Lars Magnus Valnes

Patient-specific Modeling of the Human Brain using Magnetic Resonance Imaging

Thesis submitted for the degree of Philosophiae Doctor

Department of Mathematics
The Faculty of Mathematics and Natural Sciences

Department of Numerical Analysis and Scientific Computing
Simula Research Laboratory

2019
Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of Philosophiae Doctor at the University of Oslo. The research presented here is conducted under the supervision of professor Kent-Andre Mardal, PhD Erika Lindstrøm, professor Per Kristian Eide, chief research scientist Marie E. Rognes and associate professor Unn Haukvik.

The thesis is a collection of three papers and a book excerpt that are presented in a practical order. The recurring theme is the usage of magnetic resonance imaging (MRI) in computational modeling of the human brain. The papers and the book excerpt are preceded by an introductory chapter that relates them together and provides background information and motivation for the work. The first and second paper cover interdisciplinary studies with MRI analysis performed by the author. Then, we have the book excerpt focusing on the approaches and software that were developed for this thesis and future work in bio-mechanical brain modeling. The third paper combines the first paper and the book excerpt into a computational relevant study.

Acknowledgements

This thesis is the result of four years at Department of Mathematics. It has been a time with great colleagues, ski trips and social events. Firstly I would like to thank my supervisors, Kent-Andre Mardal, Erika Lindstrøm, Marie E.Rognes, Per Kristian Eide and Unn Haukvik. I would like to thank Marie E.Rognes and Simula Research Laboratory for arranging a weekly journal club that was interesting and educational for my thesis. Furthermore, I would like to thank PhD Travis Thompson for his efforts in co-writing the initial book manuscript. I would also like to thank PhD Anders Dale for providing guidance during my visits to his lab in San Diego. Next, I would like to thank my office neighbours, PhD Diako Darian and Tormod Landet for great company. Finally, I would like to thank my family and friends for unconditional support.

Lars Magnus Valnes
Oslo, December 2019
List of Papers & Book Excerpt

**Paper I**


**Paper II**


**Book Excerpt III**


**Paper IV**


The published papers are reprinted with permission from <publisher(s)>. All rights reserved.
Contents

Preface i

List of Papers & Book Excerpt iii

Contents v

1 Introduction 1

1.1 Motivation and aim 1
1.2 Brain anatomy 2
1.3 Normal pressure hydrocephalus 6
1.4 Possible pathology of dementia 6
1.5 Magnetic resonance imaging 8
1.6 MRI in neuroscience 9
1.7 Bio-mechanical modeling 9
1.8 Porous media modeling 13
1.9 Modeling of waste clearance 14
1.10 Diffusion equation 16
1.11 Extracellular diffusion modeling 16
1.12 Finite element method 17
1.13 Adjoint state method 17
1.14 Computational geometries 18
1.15 Summary of papers 19
1.16 Other contributions 25
1.17 Future work and outlook 25
References 26

Papers & Book Excerpt 38

I Brain-wide glymphatic enhancement and clearance in humans assessed with MRI 39

II Delayed clearance of cerebrospinal fluid tracer from choroid plexus in idiopathic normal pressure hydrocephalus 57

III An introduction to basic meshing and mathematical modeling for the human brain 69

IV Can diffusion alone explain brain-wide distribution of CSF tracers within 24 hours? 163
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC</td>
<td>Apparent Diffusion Coefficient</td>
</tr>
<tr>
<td>ASL</td>
<td>Arterial Spin Labeling</td>
</tr>
<tr>
<td>CNS</td>
<td>Central Nervous System</td>
</tr>
<tr>
<td>CSF</td>
<td>Cerebrospinal Fluid</td>
</tr>
<tr>
<td>CT</td>
<td>Computerized Tomography</td>
</tr>
<tr>
<td>DTI</td>
<td>Diffusion Tensor Imaging</td>
</tr>
<tr>
<td>ECS</td>
<td>Extracellular Space</td>
</tr>
<tr>
<td>EFV</td>
<td>Endoscopic Fourth Ventriclestomy</td>
</tr>
<tr>
<td>ETV</td>
<td>Endoscopic Third Ventriclestomy</td>
</tr>
<tr>
<td>FEM</td>
<td>Finite Element Method</td>
</tr>
<tr>
<td>fMRI</td>
<td>functional Magnetic Resonance Imaging</td>
</tr>
<tr>
<td>FSI</td>
<td>Fluid-Structure Interaction</td>
</tr>
<tr>
<td>iNPH</td>
<td>idiopathic Normal Pressure Hydrocephalus</td>
</tr>
<tr>
<td>IPAD</td>
<td>Intramural Perartierial Drainage</td>
</tr>
<tr>
<td>ISF</td>
<td>Interstitial Fluid</td>
</tr>
<tr>
<td>MRI</td>
<td>Magnetic Resonance Imaging</td>
</tr>
<tr>
<td>MPET</td>
<td>Multi-network Poroelasticity Theory</td>
</tr>
<tr>
<td>PC-MRI</td>
<td>Phase Contrast Magnetic Resonance Imaging</td>
</tr>
<tr>
<td>PDE</td>
<td>Partial Differential Equations</td>
</tr>
<tr>
<td>PVS</td>
<td>Paravascular Space</td>
</tr>
<tr>
<td>REF</td>
<td>Reference Group</td>
</tr>
<tr>
<td>SAS</td>
<td>Subarachnoid Space</td>
</tr>
<tr>
<td>SMC</td>
<td>Smooth Muscle Cells</td>
</tr>
<tr>
<td>SVMTK</td>
<td>Surface Volume Meshing Toolkit</td>
</tr>
<tr>
<td>VMTK</td>
<td>Vascular Modelling Toolkit</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation and aim

Dementia is a devastating neurodegenerative disease that had an estimated global cost of 818 billion USD in 2015 [76]. It was estimated in [85] that the number of patients diagnosed with dementia in Europe will increase from 6 million in 2010 to 14 million in 2050. This makes dementia a major concern for the future health-care, and continued research can help to combat this trend.

Dementia research covers many different forms of research fields, for example genetics [53], clinical trials [132], and pharmacology [90]. The research often utilizes imaging techniques to visualize the brain, and larger studies often include patient cohorts by the hundreds. Thus, processing software have been used to analyze larger cohorts, for instance 1100 participants in [38]. The processing software also allows for accurate data comparison between studies, since it removes human bias and errors.

A new hypothesis has been proposed concerning the cause of dementia [62], and it suggests that dementia is essentially a fluid mechanical problem. Although controversial, this hypothesis is widespread, with over 1500 citations since it was published in 2012 [47], creating new avenues for bio-mechanical research of the brain.

The approach of computational bio-mechanics has become well established in cardiovascular research through the years, with studies modeling the heart [6], aneurysms [70] and blood clots [83]. Modeling can provide supplementary information that can be useful in evaluating problems in a clinical setting. The research has also developed tools and methods, which can be applied to other fields of bio-mechanics. Therefore, with the new hypothesis, it seems appropriate to establish and use bio-mechanical modeling in the brain.

Currently, there is a gap between bio-mechanical research and other research fields focusing on the brain, such as biochemistry and electrophysiology, which have been well established. In particular, the advanced imaging processing tools have not been fully utilized. Therefore, this thesis aims to shorten this gap, by introducing methods and proceedings that can be of aid. This thesis has three main objectives:

- Use semi-automatic processing software with high-throughput to analyze data [Paper I] and [Paper II].

- Develop tools and approaches concerning medical images and processing software to make bio-mechanical models with patient-specific meshes for large cohorts [Book Excerpt III].
1. Introduction

- Make use of the developed tools and approaches to perform relevant biomechanical simulations (Paper IV).

The introduction will contain information that can be of use when reading this thesis. We will include a brief introduction to the brain anatomy, terminology and brain related characteristics in the next section. This will mainly be based on medical textbooks, but also referenced articles. Then, we will introduce two different waste clearing hypotheses by Iliff et al. [62] and Carare et al. [18].

We will provide a short introduction on the imaging technique magnetic resonance imaging (MRI), and on the MRI processing software FreeSurfer. However, these elements will be covered in more detail in Book Excerpt III. Finally, we will also describe the current state of bio-mechanical modeling and waste clearance in the brain.

1.2 Brain anatomy

In this section, we will go through some basic brain anatomy. We start with the brain anatomy and terminology, before continuing with the fluid flow in and around the brain. Most of the information provided in this section is from the medical textbook "Human Anatomy and Physiology" [78] if not otherwise specified.

![Brain anatomy diagram](Image taken from Bear [9])

Figure 1.1: The image shows the main components of the brain: Cerebrum, Cerebellum and Brain stem. (Image taken from Bear [9])
The brain is an organ of the central nervous system (CNS), and consists of the cerebrum, the cerebellum and the brain stem, see Figure 1.1. The brain has high metabolism and constitutes up to 20% of the basal metabolism [82]. Furthermore, it receives 15% of the cardiac output and 20% of total body basal oxygen consumption [109]. The brain together with the spinal cord makes up the human CNS.

The cerebrum is the largest part of the brain, and consists of two cerebral hemispheres. These hemispheres are joined at the brain stem, and are connected via commissural nerve tracts, with the largest being the corpus callosum. We divide each hemisphere into four regions; frontal, occipital, parietal and temporal lobe. Additionally, MRI brain images are represented as a collection of planar images slices of the brain, and the labeling of these slices is dependent on the slice direction, see Figure 1.2. The side-view of the brain is known as the sagittal view, the front view is known as the coronal view, and the above view is known as the axial view.

**Brain tissue**

We distinguish between two types of tissue in the cerebrum, namely white and gray matter. The cortical gray matter is often referred to as the cerebral cortex, and is formed as a layer around the white matter. We also have sub-cortical gray matter, which is also found within the white matter and close to the brain stem. The sub-cortical gray matter consists of important structures, such as the basal ganglia, the hippocampus and amygdalae to name a few. In general, the gray matter tissue consists of neurons, arteries, veins and supportive glia cells, such as astrocytes and ependymal cells. The cerebral cortex has a unique structure with multiple folds, and the ridge and furrow are respectively known as gyrus and sulcus, see Figure 1.3.

In the white matter tissue we find axons, which connect the neurons in the gray matter. These axons create tracts inside the white matter, resulting in an anisotropic structure in the white matter [20]. We often denote the white and gray tissue as the brain parenchyma. The brain cells make up a solid structure in
1. Introduction

Figure 1.3: The image shows the folding of the cerebral cortex with sulcus and gyrus. (Image from Wikimedia Commons [130])

the tissue, and the narrow spaces between the cells are known as the extracellular space (ECS), which is filled with interstitial fluid (ISF).

Meninges

The brain is protected by the skull and layers of meninges, which are located between the skull and the brain. The meninges consist of three membranes; the pial, the arachnoid and the dura, which are shown in Figure 1.4.

The region between the arachnoid and the pial membrane is known as the subarachnoid space (SAS). The SAS space is filled with cerebrospinal fluid (CSF), which consists of 99% water [17], and is important for regulating the brain functions [110]. The SAS stretches along the spine down to the lumbar region, and it is also connected to the ventricular system in the cerebrum through the cisterna magna.

Ventricular system

The ventricular system consists of four connected compartments that are filled with CSF. These compartments are known as the two lateral, 3rd and 4th ventricles, as shown in Figure 1.5. Figure 1.5 also illustrates the perceived understanding of the CSF circulation in the brain. The production of CSF was estimated to be around 500 ml/day [49], and the CSF is thought to be mainly produced in the choroid plexus. The choroid plexus consists of blood vessels and specialized ependymal cells and is present in each of the ventricles. We will go in more detail about the functions of the choroid plexus in Paper II. In the traditional view by Cushing et al. [23], the CSF is produced by the choroid plexus in the lateral ventricles. It flows into the 3rd ventricle and into the cerebral aqueduct, which is the narrow channel between the 3rd and 4th ventricles. Then the CSF flows from the 4th ventricle and into the SAS, and is later absorbed by
Figure 1.4: The image shows the structure from the skull to the cerebrum, including the meninges; dura mater, arachnoid and pia mater. (Image from Wikimedia Commons [130])

the arachnoid granulations near the upper skull, and also into superior sagittal sinus.

**Peri- and paravascular spaces**

Peri- and paravascular spaces are small fluid-filled spaces that exist around the perforating vessels. These spaces are also called Virchow-Robin spaces after being described by Virchow [129] and Robin [99] in the middle of the 19th century, and knowledge has since improved and changed. The terms perivascular and paravascular are often used interchangeably, which can cause some confusion, since the terms can also describe two different types of fluid-filled pathways. In Bakker et al. [7], the perivascular space is described as a space within basement membrane between the smooth muscle cells (SCM) of arterioles and arteries. While the paravascular (PVS) describes the channels outside the SMC, enclosed by the pial membrane and glial limitans formed by astrocyte end feet, which separates the PVS from the brain parenchyma. The reported characteristic gap height of the PVS spaces is 10µm [5, 87], while the basement membrane has a reported thickness of approximately 100 nm [28, 105]. There is still some debate about the connection between the SAS and the PVS. In Iliff et al. [62], the SAS and PVS were reported to be separated by a thin pial sheet. While in the work of Bedussi et al. [10], the SAS and PVS were considered a single compartment, and the PVS were regions in the SAS with lower resistance to flow.
1. Introduction

Figure 1.5: The image illustrates the traditional view by Cushing et al. [23] on the circulation of the CSF in and around the brain parenchyma. (Image from Wikimedia Commons [130])

1.3 Normal pressure hydrocephalus

In this thesis we will mainly focus on a patient-cohort, diagnosed with a subtype of dementia known as idiopathic Normal Pressure Hydrocephalus (iNPH). The disease is most common in elderly adults, with a reported prevalence of probable iNPH 21.9/100 000 in the general population, and 181.7/100 000 for the age-group 70-79 years [15]. Furthermore, it has similarities with other neurodegenerative diseases, such as Alzheimer’s disease [107]. Patients with iNPH have excess amounts of CSF in the lateral ventricles and that causes them to expand, shown in Figure 1.6. There are three symptoms associated with iNPH, namely gait disturbance, urinary incontinence and dementia [96].

Unlike most types of dementia, early onset of iNPH can be reversed, but requires surgery, known as shunting. The surgeon places a catheter in the lateral ventricles that drains the excess CSF into the abdomen. The patients with shunting surgery had a varied response rate from 15% to 98% [50, 123], and the mean rate of complications related to shunting was estimated to be 38% based on 44 articles [52]. Shunting surgery is highly risky, and finding a new bio-marker can greatly improve the shunting success rate.

1.4 Possible pathology of dementia

The cause of dementia and Alzheimer’s disease is still unknown, but studies have reported that Alzheimer’s disease is linked to the accumulation of harmful
proteins, such as amyloid-beta and tau in the brain [8, 51]. This accumulation of amyloid-beta will create amyloid-beta plaques, which are believed to be responsible for the pathology of Alzheimer’s disease. The presence of amyloid-beta plaques was detected in 42% of patients with possible iNPH [73], and 75% with severe iNPH dementia [46].

We produce these proteins as a by-product of the metabolism, but the proteins are normally cleared from the brain. In the rest of the body, the lymphatic vessels clear the waste proteins, but there are no lymphatic vessel in the brain parenchyma [2]. Since the brain has a high metabolism [82], there must be some mechanism for clearing the harmful waste proteins.

In 2012, Iliff et al. [61] proposed a new hypothesis of the waste clearance from the brain, denoted as the glia-lymphatic or "glymphatic" system. The paper details the observations that tracers injected into the cistern magna in the SAS, which quickly entered the cerebrum tissue, and similar observations was also reported by Rennels et al. [97] in 1985.

The proposed hypothesis [65] claims that there is a CSF influx in the PVS along the arteries in the brain. In the PVS, the CSF and ISF are mixed through low-resistant pathways known as astroglial AQP-4 channels. Then, in cerebrum tissue, a bulk flow in the ECS transports the waste from arterial capillaries to venous capillaries. The waste proteins are cleared along the paravenous spaces, the hypothesis is illustrated in Figure 1.7. The influx along the penetrating arteries have recently been supported by in-vivo observations in rats [80].
1. Introduction

Figure 1.7: The image illustrates the waste clearance according to the glymphatic hypothesis. The glymphatic system works with the following process; a CSF influx from the SAS along the para-arterial space of an artery; mixing of CSF and ISF through AQP-4 channels; a convective flow from the artery to the vein that transports waste from the ECS to the paravenous space; clearing of waste from the brain parenchyma with a paravenous efflux. (Image taken from Jessen et al. [65])

The glymphatic hypothesis suggests that dementia can be caused by insufficient clearing of waste from the brain parenchyma, such as amyloid-beta, which causes the creation of plaques. Thus, understanding the fluid interaction in the brain can help create new methods to counteract the progression of dementia. However, the waste clearance of the brain is still debated, and there are conflicting hypotheses. One of these hypotheses was based on the work of Carare et al. [18], and called the intramural periartrial drainage (IPAD). This hypothesis states that the waste and interstitial fluid are eliminated from the brain along the basement membranes of capillaries and arteries, thus along the periarterial space and not along the PVS. In this hypothesis, it is believed that Alzheimer’s disease is caused by the stiffening of the arterial walls. This is mostly due to aging and arteriosclerosis, and causes the waste clearance of IPAD to fail. This leads to accumulation of amyloid-beta in the arterial walls, which is supported by observations of vascular deposition of the amyloid-beta protein in the perivascular spaces [68].

1.5 Magnetic resonance imaging

In this section, we will go through some basic principles of MRI, based on the books [9, 74]. Furthermore, we will only mention a few applications, since the applications will be more detailed in Book Excerpt III.
MRI is a tomographic technique, which uses a stable hydrogen nuclei as the source for obtaining information. This makes MRI preferable to other techniques, such as X-ray and computerized tomography (CT), which uses radiation to obtain information. The MRI technique utilizes the fact that the hydrogen nuclei has two energy states related to the magnetic moment of the nuclei. In general, these energy states are evenly distributed, but by applying a strong external magnetic field, the magnetic moments of the nuclei will become parallel after a time $T_1$, known as the longitudinal relaxation time. Then we use electromagnetic waves with the Larmor resonance frequency to excite hydrogen atoms so that we can measure the spatial distribution of hydrogen atoms. This causes the MRI-signal to have different signal values based on the hydrogen content, which makes it suitable to distinguish between different types of soft tissues, like gray and white matter.

MRI was introduced in the 1980's as a diagnostic tool, and has become important in diagnosing patients with neurodegenerative disease. The MRI technique has also provided different modalities of MRI images, such as T1-weighted MRI, functional-MRI (fMRI), Diffusion Tensor Imaging (DTI), Arterial Spin Labeling (ASL) and Phase Contrast MRI (PC-MRI), which provide additional information. As mentioned, we will go in more detail about the MRI modalities in Book Excerpt III.

1.6 MRI in neuroscience

The different modalities make MRI attractive for neuroscience, and since it is non-invasive, it allows for longitudinal studies with many participants. However, the manual analysis of a single participant can take a long time, therefore image analysis and software has been developed. Among the developed tools, we have 3D-Slicer [35], FSL [64], and FreeSurfer [37], to name a few.

