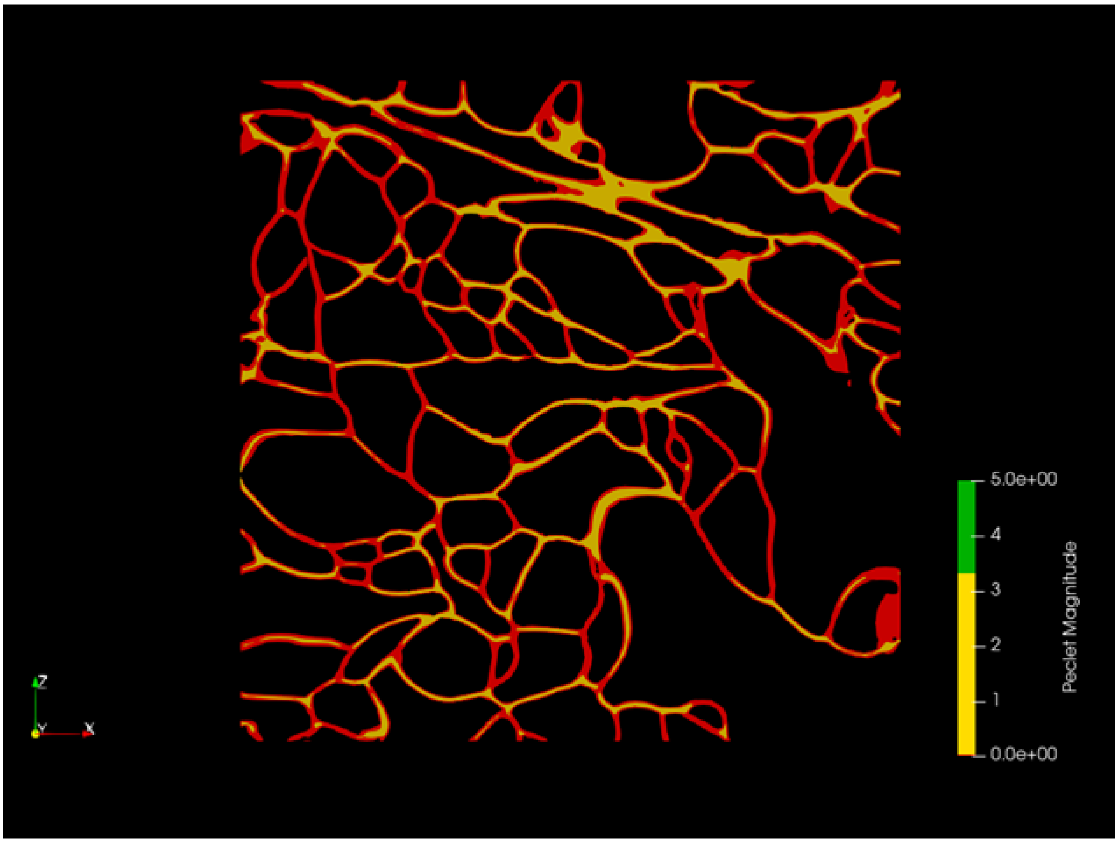


Figure 5.27: Péclet number for monomers in one of the neuropil slices with a lower velocity

A table with the interesting flow properties for both the average, high and low bulk flow velocity can be found in Table 5.5. For the simulation with a higher velocity, the values for the Péclet number are close to 1. For the simulation in the awake state, the values of the Péclet number are very low. What is interesting to mention is that in a research on the same geometry using conventional CFD methods [30], a comparable velocity was reached of $12.2 nm/s$. This is quite coincidental, since they did not take the pressure gradient due to pulsivity into account, but only a fairly low constant gradient. However you would expect that when the velocities in the exact same geometry are almost the same, the pressure gradient would also be almost the same. The pressure gradient in this research was put to 1 $mmHg/mm$. With a 20% higher pressure gradient, the velocity in this research was a bit lower. The question arises which one is right. It might be interesting to do a simulation with a finer grid to see whether the results will match more closely. This will however be left as a recommendation for future research, due to time constraints.

Average velocity $[nm/s]$	160.8	214.3	11.26
pressure gradient $[mmHg/mm]$	17.25	25.88	1.21
Average Péclet number monomers	0.1808	0.2711	0.0127
Average Péclet umber oligomers	0.5213	0.7820	0.0365

Table 5.5: Caption

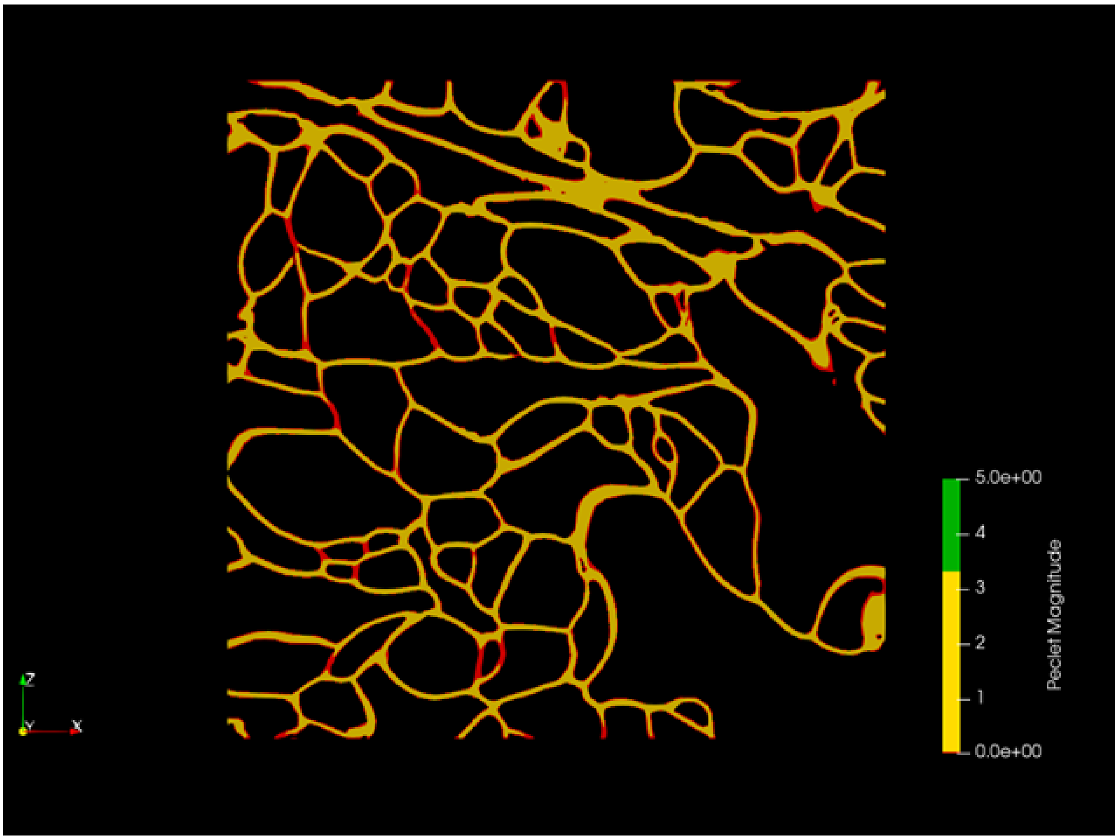


Figure 5.28: Péclet number for oligomers in one of the neuropil slices with a lower velocity

5.3.4 Simulation 4: Pulsatile flow with average velocity for a sleeping person

In the glymphatic system, bulk flow is generated by the pulsation of the cardiac cycle. To test the effect of a pulsatile behaviour in the flow, the constant velocity is replaced by a simple sine wave with the frequency of $1Hz$ which is in the range of a healthy heart rate. This does not necessarily resemble the actual pulse since the actual shape of this pulsation, if it even is there, is not defined. This simulation is merely to see what the effect would be of a pulsatile flow on the results.

To be able to compare the results of this simulation with the results of the simulation with the average constant sleeping bulk flow. The velocity in the trackers should be evaluated. Comparing the results when the velocity of the pulse simulation is at a peak, would not make sense. The velocities of the two simulations would be totally different, so no conclusions can be drawn.

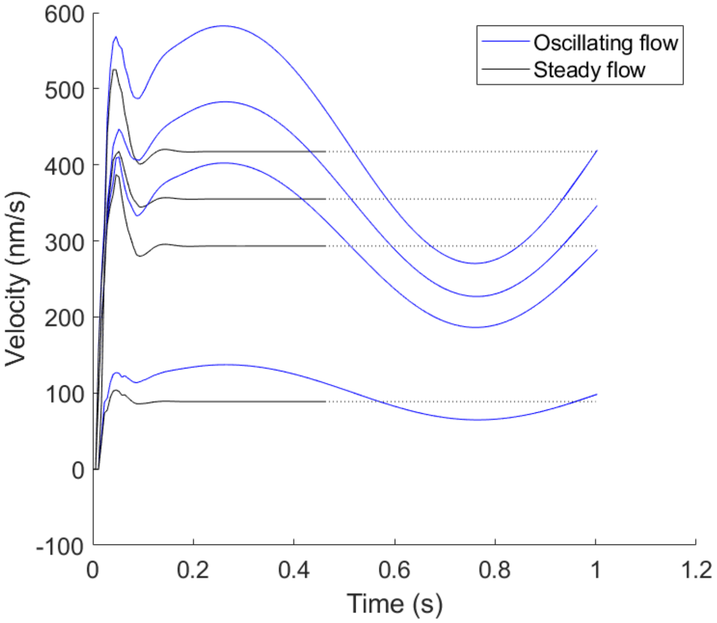


Figure 5.29: The velocity at the location of various trackers in a steady flow and an oscillating flow

Figure 5.29 Shows the velocity at the location of various trackers. The data for the steady flow did not go as far in time as the oscillating flow, so the last value of each data set has been plotted over the dotted line. The oscillating flow reaches the steady flow velocity around t=1s. This is logical, since a simple sine wave is used as the oscillating offset, and the period of this sine was chosen to be 1 second. At this time, the results of the simulations are compared. Only the velocity profile will be evaluated, since differences in the velocity will translate into the same differences in Péclet number. Figure 5.30 shows the difference between the velocity in the steady simulation and the velocity in the oscillating simulation for one of the slices.