The software FreeSurfer is a central part of this thesis, as analyzing tool and a starting point for developing additional tools for bio-mechanical modeling. FreeSurfer provides a highly autonomous process that divides the brain into different regions, displayed in Figure 1.8, called segmentation [27]. This makes it possible to effectively analyze larger data sets, and also removing the cognitive bias in the segmentation. Additionally, FreeSurfer can also be used for estimating the thickness of the cerebral cortex, longitudinal analysis, fMRI analysis and group analysis. This has contributed to the use of FreeSurfer in a variety of research, ranging from schizophrenia [111] and dementia [72] to mild traumatic brain injury [138].

1.7 Bio-mechanical modeling

Bio-mechanical modeling is a valuable resource that can provide a unique perspective on the understanding of human physiology. It helps medical doctors and researchers to evaluate surgical implants, to tailor new medical equipment and to determine the best possible treatment. This is based on the fact that
computational models does not pose a risk to the patient and can effectively provide supplementary information that can be used for disease investigations and risk assessments.

The fundamental framework of bio-mechanical modeling has been detailed in the books by Fung [39, 40], who pioneered the field. This framework has since lead to bio-mechanics becoming a well-established tool in cardiovascular research with books by Quarteroni [93] and Holzapfel et al. [58] that detail the bio-mechanical methods for cardiovascular modeling.

Developed methods in cardiovascular modeling have dealt with similar mathematical difficulties that are currently present for brain modeling, such as fluid flow, fluid-structure interactions (FSI), medical parameters, boundary conditions and tissue modeling. Therefore, we will demonstrate bio-mechanical methods for brain modeling by using established cardiovascular models as reference. We will focus on models constructed using partial differential equations (PDE), which consist of unknown functions with multiple variables and their partial derivatives. PDEs are commonly used to model mechanical problems, such as fluid flow with incompressible Navier-Stokes equations and solid deformations with linear elasticity. We can solve PDEs with different methods, like finite differences, finite elements and finite volumes to name a few, which are well documented in the mathematical literature and a brief overview of the finite element method will be given in Section 1.12.

Soft tissue deformations are a central part of bio-mechanics, and in this section we will detail some of the developed models. The construction of tissue models requires solid mathematical foundation and we have two different types of theory for handling deformations; infinitesimal elasticity theory for small deformation and finite elasticity theory for larger deformations. For small deformations, soft tissue can be approximated with linear elasticity and the infinitesimal elasticity theory [113]. However, this is not the case for larger deformations of soft tissue, which exhibit non-linear mechanical properties. The mechanical properties of the
Brain tissue have been investigated with experimental studies, and have reported that the brain tissue can be considered extremely soft, i.e. highly non-linear and viscoelastic \cite{16}. This is due to the fact that the brain tissue can be considered biphasic with 79% water by weight content \cite{137} and exhibits hyperelastic properties \cite{67}. The Ogden model is a hyperelastic model that is frequently used to model tissue by expressing the strain-energy as a non-linear summation of the principal stretches \cite{89}. We also have more tailored models, such as the Holzapfel-Ogden model \cite{59}, which is used to model soft fiber tissue, for example the heart wall (myocardium) \cite{131}. Furthermore, the water content in soft tissue makes it exhibit both fluid and solid mechanical behavior, like creep under constant load and relaxation under constant displacement. Therefore, viscoelastic tissue models have also been developed and used, for example the quasi-linear viscoelasticity model \cite{41,92} and generalized elastic-Maxwell model \cite{56,57,136}. Developed soft tissue models have been used to simulate deformations in the brain tissue in \cite{81,84,100}.

In this thesis, we want to create a model for the waste clearance in the brain, and therefore we need to determine a model that will fit with the soft tissue deformations. The model will be used to investigate the waste clearance over several cardiac cycles, so the blood flow will cause deformations due to pressure differences in the brain. Previously, we described the brain as extremely soft, but the solid deformations in the brain during a cardiac cycle are small. This is based on recent measurements, which reported that the Young’s modulus for the brain is in the range 1-4.2 kPa \cite{16,67} and that the pulsatile pressure difference over a cardiac cycle within the brain did not exceed 50 Pa \cite{128}. By Hooke’s Law, the maximum strain can be approximated to be around 0.05, which is in the regime of small deformations. Thus, we can model the waste clearance in the brain using linear elasticity and infinitesimal elastic theory. This means that we can simulate the flow and deformations in the brain with a linear poro-elastic model, which we will cover in Section 1.8.

Blood flow is frequently modeled in cardiovascular research, as it transports nutrients and waste throughout the human body. Hemorheology is the science field that covers blood flow and the interaction in a blood vessel. In \cite{116} the blood flow was modeled using the incompressible Navier-Stokes equations and solved using the finite element method. We can adapt the blood flow models to simulate the CSF flow in the narrow channels in the brain, such as the cerebral aqueduct and the PVS. This can help us understand the consequences of CSF flow obstructions in the brain, such as aqueductal stenosis \cite{21}, Chiari malformation \cite{80} and cysts \cite{75}, and evaluate different methods of treatment \cite{69,104,125}. Recently, flow modeling in the ECS, PVS and periarterial spaces have become more popular due to the waste clearance hypothesis \cite{28,55,106}, which we will cover in Section 1.9.

Unlike fluid flow in pipes, the blood vessels consist of a tissue wall that deforms with the pulsative flow. This problem is also present in lymphatic hypothesis, as shown in Figure 1.7 with the PVS encapsulating the arteries. The method to include the solid deformation is known as fluid-structure interaction (FSI) modeling, and a typical FSI model was used in Crosetto et al. \cite{22}. The
1. Introduction

study constructed a FSI model to simulate, using the finite element method, the blood flow and arterial wall deformation of a healthy aorta. It was reported that the simulation provided better comparison to experimental data than the usage of a rigid wall model. More complex FSI models exist, for example in Borazjani [13] FSI modeling was used to simulate the deformation in the heart during a cardiac cycle. This was done by combining the FSI model with a non-linear elastic model and simulated with the finite element method. Recently, FSI models were implemented to simulate CSF flow in the ECS by solid deformations in [126] and in the PVS due to arterial pulsation in [28]. These studies will be presented in more detail in Section 1.8 and Section 1.9, respectively.

Bio-mechanical models have been further developed with the intent of personalized medicine, which was covered in the review by Taylor et al. [115]. This review details general problems of patient-specific modeling, such as geometries, parameters and boundary conditions, and also highlights application areas like disease research, risk assessments, patient-specific device design and evaluation. Patient-specific risk assessment of aneurysms was studied in Gasser et al. [43] by evaluating the peak wall stress and peak wall rupture risk. This exemplifies that bio-mechanical models can help with assessing the medical situation, but we need methods to handle multiple patients. Large cohort studies have also been done with the risk assessment of the hemodynamics in cerebral artery aneurysms, with 38 aneurysms in [31] and 119 aneurysms in [135]. The first study used a processing software called the vascular modelling toolkit (VMTK) [4] that was developed with the purpose of making computational vascular geometries from medical images. We require similar processing software for the brain in order to perform larger cohort studies. This is in fact one of the aims of this thesis, and we will present the issue in Section 1.14 and more details in Book Excerpt III.

Bio-mechanical models need appropriate material parameters to provide meaningful information, and these parameters are often obtained from animal experiments and ex-vivo mechanical tests. However, mechanical properties in soft tissue may change over time because of aging and diseases, like stiffening (arteriosclerosis) of the arterial wall [79] and accumulation of plaques (atherosclerosis) in the arteries [134]. In this regard, elastography is a highly sought tool, since it allows for measuring mechanical properties in a non-invasive manner. Elastography works by applying a pressure wave into the soft tissue, measuring the response with a tomographic technique and estimating the mechanical properties based on the response. The method has been used with MR [101, 108] and ultrasound [25, 44] on different types of soft tissue, involving the brain [54]. The use of MR-elastography to obtain appropriate material properties may be important, since studies have reported stiffness changes in the brain due to neurodegenerative disease [86], such as multiple sclerosis [133], NPH [34] and different types of dementia [29].

Computer simulations can provide relevant information, but these require that the appropriate initial- and boundary conditions are considered. Boundary conditions for the Navier-Stokes equations include pressure or traction and prescribed velocity profiles at the boundary surface. In many cases, the reported values in the literature are sufficient. However, patient-specific measurements
Porous media modeling can provide additional information that can be relevant. In Vinje et al. [128], it was reported that breathing had significant impact on the CSF flow in the cerebral aqueduct. This was investigated by modeling the fluid flow with Navier-Stokes in the cerebral aqueduct using idealized geometries and comparing to both PC-MRI measurements and patient-specific geometries. The study used pressure measurements obtained from two implanted sensors in the brain to impose the boundary conditions.

The measured quantities can also be composite variables, like the flux or the averaged flow rate. In Vignon-Clementel et al. [127] the average flow rate was mapped to a parabolic velocity profile and then used as the inlet boundary condition for the velocity. The appropriate material parameters and boundary conditions can be estimated by testing different values and comparing the results with observations. However, solving the inverse problem can be computational expensive in terms of time, and variational data assimilation is an alternative.

Variational data assimilation constructs an optimization problem that tries to minimize the difference between the computational simulations and the observational data [19]. The minimization can be solved effectively by computing the descending gradient using the adjoint state method, and variational data assimilation methods have been used in cardiovascular research. In the work of Funke et al. [42] variational data assimilation was used to reconstruct the boundary conditions for blood flow in aneurysms. The assimilation robustness was tested by applying noise to the observations, and it was demonstrated that with appropriate regularization, the model accurately reconstructed the flow even with significant noise. In the work of Finsberg et al. [36], variational data assimilation was used to determine the mechanical properties in a patient-specific model of the heart. This was done for a cohort of patients with heart failure and compared with a healthy control group, and significant difference was reported in estimated contractility between the cohorts. The usage of variational data assimilation and brain modeling will be exemplified in Paper IV and the adjoint state method will be outlined in Section 1.13.

1.8 Porous media modeling

In this section, we will cover porous media modeling of brain. The solid deformation and fluid flow in porous media is often modeled using Biot’s equations for linear poroelasticity, which are derived from the equations for linear elasticity, Darcy’s law, Navier-Stokes and mass conservation. Biot’s equations were built so that the solid and the fluid were mixed and existed within the same domain [12], which made it suitable for macroscopic model of porous tissue, like the brain.

Biot’s equations can be used to model the brain, but it does not include the interaction between the different fluid compartments in the brain, like CSF, arteries, capillaries and veins. Therefore, in Vardakis et al. [125] a quasi-static extension of Biot’s equations was presented and used for brain modeling, called the multi-network poroelasticity theory (MPET). The MPET model consists of momentum balance in the porous medium coupled with continuity equations for
1. Introduction

each fluid compartment. The flow between different compartments is regulated by loading terms and it is assumed that the fluid compartments exist everywhere in the computational domain. The MPET model was used in Vardakis et al. \cite{124} with an idealized geometry of the brain and four fluid compartments, CSF, arteries, capillaries and veins. This was done to demonstrate the effect of cerebral aqueduct stenosis and fluid outlet obstruction in 4th ventricle, and the treatment with endoscopic third (ETV) and endoscopic fourth ventriculostomy (EFV). The study concluded that the treatment of EFV was preferred over ETV.

MPET has also been combined with two patient-specific models in Guo et al. \cite{48} to explore the risk factors of early onset of Alzheimer’s disease by investigating obstructions of CSF transport. Measurements of the blood flow using ultrasound were used to impose boundary conditions on the arterial blood flow. The study gave detailed description of patient-specific modeling pipeline, with construction of computational meshes using MRI.

Recently, FSI has been implemented with the MPET model in \cite{126} to investigate Alzheimer’s disease effects on the hippocampus. The FSI model was used to simulate CSF flow causing solid deformations.

Analysis of the MPET was investigated in \cite{71} by introducing the total pressure, and proved that the MPET was robust for a wide range of equation parameters. In \cite{60} a similar MPET analysis also reported stability for the equation parameters, but used strongly mass-conservative discretizations to obtain the robustness.

1.9 Modeling of waste clearance

Waste clearance hypotheses are difficult to verify using in-vivo experiments with human brains, therefore computational models have been used to explore the mechanism of waste clearance. The models often focus on a single element of the waste clearance circulation, and focus mostly one hypothesis. We will look at modeling of the waste clearance hypothesis that involves solute transport in the PVS, periarterial space, SAS and in the ECS.

Solute transport in the ECS was modeled by solving the Navier-Stokes and convection-diffusion equation in Jin et al. \cite{66} with a realistic geometry of the ECS. In the work of Holter et al. \cite{55} the Stokes equations were solved using a specific geometry obtained by scanning a (4070nm, 3690nm, 4080nm) tissue cube from the cerebral cortex of a rat. Both studies tested the required pressure difference to obtain a bulk flow that could clear the waste proteins, and both concluded that the pressure difference required to transport the waste, was unlikely to occur in the ECS. Recently, in the work of Ray et al. \cite{95} simulation was used to estimate the upper limit for superficial convective flow, which was found to be approximately 50 $\mu$m/min. The paper suggests that bulk flow still can contribute in the transportation of large molecules, since the diffusivity decrease with increased molecular mass.

Studies have also investigated the peri- and paravascular spaces and the proposed methods on how the waste is actually cleared from the brain itself.
In the work of Faghih et al. [32], the plausibility of bulk flow was studied in perivascular, paravascular and paravenous spaces. This was done using a one-dimensional branching model, and it was concluded that circulation driven by steady pressure was implausible in the current model. The study of Rey et al. [98] looked at the solute transport in the PVS by using resistance network model. It was concluded that the diffusion would dominate in the PVS, and contribution from bulk flow would be negligible.

The role of the arterial pulsation was addressed in the work of Iliff et al. [63], which suggests that arterial pulsation facilitates the interaction between the paravascular CSF and the ISF. Thus, several studies have investigated if arterial pulsation could be the driving force for solute transport, and the effect of local arterial pulsation in the spinal cords PVS was examined in Bilston et al. [11]. The fluid dynamics were modeled by the Navier-Stokes equation, and the arterial deformation was modeled as propagating wave resembling systolic pulsation. The study concluded that the arterial pulsation could induce fluid flow in the PVS in the propagating direction of arterial flow.

In Schley et al. [102], the waste clearance purposed by the IPAD hypothesis was investigated. The study used a thin-film flow approximation of incompressible Navier-Stokes equations with two arterial pulsation models; propagating pressure wave and oscillating pressure gradient. The results indicated that during each pulse cycle, the solutes and ISF were transported along the PVS in the reverse direction of the blood flow. In Diem et al. [28] the hypothesis that the arterial pulsations could drive IPAD was tested. For this study, the basement membrane was modeled as a porous medium and approximated with lubrication theory. The model also included a purposed valve mechanism of directional permeability in the porous medium that would ensure net flow in the opposite direction of the blood flow. It was demonstrated that the arterial pulsation created velocities significantly smaller than experimental observations.

Dispersion effects in the PVS were investigated in the work of Asgari et al. [5] with a modified Navier-Stokes equations with Darcy’s law for the porous medium coupled with an advection-diffusion equation. The results showed 16% to 50% increase of solute transport in the PVS due to dispersion. Recently, the dispersion enhancement of solute transport in intrathecal, periarterial and paraarterial spaces was studied by Sharp et al. [105]. The study concluded that the solute transport would have a negligible enhancement in the periarterial space. In the PVS, the paraarterial space had more potential for enhancement, while the solute transport in the paravenous spaces was predicted to be unlikely due to low pulse pressure.

Recently, the IPAD hypothesis was tested with the addition of vasomotion of blood vessels in Aldea et al. [3]. Vasomotion describes the spontaneous oscillation in the blood vessel walls that is independent of the heart beat [8]. The study used a porous-elastic model of the basement membrane with variable permeability, and a vascular tone wave was used to model the vasomotion. The study concludes that vasomotion driven IPAD is the only mechanism that could produce the resulting flow rates that could explain the current experimental observations.
1. Introduction

It was reported in [14] that for experiments performed on rabbits, the CSF is drained into the cervical lymph nodes, located in the neck. This connects the SAS and the lymphatic system, which is a possible pathway for clearing waste transported from brain into the SAS. Patient-specific modeling of the CSF flow in the inferior cranial space of the SAS, which included the 4th ventricle and cerebral aqueduct, was done in [69]. The results demonstrated that there was negligible net flow over one cardiac cycle in the cerebral aqueduct. In Vinje et al. [128] the respiration was reported to be the main cause for fluid movement in the cerebral aqueduct, and not the arterial pulsation.

We have outlined studies that have modeled the waste clearance in the brain, and this was done by investigating the solute transport. The solute transport was first hypothesized to be caused by bulk flow, but the focus has since shifted to enhanced diffusion due to solid deformations. This was studied in the PVS with a FSI model, in the ECS and the periarterial space with a poro-elastic model. There are still questions regarding the strength of the diffusion enhancement, thus we will in Paper IV investigate if solute diffusion alone can explain the solute transport in the brain parenchyma.

1.10 Diffusion equation

We will model the transport of a solute using the diffusion equation with no convective flow. The diffusion equation can be derived from the continuity equation and Fick’s first law. Fick’s first law states that the transport of a solute is defined as

$$j = -D \nabla c$$  \hspace{1cm} (1.1)

with $c$ denoting the solute concentration, $j$ the concentration flux and $D$ the diffusion coefficient. Combined with the continuity equation

$$\frac{\partial c}{\partial t} + \nabla \cdot j = 0,$$  \hspace{1cm} (1.2)

we can derive the diffusion equation

$$\frac{\partial c}{\partial t} = \nabla \cdot (D \nabla c).$$  \hspace{1cm} (1.3)

1.11 Extracellular diffusion modeling

The CSF tracer analyzed in Paper I gave a strong indication of a brain-wide waste clearance. This means that an evaluation of the lymphatic hypothesis can be done with a macroscopic model. The diffusion of solutes in the extracellular space has been studied in pharmacology, and an extensive framework with experimental verification was detailed in the seminal paper by Syková et al. [114].

In the seminal paper, the diffusion in a porous media is considered a hindered diffusion in the macroscale, with its own diffusion coefficient. This coefficient is
Finite element method

referred to as the effective or apparent diffusion coefficient (ADC), and it can be derived from experiments. The apparent and the free diffusion coefficients are related by

\[ \lambda^2 = \frac{D_{\text{free}}}{D_{\text{ADC}}}, \]

(1.4)

with \( \lambda \) known as the tortuosity. The tortuosity is a material parameter, and it has been measured to be around 1.5-1.7 in the white matter tissue [114].

1.12 Finite element method

We will use the finite element method (FEM) with the software FEniCS [77] to solve (1.3). FEM is detailed in numerous books, for example [94], so we will only give a brief description of the fundamental steps for the method. It is a numerical method used for solving PDEs by constructing a linear system, which is done by finding the weak formulation and the discretization of the computational domain. We can find the weak formulation of a PDE by multiplying the equation with a partial continuous function, and doing integration by parts. The computational domain may exist in both a spatial and temporal sense, and the discretization of the temporal domain can for example be done with backwards Euler method. The spatial discretization with FEM consists of dividing the computational geometry into smaller elements, like tetrahedrons and cubes, which are connected by shared vertices. On each element, the PDE function is approximated with a polynomial, for example first order Lagrangian. Then, by utilizing the Gaussian quadrature rules, we can integrate the weak formulation over each element. The connections between elements make it possible to construct a linear system that we can solve with numerical methods. The details on modeling and solving the diffusion equation with FEniCS will be provided in [Book Excerpt III].

1.13 Adjoint state method

In [Paper IV] we will exemplify the usage of variational assimilation with PDE-constraint optimization. PDE-constraint optimization methods have been well documented with several books, like [26], for minimizing a functional subjected to a PDE for a set of variables. The PDE may also depend on the optimization parameters, so it is preferable to use the reduced functional. Therefore, an effective method was needed for minimizing a reduced functional, and iterative optimization algorithms can do this with the descending gradient, like the conjugate gradient method. The numerical approximation of descending gradient, the tangent linear solutions, would require solving a linear system for each optimization variable, which can be computational expensive compared to the adjoint state method. The adjoint state method works by solving a single linear system to find the adjoint solution, which can be used to obtain the descending gradient. We used dolfin-adjoint [33] to construct and solve the adjoint problem.
1. Introduction

1.14 Computational geometries

The FEM requires a computational grid (mesh) of smaller elements that represents an approximation of the geometry, and the cerebral cortex has a complex structure with gyri and sulci, see Figure [1.3]. These details are hard to include in the mesh, and often require higher resolution to accurately depict. Furthermore, although the shape of the brain is similar between persons, there are possible anomalies such as cysts, tumors and obstructions that might impact the computational results. Therefore, creating patient-specific meshes were highly preferable, and methods to make high detailed meshes were investigated. The segmentation process in FreeSurfer generates surfaces of the white and cortical gray matter for each cerebral hemisphere that can be used to measure the cerebral cortex [24], see Figure [1.9]. We considered these surfaces as the starting point for creating patient-specific volume meshes.

There are different programs that can be used for creating computational volume meshes, such as Gmsh [45], ITK [103] and previously mentioned VMTK. However, they are not specifically made for creating patient-specific computational mesh of the brain. Therefore, we developed a Python module called surface volume meshing tool-kit (SVMTK) [117] for this purpose and thesis. We developed the module with the intent of handling large cohorts with semi-automatic scripts to create computational meshes. The module offers an effective Python wrapping of necessary C++ functions and classes in the computational geometry algorithms library (CGAL) [91] to construct a volume mesh from surfaces. In our
case, these surfaces were created by FreeSurfer, and we combined these surface to create subdomains representing white and gray matter in the constructed mesh. However, the surfaces generated by FreeSurfer can be unsuitable for constructing meshes, and thus the module was made to handle different types of problems that can occur. These can for instance be: gray matter surfaces extending into the dura mater, non-anatomical features due to surface self-intersections, overlapping surfaces and sharp edges. We will go through the entire process in more detail in [Book Excerpt III] including necessary repairing steps.

1.15 Summary of papers

**Paper I: Brain-wide glymphatic enhancement and clearance in humans assessed with MRI**

Administration of drugs into the SAS can be a great opportunity to increase the effectiveness of the drug delivery. In fact, the administration of drugs intravenously has low success rate because of the blood-brain barrier and the fact that the blood vessels only occupy 3% of the brain. Thus intrathecal treatment may provide an effective method for drug delivery into the brain.

Animal studies have reported that there is communication between the SAS and the perivascular compartments of the brain. The observations in animal studies have not been translated to humans, and have not demonstrated enhancement in deep brain white matter.

The study consisted of 17 individuals that were divided into 2 cohorts. The first cohort was made of 8 individuals that were considered close to healthy and were used as a reference group (REF), while the second cohort had 9 patients who were diagnosed with iNPH. The participants were subjected to a baseline MRI scan before the CSF tracer was administrated with a lumbar puncture. Then, the participants underwent several MRI scans 1-2 hours after the injection, followed by MRI scans at 2-4 hours, 4-6 hours, 6-9 hours, 24 hours and up to 48 hours. The baseline MRI was segmented using the software FreeSurfer, and all MRIs for each participant were aligned to the baseline. This made it possible to measure signal change in several regions of the brain, and determine the presence of tracer using statistical analysis.

The results suggest a brain-wide clearance of the brain. Furthermore, there were some important differences between reference subjects and the patients with iNPH. The MRI of the reference participants indicated no tracer after 48 hours, but in iNPH there were still CSF tracer present after 48 hours.

In conclusion, the study demonstrated a brain-wide enrichment of a CSF tracer that was administered intrathecally in humans. The CSF tracer distributed centripetally from the cerebrum surface towards deeper structures.
1. Introduction

Figure 1.10: CSF tracer enrichment over time in one REF subject. The brain-wide enrichment of CSF tracer over time is shown for one REF individual. The percentage change in signal unit ratio is indicated at the color scale. CSF tracer enrichment occurred in a centripetal pattern at a rate exceeding the expected for pure diffusion, suggesting an important role of bulk flow for transport of substances through the brain.

in the brain. Vascular pulsations seem to play an important role for the tracer entry into the brain parenchyma. The delayed clearance of tracers from the brain parenchyma in patients diagnosed with iNPH, suggests that intrathecal contrast-enhanced MRI could be used to diagnose preclinical neurodegenerative diseases.

**Paper II** Delayed clearance of cerebrospinal fluid tracer from choroid plexus in idiopathic normal pressure hydrocephalus

Impaired clearance of amyloid-beta from choroid plexus is one proposed mechanism behind amyloid deposition in Alzheimer’s disease. This study investigated if the clearance from choroid plexus is altered in iNPH patients by using CSF tracer as a surrogate marker of a metabolic waste product.

The study participants were divided into two cohorts, the first cohort consisted of 8 participants who were considered close to healthy and used as a reference group (REF), while the second cohort had 9 patients who were diagnosed with iNPH. The participants all started with a baseline MRI scan before the CSF tracer Gadovist was administrated with a lumbar
puncture. Then, the participants underwent several MRI scans within 1-2 hours after the injection, followed by MRI scans in the time intervals 2-4 hours, 4-6 hours, 6-9 hours, 24 hours and up to 48 hours. The baseline MR images were segmented with the software FreeSurfer, and the MRI images for each participant were aligned with the baseline MRI. In cases with iNPH patients, T2-weighted MRI images were used to correct segmentation error in the lateral ventricles.
The results showed that the normalized T1 signal increased to the maximum values within the choroid plexus and the CSF in the lateral ventricles 6-9 hours after injection of gadobutrol. There were significant differences between the cohorts during the enhancement phase after 4-6 hours and 6-9 hours, and during the clearance phase at 24 hours after injection. In particular, the difference after 24 hours indicated that there was a delayed clearance of CSF tracer from the choroid plexus in iNPH.

In conclusion, the results suggest that the CSF tracer agent gadobutrol is absorbed by the choroid plexus. Furthermore, patients with iNPH had the CSF tracer cleared with a slower rate compared to the reference cohort. This suggests that neurodegeneration in iNPH may be caused by delayed CSF clearance of brain metabolism through the choroid plexus. There were discussions of a follow up computational study, like [Paper IV] for [Paper I], to simulate the enrichment of the choroid plexus. However, no significant delay was observed between the enrichment of lateral ventricles and choroid plexus, which made it difficult to construct a computational model.

Book Excerpt III: An introduction to meshing and mathematical modeling for the human brain: From magnetic resonance images to finite element simulations

This text is a book manuscript excerpt covering three chapters, numbered 2, 3 and 4, which describes methods using MRI images to create computational models of the human brain. It consists of tools, code and data so that the reader can reproduce the results presented in the book, but the MRI data will not be provided in this excerpt, due to privacy requirements.

Chapter 2 begins with a mathematical introduction to the diffusion equation as a bio-mechanical model relating the different parameters to measurable data. We continue with a brief of brain anatomy, illustrated using MRI to familiarize the reader with the format. Then, we present some of the MRI modalities, and how it can be used in relation to the bio-mechanical modeling. The chapter finishes with an overview of the tools, like FreeSurfer and FEniCS, which were used throughout the book. This section includes a detailed installation guide with code and references.

Chapter 3 begins with the details of the procedure on how to create computational meshes from a T1-weighted MRI. T1-weighted MRI is used to create segmentation and surfaces by using the software FreeSurfer, but the resulting surfaces can have variable quality. Thus, we continue with different ways to improve the surfaces with SVMTK, which include smoothing, remeshing and boolean operations to name a few. Then we go through how the Python module SVMTK is used to create a volume mesh from the surfaces. The chapter ends with a detailed example on the computational modeling of the diffusion equation with the produced volume mesh, and a detailed description on the implementation of SVMTK.

Chapter 4 details the procedure to create a mesh with multiple subdomains. This procedure uses the surfaces of the cerebral cortex and the white matter
Figure 1.12: The image shows a clip of the mesh constructed using the pial and white surfaces for each hemisphere, with the lateral ventricles removed.

of each hemisphere generated by FreeSurfer. The lateral ventricle surfaces are not automatically produced by FreeSurfer, therefore the chapter will cover this procedure in detail. Then, we will go through the subdomain function and how to assign each subdomain with an integer tag. The chapter will also cover how to remove a subdomain from the volume mesh, like the lateral ventricles. The last section covers the methods to convert the subdomain cell marking and the facet marking between subdomains into FEniCS using mesh conversion tools.

**Paper IV**: Can diffusion alone explain brain-wide distribution of CSF tracers within 24 hours?

Waste clearance of the brain has received more attention after the hypothesis of the glymphatic system. The hypothesis links neurodegenerative diseases, like Alzheimer’s disease and iNPH, to the inadequate clearance of harmful proteins from the brain parenchyma.

The waste clearance in the glymphatic hypothesis is based on the transportation of waste through a bulk flow in extracellular space in the brain. Recently, computational studies looked at the possibility of bulk flow in extracellular and paravascular spaces, and the results indicate that a bulk flow was very unlikely. Thus, the purpose for this paper is to explore if the CSF tracer distribution seen in **Paper I** can be explained by diffusion alone.
1. Introduction

This study obtained brain MRI observation of CSF tracer in one patient diagnosed with iNPH at several time points. Boundary constrained PDE optimization was selected to compute the ADC based on the MRI observations. The optimization was done by minimizing a functional that was constructed with three regularization parameters; spatial, temporal and gradient regularization. The finite element method software FEniCS in combination with dolfin-adjoint was used to solve the problem, and meshes were constructed using the baseline MRI.

The robustness of the method was tested using a synthetic test case with a large range of computational parameters, which also included adding artificial noise to the manufactured simulations. Additionally, methods for reducing the noise in the MRI data were also tested with Gaussian smoothing and value projection onto the boundary. We estimated the concentration distributions based on MR images obtained at different points in times. The method was used with the concentration distributions to compute the ADC for gray and white matter, which was compared with the ADC estimated with free diffusion coefficient and the tortuosity obtained from DTI.

The results of the synthetic test case showed high accuracy reconstruction of the ADC and robustness to noise. The results were consistent for a large range of regularization parameters and showed good convergence in the relative error to less than 5%. ADC computed with the MRI observation were $2.2 \times 10^{-4}$ mm$^2$/s in the white matter and $1.8 \times 10^{-4}$ mm$^2$/s in the gray matter. These values were compared to the ADC estimated using...
DTI, which were $1.10 \times 10^{-4} \text{ mm}^2/\text{s}$ in white matter and $1.26 \times 10^{-4} \text{ mm}^2/\text{s}$ in gray matter. As an effect of the Gaussian smoothing, the computed ADC in the gray matter was on average 250% times higher than ADC estimated using DTI. While the values in the white matter were more consistent, the method of projecting values onto the boundary gave ADC of 5% lower ADC than estimated with DTI.

In conclusion, the methodology was tested with a synthetic test case and the results showed robustness for a large range of parameters and noise. The computed ADC in gray matter and the white matter were respectively 23% and 82% higher than the estimated ADC using DTI, indicating enhancement of solute transport. This could be due to the iNPH diagnosis of the patient, which caused large variance of values in the DTI. Therefore, the next step is to determine to what extent the methodology is able to discriminate between healthy and impaired clearance.

1.16 Other contributions

In addition to the presented papers and the book excerpt, the PhD also included co-authoring the following articles [111, 112, 128] proceedings [112], poster presentation [118], oral presentation at international conferences [119, 122] and at national workshops [120, 121].

1.17 Future work and outlook

In this thesis, we used MRI processing software FreeSurfer, which we later combined with bio-mechanical modeling. Although we have covered the objectives of this thesis, there is still a gap between bio-mechanical brain modeling and other bio-mechanical fields of modeling. Therefore, we will in this section cover some of the future additions and improvements. The papers Paper I and Paper II had 17 participants, so increasing the number of participants would be a natural step forward. There are also other improvements possible, such as the distribution of MRI during the first 48 hours period. This can be more difficult to achieve, due to logistics, but optimization of the scan times can provide more information. In the Book Excerpt III, we detailed the process of making a patient-specific geometry, but we did not include different MRI modalities, like elastography and DTI, in the computational model. Therefore, the next step should aim at including different MRI modalities to bio-mechanical model. This presents another problem of MRI resolution that is dependent on the MRI modality. Therefore, we need to have a proper approach to combining different MRI modalities. The methodology presented in Paper IV was only applied to a single person, thus the next step would involve applying the same methodology on two different cohorts, a reference cohort and an iNPH cohort, to see if the methodology can discriminate between the cohorts.
1. Introduction

References


1. Introduction


[31] Evju, Ø. et al. “Robustness of common hemodynamic indicators with respect to numerical resolution in 38 middle cerebral artery aneurysms”. In: PloS one vol. 12, no. 6 (2017), e0177566.


1. Introduction


1. Introduction


1. Introduction


Papers & Book Excerpt
Paper I

Brain-wide glymphatic enhancement and clearance in humans assessed with MRI

Brain-wide glymphatic enhancement and clearance in humans assessed with MRI

Geir Ringstad,1,2 Lars M. Valnes,3 Anders M. Dale,4,5,6 Are H. Pripp,7 Svein-Are S. Vatnehol,8 Kyrre E. Emblem,9 Kent-Andre Mardal,3,10 and Per K. Eide2,11

1Division of Radiology and Nuclear Medicine, Department of Radiology, Oslo University Hospital — Rikshospitalet, Oslo, Norway. 2Institute of Clinical Medicine, Faculty of Medicine, and 3Department of Mathematics, University of Oslo, Oslo, Norway. 4Department of Neurosciences, 5Department of Radiology, and 6Multimodal Imaging Laboratory, UCSD, La Jolla, California, USA. 7Oslo Centre of Biostatistics and Epidemiology, Research Support Services, Oslo University Hospital, Oslo, Norway. 8The Intervention Centre, Oslo University Hospital — Rikshospitalet, Oslo, Norway. 9Department of Diagnostic Physics, Oslo University Hospital, Oslo, Norway. 10Center for Biomedical Computing, Simula Research Laboratory, Lysaker, Norway. 11Department of Neurosurgery, Oslo University Hospital — Rikshospitalet, Oslo, Norway.

To what extent does the subarachnoid cerebrospinal fluid (CSF) compartment communicate directly with the extravascular compartment of human brain tissue? Interconnection between the subarachnoid CSF compartment and brain perivascular spaces is reported in some animal studies, but with controversy, and in vivo CSF tracer studies in humans are lacking. In the present work, we examined the distribution of a CSF tracer in the human brain by MRI over a prolonged time span. For this, we included a reference cohort, representing close to healthy individuals, and a cohort of patients with dementia and anticipated compromise of CSF circulation (idiopathic normal pressure hydrocephalus). The MRI contrast agent gadobutrol, which is confined to the extravascular brain compartment by the intact blood-brain barrier, was used as a CSF tracer. Standardized T1-weighted MRI scans were performed before and after intrathecal gadobutrol at defined time points, including at 24 hours, 48 hours, and 4 weeks. All MRI scans were aligned and brain regions were segmented using FreeSurfer, and changes in normalized T1 signals over time were quantified as percentage change from baseline. The study provides in vivo evidence of access to all human brain subregions of a substance administered intrathecally. Clearance of the tracer substance was delayed in the dementia cohort. These observations translate previous findings in animal studies into humans and open new prospects concerning intrathecal treatment regimens, extravascular contrast-enhanced MRI, and assessment of brain clearance function.

Introduction

Knowledge about the access of substances administered in the subarachnoid space to human brain as whole could potentially make new treatments of brain disease. The blood-brain barrier (BBB) represents one of the largest obstacles to effective CNS drug delivery (1), and the compartment within blood vessels merely occupies less than 3% of the total brain volume (2). Thus, new therapeutic CNS drugs generally show lower success rates than those for non-CNS indications (3), while intrathecal treatment regimens have emerged with great promise (4, 5).

Previous animal studies have shown communication between the subarachnoid cerebrospinal fluid (CSF) space and perivascular compartments of the brain (6–10) and spinal cord (11). However, literature reporting human in vivo CSF tracer studies is lacking, and observations made in animals have not been translated into humans. Two recent human studies demonstrated brain parenchyma enhancement subsequent to subarachnoid (intrathecal) administration of a MRI contrast agent, but observations were limited to selected ROIs (12, 13). An MRI study of rats failed to demonstrate enhancement in deep brain white matter (14). Moreover, animal studies report diverging findings regarding the sites of perivascular brain influx and efflux and direction of perivascular flow (7, 15, 16). Animal in vivo observations typically cover extremely limited fields of view, as when utilizing 2-photon microscopy (7).

Furthermore, mechanisms behind transport and clearance of substances within the brain interstitial space are controversial. For a long time, size-dependent diffusion was considered to explain
interstitial movement of molecules (17). In 2012, a brain-wide pathway for convective transport of waste solutes from the brain was first described and denoted the glymphatic system (7). Net water and solute transport through the extracellular compartment from arterial to venous paravascular spaces was proposed to be dependent on aquaporin-4 (AQP4) water channels and mediated by arterial pulsations (18). Impaired glymphatic function has been suggested to be instrumental in a range of brain diseases; this has been illustrated most in Alzheimer’s dementia (7, 19, 20) but has also been shown to be relevant in posttraumatic encephalopathy (21), ageing (22), sleep (23, 24), depression (20), and exercise (25). However, the glymphatic concept has been challenged by several modeling studies that have opposed CSF pulsations as the explanation for net convective interstitial flow (26, 27). Moreover, a later animal study utilized a similar set of experiments as those bringing evidence for a glymphatic system and found, contrary to the previous observations, that interstitial flow could be explained by diffusion alone, independent of AQP4 status (9). Later, several independent groups provided evidence for an important role of AQP4 in glymphatic circulation (28).