It is visible that the difference is very low at most of the points. The maximum difference lays around 30 nm/s, but that is mainly on the parts where the velocity is high. It might be more interesting to look at the relative difference. This difference can be seen in Figure 5.31, it is given by $\frac{u_{oscil}-u_{steady}}{u_{oscil}} \cdot 100\%$. Again the differences are the largest at the locations of the highest velocity, but overall the difference is not that big. At most, is is around 5%. It is not surprising that this difference is not very large, since the biggest difference is expected to be because particles might still have momentum in a certain direction when the velocity changes. However, because the Reynolds numbers are very low, the inertial forces are negligible compared to the viscous forces. Any inertia particles might have when the velocity input changes can be neglected. It is important to note that this simulation was done with the wrong viscosity. In reality, the viscosity is much higher and the Reynolds number is much lower. The difference between the two simulations is expected to be even smaller in reality.

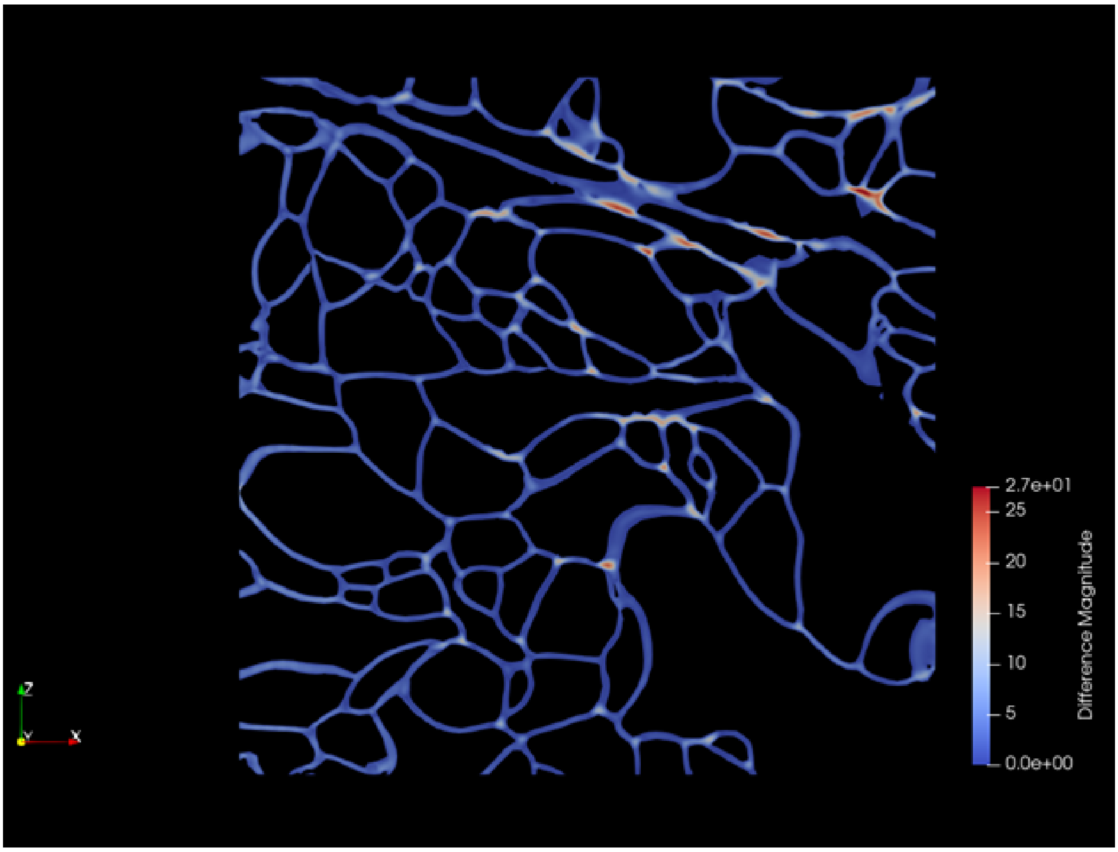


Figure 5.30: The absolute difference in velocity between the steady and oscillating simulation. The velocities are given in nm/s

Next to looking at the difference between the oscillating flow and the steady flow, it is also important to look at whether mixing is taking place. Especially when the inflow increases, the fluid at the inlet has a higher velocity than the fluid right in front of it, this might effect the nature of the flow. Mixing might take place, which would influence the results. The diffusion constant in turbulent areas is different that in laminar areas. The oscillation in the simulation is a sine wave, which has its larges acceleration at the start or end of each period. That is fortunate, since for this moment in time, the differences between the oscillating and the steady flow are known. On the location with the largest differences, closer inspection will take place to see if there is any sign of vortices, which would be a indication of mixing. Figure 5.32 and 5.33 show the velocity field with vector arrows on two zoomed in parts of the neuropil slice. The red rectangles in Figure 5.31 indicate where the zoomed in parts are located. The closer inspection lead to no evidence of any mixing. More figures of zoomed in sections as well as

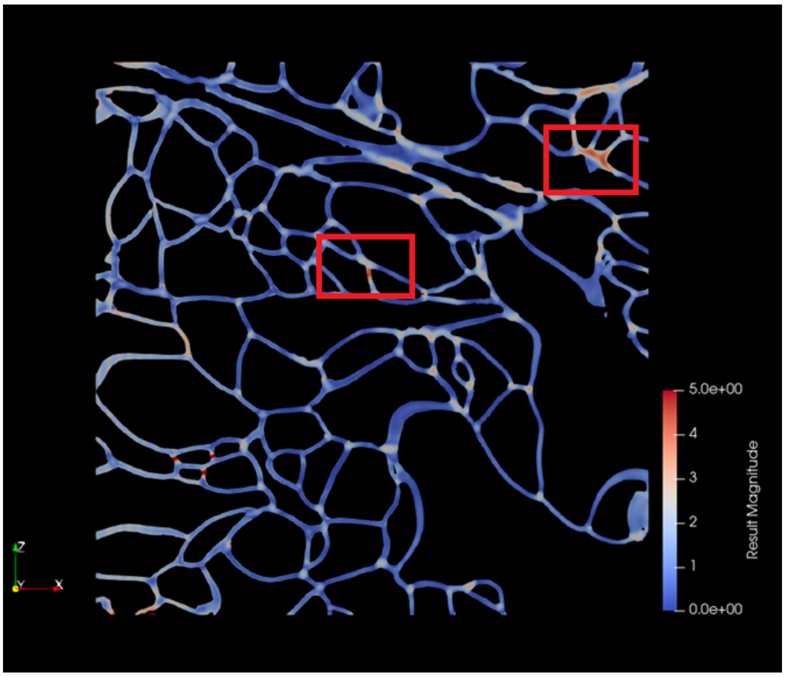


Figure 5.31: The relative difference in velocity as a percentage of the velocity of the oscillating simulation.

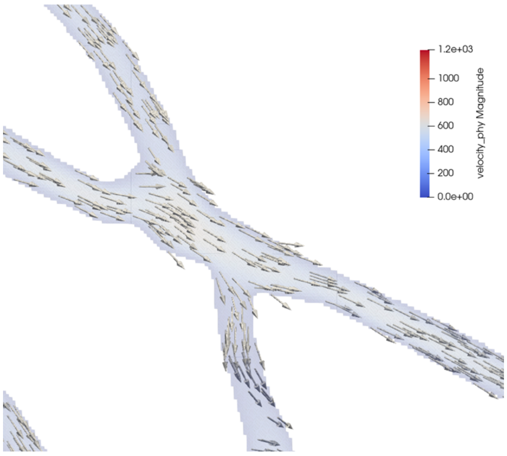


Figure 5.32: Velocity field of a zoomed in part of the neuropil geometry

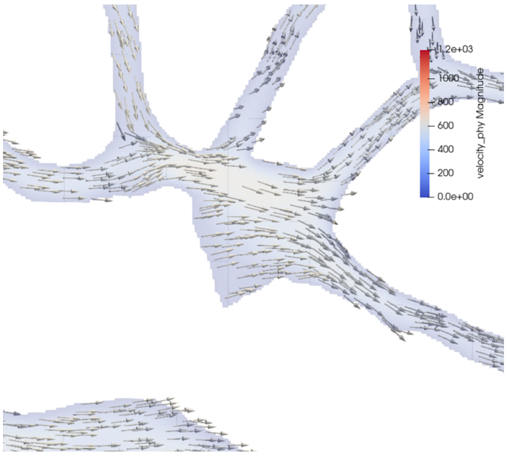


Figure 5.33: Velocity field of a zoomed in part of the neuropil geometry

Chapter 6

Discussion, conclusions and Recommendations

In this chapter, the results are discussed, after that, conclusions are drawn from the results based on the research question ad sub questions stated in the introduction. Finally, recommendations for future research are given.