To this end, we here show for what we believe to be the first time brain-wide CSF tracer enhancement and clearance in humans. For this, we administered an MRI contrast agent in the subarachnoid CSF compartment, followed by repeated MRI scans at 24 and 48 hours and after 4 weeks. In addition, we found delayed clearance of CSF tracer from the brain in a cohort of patients with dementia and expected CSF circulation failure (idiopathic normal pressure hydrocephalus [iNPH]).

Results

Brain-wide distribution of CSF tracer within brain parenchyma in reference subjects. The study includes 8 reference (REF) individuals who underwent MRI for tentative idiopathic intracranial hypotension due to...
RESEARCH ARTICLE

CSF leakage (Supplemental Table 1; supplemental material available online with this article; https://doi.org/10.1172/jci.insight.121537DS1). CSF leakage was diagnosed in 3 of 8 individuals; no CSF circulation disturbance was identified in the remaining 5 individuals. Even though the REF individuals had some complaints (Supplemental Table 1), we considered this cohort as close to healthy.

Figure 1 illustrates the distribution of CSF tracer in the brain parenchyma of one REF individual (see Supplemental Video 1). The tracer distributed centripetally from cortex to deeper brain regions. Supplemental Figure 1 shows the change in the T1 signal within the CSF space and brain parenchyma following intrathecal gadobutrol. Table 1 presents the percentage change in normalized T1 signal from before CSF tracer administration, including percentage change for cerebral cortex (gray matter), cerebral white matter, basal ganglia, thalamus, limbic structures (hippocampus, amygdala, nucleus accumbens, and entorhinal cortex), cerebellar cortex, and cerebellar white matter. The MRI signal increase was significant at all main locations (Table 1). Figure 2 shows the CSF contrast enrichment after 6–9 hours in coronal MRI sections of 8 REF individuals. As illustrated, the CSF tracer enrichment occurred in a centripetal pattern and primarily in brain regions adjacent to large artery trunks at the surface, i.e., the anterior, middle, and posterior cerebral arteries.

Figure 3 shows the percentage change in normalized T1 signal over time for selected brain subregions, including cerebral cortex (Figure 3A), cerebral white matter (Figure 3B), cerebellar cortex (Figure 3C), cerebellar white matter (Figure 3D), limbic structures (Figure 3E), basal ganglia (Figure 3F), corpus callosum (Figure 3G), and hippocampus (Figure 3H).

The REF individuals showed CSF tracer enrichment in close to all the brain regions studied, as further detailed in Supplemental Table 2.

Brain-wide distribution of CSF tracer in individuals with iNPH dementia. The study also includes 9 individuals with a subtype of dementia, denoted as iNPH (Supplemental Table 1). The iNPH patients were older than the REF subjects and presented with other symptoms (Supplemental Table 1). Figure 4 illustrates the color-coded enrichment of CSF tracer in brain parenchyma over time for one iNPH patient (see Supplemental Video 2). Enrichment in the periventricular white matter is a feature typical of iNPH, due to ventricular tracer reflux and transependymal migration, as previously reported (13). Figure 5 shows the CSF contrast enrichment after 6–9 hours in coronal MRI sections of 8 iNPH individuals. As for REF subjects, the CSF tracer enrichment occurred in a centripetal pattern and primarily in brain regions adjacent to large artery trunks at the surface. Table 2 presents the percentage of change in CSF tracer enrichment over time for selected brain regions of the iNPH subjects, including cerebral cortex (gray matter), cerebral white matter, basal ganglia, thalamus, and limbic structures as well as the cerebellar gray and white matter. CSF tracer enrichment was found in all these regions, and all subregions were defined in FreeSurfer (Supplemental Table 3).

Clearance of CSF tracer in REF subjects and iNPH patients. Our findings point to some important differences between REF subjects and individuals with iNPH dementia.

As illustrated in Figure 6, as compared with REF subjects, the CSF tracer enrichment in iNPH patients

| Table 1. Percentage change in signal unit ratio at various time points after i.th. gadobutrol in REF individuals |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Anatomical region |
|                  | 1.5–2 hours     | 2–4 hours       | 4–6 hours       | 6–9 hours       | 24 hours        | 48 hours        |
|                  | 16 ± 20         | 101 ± 116       | 354 ± 309       | 500 ± 421       | 157 ± 164       | 177 ± 177       |
| CSF [24]         |
| Cerebral cortex (gray matter) [1,000–1,035 (left) + 2,000–2,035 (right)] |
|                  | −3 ± 7          | −1 ± 15         | 26 ± 44         | 44 ± 50         | 52 ± 54         | 61 ± 56         |
|                  | <0.001          |                 |                 |                 |                 |                 |
| Cerebral white matter [2, 41] |
|                  | −3 ± 9          | −1 ± 18         | 26 ± 40         | 34 ± 46         | 27 ± 33         | 32 ± 10         |
|                  | 0.005           |                 |                 |                 |                 |                 |
| Basal ganglia [11, 12, 13, 50, 51, 52] |
|                  | −3 ± 8          | −6 ± 14         | 1 ± 25          | −4 ± 14         | 4 ± 18          | 32 ± 10         |
|                  | 0.014           |                 |                 |                 |                 |                 |
| Thalamus [10, 49] |
|                  | −2 ± 8          | −4 ± 15         | 5 ± 26          | 2 ± 18          | 11 ± 21         | 24 ± 8 ± 18     |
|                  | 0.008           |                 |                 |                 |                 |                 |
| Hippocampus, amygdala, accumbens, entorhinal [17, 18, 26, 34, 50, 506, 1006] |
|                  | −2 ± 7          | −1 ± 15         | 28 ± 38         | 43 ± 43         | 42 ± 39         | 40 ± 8 ± 17     |
|                  | 0.004           |                 |                 |                 |                 |                 |
| Cerebellar cortex [8, 47] |
|                  | 6 ± 6           | 12 ± 27         | 45 ± 58         | 67 ± 80         | 53 ± 47         | 90 ± 5 ± 18     |
|                  | 0.003           |                 |                 |                 |                 |                 |
| Cerebellar white matter [7, 46] |
|                  | 1 ± 9           | 3 ± 14          | 3 ± 23          | 2 ± 19          | 15 ± 23         | 33 ± 7 ± 19     |
| Continuous variables are presented as mean ± SD. Statistical differences were determined by a linear mixed model for repeated measurements. n = 8. i.th., intrathecal. Numbers given in brackets refer to FreeSurfer predefined numbers for specific brain regions.
at 24 hours was significantly higher in cerebral white matter (Figure 6C), basal ganglia (Figure 6D), corpus callosum (Figure 6F), limbic system (Figure 6G), and hippocampus (Figure 6H), indicative of reduced tracer clearance. Supplemental Table 4 shows differences in percentages of change over time of normalized T1 signal for a wide range of brain regions.

CSF tracer enrichment after 4 weeks. We also compared the normalized T1 signal before and 4 weeks after intrathecal CSF tracer to look for signs of CSF tracer retention after 4 weeks. This was not found in any brain region examined for either REF or iNPH (Supplemental Table 5).

In a wide range of brain regions, the normalized T1 signal was consistently lower in iNPH individuals than in REF individuals before intrathecal gadobutrol and reached statistical significance for some of the regions (Supplemental Table 5). A lower T1 signal may be interpreted as indicative of higher brain parenchymal water content.

Correlation between CSF tracer enrichment and brain volume. The Pearson correlation coefficients between percentage change in signal unit ratios and tissue volume for white matter, gray matter, and CSF of lateral ventricles for REF and iNPH subjects are presented in Table 3. The correlation coefficients differ between REF and iNPH cohorts. In iNPH subjects, the volume of gray matter correlated negatively with CSF tracer enhancement in gray and white matter, i.e., with increasing volume the CSF tracer enhancement was lower.

Discussion

Brain-wide enhancement of CSF tracer in human brain. In this human CSF tracer study, repeated MR imaging provides in vivo evidence of a route for macromolecules administered in CSF from the subarachnoid compartment to the entire brain. Any substance with features similar to the studied tracer, gadobutrol, should therefore be expected to access all brain regions when administered intrathecally. We have previously demonstrated that an equally small amount of MRI contrast agent injected intrathecally at the lumbar level typically reaches the intracranial subarachnoid compartment in approximately 20 minutes, where it primarily distributes along large artery trunks at the brain surface (13). These observations were reproduced in the present work. Moreover, we reproduced the finding of primary tracer entry into the brain in tissue adjacent to major artery trunks, indicating an important role of CSF pulsations for macromolecular transport within brain tissue. Here, we have extended our assessment to cover all brain subregions, demonstrating tracer enhancement in locations even several centimeters deep to the brain surface and tracer bulk flow through the cerebral mantle. The CSF tracer enhancement within the brain occurred in a centripetal pattern, with some regional differences, and peak
Figure 3. CSF tracer enrichment in some brain regions of the REF cohort. Trend plots of percentage change in signal unit ratio are presented for main regions within the supratentorial and infratentorial compartments, including (A) cerebral cortex (gray matter), (B) cerebral white matter, (C) cerebellar cortex, (D) cerebellar white matter, (E) limbic system, (F) basal ganglia, (G) corpus callosum, and (H) hippocampus. Trend plots are presented with mean ± standard error (SE). n = 8.
enhancement typically occurred at 24 hours after tracer injection. Thereafter, the level of tracer in parenchyma declined, and tracer had been cleared from the brain after 4 weeks.

Mechanisms behind brain cortex tracer enrichment. The hydrodynamic diameter of gadobutrol has, to our knowledge, not been estimated previously. However, prediction of hydraulic diameter based on molecular weight (MW) is feasible (29). With a MW of 3,000 Da, dextran 3 is comparable to the highly hydrophilic molecule gadobutrol with respect to MW (MW = 604 Da), and the hydrodynamic diameter of dextran 3 has been experimentally calculated as 26 Å or 2.6 nm (30). Hydraulic diameter of gadobutrol is therefore not expected to exceed this, and further distribution within the brain cortex interstitial compartment should accordingly be expected to be predominantly driven by diffusive flux (1). Average in vivo width of extracellular space has been reported to be between 38 and 64 nm (31). In ref. 17, the timescale of the relative diffusive tracer distribution ($C$) is estimated as follows: $C = \text{erfc}(x/(2 \times \sqrt{D \times t}))$. Using this formula, assuming a diffusivity of $D = 12 \times 10^{-7}$ cm$^2$/s, with $x$ and $t$ corresponding to the length in cm and time in seconds, respectively, a 50% saturation of the extracellular space can be estimated to occur at around 55 hours. The assumptions underlying this formula are, however, that the length scale is such that the cortex can be considered flat and the tracer distribution uniform. These assumptions are not valid in our case, but it seems unlikely that diffusion alone explains the brain-wide distribution. Further, loss of the molecule across the BBB, uptake into cells, or binding to receptors all may modify diffusion (17).

It has been proposed that the perivascular compartment of human cortex is merely a potential space for movement of solutes (32–36), a view that recently was maintained based on observations made in fixed tissue specimens (37). In contrast, others have reported that the intrinsic hydraulic permeability of the human perivascular space is far greater than the width of the cortical extracellular space (1). The width of the perivascular space is suggested to be at least 2 orders of magnitude greater than neocortical extracellular...
space width, and intrinsic hydraulic permeability of the perivascular space is suggested to be at least 10,000-fold higher than the extracellular space of the neuropil. It is therefore reasonable to assume that the cortical tracer enhancement in our human cohorts derives from the same mechanism as that reported in animal studies, where perivascular CSF tracer distribution in animal cortex is consistently demonstrated (6–8). Human cortical enhancement was, however, a much more durable process (days) than that of animals (hours) (14). The role of vascular pulsations being mediated to CSF for enrichment of perivascular spaces is underlined by the rich and early enhancement in parenchyma adjacent to large arterial vessels at the brain surface (Figures 1, 2, 4, and 5). Here, diffusion likely adds to advective pulsatile to-and-fro flow, i.e., dispersion (38). Dispersion by itself does not require net, convective flow. Based on animal studies (6, 7, 39), we expected further transport of tracer into the interstitial space to occur at the neurovascular interface through astrocytic inter-endfeet gaps, which allow for passage of solutes of up to 20-nm hydraulic diameter (38).

**Tracer enrichment in deep brain structures.** CSF tracer also propagated in brain tissue deeper than what can be attributed to perivascular spaces connected to the brain surface only. The present in vivo observations of contrast enhancement in deep white matter extends evidence from a similarly designed MRI study of rats, where lack of enhancement in deep portions of the brain white matter was attributed to concentrations of contrast agent probably being too small to be detected. Image resolution of the MRI scan was 1 mm, while the sizes of paravascular and interstitial spaces are in the order of μm. A detailed assessment of molecular motion at a microscopic level was therefore beyond reach in this study.

Previous animal studies have indicated that axon tracts, in addition to perivascular spaces, can also act as preferential, low-resistance pathways for solutes through the brain (40). White matter has been identified as a major site for convective flow under normal conditions (41). In a recent study, interconnections between perivascular compartments of arteries and veins, allowing small and large molecules to bypass the interstitial space, were demonstrated (42). Diffusion alone has also been proven inefficient at distances larger than a few millimeters, as in the human brain (43). Furthermore, tracers of different MW are cleared at similar speeds in the brain, which is not consistent with diffusion, but rather convective flow (44). Our study observations, with enrichment of CSF tracer over a range of several centimeters within 24 hours, may suggest that convective forces and bulk flow in general are more important for brain-wide tracer distribution than diffusion alone. Bulk flow has previously proven difficult to quantify (17) but is considered most likely to be restricted to the perivascular spaces (40). One study estimated the velocity of white matter bulk flow toward the ventricles as 10.5 μm/min (approximately 15 mm per day) (41). In our REF subjects, enhancement in regions several cm deep to the brain surface was found at 24 hours (Figures 1–3; Table 1; and Supplemental Table 2), which by far exceeds previous estimates for bulk flow. Parenchymal migration of a substance over a distance much farther than previously shown in both animals and humans also has important implications for the potential of neurotransmitters at a synapse to reach receptors outside the immediate postsynaptic site. Such extrasympathetic volume transmission has not been explicitly identified before and has therefore remained an elusive concept with respect to intercellular communication in the CNS (17).

### Table 2. Percentage change in signal unit ratio at various time points after i.th. gadobutrol in INPH individuals

<table>
<thead>
<tr>
<th>Anatomical region</th>
<th>Time after i.th. gadobutrol</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>[FreeSurfer]</td>
<td>1.5–2 hours</td>
<td></td>
</tr>
<tr>
<td>CSF [24]</td>
<td>23 ± 29</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Cerebral cortex (gray matter [1,000–1,035 (left) + 2,000–2,035 (right)])</td>
<td>4 ± 20</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Cerebral white matter [2, 41]</td>
<td>3 ± 20</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Basal ganglia [11, 12, 13, 50, 51, 52]</td>
<td>4 ± 21</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Thalamus [10, 49]</td>
<td>5 ± 21</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Hippocampus, amygdala, accumbens, entorhinal [17, 18, 26, 53, 54, 58, 1,006, 2,006]</td>
<td>5 ± 22</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Cerebellar cortex [8, 47]</td>
<td>7 ± 18</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Cerebellar white matter [7, 46]</td>
<td>4 ± 19</td>
<td>&lt;0.001</td>
</tr>
</tbody>
</table>

Continuous variables given as mean ± SD. Statistical differences determined by a linear mixed model for repeated measurements. n = 9. i.th., intrathecal. Numbers given in brackets refer to FreeSurfer predefined numbers for specific brain regions.
Effect of sleep. One mechanism behind the observed high bulk flow rate could be the effect of sleep. Natural sleep in animal studies has been shown to associate with increased size of the interstitial space, facilitating increased convective exchange of CSF with interstitial fluid and clearance of macromolecules (23). While we notice that the observed tracer enhancement in brain generally peaked after the first night (at 24 hours), which is remarkable, we do not have a sufficient number of MRIs in the time interval from 6–9 hours to 48 hours to conclude the exact time point of when peak enhancement truly occurred. It should also be noted that tracer enrichment of cortex and white matter was in progress before onset of sleep (Figures 1–3; Table 1; and Supplemental Table 2). Peak enhancement occurring at 24 hours differs significantly from that found in studies of rats, which typically was at 1–3 hours (14, 45).

Tracer clearance routes from CSF and brain parenchyma. The centripetal pattern of brain enhancement seems to contradict studies in which intraparenchymal tracers have been injected and described as leaving the brain centrifugally along the walls of arteries (15, 46). The findings made in these previous tracer studies have led to the hypothesis that pathological aggregates of amyloid-β in walls of cortical arteries are indicative of AD pathogenesis. Widespread penetration of substances from CSF into brain parenchyma along perivascular spaces of arteries is, however, well supported by animal studies with a similar experimental setup to ours (1, 47–49). While our present study demonstrated declining levels of the tracer molecule after 24 hours, and no remains at 4 weeks, the routes for tracer clearance are still elusive. Uptake into arachnoid villi, as proposed more than one hundred years ago, seems to be of minor importance in humans, as reviewed by Brinker et al. (50) and as indicated in our recent study (13). In 2015, lymphatic vessels draining from CSF to neck lymph nodes were for the first time detected in walls of dural venous sinuses (51, 52). The full significance of these lymphatic drainage pathways has yet to be shown, but we have found that neck lymph node and brain tracer enhancement coincided in time, indicating this drainage is from the brain rather than directly from CSF (53). Should lymphatic drainage be directly from CSF, as suggested in a recent animal study (54), this would then occur much slower in humans than reported in animals. Our present findings of peak tracer enhancement in CSF at 6–9 hours, followed by a decline before peak enhancement in brain occurs (24 hours; Figure 3 and Figure 6), contradicts that the levels of tracer in CSF and parenchyma are merely balanced by diffusion between these compartments. Because CSF tracer clearance along veins is suggested in several studies (6–8, 39), we found it reasonable to hypothesize that perivenous compartments and lymphatic vessels are interconnected.
Figure 6. Comparison of CSF tracer enrichment between REF and iNPH subjects within selected brain regions. Trend plots of percentage change in signal unit ratio are presented for different regions within the supratentorial and infratentorial compartments, including (A) CSF, (B) cerebral cortex (gray matter), (C) cerebral white matter, (D) basal ganglia, (E) cerebellar cortex, (F) corpus callosum, (G) limbic system, and (H) hippocampus. Significant differences between REF (blue line) and iNPH (green line) groups were determined by linear mixed models for repeated measurements. *P < 0.05, **P < 0.01, ***P < 0.001. Trend plots are presented with mean ± standard error (SE). n = 8, REF; n = 9, iNPH.
Therapeutic and diagnostic implications of brain-wide tracer enhancement. The potential for administrating drugs via the intrathecal route has been considered to be limited by some (55). Compared with the intravenous route, intrathecal delivery of drugs in treatment of neurological disease is still rare, and few drugs have been approved for this use. Because many neurological conditions primarily are diseases of the interstitial space, and do not cause breakdown of the BBB, brain-wide access to the extravascular CNS compartment via intrathecal delivery may show to have many benefits. An already established treatment regime is that of intrathecal monoclonal antibodies against multiple sclerosis (56). Other medications approved by the US Food and Drug Administration for intrathecal use include morphine, ziconotide, and baclofen (57). Studies of intrathecal treatment for infantile-onset spinal muscular atrophy are emerging with promising results (4), and intrathecal delivery of recombinant enzyme to patients with mucopolysaccharidosis type II (Hunter syndrome) has been proposed (47).