6.1 Discussion

The first simulation was done with a steady flow and an average ISF velocity for a sleeping person. In this simulation, the velocity profile looked logical. The velocity was lower in narrow channels and near walls and in larger tunnels the velocity was much higher. Velocities of around 800 nm/s are not uncommon, while the mean velocity is 160.8 nm/s. The maximum velocity in the neuropil is about 5 times as high as the mean velocity. The pressure gradient of 17.25 $mmHg/mm$ is in line with values in literature that state that the pressure gradient varies between 1 $mmHg/mm$ and 43 $mmHg/mm$. Given the fact that this varying is due to pulsations of arteries, and this simulation looked at the mean velocity, a value for the pressure gradient somewhere in the middle between 1 and 43 $mmHg/mm$ is to be expected.
Increasing or decreasing the inlet velocity by a certain factor has almost exactly the same effect as just multiplying the velocity in every point with that same factor. This is not surprising, as the flow stays in the laminar regime for all the simulated velocities on all locations.
In a study on the same geometry that uses common CFD techniques, a pressure gradient of 1 $mmHg/mm$ is used[30]. With a simulation with 84 million elements, their average velocity was 12.2 nm/s. In the low velocity simulation in this research with 385 elements, with a pressure gradient of 1.21 $mmHg/mm$, the average velocity was 11.26 nm/s. With a 20% higher pressure drop in this research, the velocity still was around 8% lower. It is probably not entirely fair to compare the amount of elements one on one, but the large difference in number of elements would be in favour of the results in this research.
The results of the oscillating flow simulation were very similar to the results of the steady flow. The results differed at most around 5% compared to the velocity values of the oscillating flow. This small difference even was from the simulation with the much lower viscosity. The difference will presumably be even smaller when the correct viscosity is considered. The inertia terms will become even less significant and the flow will follow changes more directly.

Before drawing conclusions from the results, it is important to point out some possible uncertainties and noteworthy comments and judge whether they would change the conclusion significantly.

To start, in this research the neuropil cube is considered to be located in the middle between the blood vessels. This is because waste products like amyloid β are produced everywhere in the parenchyma, so the average distance a freshly produced waste molecule has to travel is half the vascular separation. It is however possible to assume the cube is closer or further away from the periveinous space. In this case the L in the Péclet changes. The Péclet number can either become twice as high for the waste products created twice as far, or close to zero for the waste products created very close to the periveinous space. Meaning the Péclet number in the high velocity case for monomers could get as high as 0.54, which would already be quite balanced between diffusion and advection driven transport. The Péclet number for oligomers can get as high as 1.56. In that situation the ratio of diffusion and advection is shifted towards advection, but it would still be considered quite balanced. In all cases, the transport of amyloid β is never convincingly advection driven.

It has already been mentioned before, but the velocity used in the Péclet formula is the velocity of the flow, and not the velocity of the solute molecules. The difference in these velocities is thought to be insignificant, since the Reynolds number of the flow is very low, meaning the inertia terms are negligible compared to the viscous terms. Intuitively this can be seen as a pebble following a flow in water and in honey. The Reynolds number in the flow of honey would be lower than in the flow of water. The pebble would follow the flow in honey much better. In this thesis, the Reynolds numbers are very low, so the amyloid β molecules would probably follow the flow very well. Another thing that might slow the molecules down compared to the fluid flow, would be restrictions from either tangling oligomers that get stuck, or long oligomers that wrap around a corner. This restriction cannot be neglected because of the low Reynolds number. It is uncertain whether this has a significant impact. Another problem that was already mentioned, is that the local Péclet number does not really say a lot. The Péclet number says something about the dominant transport mechanism over the entire travel distance L. So if there is a small part on that route where the the velocity is higher, this will not influence the the transport rate over the entire length all that much. The average Péclet number was used to tell something useful about the results. The sad truth about this is that this average Péclet number can be calculated with the average velocity, which is known before the simulation is even preformed, making the simulations not that useful.
A possible way to solve this problem would be to simulate the molecules in the flow separately. This is not possible in the current framework, but it is being worked at. This would however be a cumbersome problem, because the molecules are quite long and flexible. Just simulating rigid particles would not allow for tangling or wrapping of molecules around each other or corners. However, the overall Péclet number already points towards a mostly diffusion driven transport. When the molecules are restricted, their velocity decreases and the Péclet number will point to diffusion even more. So these restrictions will not change the conclusion all that much. Next to that, it might already be a big step in the right direction to use small rigid particles for the solute that have the same diffusion constant instead of the flexible ones. These results would show whether these molecules would follow the fluid flow as well as expected. It might also be the case that these rigid molecule replacements are mainly found in high velocity regions and barely in the low velocity regions, of the other way around. This would already give some insight without having to simulate the complicated long flexible molecules.

In this report, the neuropil cube was placed in the middle of an artery vein pair. It is however not the case that these blood vessels are plane source and sink, instead they are better considered as line source and sink. This changes the bulk velocity shape between the source and the sink. With a plane source and sink the flow lines go straight from the source to the sink. With a line source and sink, it looks like Figure 6.1. Here the velocities near the artery and vein are higher than in the middle. The velocity in the middle where the neuropil cube was put would not have the average velocity, but rather the minimum velocity. However, when recalling what the Péclet number represents, it gives the diffusion time over the advection time. The average advection time is given by the average distance divided by the average velocity. The average distance might not have been half the vascular separation, maybe a bit more, but the average velocity is used correctly in this report. Considering that the average distance solutes have to travel becomes a bit larger, the average Péclet number increases a bit. How much it increases depends on the exact layout of arteries and veins in the brains parenchyma, which is not known.

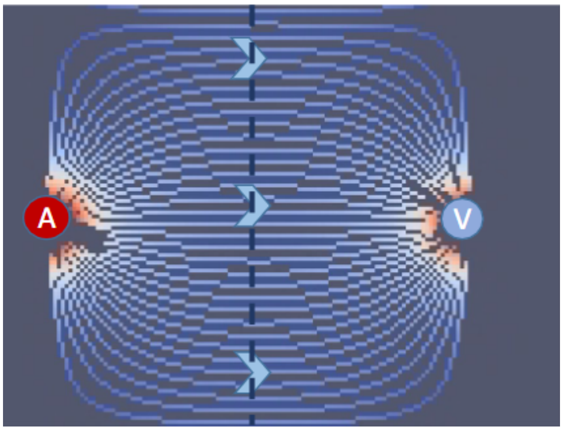


Figure 6.1: Streamlines of a line source and sink

Continuing on the line sources and sinks, the geometry in this report lends itself for a nice opportunity. The hole throughout the entire geometry is presumably because either an axon is located there. While capillaries are much bigger than this hole (around 8-10 μm in diameter), it would still be a nice follow-up study to investigate the flow for the situation when the flow inlet or outlet is located as a line source in this hole. This way the flow close to an artery or vein can be studied as well. It is debatable how well this approximates reality however, because the flow will be relatively very convergent or divergent close to the line sink or source. Normally the diameter of a capillary is more than twice as big as a side of the geometry, leading to a more unidirectional flow.

Another noteworthy thing about the used geometry, or rather the mesh is the type of boundary condition on the side walls of the cube. In this report a bounce-back boundary condition is used, meaning that the velocity is zero at those locations. This simulates a no-slip condition you would normally find at walls. However, the walls of the containing cube are not there in reality. There is no reason for the flow parallel to the wall to be zero, so instead of particles bouncing back to the node they came from, they should bounce forward in the direction they were going. When looking at Figure 6.2, a better alternative would be that from step B to step C, the velocity in the direction parallel to the wall would not change, only the velocity perpendicular to the wall. So the arrows in step C would not all point to the middle element but rather the left arrow would point to the left most element, and the right arrow would point to the right most element. This way the velocity at the sides of the cube would not be zero. The effect of choosing this boundary condition is probably insignificant, since the walls of the geometry do have the right no slip boundary condition, and their effect is in most cases more dominant than the effect of the walls at the side of the cube. This is because most points are simply far away from the sides of the cube compared to their proximity to the walls of the geometry.

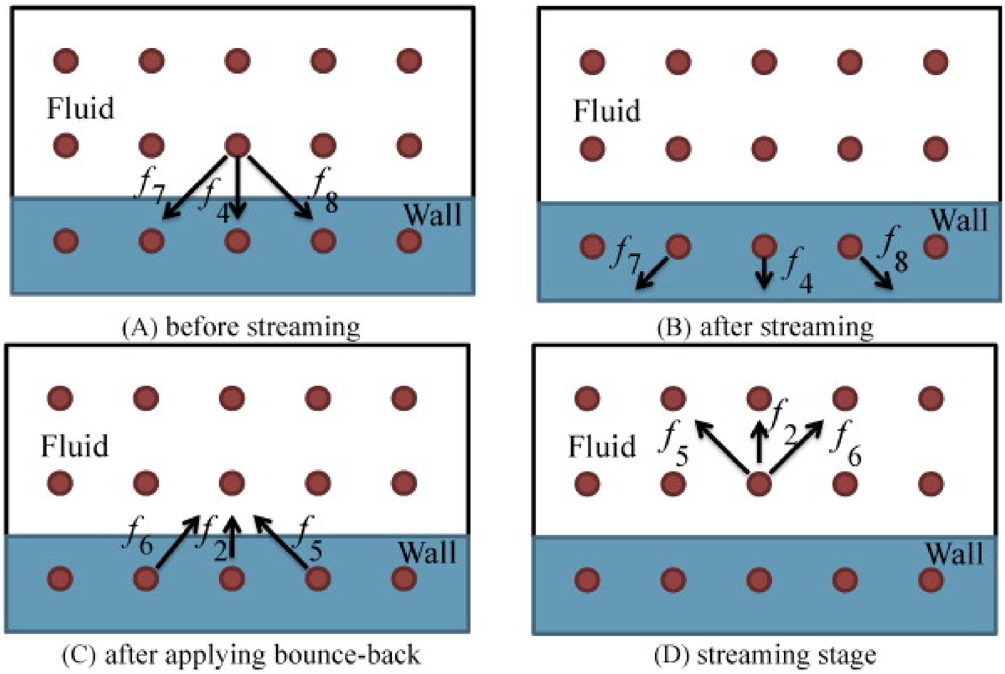


Figure 6.2: Velocity bounce-back boundary condition

Another small note on the mesh is that it is rigid. This is not something that can be changed, and it is not really a problem in the steady state situations, since in that case, the geometry would not change over time. But in the pulsatile flow it might have had some effect.
Also, when looking at the awake state, the velocity is lower. In this report this is a result of a lower velocity as the inlet boundary condition. In reality this is a result of a lower volume fraction. In the awake state, the channels in the geometry should have been narrower. Because the most relevant parameter for the Péclet number is the average velocity, and that did not change, the Péclet number is not really influenced by this inaccuracy.

6.2 Conclusions

In order to draw the conclusions, the research question and sub questions are recalled:
"Is the transport of amyloid β in the interstitial space in the brain mainly diffusion or bulk flow driven?"

1. **How can the Peclèt number be derived from the LBM simulations?**

2. **Using what boundary conditions can the flow in the in the ISS be simulated realistically?**

3. **Does the velocity profile of interstitial fluid represent the velocity of amyloid β molecules ?**

4. **What effect do external periodical influences have on the results**

The first sub question has already been answered in an earlier part in the report. Values for the diffusion constant of amyloid β in the interstitial fluid can be found in literature. The tortuosity of the geometry had to be kept in mind in the calculations of the Péclet number, which has been done.

The second sub question has also already been elaborated. However, the simple bounce back boundary condition on the walls of the cube might not have been the right choice as argued in this chapter, but the effect of this choice are probably insignificant.

Because of the low Reynolds numbers in the flow, it is believed that velocities of solutes in the flow will match the velocity profile closely. Apart from the intertwining of molecules and wrapping of molecules around corners, the simulations are believed to represent the velocity of the molecules well enough to be able to draw conclusions based on the results.

The oscillatory behaviour of the flow is believed to have a negligible effect on the results. The very low Reynolds numbers indicate that the inertia of the flow is very low compared to the viscous terms, so changes in the input velocity will probably be translated directly into the flow. Therefor it can be concluded that averaging over the oscillation does not have a big impact on the results.

For the steady state simulations, it can be concluded that the transport of amyloid β molecules is in no case dominated by advection. At most the Péclet number reaches values around 1.5, for oligomers close to arteries. Apart from this specific case, the majority of the solutes will be transported mainly because of diffusion. The results are however not as convincing to be able to rule out the effect advection has on the transport of solutes.

6.3 Recommendations

For future research it would be recommended to simulate the amyloid β molecules as separate particles with the right shape and flexibility in the flow. In that case it is possible to see whether the solutes will have the same velocity as the flow at that location, but also whether the solutes are located predominantly in either the high or low velocity regions. Furthermore, when the molecules in the simulation have the right shape and flexibility, it is possible to see the effect of tangling and folding around corners.

Furthermore, for future simulations, it might be a good idea to look into using a different boundary conditions for the walls of the cube, as mentioned in the discussion section of this chapter. The boundary condition used in this research is very simple and might cause for less required computational power than the other boundary condition. Also the effect that the current boundary condition has on the results is probably not that big. But since the other boundary condition, mentioned in the discussion part of this chapter, represents reality more accurately, it might be worth looking into.

Another idea for a follow up research is to study the flow close to an artery or vein using the tunnel in the geometry. Although this hole is too small to realistically represent a capillary, such a simulation might give useful insights. This, in combination with the velocity shape between a line source and sink as shown in Figure 6.1, will give a depiction of the flow on different interesting locations.

The results from the low flow velocity simulation were comparable with results of a paper that did flow simulations on exactly the same geometry. However, there was still a significant difference between their results and the results in this report. The mesh used in this report had significantly more elements than the aforementioned paper. However, the methods used are different so a detailed conclusion on which results are more accurate is not possible. However, a simulation with a finer mesh might shed more light on this problem. As of this moment, a simulation with 1.3 billion cells is near to impossible, but in a future with higher computing power it might be worth looking in to.

Bibliography

[1] W. H. Organization, "Dementia: https://www.who.int/news-room/fact-sheets/detail/dementia." [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/dementia

[2] C. Reitz and R. Mayeux, "Alzheimer disease: Epidemiology, diagnostic criteria, risk factors and biomarkers," *Biochemical Pharmacology*, vol. 88, no. 4, p. 640–651, 2014. [Online]. Available: https://dx.doi.org/10.1016/j.bcp.2013.12.024

[3] D. J. Selkoe, "Aging, amyloid, and alzheimer's disease: A perspectivein honor of carl cotman," *Neurochemical Research*, vol. 28, no. 11, p. 1705–1713, 2003.

[4] J. M. Tarasoff-Conway, R. O. Carare, R. S. Osorio, L. Glodzik, T. Butler, E. Fieremans, L. Axel, H. Rusinek, C. Nicholson, B. V. Zlokovic, and et al., "Clearance systems in the brain—implications for alzheimer disease," *Nature Reviews Neurology*, vol. 11, no. 8, p. 457–470, 2015. [Online]. Available: https://dx.doi.org/10.1038/nrneurol.2015.119

[5] L. Fan, C. Mao, X. Hu, S. Zhang, Z. Yang, Z. Hu, H. Sun, Y. Fan, Y. Dong, J. Yang, C. Shi, and Y. Xu, "New insights into the pathogenesis of alzheimer's disease," *Frontiers in Neurology*, vol. 10, p. 1312, 2020. [Online]. Available: https://www.frontiersin.org/article/10.3389/fneur.2019.01312

[6] J. Hardy and D. Allsop, "Amyloid deposition as the central event in the aetiology of alzheimer's disease," *Trends in pharmacological sciences*, vol. 12, pp. 383–388, 1991.

[7] M. A. Deture and D. W. Dickson, "The neuropathological diagnosis of alzheimer's disease," *Molecular Neurodegeneration*, vol. 14, no. 1, 2019. [Online]. Available: https://dx.doi.org/10.1186/s13024-019-0333-5

[8] W.-P. Chang, G. Koelsch, S. Wong, D. Downs, H. Da, V. Weerasena, B. Gordon, T. Devasamudram, G. Bilcer, A. K. Ghosh *et al.*, "In vivo inhibition of aβ production by memapsin 2 (β-secretase) inhibitors," *Journal of neurochemistry*, vol. 89, no. 6, pp. 1409–1416, 2004.

[9] L. McConlogue, M. Buttini, J. P. Anderson, E. F. Brigham, K. S. Chen, S. B. Freedman, D. Games, K. Johnson-Wood, M. Lee, M. Zeller *et al.*, "Partial reduction of bace1 has dramatic effects on alzheimer plaque and synaptic pathology in app transgenic mice," *Journal of Biological Chemistry*, vol. 282, no. 36, pp. 26 326–26 334, 2007.

[10] G. T. Wong, D. Manfra, F. M. Poulet, Q. Zhang, H. Josien, T. Bara, L. Engstrom, M. Pinzon-Ortiz, J. S. Fine, H.-J. J. Lee *et al.*, "Chronic treatment with the γ-secretase inhibitor ly-411,575 inhibits β-amyloid peptide production and alters lymphopoiesis and intestinal cell differentiation," *Journal of Biological Chemistry*, vol. 279, no. 13, pp. 12 876–12 882, 2004.

[11] C. A. Hawkes, L.-H. Deng, J. E. Shaw, M. Nitz, and J. McLaurin, "Small molecule β-amyloid inhibitors that stabilize protofibrillar structures in vitro improve cognition and pathology in a mouse model of alzheimer's disease," *European Journal of Neuroscience*, vol. 31, no. 2, pp. 203–213, 2010.

[12] Y.-H. Liu, Y.-R. Wang, Y. Xiang, H.-D. Zhou, B. Giunta, N. B. Mañucat-Tan, J. Tan, X.-F. Zhou, and Y.-J. Wang, "Clearance of amyloid-beta in alzheimer's disease: shifting the action site from center to periphery," *Molecular neurobiology*, vol. 51, no. 1, pp. 1–7, 2015.

[13] S.-H. Xin, L. Tan, X. Cao, J.-T. Yu, and L. Tan, "Clearance of amyloid beta and tau in alzheimer's disease: from mechanisms to therapy," *Neurotoxicity research*, vol. 34, no. 3, pp. 733–748, 2018.

[14] F. G. De Felice and S. T. Ferreira, "β-amyloid production, aggregation, and clearance as targets for therapy in alzheimer's disease," *Cellular and molecular neurobiology*, vol. 22, no. 5, pp. 545–563, 2002.

[15] G. Pocock and C. D. Richards, *Human Physiology: The Basis of Medicine (Oxford Core Texts)*, paperback ed. Oxford University Press, 3 2006. [Online]. Available: https://lead.to/amazon/com/?op=bt&la=en&cu=usd&key=0198568789

[16] M. E. Watts, R. Pocock, and C. Claudianos, "Brain energy and oxygen metabolism: Emerging role in normal function and disease," *Frontiers in Molecular Neuroscience*, vol. 11, p. 216, 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnmol.2018.00216

[17] D. Raper, A. Louveau, and J. Kipnis, "How do meningeal lymphatic vessels drain the cns?" *Trends in Neurosciences*, vol. 39, no. 9, p. 581–586, 2016. [Online]. Available: https://dx.doi.org/10.1016/j.tins.2016.07.001

[18] S. B. Hladky and M. A. Barrand, "Elimination of substances from the brain parenchyma: efflux via perivascular pathways and via the blood–brain barrier," *Fluids and Barriers of the CNS*, vol. 15, no. 1, 2018. [Online]. Available: https://dx.doi.org/10.1186/s12987-018-0113-6

[19] B. Engelhardt, R. O. Carare, I. Bechmann, A. Flügel, J. D. Laman, and R. O. Weller, "Vascular, glial, and lymphatic immune gateways of the central nervous system," *Acta Neuropathologica*, vol. 132, no. 3, p. 317–338, 2016. [Online]. Available: https://dx.doi.org/10.1007/s00401-016-1606-5

[20] M. L. Barr and J. A. Kiernan, *Barr's the human nervous system: An anatomical viewpoint*, 8th ed. Philadelphia, PA: Lippincott Williams and Wilkins, 2004.