Knowledge of what extent an intrathecal substance with certain size and properties can distribute throughout the entire human brain may help pave the way for new treatment options. It is likely that monoclonal antibodies and viral vectors for gene therapy could provide new prospects for treatment of neurodegenerative disorders, such as Alzheimer’s and Parkinson’s disease (1, 58). Furthermore, neoplastic and inflammatory processes that primarily occur in the extravascular compartment could more readily be targeted for diagnostic purposes. For the latter, contrast-enhanced imaging of the extravascular space may increase sensitivity for detection of extravascular inflammatory (59) and neoplastic diseases (60), which probably are most likely missed with conventional MRI techniques. With contrast-enhanced imaging, disease-targeting substances may be conjugated with contrast agents to mark disease and monitor treatment.

There are to date no gadolinium-based MRI contrast agents approved for intrathecal use. In the present study, gadobutrol was used after special permission granted from the National Medicine Agency of Norway. Gadolinium accumulation in the brain after multiple intravenous administrations has been a major concern since it was first reported in 2014 (61). However, there have been no reports demonstrating brain toxicity (62); the US Food and Drug Administration and American College of Radiology have declared that there is no evidence to date that gadolinium accumulation in the brain is harmful and that there is no need to restrict its intravenous usage. In the present study, where intrathecal administration of an MRI contrast agent was utilized, no serious adverse events were noted among our study subjects, and at the 4-week MRI follow-up, no evidence of remaining gadobutrol within brain parenchyma or CSF was found (Supplemental Table 5). Interestingly, gadolinium deposits in brain tissue have recently been attributed to leakage from blood into CSF through the choroid plexus and entrance into the brain from the surface along perivascular spaces (62). In rats, gadolinium concentration has been shown to be higher in CSF than blood 4.5 hours after intravenous distribution and highest in brain after 24 hours (63). In principle, an intravenous dose could therefore potentially expose the brain perivascular compartment to a larger amount of contrast agent than a very small intrathecal dose, as used in this study. Macrocyclic contrast agents (including gadobutrol) are now generally preferred above linear agents, which are chemically less stable compounds and more susceptible to release into tissue (64, 65).
Reduced clearance of CSF tracer in iNPH. The other main finding of this study is the demonstration of reduced clearance of gadobutrol used as CSF tracer in a cohort of patients with dementia. Clearance of MRI contrast agents, such as gadobutrol, has previously been proposed to potentially resemble clearance of brain macromolecules with similar properties (14), including amyloid-β (7) and tau (21, 66). These proteins are found pathologically accumulated as insoluble amyloid-β and as component of neurofibrillary tangles, respectively, in brains of patients with Alzheimer’s disease (67). There is large overlap of typical findings in brain specimens of patients with Alzheimer’s disease and iNPH (68). We have recently demonstrated volume loss and reduced tracer clearance in an iNPH dementia cohort at the entorhinal area (69), where the first signs of neurodegeneration appear in Alzheimer’s disease (70). The apparent important role of artery pulsations in parenchymal tracer propagation compares well with findings of vascular comorbidity (71) and reduced subarachnoid space propagation of CSF tracer in iNPH (13). Furthermore, brain subregions being closest to large artery trunks at the brain surface are regions traditionally considered as part of the limbic system (entorhinal cortex, hypothalamus, insula, amygdala, cingulum, etc.), representing the phylogenetically oldest part of the cerebral cortex (allocortex). On the MRIs, these regions enhanced earlier and enhancement was richer compared with other brain regions (Figures 1,2, 4 and 5; Table 2; Supplemental Table 3). We find it compelling how this pattern of time-progressive “limbic enhancement” resembles the pattern of neurofibrillary tangle advancement in evolving Alzheimer’s disease (72), which to date is incompletely understood. A contributing cause to disease evolution might therefore be that brain regions being most dependent on CSF pulsations for clearance of brain macromolecules are also the ones most susceptible to disease when pulsations are restricted or when paravascular pathways are blocked by any cause. To what extent restricted CSF pulsations and restricted paravascular flow may be instrumental in the pathogenesis behind Alzheimer’s disease, iNPH, and other brain proteinopathies should be subject to future studies.

In the present study, the 24-hour time point showed the best discrimination between patients and REF individuals. Assessment of cerebral clearance with MRI by absolute quantities is, however, not possible at this level, as normalized T1 signal cannot be assumed to be strictly proportional with amount of contrast agent in each image voxel. Another important limitation of our comparison of iNPH patients with REF individuals is their different ages, and paravascular clearance has been shown to be impaired with ageing (22). Nevertheless, CSF flow patterns in iNPH patients also differ clearly from those of REF individuals by means of an early and persisting ventricular regurgitation of tracer, as well as by enhancement of periventricular tissue across the ependymal ventricular lining, which was particularly prominent on late scans (Figures 1, 2, 4 and 5). Net tracer flux into brain ventricles may be hypothesized to express a salvage route for CSF flow along the pathway of least resistance and subsequently lead to compromised brain paravascular flow. AQP4 density and its anchoring protein dystrophin 71 have been found to be reduced at astrocytic end feet in iNPH (73). AQP4 loss may cause swelling of end feet and thereby may reduce the distance of inter end feet gaps (74), which may in its turn deteriorate permeability for solutes. The 1-mm image resolution of our MRI exams prevents this direct observation and therefore remains speculative. Furthermore, we note that parenchymal enhancement in iNPH patients was comparable with that of REF individuals (Figure 6) and, in some regions, was more pronounced (Figure 6 and Supplemental Table 4). Increased width of perivascular spaces has been shown to be associated with increasing age (75) and may thus be a determining factor for parenchymal tracer enrichment more than clearance, which has rather been proposed to be dependent on convective forces and AQP4 status in the glymphatic system (7). Our finding is also analogous to those of an MRI study of rats with diabetes mellitus type 2, where perivascular tracer influx was comparable to that in nondiabetes mellitus rats (45). Based on previous data (14), it was suggested that diabetes mellitus type 2 rats have enlarged perivascular spaces owing to increased risk of small vessel disease. This enlargement may therefore reduce resistance to perivascular tracer influx (76).

Moreover, we also found that the correlation between percentage change in tracer enrichment after 24 hours and volume of brain region (gray and white matter and CSF; Table 3) was different between REF and iNPH individuals. For gray matter of iNPH patients, there were negative correlations between CSF tracer enrichment within the respective region and volume of the specific region, i.e., increased volume of gray matter was associated with lower CSF contrast enrichment. Reduced brain tissue volume in iNPH patients may therefore also have contributed to the observed stronger tracer enrichment in this group at 24 hours. In future studies, CSF tracer clearance assessment with MRI may preferably be compared in patients with minimal cognitive impairment and age-matched healthy controls. This would further reveal the full potential of MRI-based assessment of brain metabolic function to detect neurodegenerative disease in the presymptomatic phase.
Conclusions. In this study, we have used repeated MRI to demonstrate human brain-wide enrichment of a CSF tracer administered intrathecally. Our observations in human brain add to previous evidence from animal studies, with important distinctions. In humans, CSF tracer distributed centripetally from the surface toward structures in the deep parts of the brain, but over a much longer time course than that reported in animals. Vascular pulsations mediated to CSF seem to play an important role for tracer entry into brain parenchyma. Access of substances to the entire brain extravascular compartment may potentially have large implications for new therapeutic and diagnostic approaches to neurological disease. Delayed tracer clearance from brain parenchyma in patients with iNPH dementia suggests that intrathecal contrast-enhanced MRI could be used to diagnose preclinical neurodegenerative disease.

Methods

The study was prospective and observational, including consecutive patients with tentative CSF leakage and idiopathic intracranial hypotension and patients with iNPH who underwent MRI before and at several time points following intrathecal lumbar injection of the MRI contrast agent gadobutrol. They were prospectively enrolled from October 2015 to September 2016. Exclusion criteria were history of hypersensitive reactions to contrast agents, history of severe allergy reactions in general, evidence of renal dysfunction, and age <18 or >80 years as well as pregnancy or breastfeeding for women.

REF subjects. The individuals with idiopathic intracranial hypotension were referred to the Department of Neurosurgery, Oslo University Hospital — Rikshospitalet, from local neurological departments, based on clinical symptoms and imaging findings indicative of idiopathic intracranial hypotension. They were referred for clinical workup of suspected CSF leakage and underwent MRI with intrathecal gadobutrol, with the primary goal to define site of CSF leakage. These patients were recruited prospectively and consecutively in parallel with iNPH patients. Five of eight REF individuals (REF nos. 1, 2, 4, 5, and 6) have been reported in a previous study (13), though with another method of assessing CSF contrast enrichment and a limited selection of brain regions.

Patients with iNPH dementia. The iNPH patients were referred to the Department of Neurosurgery, Oslo University Hospital — Rikshospitalet, from local neurological departments, based on clinical symptoms and findings indicative of iNPH and imaging findings of ventriculomegaly. Within the Department of Neurosurgery, a clinical assessment was done, and clinical severity was graded based on a previously described NPH grading scale (77, 78). Patients were selected for CSF diversion surgery based on a combination of clinical symptoms and findings indicative of iNPH, imaging findings, comorbidity, and results of intracranial pressure monitoring, as previously described (77, 78). Four of nine iNPH subjects (iNPH nos. 1, 2, 3, and 4) have been reported in a previous study (13), though with another method of assessing CSF contrast enrichment and a limited selection of brain regions.

MRI protocol. We used a 3 Tesla Philips Ingenia MRI scanner (Philips Medical Systems) with equal imaging protocol settings at all time points to acquire sagittal 3D T1-weighted volume scans. The imaging parameters were as follows: repetition time, “shortest” (typically 5.1 ms); echo time, “shortest” (typically 2.3 ms); Flip angle, 8 degrees; field of view, 256 × 256 cm; and matrix, 256 × 256 pixels (reconstructed 512 × 512). We sampled 184 overcontiguous (overlapping) slices with 1-mm thickness, which was automatically reconstructed to 368 slices with 0.5-mm thickness. The total duration of each image acquisition was 6 minutes and 29 seconds. At each time point, imaging stacks were planned using an automated anatomy recognition protocol based on landmark detection in MRI data (SmartExam, Philips Medical Systems) to secure consistency and reproducibility of the MRI slice placement and orientation.

Before gadobutrol administration, we also scanned patients with a sagittal 3D FLAIR volume sequence, where the main imaging parameters were as follows: repetition time = 4,800 ms; echo time, “shortest” (typically 318 ms); inversion recovery time, 1,650 ms; field of view, 250 × 250 mm; and matrix, 250 × 250 pixels (reconstructed 512 × 512). We sampled 184 overcontiguous slices with 1-mm thickness, which was automatically reconstructed to 365 slices with 0.5-mm thickness.

Intrathecal administration of gadobutrol. After the precontrast MRI, which was acquired approximately 8 a.m., an interventional neuroradiologist performed x-ray guided lumbar puncture. Correct position of the syringe tip in the SAS was verified by CSF backflow from the puncture needle, and injecting of a small amount (typically 3 ml) of 270 mg I/ml ioxixanol (Visipaque, GE Healthcare) confirmed unrestricted distribution of radiopaque contrast agent in the lumbar SAS. Thereafter, intrathecal injection of 0.5 ml of 1.0 mmol/ml gadobutrol (Gadovist, Bayer) was performed. After needle removal, the study subjects were
instructed to rotate themselves around the long axis of the body once before transportation back to the MRI suite, while remaining in the supine position. Supplemental Video 3 illustrates the distribution of CSF tracer within the CSF space following intrathecal administration.

**Postcontrast MRI acquisitions.** Consecutive, identical MRI acquisitions using the previously outlined MRI protocol parameters were performed as soon as possible after intrathecal gadobutrol administration (typically with approximately 10 minutes delay) and performed approximately every 10 minutes during the first hour after contrast agent injection. The study participants were thereafter instructed to remain supine in bed. One pillow under the head was allowed, allowing for 15-degrees head elevation. Repeated, identical image acquisitions were then performed approximately every 2 hours after intrathecal gadobutrol administration until afternoon, at approximately 4 p.m. All transfer of study subjects between the neurosurgical department and the MRI suite, and between the bed and the MRI table, was performed by the hospital staff to allow for the patient to remain in the supine position. The final MRI scanning was performed next morning (approximately 24 hours after contrast agent injection). Patients and controls were allowed to move without any restrictions between the 4 p.m. examination at the end of day 1 and the 24-hour scan next morning.

While the MRI exams, for practical reasons, could not be obtained at identical time points for every study subject, all exams were categorized into the following time intervals: precontrast, 1–2 hours, 2–4 hours, 4–6 hours, 6–9 hours, 24 hours, 48 hours, and after 4 weeks.

**Image analysis.** The FreeSurfer software (version 6.0) (http://surfer.nmr.mgh.harvard.edu/) was used for segmentation, parcellation, and registration/alignment of the longitudinal data (Supplemental Figures 2 and 3). The segmentation and parcellation acquired from FreeSurfer were used to investigate the increase of T1 intensity due to CSF tracer. The methods are documented in a review (79). This includes removal of nonbrain tissue using a hybrid watershed/surface deformation procedure (80), automated Talairach transformation, and segmentation of the subcortical white matter and deep gray matter volumetric structures (including hippocampus, amygdala, caudate, putamen, and ventricles) (81, 82).

The MR images of each patient were used to create a median template registered to the baseline, a process has been previously described (83). Hence, for each patient, the MR images were registered to the corresponding template using a rigid transformation (83). The registrations were subsequently checked manually by LMV, and no significant errors were visible.

The T2 images (for all iNPH patients except 3) were also used for the segmentation with FreeSurfer. Additionally, the specification of large ventricles was added to the segmentation processes for the NPH patients.

The segmentation of 7 iNPH patients was edited due to segmentation errors. These editions were due to severe segmentation errors, such as mislabeling of ventricle volume as white matter and the inclusion of dura as part of the parenchyma; other minor errors were not considered. These corrections were automatic based on the patient T2 image, except in the case of 1 iNPH patient, which was done manually.

**Parameters derived from gadobutrol enhancement.** For each segmented area, we computed the median T1 signal unit for each time point. Further, we divided the median signal unit against the signal unit of a REF ROI placed within the posterior part of the superior sagittal sinus in axially reconstructed images from the same T1 volume scan (Supplemental Figure 4). We referred to the result of this as normalized T1 signal units; this process corrects for any baseline changes of image gray scale due to image scaling. Previous observations indicate no measurable enhancement of contrast agent at MRI in the REF location after intrathecal injection of gadobutrol at this concentration (13).

**CSF contrast enrichment and volume of brain region.** For REF and iNPH subjects, we determined Pearson correlation coefficients by comparing the percentage change in signal unit ratio after 24 hours and the volume of the specific brain region (white and gray matter and CSF of lateral ventricles). Volumes were computed by summation of the number of voxels of every subregion in the segmentation and multiplication of the result by the voxel size of 1.0 mm³.

**Statistics.** Categorical data are shown as the number of observations (percentage) and continuous data are shown as mean (SD) or mean (SEM), as appropriate. We estimated from the image analysis the mean response for each patient at 0 (precontrast), 1.5–2.0, 2–4, 4–6, 6–9, 24, and 48 hours and at 4 weeks follow-up. The repeated measurements were assessed with linear mixed models using a random intercept, robust SEM, and maximum likelihood estimation. Due to few observations, the results at 48 hours were omitted from the statistical model. Using a linear combination from the statistical model, we tested the difference between the REF and iNPH groups at the different points of follow-up.
For the statistical analysis, we used SPSS version 22 (IBM Corporation) or Stata/SE 15.0 (StataCorp LLC). Statistical significance was accepted at the 0.05 level (2-tailed).

Study approval. The Regional Committee for Medical and Health Research Ethics of Health, South East Region, Norway (2015/96); the Institutional Review Board of Oslo University Hospital (2015/1868); and the National Medicines Agency, Norway (15/04932-7) approved the study. The study participants were included after written and oral informed consent.

Author contributions
GR, LMV, KAM, and PKE conceptualized and designed the study. GR, LMV, AMD, AHP, SASV, KEE, KAM, and PKE provided investigation and formal analysis. GR, KAM, and PKE supervised and administrated the study and wrote the original draft. GR, LMV, AMD, AHP, SASV, KEE, KAM, and PKE wrote, reviewed, and edited the manuscript. All authors (GR, LMV, AMD, AHP, SASV, KEE, KAM, and PKE) approved the final manuscript.

Acknowledgments
We thank Øivind Gjertsen, Bård Nedregaard, and Ruth Sletteberg from the Department of Radiology, Oslo University Hospital — Rikshospitalet, who performed the intrathecal gadobutrol injections in all study subjects. We also sincerely thank the Intervention Centre and Department of Neurosurgery at Oslo University Hospital — Rikshospitalet for providing valuable support with MR scanning and care for all study subjects throughout the study. Finally, we sincerely thank the Nursing Staff and Hydrocephalus Outward Clinic, Department of Neurosurgery Oslo University Hospital — Rikshospitalet, for care of all study subjects throughout the study.

Address correspondence to: Per K. Eide, Department of Neurosurgery, Oslo University Hospital — Rikshospitalet, Pb 4950 Nydalen, N-0424 Oslo, Norway. Phone: 47.2307.4321; Email: p.k.eide@medisin.uio.no or peide@ous-hf.no.

Paper II

Delayed clearance of cerebrospinal fluid tracer from choroid plexus in idiopathic normal pressure hydrocephalus

Per K. Eide, Lars M. Valnes, Are H. Pripp, Kent-Andre Mardal, Geir Ringstad
Delayed clearance of cerebrospinal fluid tracer from choroid plexus in idiopathic normal pressure hydrocephalus

Per Kristian Eide1,2, Lars Magnus Valnes3, Are Hugo Pripp4, Kent-Andre Mardal3,5 and Geir Ringstad2,6

Abstract
Impaired clearance of amyloid-β from choroid plexus is one proposed mechanism behind amyloid deposition in Alzheimer’s disease. The present study examined whether clearance from choroid plexus of a cerebrospinal fluid tracer, serving as a surrogate marker of a metabolic waste product, is altered in idiopathic normal pressure hydrocephalus (iNPH), one subtype of dementia. In a prospective observational study of close to healthy individuals (reference cohort; REF) and individuals with iNPH, we performed standardized T1-weighted magnetic resonance imaging scans before and through 24 h after intrathecal administration of a cerebrospinal fluid tracer (the magnetic resonance imaging contrast agent gadobutrol). Changes in normalized T1 signal within the choroid plexus and cerebrospinal fluid of lateral ventricles were quantified using FreeSurfer. The normalized T1 signal increased to maximum within choroid plexus and cerebrospinal fluid of lateral ventricles 6–9 h after intrathecal gadobutrol in both the REF and iNPH cohorts (enrichment phase). Peak difference in normalized T1 signals between REF and iNPH individuals occurred after 24 h (clearance phase). The results gave evidence for gadobutrol resorption from cerebrospinal fluid by choroid plexus, but with delay in iNPH patients. Whether choroid plexus has a role in iNPH pathogenesis in terms of delayed clearance of amyloid-β remains to be shown.