[21] I. Verheggen, M. Van Boxtel, F. Verhey, J. Jansen, and W. Backes, "Interaction between blood-brain barrier and glymphatic system in solute clearance," *Neuroscience & Biobehavioral Reviews*, vol. 90, p. 26–33, 2018. [Online]. Available: https://dx.doi.org/10.1016/j.neubiorev.2018.03.028

[22] T. Taoka and S. Naganawa, "Gadolinium-based contrast media, cerebrospinal fluid and the glymphatic system: Possible mechanisms for the deposition of gadolinium in the brain," *Magnetic Resonance in Medical Sciences*, vol. 17, no. 2, p. 111–119, 2018.

[23] J. J. Iliff, M. Wang, Y. Liao, B. A. Plogg, W. Peng, G. A. Gundersen, H. Benveniste, G. E. Vates, R. Deane, S. A. Goldman, and et al., "A paravascular pathway facilitates csf flow through the brain parenchyma and the clearance of interstitial solutes, including amyloid β," *Science Translational Medicine*, vol. 4, no. 147, p. 147ra111–147ra1, 2012. [Online]. Available: https://dx.doi.org/10.1126/scitranslmed.3003748

[24] A. J. Smith and A. S. Verkman, "The "glymphatic" mechanism for solute clearance in alzheimer's disease: game changer or unproven speculation?" *The FASEB Journal*, vol. 32, no. 2, p. 543–551, 2018. [Online]. Available: https://dx.doi.org/10.1096/fj.201700999

[25] Y. Lei, H. Han, F. Yuan, A. Javeed, and Y. Zhao, "The brain interstitial system: Anatomy, modeling, in vivo measurement, and applications," *Progress in Neurobiology*, vol. 157, p. 230–246, 2017. [Online]. Available: https://dx.doi.org/10.1016/j.pneurobio.2015.12.007

[26] G. Rosenberg, W. Kyner, and E. Estrada, "Bulk flow of brain interstitial fluid under normal and hyperosmolar conditions," *American Journal of Physiology-Renal Physiology*, vol. 238, no. 1, pp. F42–F49, 1980.

[27] T. Ichimura, P. Fraser, and H. F. Cserr, "Distribution of extracellular tracers in perivascular spaces of the rat brain," *Brain research*, vol. 545, no. 1-2, pp. 103–113, 1991.

[28] L. Ray, J. J. Iliff, and J. J. Heys, "Analysis of convective and diffusive transport in the brain interstitium," *Fluids and Barriers of the CNS*, vol. 16, no. 1, 2019. [Online]. Available: https://dx.doi.org/10.1186/s12987-019-0126-9

[29] L. A. Ray and J. J. Heys, "Fluid flow and mass transport in brain tissue," *Fluids*, vol. 4, no. 4, p. 196, 2019.

[30] K. E. Holter, B. Kehlet, A. Devor, T. J. Sejnowski, A. M. Dale, S. W. Omholt, O. P. Ottersen, E. A. Nagelhus, K.-A. Mardal, K. H. Pettersen, and et al., "Interstitial solute transport in 3d reconstructed neuropil occurs by diffusion rather than bulk flow," *Proceedings of the National Academy of Sciences*, vol. 114, no. 37, p. 9894–9899, 2017. [Online]. Available: https://dx.doi.org/10.1073/pnas.1706942114

[31] L. Ray, J. J. Iliff, and J. J. Heys, "Analysis of convective and diffusive transport in the brain interstitium," *Fluids and Barriers of the CNS*, vol. 16, no. 1, 2019. [Online]. Available: https://dx.doi.org/10.1186/s12987-019-0126-9

[32] B.-J. Jin, A. J. Smith, and A. S. Verkman, "Spatial model of convective solute transport in brain extracellular space does not support a "glymphatic" mechanism," *Journal of General Physiology*, vol. 148, no. 6, p. 489–501, 2016. [Online]. Available: https://dx.doi.org/10.1085/jgp.201611684

[33] J. Waters, "The concentration of soluble extracellular amyloid-β protein in acute brain slices from crnd8 mice," *PLOS ONE*, vol. 5, no. 12, p. e15709, 2010. [Online]. Available: https://dx.doi.org/10.1371/journal.pone.0015709

[34] K. Takagi and K. Negishi, "Measurement of ultrasonic relaxation time and mean free path in liquids," *The Journal of Chemical Physics*, vol. 72, no. 3, p. 1809–1812, 1980.

[35] B. E. Rapp, "Chapter 9 - fluids," in *Microfluidics: Modelling, Mechanics and Mathematics*, ser. Micro and Nano Technologies, B. E. Rapp, Ed. Oxford: Elsevier, 2017, pp. 243–263. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9781455731411500095

[36] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, and E. M. Viggen, *The Lattice Boltzmann Method - Principles and Practice*, 10 2016.

[37] J. v. d. Hoek, "Computational modelling of flow in the liver vasculature using the lattice boltzmann method to study microsphere distribution," vol. 1, p. 11, 2021.

[38] I. Newton, *Philosophiae naturalis principia mathematica.* J. Societatis Regiae ac Typis J. Streater, 1687. [Online]. Available: https://goo.gl/1ahxL2

[39] KhanAcademy, "Khanacademy: Maxwell-boltzmann distribution." [Online]. Available: https://www.khanacademy.org/science/physics/thermodynamics/temp-kinetic-theory-ideal-gas-law/a/what-is-the-maxwell-boltzmann-distribution

[40] A. Dadvand, M. Baghalnezhad, I. Mirzaee, B. C. Khoo, and S. Ghoreishi, "An immersed boundary–lattice boltzmann approach to study the dynamics of elastic membranes in viscous shear flows," *Journal of Computational Science*, vol. 5, no. 5, pp. 709–718, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187775031400074X

[41] G. Karniadakis, A. Beskok, and N. Aluru, *Microflows and nanoflows: fundamentals and simulation.* Springer Science & Business Media, 2006, vol. 29.

[42] "Adaptable poly-engineering simulator (apes)." [Online]. Available: https://apes.osdn.io/

[43] T. Bergman, F. Incropera, D. DeWitt, and A. Lavine, *Fundamentals of Heat and Mass Transfer.* Wiley, 2011. [Online]. Available: https://books.google.nl/books?id=vvyIoXEywMoC

[44] J. P. Kinney, J. Spacek, T. M. Bartol, C. L. Bajaj, K. M. Harris, and T. J. Sejnowski, "Extracellular sheets and tunnels modulate glutamate diffusion in hippocampal neuropil," *Journal of Comparative Neurology*, vol. 521, no. 2, p. 448–464, 2013. [Online]. Available: https://dx.doi.org/10.1002/cne.23181

[45] A. R. Costa, R. Pinto-Costa, S. C. Sousa, and M. M. Sousa, "The regulation of axon diameter: From axonal circumferential contractility to activity-dependent axon swelling," *Frontiers in Molecular Neuroscience*, vol. 11, p. 319, 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnmol.2018.00319

Appendix A

Appendix A

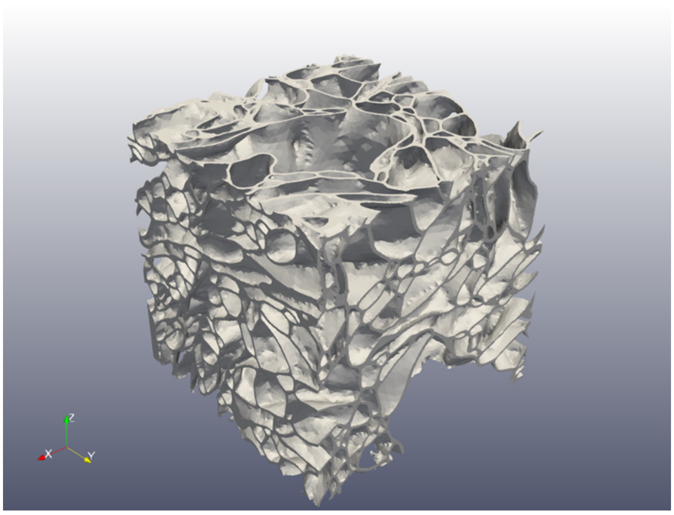


Figure A.1: Different view of the neuropil EM scan.