Keywords
Idiopathic normal pressure hydrocephalus, choroid plexus, cerebrospinal fluid, CSF tracer, clearance, magnetic resonance imaging, intrathecal contrast agents, glymphatic, humans

Received 21 December 2018; Revised 29 June 2019; Accepted 1 July 2019

Introduction
The choroid plexus is a highly vascularized tissue located in the cerebral ventricles and has traditionally been considered a main source of cerebrospinal fluid (CSF) production, thereby being a driving force behind CSF bulk flow from the ventricular system to the subarachnoid space.1–3 More recently, the choroid plexus has emerged as a vital structure for neuroprotection, neuro-immune regulation, and homeostasis of the brain’s chemical milieu in a broad sense.4 Experimental research also provides evidence that the choroid plexus has a function in clearance of amyloid-β from the CSF, thus playing a role in the pathogenesis of Alzheimer’s disease.5,6

The description of the cerebral glymphatic system in 20127 and the demonstration of dural lymphatic vessels in 20158,9 have facilitated renewed interest in the mechanisms behind clearance of metabolic waste solutes from the brain. Impaired glymphatic7 and lymphatic10 function may both lead to impaired clearance of metabolic waste products, such as amyloid-β, which in

1Department of Neurosurgery, Oslo University Hospital – Rikshospitalet, Oslo, Norway
2Faculty of Medicine, Institute of Clinical Medicine, University of Oslo, Oslo, Norway
3Department of Mathematics, University of Oslo, Oslo, Norway
4Oslo Centre of Biostatistics and Epidemiology, Research Support Services, Oslo University Hospital, Oslo, Norway
5Center for Biomedical Computing, Simula Research Laboratory, Lysaker, Norway
6Division of Radiology and Nuclear Medicine, Department of Radiology, Oslo University Hospital – Rikshospitalet, Oslo, Norway

Corresponding author:
Per Kristian Eide, Department of Neurosurgery, Oslo University Hospital – Rikshospitalet, PB 4950 Nydalen, Oslo 0424, Norway.
Email: peide@ous-hf.no
neuro-degenerative disease may accumulate and cause toxic damage to the brain. We previously demonstrated impaired glymphatic clearance of a magnetic resonance imaging (MRI) contrast agent (gadobutrol), serving as a CSF tracer in individuals with idiopathic normal pressure hydrocephalus (iNPH).\textsuperscript{11–13} This CSF tracer is a molecule of size 604 Da that may serve as a surrogate marker of clearance of metabolic waste products from brain. The disease iNPH is characterized by dementia, gait ataxia, urinary incontinence and enlarged size of the cerebral ventricles,\textsuperscript{14} and is characterized by neuro-degeneration and deposition of amyloid-\(\beta\) in a significant proportion of the patients.\textsuperscript{15} Another feature of this entity is the significant alterations of the CSF circulation. Therefore, CSF diversion surgery may provide sustained clinical improvement even though with some reduction of treatment effect over time.\textsuperscript{16}

Given the important role of impaired cerebral clearance function for evolvement of neuro-degeneration and dementia, characterizing the underlying mechanisms seems crucial. While glymphatic clearance seems reduced in iNPH patients,\textsuperscript{11–13} it remains unknown whether or not clearance from the choroid plexus is reduced. In this present study, we for the first time utilized a CSF tracer for in vivo assessment of choroid plexus clearance function in references and a cohort with a dementia disease.

\textbf{Materials and methods}

\textit{Patients and study design}

The study was approved by the Regional Committee for Medical and Health Research Ethics (REK) of Health Region South-East, Norway (2015/96), the Institutional Review Board of Oslo university hospital (2015/1868), and the National Medicines Agency (15/04932-7). The conduct of the study was governed by ethical standards according to the Helsinki Declaration of 1975 (and as revised in 1983). Study participants were included after written and oral informed consent.

A prospective and observational study design was used, and included consecutive patients with suspected CSF leakage, causative of idiopathic intracranial hypotension, and patients with iNPH. All individuals underwent MRI before and at several time points following intrathecal lumbar injection of the MRI contrast agent gadobutrol during a study period from October 2015 to September 2016.

Exclusion criteria were: History of hypersensitivity reactions to contrast agents, history of severe allergy reactions in general, evidence of renal dysfunction, pregnant or breastfeeding women, and age <18 or > 80 years.

The reference (REF) subjects had a tentative diagnosis of idiopathic intracranial hypotension and were referred to the Department of Neurosurgery, Oslo University Hospital – Rikshospitalet, Oslo, Norway, from local neurological departments for clinical work-up of suspected CSF leakage. They underwent MRI with intrathecal gadobutrol with the primary indication to define site of CSF leakage, and were recruited prospectively and consecutively in parallel with iNPH patients. Patients with iNPH were referred to the Department of Neurosurgery, Oslo University Hospital – Rikshospitalet, Oslo, Norway, from local neurological departments, based on clinical symptoms and findings indicative of iNPH, and imaging findings of ventriculomegaly. Within the Department of Neurosurgery, clinical severity was graded based on a previously described NPH grading scale.\textsuperscript{16,17} Patients were selected for shunt surgery based on assessing clinical symptoms and findings, imaging findings, co-morbidity, and results of intracranial pressure (ICP) monitoring, as previously described.\textsuperscript{16,17}

\textit{MRI protocol}

A 3 Tesla Philips Ingenia MRI scanner (Philips Medical systems, Best, The Netherlands) with equal imaging protocol settings at all the time points was applied to acquire sagittal 3D T1-weighted volume scans with ultrafast gradient echo and preparation pulse (T1 FFE). The following imaging parameters were used: repetition time (TR) = “shortest” (typically 5.1 ms), echo time (TE) = “shortest” (typically 2.3 ms), inversion time = 853 ms, flip angle (FA) = 8 degrees, field of view (FOV) = 256 x 256 mm, and matrix = 256 x 256 pixels (reconstructed 512 x 512). We sampled 184 over-contiguous (overlapping) slices with 1 mm thickness, which was automatically reconstructed to 368 slices with 0.5 mm thickness.

In addition, a T2-weighted volume acquisition (T2 VISTA) was obtained with the following parameters: TR = 2500 ms, TE = 330 ms, FA = 90 degrees, FOV = 250 x 250 mm, matrix size = 252 x 250 pixels.

\textit{Intrathecal administration of gadobutrol}

After the pre-contrast MRI exam, an interventional neuroradiologist performed X-ray-guided lumbar puncture. Correct position of the syringe tip in the subarachnoid space was verified by CSF backflow from the puncture needle, and a small amount (typically 3 ml) of 270 mg 1/ml iodixanol (Visipaque\textsuperscript{TM}, GE Healthcare, USA) was injected to confirm unrestricted distribution of radiopaque contrast agent in the lumbar SAS. Then, 0.5 ml of 1.0 mmol/ml gadobutrol (Gadovist\textsuperscript{TM}, Bayer Pharma AG, Berlin, Germany) was injected.
intrathecally through the same needle. Following removal of the needle, the study subjects were instructed to rotate themselves around the long axis of the body once before transportation back to the MRI suite, while remaining in the supine position.

Post-contrast MRI acquisitions

Consecutive, identical MRI acquisitions using the previously outlined MRI protocol parameters were performed after intrathecal gadobutrol administration. The study participants were instructed to remain supine in bed. One pillow under the head allowed for up to 15 degrees head elevation, and all transfer of study subjects between the neurosurgical department and the MRI suite, and between the bed and the MRI table, was performed by the hospital staff to help the patient remain in the supine position. Patients and controls were allowed to move without any restrictions between the 4 p.m. examination at the end of day one and the 24 h scan next morning.

While the MRI exams, for practical reasons, could not be obtained at identical time points for every study subject, all exams were categorized into the following time intervals: Pre-contrast, 1.5–2 h, 2–4 h, 4–6 h, 6–9 h, 24 h and 48 h.

Image analysis

The FreeSurfer software (version 6.0) (http://surfer.nmr.mgh.harvard.edu/) was used for segmentation, parcellation, and registration/alignment of the longitudinal data. The segmentation and parcellation acquired from FreeSurfer were used to investigate the increase of T1 signal intensity due to CSF tracer enhancement. The methods are documented in a review.18 In this study, we segmented the choroid plexus within the lateral ventricles as well as the CSF within the lateral ventricles.

The MR images of each patient were used to create a median template registered to the baseline, the process of which has been previously described.19 Hence, for each patient, the MR images were registered into the corresponding template using a rigid transformation.19 The registrations were subsequently checked manually by one of the co-authors (LMV), and no significant errors were visible.

The T2-weighted images (except for three iNPH patients who had no T2 images) were also used for the segmentation with FreeSurfer. Additionally, the specification of large ventricles was added to the segmentation processes for the iNPH patients.

The segmentation of seven iNPH patients was edited due to segmentation errors. The corrections were automatic based on the patients T2 image, except in one iNPH patient who had no T2 image, which required manual editing to correct. Notably, the segmentation procedure did not impact the measured T1 signal units within the regions of interest (Supplementary Figure 1).

We determined the volume of choroid plexus and lateral ventricles, which was done by summarizing all voxels with the same segmentation. The corrections of the lateral ventricle volumes were done by labeling voxels in the near vicinity of the lateral ventricles, such as white matter and choroid plexus.

T1 signal derived parameters

For each segmented area, the median T1 signal unit was computed for each time point. Further, the median signal unit was divided against the signal unit of a reference ROI placed within the posterior part of the superior sagittal sinus in axially reconstructed images from the same T1 volume scan. The ratio refers to as normalized T1 signal units and corrects for any baseline changes of image grey scale due to potential image scaling between single scans. Previous observations indicate no measureable enhancement of contrast agent at MRI in the reference location after intrathecal injection of gadobutrol at this concentration.11

Estimation of CSF tracer concentration in lateral ventricle

The CSF tracer concentration (C) causes the T1 time to be shortened with the following relation

\[
1/T1 = 1/T1(0) + R \times C
\]  

(1)

According to Lu et al.,20 we assumed that T1(0) = 4300 ms, and with reference to Rohrer et al.,21 the relaxivity constant \( R = 3.2 \text{mM}^{-1} \text{s}^{-1} \). Since imaging parameters are the same for each T1 image, the T1-time becomes the only variable in the T1 protocol signal equation (equation (1)) in Gowland et al.22 Then the concentration C is estimated by computing the normalized T1 signal increase from baseline that is needed to achieve the average signal increase.

Statistical analysis

A linear mixed model for repeated measurements with a random intercept for study participant using maximum likelihood estimation with robust standard errors analyzed the effect of segment (i.e. choroid plexus versus CSF) and cohort (i.e. REF versus iNPH) on normalized T1 signal units at the time interval from 4 to 24 h. In addition to the main fixed effects of segment, cohort and age, the statistical model included an interaction term between segment and cohort. The interaction term assessed the modifying effect of cohort on normalized
T1 signal units in segment. Thus, a statistical significant interaction term could indicate that the mean difference in normalized T1 signal units between choroid plexus and CSF was different in the iNPH and REF cohort.

Statistical significance was accepted at the 0.05 level. The descriptive statistics was performed using the SPSS software version 20 (IBM Corporation, Armonk, NY), and the linear mixed model analysis was performed with Stata/SE 15.0 for Windows (Stata Corp LLC, College Station, TX).

Results

The study included two cohorts of eight REF and nine iNPH patients, respectively (Table 1). The REF individuals had a tentative diagnosis of idiopathic intracranial hypotension due to CSF leakage, which was confirmed in 3/8 individuals. No CSF leakage was seen in 5/8 individuals, who we consider close to healthy. However, the patient cohorts differed in several respects (Table 1).

Table 2 presents measures of ventriculomegaly/CSF circulation failure (Evans index, callosal angle, and disproportionately enlarged subarachnoid space hydrocephalus, DESH), as well as volume of choroid plexus in lateral ventricle and lateral ventricular volume size in REF and iNPH individuals. We found no differences in volume of choroid plexus between patient cohorts, while volumes of CSF spaces were larger in the iNPH cohort (Table 2).

After intrathecal administration of the CSF tracer (i.e. MRI contrast agent), tracer propagated into the ventricular system and enriched the choroid plexus of both REF and iNPH individuals (Figure 1). This was, however, markedly more pronounced in iNPH than REF subjects. The changes in signal unit ratios within choroid plexus of lateral ventricles and the CSF of lateral ventricles of the (a) REF and (b) iNPH cohorts are illustrated in Figure 2. For both locations, the tracer enrichment was more pronounced in the iNPH than in REF individuals (Figure 2). Moreover, Table 3 presents the percentage change in normalized T1 signal from before CSF tracer administration within choroid plexus and lateral ventricle of the REF and iNPH cohorts. While tracer enrichment within the lateral ventricle of REF individuals was close to significant (P = 0.052), the MRI signal increased significantly (P < 0.001) at the other locations (Table 3).

While a CSF leakage was verified in 3/8 REF patients, the presence of CSF leakage seemed not to impact the measured T1 signal units. We found no differences in normalized T1 signal units between those individuals with/without verified leakage at any time point.

Table 1. Demographic and clinical information about the REF and iNPH cohorts at the time MRI.

<table>
<thead>
<tr>
<th></th>
<th>REF</th>
<th>iNPH</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>8</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Sex (F/M)</td>
<td>6/2</td>
<td>1/8</td>
<td>0.008</td>
</tr>
<tr>
<td>Age (years)</td>
<td>38.4 ± 17.3</td>
<td>68.3 ± 9.5</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Symptoms reported by patient and/or family</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gait disturbance</td>
<td>2 (25%)</td>
<td>9 (100%)</td>
<td>0.001</td>
</tr>
<tr>
<td>Urinary incontinence</td>
<td>0 (0%)</td>
<td>6 (67%)</td>
<td>0.004</td>
</tr>
<tr>
<td>Cognitive impairment</td>
<td>0 (0%)</td>
<td>8 (89%)</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Duration of symptoms (months)</td>
<td>42.4 ± 40.6</td>
<td>21.3 ± 10.4</td>
<td>ns</td>
</tr>
<tr>
<td>NPH score</td>
<td>15 (14–15)</td>
<td>13 (11–13)</td>
<td>&lt;0.001</td>
</tr>
</tbody>
</table>

Note: Categorical data presented as numbers; continuous data presented as mean ± standard deviation. Significant differences between groups were determined by Pearson Chi-square test for categorical data and by independent samples t-test for continuous data. iNPH: idiopathic normal pressure hydrocephalus; REF: reference patients.

Table 2. Radiological measures and volumes of cerebral ventricles and choroid plexus of REF and iNPH cohorts.

<table>
<thead>
<tr>
<th></th>
<th>REF</th>
<th>iNPH</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evans index</td>
<td>0.30 ± 0.05</td>
<td>0.38 ± 0.04</td>
<td>0.001</td>
</tr>
<tr>
<td>DESH (Y/N)</td>
<td>0/8</td>
<td>6/3</td>
<td>ns</td>
</tr>
<tr>
<td>Callosal angle (degrees)</td>
<td>117.3 ± 14.3</td>
<td>76.3 ± 29.0</td>
<td>0.003</td>
</tr>
<tr>
<td>Volume measures (ml)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Choroid plexus</td>
<td>0.75 ± 0.22</td>
<td>0.88 ± 0.23</td>
<td>ns</td>
</tr>
<tr>
<td>CSF, lateral ventricle</td>
<td>14.83 ± 14.43</td>
<td>68.85 ± 18.52</td>
<td>&lt;0.001</td>
</tr>
</tbody>
</table>

Note: Categorical data presented as numbers; continuous data presented as mean ± standard deviation. Significant differences between groups were determined by Pearson Chi-square test for categorical data and by independent samples t-test for continuous data. CSF: cerebrospinal fluid; DESH: disproportionate enlargement of subarachnoid spaces hydrocephalus; iNPH: idiopathic normal pressure hydrocephalus; REF: reference patients.
point, neither the choroid plexus (lowest P-value 0.34) nor the CSF of the lateral ventricle (lowest P-value 0.21) (data not shown).

Figure 2 indicates that both choroid plexus enhancement and clearance of CSF tracer was different from REFs in the iNPH patients. Significant differences were seen during the enhancement phase after 4–6 and 6–9 h, as well as during the clearance phase at 24 h (Table 3). In particular, the most pronounced difference between REF and iNPH in choroid plexus was seen after 24 h, indicative of delayed clearance of CSF tracer from choroid plexus in iNPH disease.

Estimation of CSF tracer concentration and level also gave evidence for delayed clearance from lateral ventricle in iNPH. Table 4 presents the concentration and level of CSF tracer in lateral ventricle of iNPH and REF individuals, estimated according to equation (1). Concentrations for REF individuals were 0.0079 and 0.0022 mM after 6–9 and 24 h, respectively, while corresponding numbers in iNPH were 0.031 and 0.024 mM. As such, the reduction of tracer in lateral ventricles during this period in iNPH and REF was 23% and 72%, respectively, in a relative sense. On the other hand, in iNPH, the combination of increased volume of lateral ventricles and more pronounced CSF tracer enrichment resulted in increased absolute amounts of CSF tracer of 2.13 and 1.65 µmol after 6–9 and 24 h, respectively. In an absolute sense, the corresponding reduction in CSF tracer levels from 6–9 to 24 h after tracer administration was −0.48 and −0.0057 µmol, respectively.

To further test whether the clearance of CSF tracer from choroid plexus per se was significantly different in iNPH, we determined the interaction term whether the iNPH diagnosis itself modified the CSF tracer enrichment within the choroid plexus and CSF of lateral ventricle. As indicated in Figure 3, a significant interaction term provided evidence for delayed clearance of CSF tracer from choroid plexus of iNPH compared to REF. Hence, in iNPH, enhancement was higher in CSF compared to choroid plexus, while in REF subjects, enhancement was higher in choroid plexus compared to CSF (Figure 3). Moreover, after correcting the interaction presented in Figure 3 for age-differences between REF and iNPH cohorts, the interaction term between REF and iNPH individuals remained significant. That is, the differences in CSF tracer enhancement within choroid plexus could not be explained by age-differences between REF and iNPH subjects.

Discussion

The present data indicate that the MRI contrast agent gadobutrol is resorbed from CSF by the choroid plexus...
of the lateral ventricles. In iNPH patients, CSF enrichment of gadobutrol exceeded that of choroid plexus and was opposite to observations made in a cohort of healthy, or close to healthy, subjects. The findings suggest delayed clearance of molecules with similar features from choroid plexus in iNPH.

**Patients**

The iNPH cohort was older than the reference cohort. An effect of age on results of CSF tracer clearance could therefore be anticipated. However, the differences in CSF tracer enrichment within choroid plexus and CSF of REF and iNPH individuals remained consistent after adjusting for age in the statistical analysis.

### Table 3. Percentage change in signal unit ratios at various time points after intrathecal gadobutrol in REF and iNPH patients.