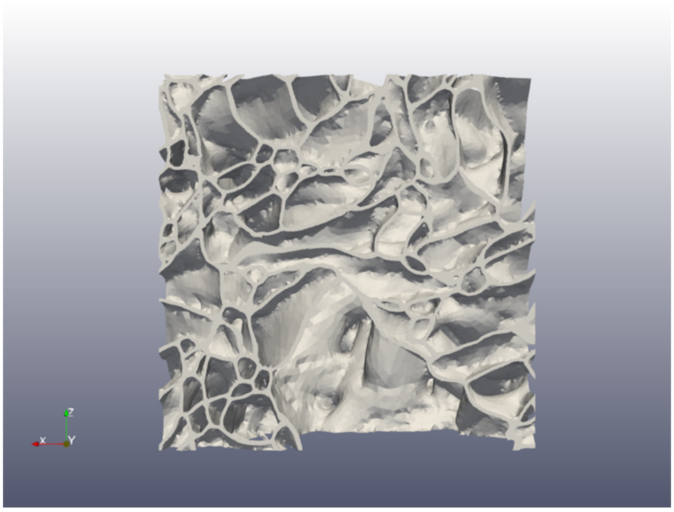


Figure A.2: View on an x,z-plane of the neuropil

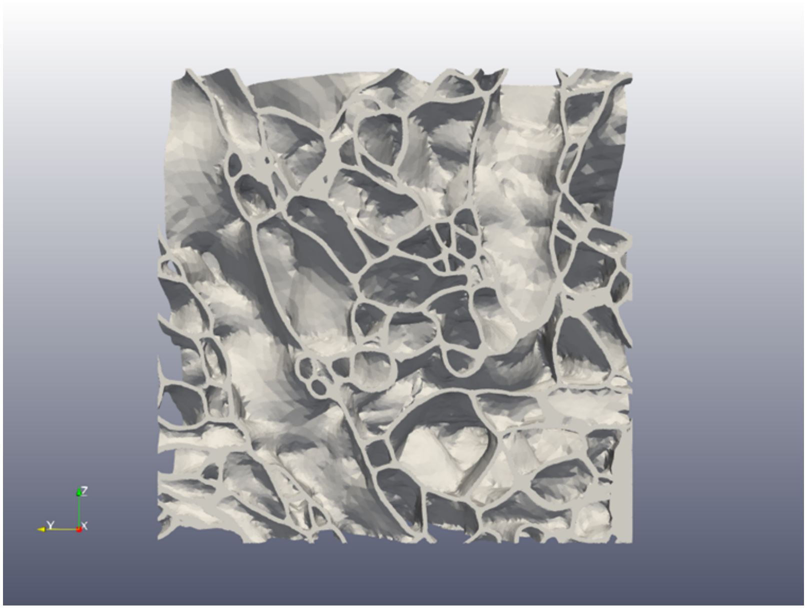


Figure A.3: View on an y,z-plane of the neuropil

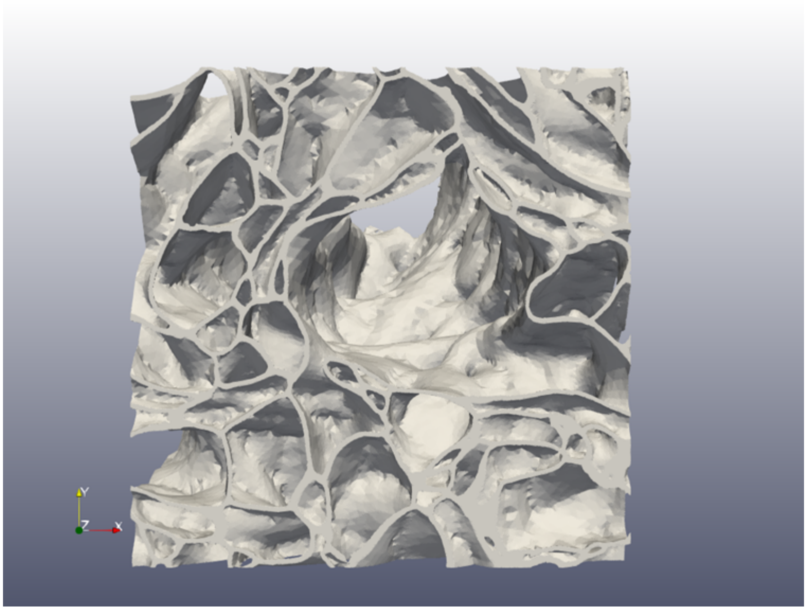


Figure A.4: View on an x,y-plane of the neuropil

Appendix B

Appendix B

More zooms on slices with the velocity profile and corresponding vector arrows