<table>
<thead>
<tr>
<th>Anatomical region</th>
<th>REF</th>
<th>iNPH</th>
<th>Overall significance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Group 1.5–2 h</td>
<td>2–4 h</td>
<td>4–6 h</td>
</tr>
<tr>
<td>Choroid plexus</td>
<td>3.1–63</td>
<td>25</td>
<td>21</td>
</tr>
<tr>
<td>CSF, lateral ventricle</td>
<td>4.43</td>
<td>79</td>
<td>340</td>
</tr>
</tbody>
</table>
Clearance of molecules from lateral ventricles

In the present study, the combined increased lateral ventricular volume and increased CSF tracer enrichment gave 19.4 and 50.0 times higher CSF tracer amounts in the lateral ventricles of iNPH patients, as compared with REF individuals after 6–9 and 24 h, respectively. The increased CSF tracer enhancement in lateral ventricles is related to CSF flow alterations in iNPH. We are currently investigating whether the retrograde aqueductal flow of CSF in iNPH patients, as previously reported by us,11,23 explains the increase.

Table 4. Estimated concentration of CSF tracer within lateral ventricles of REF and iNPH patients.

<table>
<thead>
<tr>
<th></th>
<th>REF</th>
<th>iNPH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume CSF, lateral ventricles (ml)</td>
<td>14.287</td>
<td>68.845</td>
</tr>
<tr>
<td>Time after i.th. CSF tracer</td>
<td>6–9 h</td>
<td>24 h</td>
</tr>
<tr>
<td>Signal increase from baseline (ratio)</td>
<td>7.56</td>
<td>2.73</td>
</tr>
<tr>
<td>Concentration, CSF tracer (mM)</td>
<td>0.0079</td>
<td>0.0022</td>
</tr>
<tr>
<td>Change in concentration of CSF tracer from 6–9 h to 24 h (mM)</td>
<td>−0.0057</td>
<td>−0.007</td>
</tr>
<tr>
<td>Level of CSF tracer (µmol)</td>
<td>0.11</td>
<td>0.033</td>
</tr>
<tr>
<td>Change in level of CSF tracer from 6–9 h to 24 h (µmol)</td>
<td>−0.077</td>
<td>−0.48</td>
</tr>
</tbody>
</table>

Note: The concentrations and levels of CSF tracer within the lateral ventricles were computed according to equation (1), see Materials and Methods section. CSF: cerebrospinal fluid; iNPH: idiopathic normal pressure hydrocephalus; REF: reference patients.

The present results extended previous data showing that during clearance phase (from 6–9 h to 24 h after CSF tracer administration), the relative reduction of CSF tracer concentration in lateral ventricles of iNPH was one third that of REF, but was much larger in absolute terms. Accordingly, while the relative reduction of CSF tracer was reduced in iNPH compared to REF individuals, the reduction of CSF tracer in absolute terms was higher in iNPH, due to the stronger enrichment of ventricular CSF tracer in this latter patient group. First, we assume that clearance of CSF tracer is both trans-ependymal and via choroid plexus. The trans-ependymal route is indicated by strong peri-ventricular tracer enrichment in iNPH.11,12 Second, since the CSF tracer is biologically inert, it may seem less likely that the choroid plexus has an active role in removing CSF tracer; rather, the relative reduction is by passive clearance of tracer.

Clearance of molecules from choroid plexus

Several studies have demonstrated that reduced function of choroid plexus may be involved in disease processes causing neurodegeneration and Alzheimer’s disease.4,6,24,25 The present results provide in vivo evidence of reduced clearance of a molecule from choroid plexus of iNPH individuals. While tracer supply in CSF was higher in iNPH due to ventricular reflux, tracer enhancement in plexus was proportionally lower than in REF (Figure 3). In line with this assumption, experimental studies have shown reduced clearance of amyloid-β from choroid plexus, suggesting a role in the pathogenesis of Alzheimer’s disease.5,5 We consider the CSF tracer used in this study as a surrogate marker of cerebral waste products in general, but how well the present CSF tracer depicts the behavior of soluble amyloid-β has not been determined. It has already been established that the choroid plexus has a role in absorption of a range of other macromolecules from
CSF, for example, drugs such as penicillin and other endogenous and exogenous solutes.2

While this study provides evidence for transfer of an MRI contrast agent over the CSF–blood–barrier, transport of MRI contrast agents in the other direction, i.e. from blood to CSF, has recently been seen in several studies.26–28 In rats, the concentration of MRI contrast agent in CSF was clearly higher than in blood after 4.5 h, and most likely leaked into ventricular CSF through the choroid plexus.28 It may not seem reasonable that the highly vascularized choroid plexus should be a site for one-way traffic of substances from blood to CSF. Others have previously provided evidence of bidirectional exchange of substances from the choroid plexus,29–31 i.e. both from the blood and into the CSF and from the CSF into the blood.

Absorption of CSF by the choroid plexus was first suggested almost one hundred years ago by Foley in 1921 (reviewed by McComb32) and later in hydrocephalic infants,33 but has been contradicted by others.33 A recent review34 concluded that further investigation is required to establish the role of the choroid plexus in absorption of CSF.

Role of choroid plexus in CSF production

It has traditionally been considered that the choroid plexus is the main source of CSF production,35–37 while minor contributions to CSF production have been attributed to extra-ventricular sources such as the brain ependyma and parenchyma,32,38 and spinal cord ependyma.39 However, the traditional view of choroid plexus being the main producer of CSF has been heavily criticized.3,36,40 It is now clear that the subarachnoid CSF compartment is continuous with the entire paravascular compartment of the brain and spinal cord not only in animals7,41–43 but also in humans.12 New insights of continuous bi-directional fluid exchange over the entire blood–brain barrier clearly render for extra-choroidal CSF production.36

The present observations of significantly increased enhancement and delayed clearance of CSF tracer might also be attributed to reduced CSF tracer washout due to reduced CSF production in choroid plexus of iNPH. Experimental data from ageing rats and humans with iNPH or Alzheimer’s disease have shown reduced production of CSF from choroid plexus and impaired turnover of CSF.44 After intrathecal administration of an MRI contrast agent as CSF tracer, clearance of the CSF tracer from CSF spaces was significantly reduced, indicative of reduced CSF turnover in iNPH.11–13 The reduced CSF turnover within CSF spaces in iNPH was accompanied with delayed brain-wide clearance of CSF tracers that were interpreted as impaired lymphatic clearance,11,12 and suggested a common pathway behind iNPH and Alzheimer’s dementia and explains why amyloid-β deposits in brain tissue overlap significantly in these conditions.45

Others have reported that the function of choroid plexus changes during ageing and in CSF circulation failure.40 The iNPH cohort in this study was significantly older than the REF group. Furthermore, reduced CSF production in iNPH might be associated with increased pulsatile ICP and to lesser extent, increased static ICP, which is characteristic of iNPH patients responding to CSF diversion surgery.17 Several authors have previously reported that CSF production can be reduced as response to an increased ICP caused by hydrocephalus.46–48

A link between CSF production and lymphatic function is to be expected. Probably, a certain magnitude of CSF production is required to maintain sufficient CSF pressure to drive paravascular clearance pathways throughout the brain7,49,50 and to drive lymphatic efflux of waste molecules from the craniospinal compartment.51 This view is in contrast to the previous concept that CSF production is primarily required for mechanical protection of the brain and maintenance of the electrolytic environment and acid–base balance.52

Limitations

It should be noted that different levels of CSF tracer enhancement do not directly reflect changes in the concentration levels of the CSF tracer. The present results do therefore not provide measures by absolute quantities, even though such quantities may be estimated according to equation (1). T1 maps might be used for absolute quantifications of CSF tracer in future studies. While the choroid plexus is present in all cerebral ventricles, this study only examined choroid plexus in lateral ventricles. We would, however, expect results to be similar for all compartments.

Conclusions

The present study indicates that the MRI contrast agent gadobutrol is absorbed by the choroid plexus when utilized as CSF tracer. In patients with iNPH, this CSF tracer enhanced more strongly and was cleared with slower rate, from choroid plexus as compared to in REF subjects. Delayed CSF clearance of brain metabolites through choroid plexus may be instrumental for neurodegeneration in iNPH dementia.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.
Acknowledgements
We thank Dr. Øivind Gjertsen, Dr. Bård Nedregaard and Dr. Ruth Sletteberg from the Department of Radiology, Oslo University Hospital – Rikshospitalet, who performed the intrathecal gadobutrol injections in all study subjects. We also sincerely thank the Intervention Centre and Department of Neurosurgery at the Oslo University Hospital – Rikshospitalet for providing valuable support with MR scanning and care-taking of all study subjects throughout the study. Finally, we sincerely thank the Nurse Staff and Hydrocephalus outward clinic, Department of Neurosurgery, Oslo University Hospital – Rikshospitalet for care-taking of all study subjects throughout the study.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Authors’ contributions
Conceptualization and design: PKE, LMV, KAM, GR. Investigation, formal analysis and visualization: PKE, LMV, KAM, GR. Supervision, Administration and writing – original draft: PKE, GR. Statistical analysis: AHP. Writing, review and editing: PKE, LMV, AHP, KAM, GR. All authors (PKE, LMV, AHP, KAM, GR) approved the final version of the article. Correspondence and material requests: PKE.

ORCID ID
Per Kristian Eide https://orcid.org/0000-0001-6881-9280

Supplemental material
Supplemental material for this paper can be found at the journal website: http://journals.sagepub.com/home/jcb

References
Book excerpt III

An introduction to basic meshing and mathematical modeling for the human brain

Kent-Andre Mardal, Marie E. Rognes, Travis B. Thompson, Lars M. Valnes
Chapter 2
Basic data files and tools

The purpose of this chapter is to offer a high level overview regarding the basic data files and tools needed throughout the remainder of the text. Further chapters delve into specifics related to these files and tools, as necessary, to accomplish specific goals. In order to facilitate a broad discussion it is helpful to have in mind a concrete, albeit simple, mathematical modeling problem. Suppose we would like to study the distribution of a certain solute concentration throughout a domain, \( \Omega \), consisting of brain tissue matter. This concentration could represent, for example, an injected tracer or possibly a waste protein, such as amyloid-\( \beta \) or phosphorylated tau, that we are interested in tracking. A very simple starting point for a mathematical model could then look similar to the following:

\[
- \text{div} \left( K \nabla u \right) = f \quad \text{in} \quad \Omega. \tag{2.0.1}
\]

The unknown quantity, for which we hope to gain information by solving (2.0.1), is \( u \). However, there are other quantities, namely \( K \) and \( f \), involved in such a simple expression. How can we connect these quantities to the ‘real-world’ of medical data; furthermore, how can we extract such information from the data-sets typically produced by medical procedures?

Medical data, such as those provided by magnetic resonance imaging (MRI), play an enormous practical role in the ability of applied mathematicians to extract the necessary information enabling models such as (2.0.1). Take, for example, \( K \) in (2.0.1). Mathematically, this quantity is a matrix that, at each point \( x \in \Omega \), offers information on the tendency of \( u \) to flow; that is, in what direction \( u \) is more or less free to travel. We will see that information about \( K \) can be extracted from so-called diffusion tensor image (DTI) data. Likewise, \( f \)
in (2.0.1) mathematically represents a so-called source or sink of $u$; depending on the context this could be an injected dye tracer or possibly proteins carried into the parenchymal interstitium via paravascular spaces. We will see that information about $f$ can be ascertained from so-called arterial spin labeled (ASL) image data.

A more fundamental problem, however, is one of constructing a computational representation of the brain, i.e. $\Omega$ in (2.0.1), on which we can solve (2.0.1) to begin with; $K$ and $f$ notwithstanding. A computational representation of the physical brain is referred to as a mesh. An MRI machine utilizes time varying electromagnetic fields to create images which discriminate between different areas of the brain in terms of different structure or function. We will see that images obtained by a standard protocol called T1-weighted MRI can be used to extract the data necessary to create a preliminary computational-mesh-representation of the brain. We will also see that another standard protocol called T2-weighted MRI can be used to augment this mesh generation process and enhance the resulting computational mesh when needed.

In this section a brief introduction of the primary data files, used throughout the text, is endeavored. Broadly speaking, the primary files we will need to extract modeling information, even for simplified models such as (2.0.1), are: T1 weighted images (T1), diffusion tensor images (DTI), and arterial spin labeled (ASL) images. We will also discuss T2 weighted images (T2) though these are of secondary importance and primarily used to improve the computational meshes by offering increased resolution for areas we will wish to remove from a mesh. All of the files mentioned above are so-called magnetic resonance images (MRI); before discussing each of the different image types we will first overview MRI data in general.

### 2.1 Basic Anatomy

Some basic understanding of the brain anatomy is needed in the following text. We will frequently utilize MR images in the computational problems and processing, therefore as a method to familiarize ourselves with MRI, we will use MRI to illustrate the brain anatomy. The brain consists of the cerebrum, cerebellum and the brain stem, as shown in Figure 2.1. The cerebrum is made up of two hemispheres, which are connected through commissural nerve tracts, with the largest being the corpus callosum.

We distinguish between two types of tissue in the cerebrum and cerebellum, referred to as white and gray matter. The tissue structure in the cerebrum is
2.1 Basic Anatomy

Fig. 2.1 The image shows the T1 MR image of a brain with cerebrum in blue, cerebellum in red and brain stem in yellow.

shown in Figure 2.2, which displays the white matter, the cortical and subcortical gray matter. The gray matter tissue consist mainly of neurons, arteries and veins and supportive glia cells, while in the white matter we have axons that connects the neurons. The sub-cortical gray matter includes basic and important structures, such as the hippocampus and the basal ganglia and thalamus, which are frequently studied in neuroscience research because of their involvement in memory. It is believed that Alzheimer’s disease originates in these region.

The brain is protected by three layers of meninges; the dura, the arachnoid and the pial membrane, listed from the skull to the cerebrum. The pial membrane is found directly layered outside the cortical gray matter, and that the space between the pial and arachnoid membranes is known as the subarachnoid space (SAS). The SAS is filled with cerebrospinal fluid (CSF), and it continues from the brain stem down alongside the spinal cord.
Fig. 2.2 The image shows the tissue composition of the cerebrum, with red marking the cortical gray matter, blue marking the subcortical gray matter and white marking the white matter.

The SAS is also connected to the ventricular system, which interconnected fluid compartments inside the brain. The ventricular system consists of the two lateral ventricles, the 3rd ventricle and the 4th ventricle, shown in Figure 2.3. The narrow passage between the 3rd and 4th ventricle is known as the cerebral aqueduct.

2.2 Magnetic Resonance imaging

Magnetic resonance imaging (MRI) is a rich and versatile method which is able to generate images of the human brain, with different contrasts, in a non-invasive and harmless way. MRI enables detailed investigation of both tissue structure and morphology in addition to highlighting aspects related to
2.2 Magnetic Resonance imaging  

Fig. 2.3 The image shows the T1 MR image of a brain with the lateral ventricles marked in yellow, the 3rd ventricle marked in blue and the 4th ventricle marked as red.

tissue function. There are several types of MRI modalities including: structural MRI, diffusion MRI, perfusion MRI, and dynamic contrast enhanced MRI. The modality of MRI most likely to be familiar to most readers is structural MRI, which both T1 and T2 weighted MRI are examples of. Structural MRI (sMRI) provides high resolution images of the brain with outstanding soft tissue contrast, enabling accurate and quantitative segmentation of cortical and sub-cortical structures.

Quantifying tissue structure is a powerful feature of MRI techniques. However, clinical investigation is often concerned with questions regarding the movement of water through the brain. Diffusion MRI (dMRI) quantifies the mobility of water molecules at different scales; this allows for semi-quantitative analysis of cellular and extracellular structure and volumes as well as the differentiation of extravascular versus intravascular water fractions. Diffusion tensor imaging (DTI) is an extension of dMRI which enables quantification of diffusion anisotropy, providing indirect measures of white matter axonal fiber orientation and integrity.

Finally, MRI modalities can also offer insight into the delivery, and volume of, blood to the brain. Perfusion MRI (pMRI) provides estimation of tissue hemodynamic properties, such as perfusion and volume, as well as cap-
illary permeability following injection of paramagnetic contrast agents; this is called dynamic contrast enhanced (DCE) MRI. Alternative, non-invasive pMRI methods (ASL MRI) have also recently become available. The interested reader can find more detail regarding the various MRI modalities in e.g. [3].

We will now describe some of the different image types produced by modalities mentioned above. We limit the discussion to the file types used throughout the remainder of the text.

### 2.2.1 T1 and T2 weighted images

One useful type of image generated by a magnetic resonance imaging (MRI) device is a T1 weighted image. The signal intensity difference present in such an image is quite effective, for example, at differentiating structures such as fatty tissues from, say, bone or cerebrospinal fluid. A T1 weighted image also offers sufficient contrast difference between white matter, which appears brighter, and gray matter, which appears darker, in an MRI sequence.

The purpose of a T1 weighted image is to provide contrast between white and gray matter; a T1 weighted image does not effectively highlight the cerebrospinal fluid. In particular, it may be difficult with T1 weighted images alone, to highlight, with high accuracy, structures such as the third or fourth ventricles, cerebral aqueduct or subarachnoid spaces in the brain. In order to effectively visualize these regions we might enhance our data processing of T1 images with supplementary data from T2 weighted (T2) MRI data. The specific techniques producing a T1 versus T2 weighted image is a technical matter, beyond the scope of our discussion, and we refer the interested reader to [1]. A comparison of a T1 versus T2 image, illustrating the differences in signal intensity for different regions, is shown in Figure 2.4.

Medical images, including T1 and T2 images, are typically stored in the so-called DICOM file format; DICOM is an imaging standard and stands for 'digital imaging and communications in medicine'. The format stores both the image itself as well as a comprehensive set of meta data such as imaging protocol, patient identification etc. which enables consistent and safe usage across different vendors and software packages. A computational tool to view a T1 image is discussed in Section 2.3. We will use T1 images as part of an intermediate process, called segmentation, in order to produce a mesh of a desired brain region such as the left or right hemisphere. T2 images are optional for the segmentation process and will be used, for the first time, in
Fig. 2.4 T1 weighted image (left) versus T2 weighted image (right). In the T1 weighted image, white matter is recognized in the light gray color, whereas the darker gray color that surface the brain is gray matter. T1 weighted images is used in particular as it has a sharp contrast between gray and white matter. The T2 weighted images shows the CSF as almost white and provides good contrast between the CSF and the brain, but less contrast between white and gray matter.

Chapter 4 as a means for more accurately identifying, and removing, areas such as the ventricles from a computational mesh.