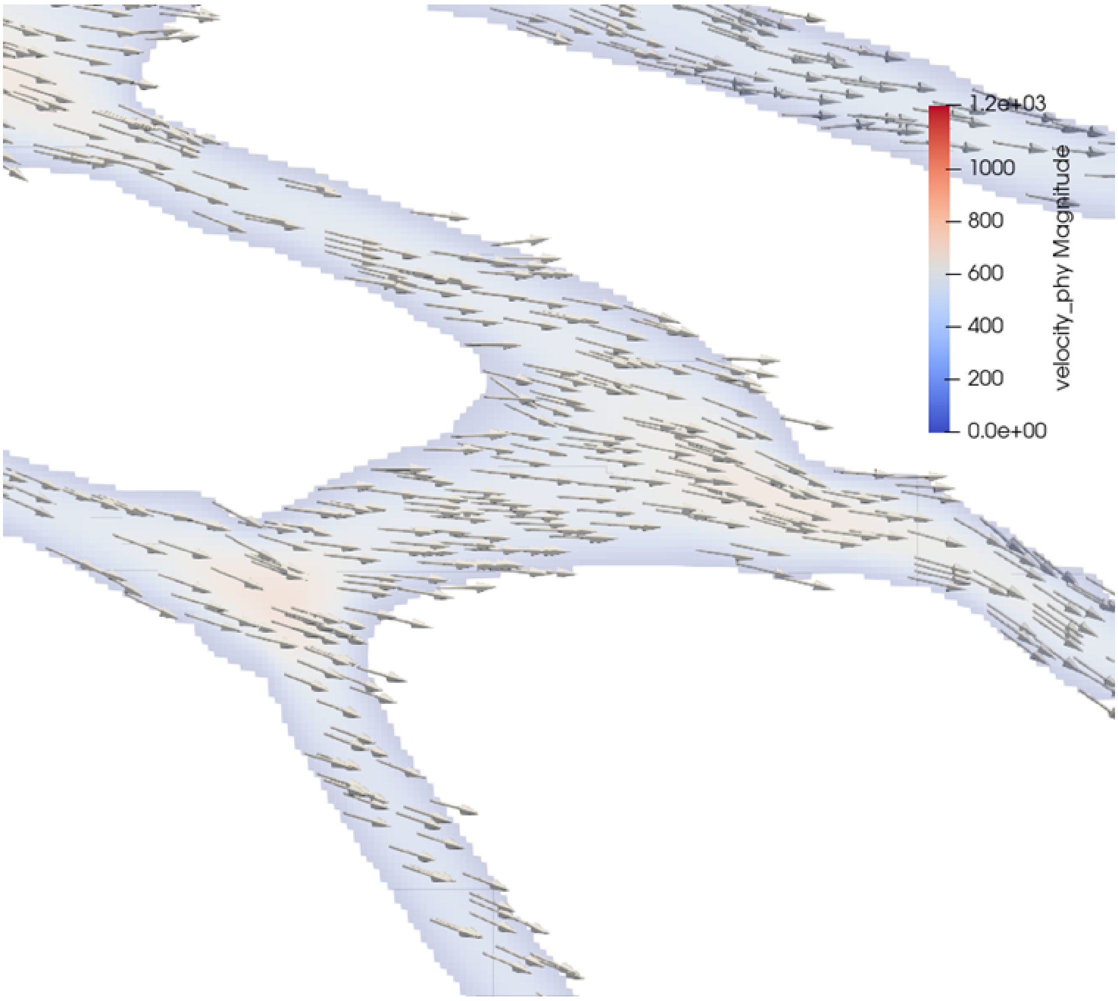


Figure B.1: Velocity field with vector arrows

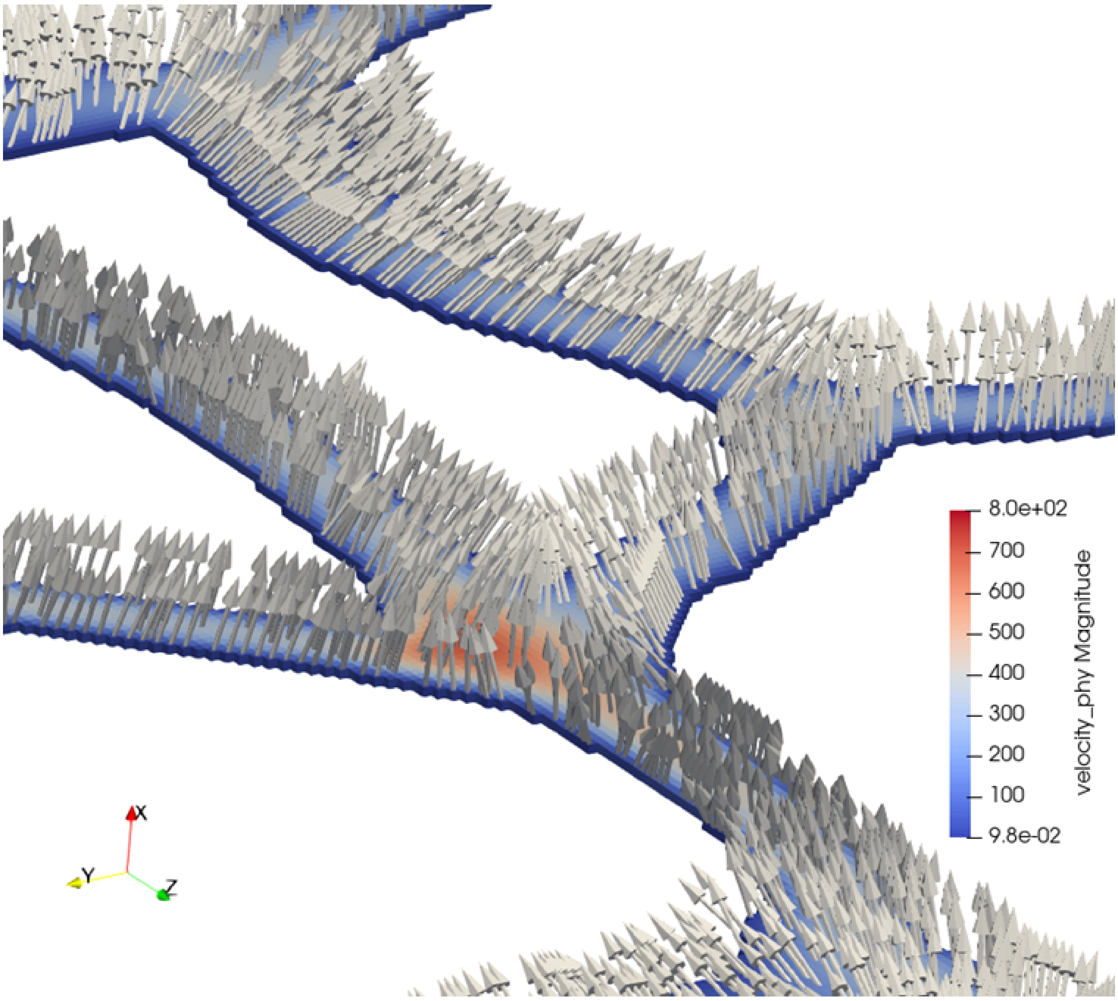


Figure B.2: Velocity field with vector arrows

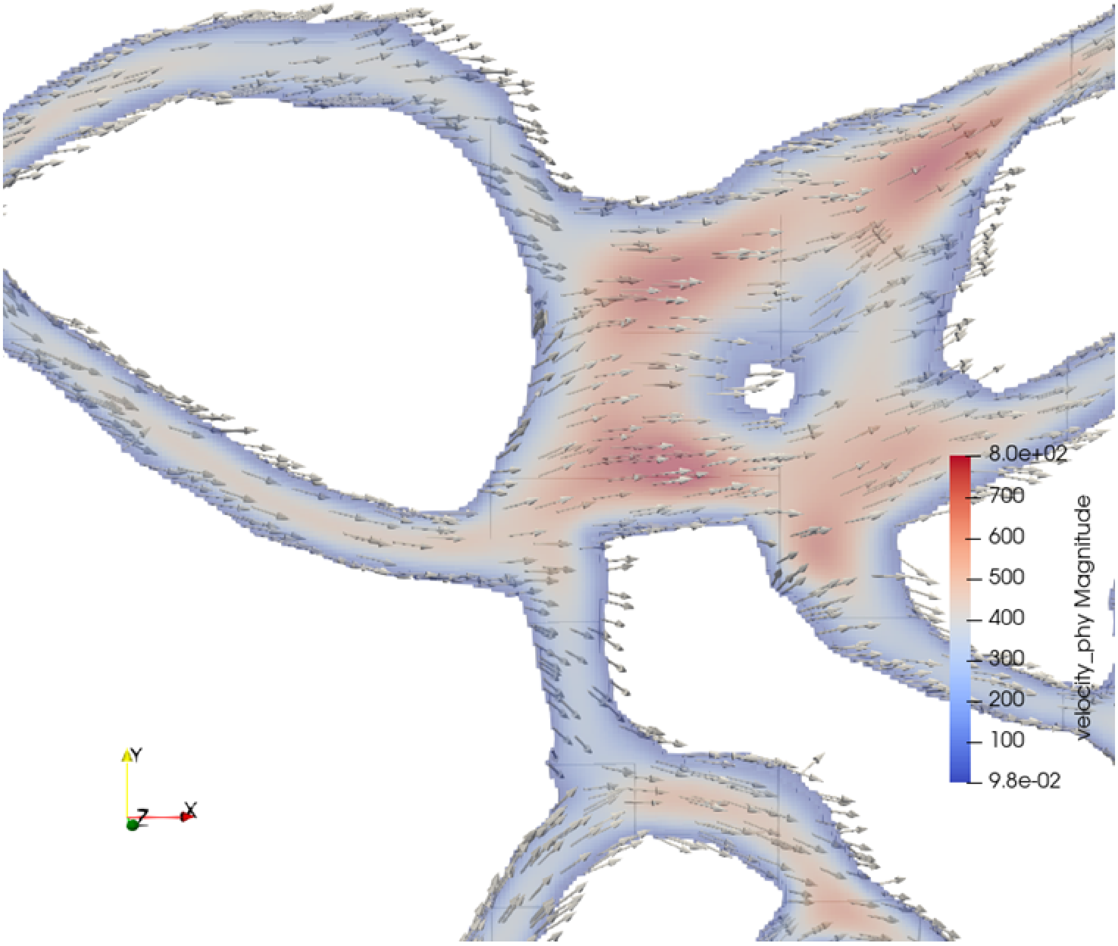


Figure B.3: Velocity field with vector arrows

Velocity in a slice in the last 0.1 second of the simulation

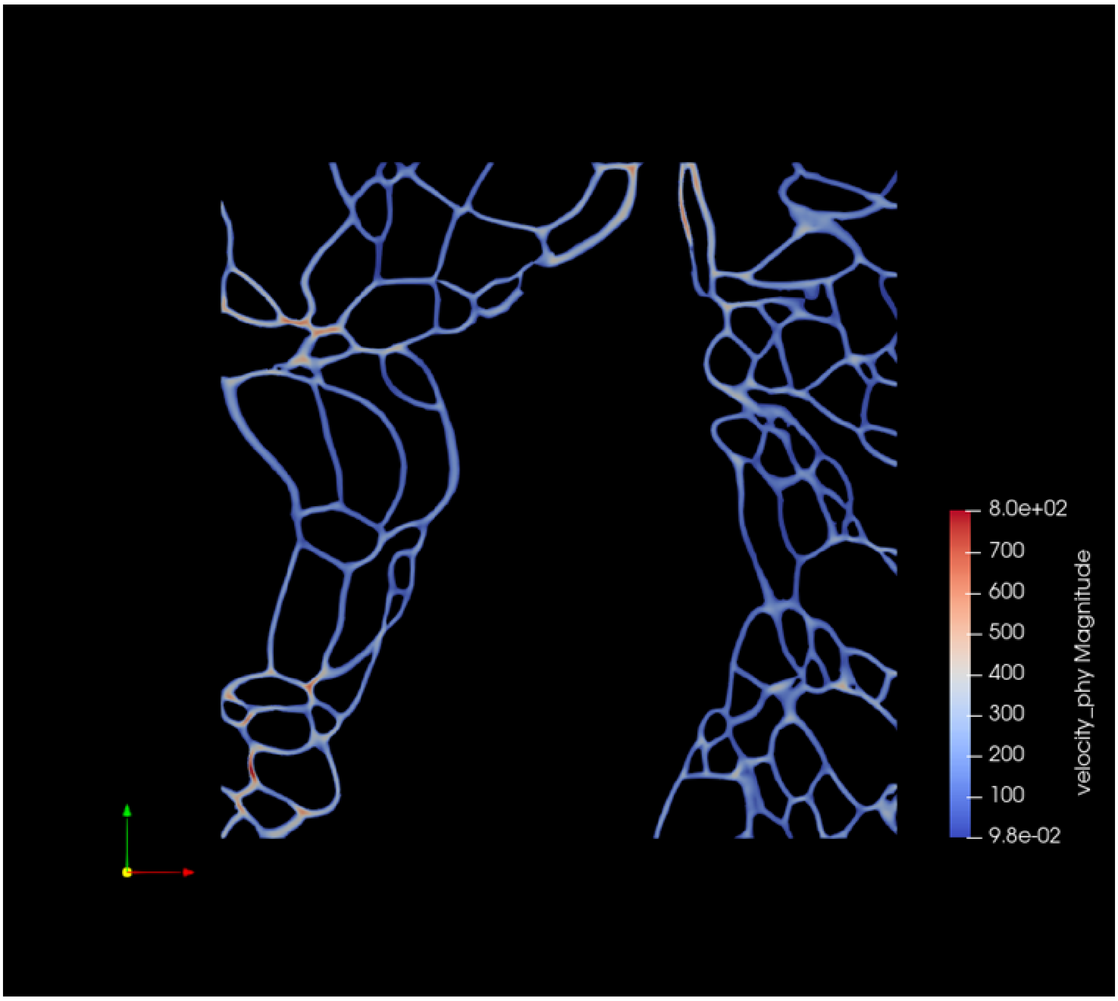


Figure B.4: Neuropil slice at t=0.93 s

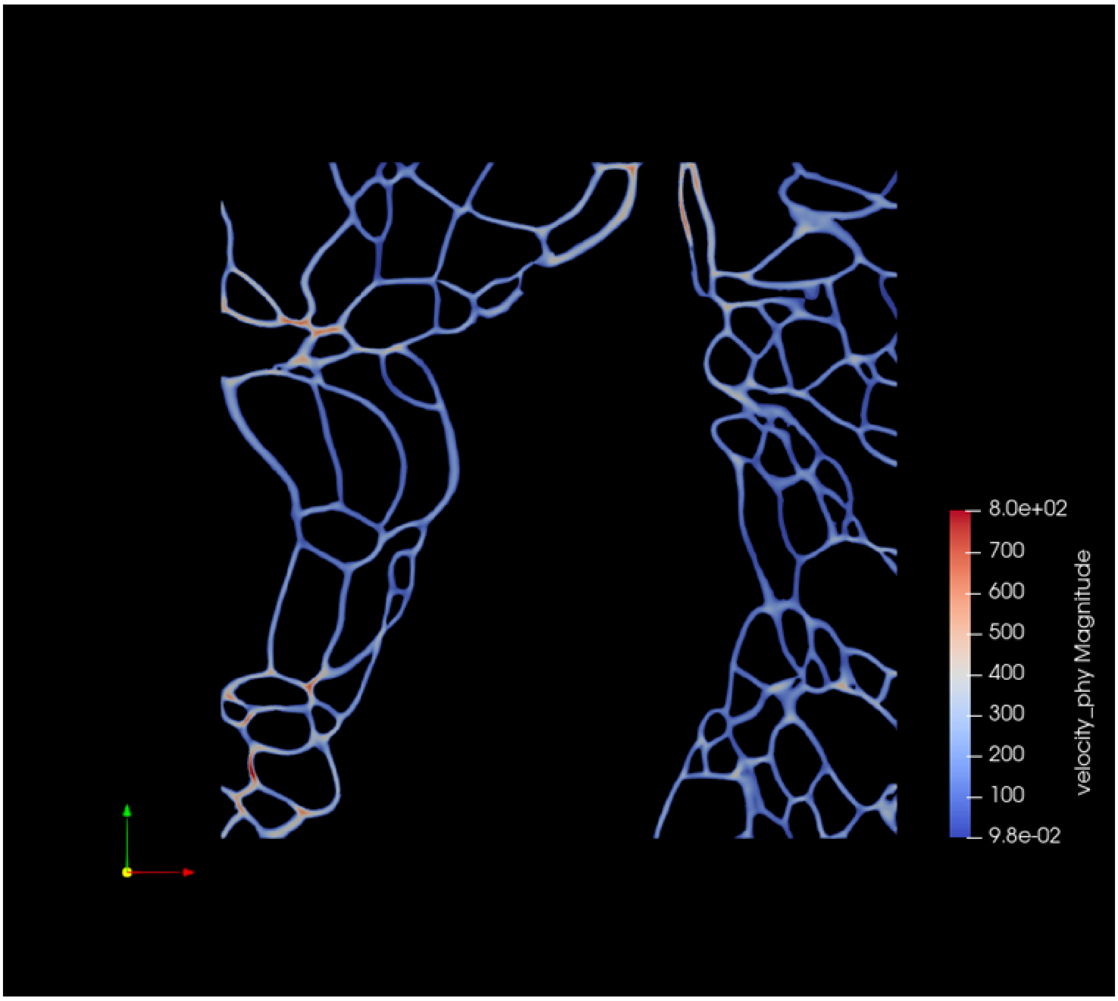


Figure B.5: Neuropil slice at t=0.95 s

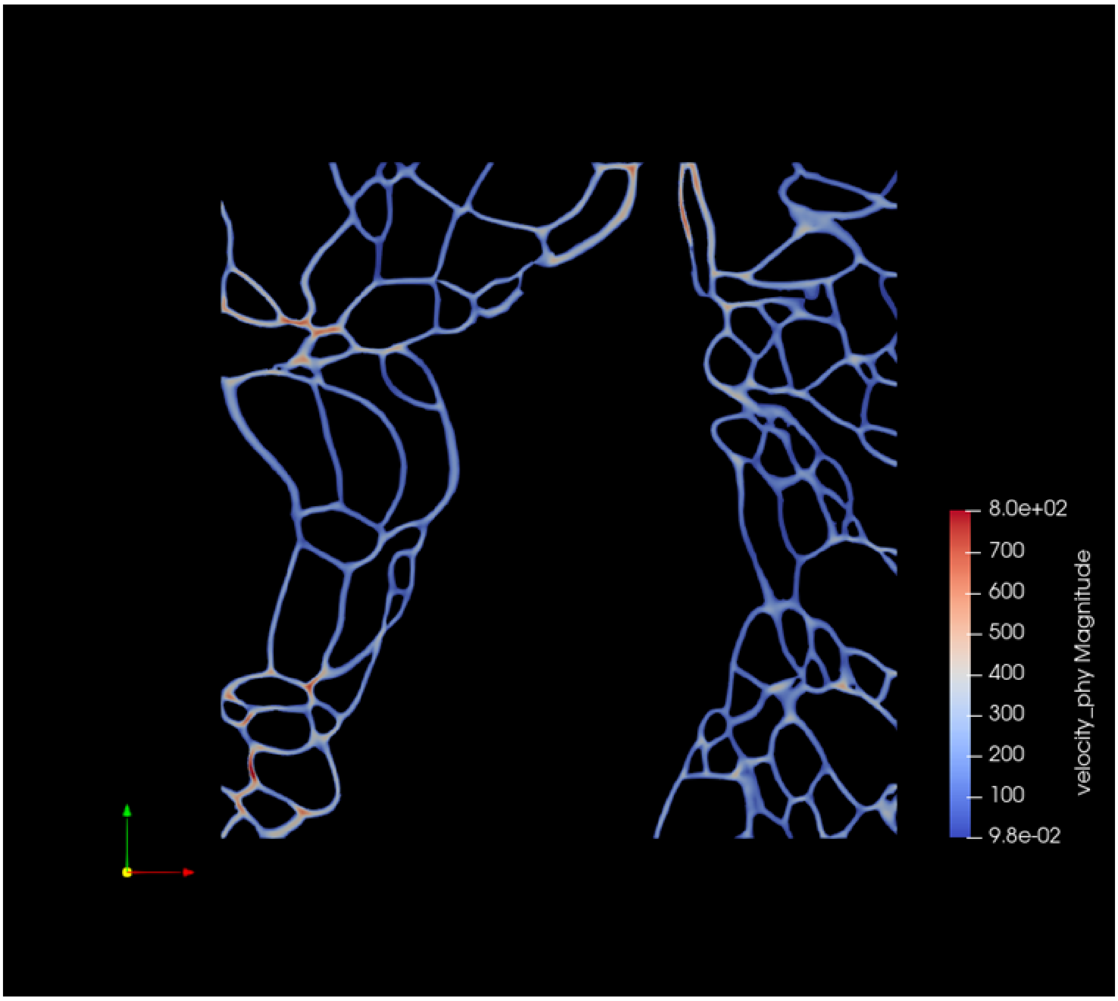


Figure B.6: Neuropil slice at t=0.97 s

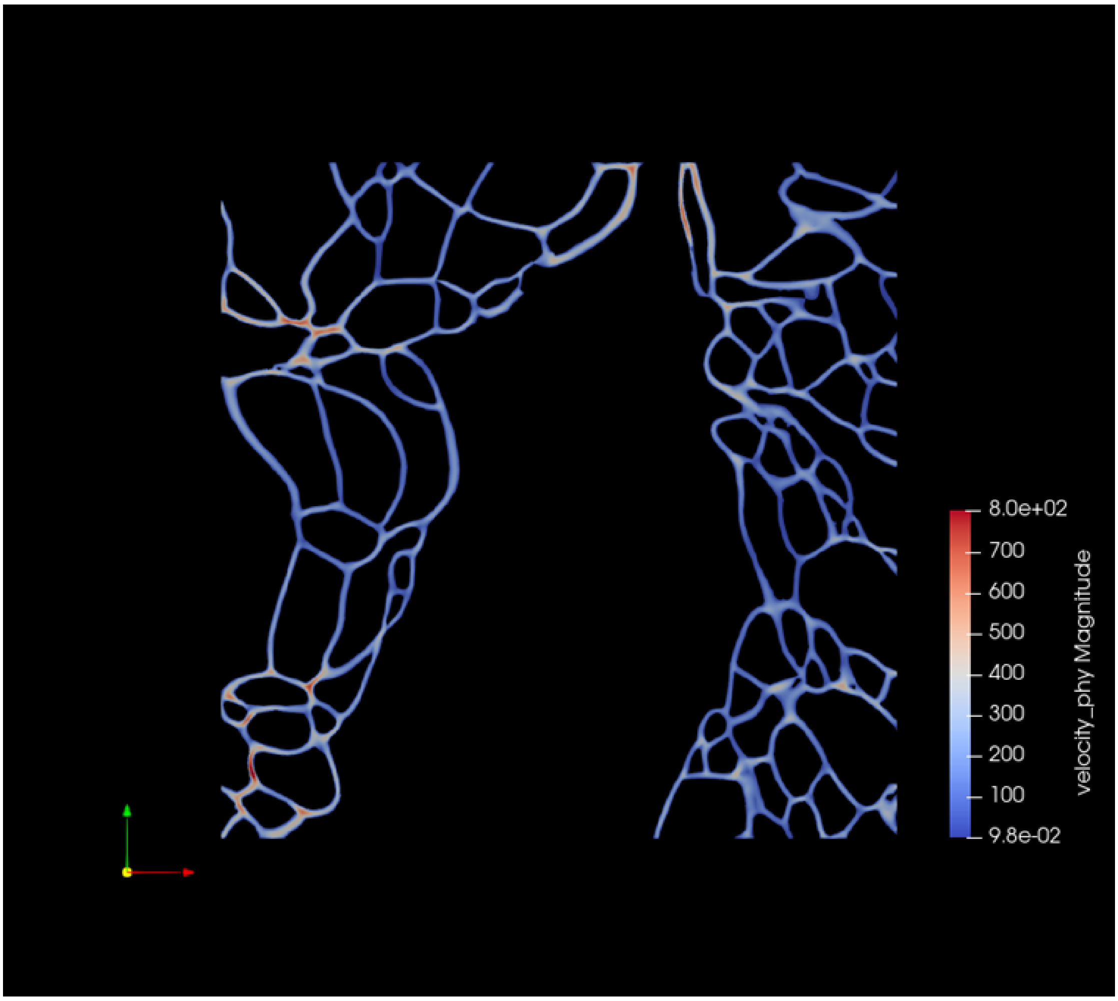


Figure B.7: Neuropil slice at t=0.99 s