2.2.2 DTI images

Diffusive fluid transport in the brain is not isotropic. Along axons, for example, transport is less encumbered than through the tortuous interstitial spaces between cells. As a result, large groups of these axons can bias the directionality of through the brain; depending on their orientation [5]. For example, in the gray matter at the surface of the brain, where neurons are quite densely packed axonal topology can be more-or-less chaotic and the net effect is, on average, more isotropic. Conversely, in the white matter the distribution of axons primarily serves to connect disparate regions of neurons. The resulting axonal pathways are not approximately homogeneous in their directionality and distinctive ‘tracts’ of axons can bias fluid flow directionality, significantly, in these brain regions [5].
A diffusion tensor weighted (DTI) image offers insight into the directionality of flow pathways in the brain and can therefore be used to inform the construction of anisotropic diffusion tensors used in mathematical models. Like T1 and T2 data, DTI data is one of the many canonical outputs of the MRI procedure and this data is also often formatted as DICOM-type files. Figure 2.5 shows an example of a raw, axially-oriented DTI image as it appears in a viewing tool; this viewing tool is discussed in Section 2.3. The use of DTI images for constructing a diffusion tensor is first discussed in Chapter 5. The reader interested in the technical aspects underlying diffusion weighted tensor imaging is referred to [7].

2.2.3 Arterial Spin Labeling (ASL) images

Perfusion is a term which, in the study of brain mechanics, essentially means the quantity of blood delivered to a region of tissue. Knowing the rate of perfusion for a specific region of the brain has been a valuable diagnostic tool for clinicians when diagnosing brain pathologies; examples include chronic thrombosis, dementia and epilepsy [4]. Perfusion data is also of use when one is interested in questions of delivery of a quantity, such as oxygen, to brain tissue.
regions. For example, if our simple model problem (2.0.1) was representative
of the diffusion of oxygen through a brain-tissue region then perfusion data
would help us improve our model by informing the right-hand side quantity $f$.

Information about perfusion can be extracted from files generated in an MRI
process; the clinical term for such files is ‘arterial spin labeled’, or ASL, files.
Arterial spin labeling is a technical process whereby arterial water protons are
‘labeled’ by radio frequency pulses; combining these images produces a cerebral
blood flow map from which the quantities of interest can be deduced. There
are multiple types of radio frequency labeling utilized in ASL techniques in
addition to a few conditions necessary to produce a cerebral blood flow map;
the interested reader can see, for example, [4] for a more detailed discussion of
specifics. A sequence of such images would then be used to create the requisite
blood flow map to inform the right hand side of a model problem such as
(2.0.1).

2.3 Overview of external tools

In this section we will briefly overview several freely available open-source
tools that the reader will find useful in producing the computational meshes
and extracting parameters needed to solve the model problems in this text.
The basic idea is that one needs to view MRI files, extract surfaces from MRI
data, generate meshes from these surfaces and then solve the data-informed
mathematical problems of interest using these meshes. We will present the
tools needed to perform the above process in the indicated order.

2.4 Viewing and extracting MRI data

We will assume that your MRI data is available to you and is stored on your
computer in a directory, called MRI, which is located in your user home path.

2.4.1 Installing the DicomBrowser software

The output from an MRI scan is a collection of files arranged in sequences.
A given file within a sequence also contains the necessary information to in-
form a viewer as to what the next file in that sequence is. The DICOM files can be accessed by a binary file named **DICOMDIR**; this file indexes the entire structure of an MRI dataset. For the purposes of this text we suggest using a DICOM viewer which also contains additional functionality that will allow for the extraction of individual sequences within a comprehensive MRI data set. One such viewing and extraction tool, developed by the Neuroinformatics Research Group (NRG), is named DicomBrowser\(^1\); this tool is provided as source\(^2\) and in a binary format\(^3\) which can be directly downloaded and installed on several operating systems.

The first step is to download DicomBrowser and install it on your machine. If you are running a version of Linux you can do this by issuing your built-in package manager. For this text we assume you are using the ‘advanced packaging tool’, or ‘apt’, package manager and that you have the appropriate privileges to use the ‘sudo’ command. Other versions of Linux, such as Redhat or SuSe, may have different package managers installed; rpm and yum respectively for these examples. If you have a different package manager please substitute the appropriate command in the lines that follow.

Assuming the binary file you choose to download, from the DicomBrowser download site, is named **dicombrowser.deb** you can install it by executing the following command from a terminal window in the directory in which the file was saved

```
Terminal window
$ sudo apt-get install -y dicombrowser.deb
```

### 2.4.2 Viewing the contents of an MRI dataset

After the installation process completes, navigate to the directory **MRI** in a terminal window and open the DicomBrowser tool by typing the following command.

\(^1\) [http://nrg.wustl.edu/software/dicom-browser/](http://nrg.wustl.edu/software/dicom-browser/)
\(^2\) [https://bitbucket.org/nrg/dicombrowser/src/default/](https://bitbucket.org/nrg/dicombrowser/src/default/)
\(^3\) [https://wiki.xnat.org/xnat-tools/dicombrowser](https://wiki.xnat.org/xnat-tools/dicombrowser)
2.5 Postprocessing MRI images with FreeSurfer

In order to accomplish the overall goal of generating a meshes of different brain regions we will need to isolate different regions from each other in MRI image data. One computational method for identifying different areas of the brain from MRI images is called segmentation. Essentially, the segmentation process identifies and labels different brain regions by utilizes the intensity data.
embedded in the MRI images. The entire process that FreeSurfer utilizes is well documented; the interested reader can consult both online documentation\(^4\) as well a published review article [6]. In the remainder of this section we discuss the procedure of downloading and installing the FreeSurfer set of tools. A bit of a warning though, the download requires almost 4.6G and consists of pre-compiled versions of more than eight hundred scripts and programs. For a technical user it might be tempting to compile the system yourself, to get familiarized with the system, but from our own experience, this may be a challenge.

The process begins by obtaining the latest FreeSurfer build from the website\(^5\). For simplicity we will assume the name of this file is freesurfer.tar.gz. In a terminal window, navigate to the directory where this file has been downloaded and type the following command to unpack it.

```
Terminal window
$ tar -C /usr/local -xzvf freesurfer.tar.gz
```

The next step is to configure FreeSurfer by adding a few lines to the file named .bashrc in your home directory; note that this filename begins with a period. Open your ~/.bashrc file in a text editor; at the bottom of the file add the following lines.

```
Text editor
export FREESURFER_HOME=/usr/local/freesurfer
source $FREESURFER_HOME/SetUpFreeSurfer.sh
```

This will connect the terminal to the installed FreeSurfer files next time you open the terminal window. The FreeSurfer programming team requires that each use acquires a license key in order to use the software; the key is free, but to acquire it one must follow the directions on the registration web page\(^6\). You will receive your license text file via an automated email at the address you provide during registration. Save this license.txt file in the /usr/local/freesurfer directory where FreeSurfer has been installed. FreeSurfer comes packaged along with a visualization tool called freeview. In

\(^{4}\) https://surfer.nmr.mgh.harvard.edu/fswiki
\(^{5}\) https://surfer.nmr.mgh.harvard.edu/fswiki/DownloadAndInstall
\(^{6}\) https://surfer.nmr.mgh.harvard.edu/registration.html
some cases, freeview may need to be updated in order to function correctly alongside the current version of FreeSurfer. You can test freeview by issuing the following command from a terminal window.

```
Terminal window
$ freeview &
```

FreeSurfer comes with a rich set of online tutorials\(^7\). The installation will provide access to all FreeSurfer commands, which can be useful to know. In general, the FreeSurfer commands are well documented, and sufficient information can be accessed using the flag `-help`. There are also good documentation found online at the FreeSurfer wiki describing the FreeSurfer commands\(^8\). We will cover the used FreeSurfer commands in this book, and provide some additional information about the FreeSurfer commands in colored boxes.

### 2.6 Meshing tools for postprocessed MRI images

In order to create computational meshes from postprocessed MRI images we will use a suite of open source tools; these tools are described in this section. Most notably we will utilize the Surface-Volume-Meshing Toolkit (SVM-Tk), a python-module that developing at Simula Research Laboratory in Fornebu, Norway. The module is a minimalistic python-wrapper of The Computational Geometry Algorithms Library (CGAL) that is needed to construct a volume mesh from a collection of triangulated surface files. To install, and work with, the packages described in this section you will need at least python version three installed on your computer. You can check what python version you currently have installed by typing the following command in a terminal window:

```
Terminal window
$ python --version
```

In section 2.6.1 we describe the installation of the Surface-Volume-Meshing toolkit, section 2.6.2 covers miscellaneous utilities that are useful alongside the SVM-Tk.

---

\(^7\) https://surfer.nmr.mgh.harvard.edu/fswiki/Tutorials
\(^8\) https://surfer.nmr.mgh.harvard.edu/fswiki/FreeSurferCommands
2.6.1 Surface-Volume-Meshing Toolkit

The SVM-Tk provides a simple python-based interface to the computational geometry algorithms library (CGAL) and allows for complex meshing operations to be carried out simply. In this section we will walk through the process of installing the SVM-Tk, and its dependencies, on your machine. Before installing packages it is important to update your package list. This can be done by typing the following command in a terminal window.

```
Terminal window
$ sudo apt-get update
```

2.6.1.1 Installing SVM-Tk dependencies

The SVM-Tk depends on several packages. Suppose that a package named ‘pkg’ is needed by the SVM-Tk; as we have seen previously, we can install this package issuing the following command in a terminal window.

```
Terminal window
$ sudo apt-get install -y pkg
```

Repeat the above command for the following package names: bzip2, cmake, curl, g++, libboost-all-dev, libeigen3-dev, libgmp3-dev, libmpfr-dev, xz-utils, zlib1g-dev, cgal-dev, and libcgal-dev. The next step is to install the latest version of CGAL. At the time of writing this is CGAL version 4.13; the latest release version number can be checked online. We will use the ‘curl’ utility to download the latest source for CGAL. We can see from the CGAL website that version 4.13 of CGAL has source code available via a github url. For the simplicity of the discussions we assume this github url is given by https://github.com/CGAL/cgal/releases/tag/releases%2FCGAL-4.13. You should substitute the actual URL when you type the following command into a terminal window.

---

10. https://www.cgal.org/2018/10/01/cgal413/
2.6 Meshing tools for postprocessed MRI images

Terminal window

curl -sL https://github.com/CGAL/4.13/CGAL-4.13.tar.xz | tar -xJf

The above command downloads the latest CGAL source code to the current directory. Now we need to build CGAL; we will do this in the same terminal window. From this terminal window type the following commands; wait for the previous command to complete before entering the next.

Terminal window

$ cmake -DWITH_Eigen3:BOOL=ON .
$ make
$ sudo make install

2.6.1.2 Installing the SVM-Tk

We now have all of the prerequisites needed to install the SVM-Tk. In order to install the SVM-Tk we first retrieve the source code by typing the following command into a terminal window.

Terminal window

$ git clone –recursive https://github.com/SVMTK/SVMTK.git

This creates a new folder called SVMTK in the current directory. To finish the installation type the following commands into the terminal window.

Terminal window

$ cd SVMTK
$ sudo python3 setup.py install

After the install process completes you can test you installation of SVM-Tk by running any of the scripts located in the SVMTK/examples directory. For instance the boolean_operations.py script could be used to test your installation by typing the following into a terminal window from the SVMTK/examples directory.
If the SVM-Tk has been installed correctly you should see output from the script. Conversely, an error from the python interpreter would indicate that something has gone wrong.

2.6.2 Visualization, mesh conversion and other tools

Throughout the text we will need several other utilities; including file conversion utilities and visualization utilities. First it is suggested to install the ParaView data analysis and visualization application. This can be accomplished by downloading it, and following the installation instructions, from the website\(^\text{12}\) or by typing the following command into a terminal window.

```
Terminal window
$ sudo apt-get install paraview
```

An optional, but quite useful, tool for visualizing the surface stl files of Chapter 3 is Gmsh\(^\text{13}\). Similar to ParaView, above, this can be installed by the following terminal window command.

```
Terminal window
$ sudo apt-get install gmsh
```

Aside from visualization we will often have other needs. One of the more important auxiliary capabilities for use alongside the SVM-Tk is the conversion of one mesh filetype to another. Luckily, the meshio\(^\text{14}\) project provides a simple command-line tool that does exactly that. The meshio project is a python module that is part of the python package index (PyPI)\(^\text{15}\), which consist of over 190,000 projects. We can convert between various mesh formats representing

\(^{12}\) https://www.paraview.org/

\(^{13}\) http://gmsh.info/#Download

\(^{14}\) https://pypi.org/project/meshio/

\(^{15}\) https://pypi.org/
unstructured meshes, like the medit file format. This file format is the default mesh format in CGAL and in SVM-Tk and has the extension .mesh. To install the meshio suite of tools, it is suggested to use the pip installer. The following terminal commands will handle the installation of everything necessary.

```
Terminal window

$ sudo apt install python3-pip
$ sudo pip install meshio
```

We mentioned that meshio is a python module, which means that we can write python scripts to convert between different mesh formats, like

```
import meshio
msh = meshio.read("ernie.mesh")
meshio.write("ernie.xml",

    meshio.Mesh(points=msh.points,

    cells={"tetra": msh.cells["tetra"]})
)
```

In this python script, we load the meshio-module and read the selected mesh that we named ernie.mesh. Then we use the write function to make a file called ernie.xml, which is one of mesh formats we will use in this book. The writing function requires that we specify what kind of information we want to write, and in this case we specified to write the mesh points and the tetrahedron cells connections. We mentioned that meshio can also be used with the command-line, and the script is equivalent to the command-line

```
Terminal window

$ meshio-convert ernie.mesh ernie.xml
```

The inclusion of MRI data, such as DTI and ASL, require that we can read the data into our models. This can be done with the python module Nibabel [8], which provides read and write access to several file format in neuroimaging. The module is part of the nipy, a community for Python users in the analysis of neuroimaging data, with several interesting projects. We can install Nibabel with the use pip and the following terminal command

```
$ pip install nibabel
```

---

16 https://nipy.org/nibabel
17 https://nipy.org/
2.7 Finite element numerical methods with FEniCS

Finite element software is a critical component for solving the model problems discussed alongside the meshing and data extraction pipelines of this text. The FEniCS\textsuperscript{18} numerical software package is an open source library that is now in use, and active development, by thousands of researchers and scientists across the world.

The FEniCS library can be used with both the C++ and Python languages; the former for large-scale problems where performance is key and the latter for smaller scale problems where teaching, demonstrating or prototyping and testing numerical methods is the focus. There are many ways to install\textsuperscript{19} FEniCS; including the use of Docker images, using pre-built Anaconda packages, or from source. The simplest way to install FEniCS is to issue the following commands from a terminal window.

Terminal window

$ sudo apt-get install software-properties-common
$ sudo add-apt-repository ppa:fenics-packages/fenics
$ sudo apt-get update
$ sudo apt-get install --no-install-recommends fenics

In this text we will use FEniCS alongside the Python interface for simplicity. Much in the way of documentation on using FEniCS is available\textsuperscript{20} freely online and it is assumed that the reader has enough working knowledge of Python and FEniCS to follow along with the included code.

\textsuperscript{18} https://fenicsproject.org/
\textsuperscript{19} https://fenicsproject.org/download/
References

In this chapter the primary goal is to demonstrate the simplest case of computational human brain modeling. We will use basic features of the tools overviewed in Chapter 2.3 to create a simple left-hemisphere human brain mesh from MRI data. We will use this mesh to solve a simple problem; namely to find $u = u(t, x)$ so that

$$\begin{align*}
\partial_t u - \text{div} \left( K \nabla u \right) &= f, \\
u &= u_d(t, x) \quad \text{on} \quad (0, T] \times \partial \Omega, \\
u(0, x) &= u_0(x) \quad \text{on} \quad \Omega.
\end{align*}$$

The problem (3.0.1) is a basic initial value problem motivated by clinical application. Namely it is a basic starting point, for example, when considering models for tracking the diffusion of a substance, such as a tracer or amyloid-$\beta$ protein concentration, through the brain. The function $u_d(t, x)$ represents the value of the substance concentration on the boundary, $x \in \partial \Omega$, for times $0 < t < T$ where $T$ is a fixed end-point, in time, of interest; for example, the end time of the tracer experiment. The function $u_0(x)$ represents the initial tracer distribution in the entire left hemisphere and $f = f(t, x)$ represents a source term for additional tracer concentration injection. In the present chapter, $\Omega$ will represent the left hemisphere of the brain and $\partial \Omega$ is the corresponding boundary.

Remark 3.1 We are essentially asking the question ‘what is the solution, $u = u(t, x)$, to the equation above when $u_d, u_0, K$ and $f$ are particular given quantities?’ The values of the known quantities $u_d, u_0$ and $f$ are derived from the experiment being carried out while $K$ comes from, in practice, the physical properties of the material through which the tracer is diffusing.
In this particular chapter we will assume that $K$ is a constant value which, in practice, it certainly is not. However, our purpose here is to produce a simple computational code to serve as a foundation for further increasing complexity using patient MRI data.

To solve (3.0.1) we will generate a computational mesh of the left hemisphere from MRI data in Section 3.1 and discuss solving (3.0.1) numerically in Section 3.2. The reader will be referred to external texts for particulars beyond the scope of the current work; this includes mathematical questions surrounding the finite element method in addition to more usage of the FEniCS software platform that falls outside the scope of our current concerns.

3.1 A basic volume mesh generation pipeline

In this section we have one primary goal: to create a mesh from a T1 weighted MRI dataset. First we must extract a T1 image sequence from an MRI dataset; we covered how to open an MRI dataset in Chapter 2.4.2. Then, we will use this extracted T1 image sequence to create a surface using FreeSurfer. Finally, we will transform this surface into a mesh using command line tools included in FreeSurfer along with the Surface-Volume-Meshing Toolkit (SVM-Tk) library. The SVM-Tk library wraps functionality from the computational geometry algorithms library (CGAL) with a script based python interface, which enables automation; the SVM-Tk has been developed specifically with the aim of providing a semi-automatic high throughput pipeline for setting up patient-specific simulations in larger patient-cohorts.

3.1.1 Extracting sample data from an MRI dataset

We will now describe the process of extracting data from a DICOM database. We describe the extraction of a sequence of T1 images; however, the general process also applies to other types of image data. There are two primary ways of approaching the extraction procedure; via the DicomBrowser graphical interface or by using command line tools. We discuss both options below; beginning with the graphical user interface.
3.1 A basic volume mesh generation pipeline

3.1.1.1 Extracting data using the DicomBrowser graphical interface

The first step is to launch the DicomBrowser viewer tool and open the MRI data sequence main DICOM file as described in Chapter 2.4.2. After opening the main MRI DICOM database file, select a patient ID and then select an MR series from the available list of MR series (c.f. 2.4.2).

Each of the MR Series contains images which adhere to a specific MR protocol. These can include the T1 protocol, the T2 protocol, etc. Click on an MR series of your choice. In the second column of the right-hand side pane there are several identifiers listed that are associated to the selected MR series. For now we scroll down until we find the Protocol Name identifier in this column of entries. Once you find this identifier you will notice that in the far right-hand column of the right-side pane the values corresponding to each tag, in the Name column, are listed. The values corresponding to the Protocol Name identifier can vary; for example we might see any of the following: T1_3D, WIP T1_3D, T1-cor-REK, WIP MO 3.6mm, WIP MIP-s3DI_MC, etc.

Find an MR series whose value indicates that it is a T1 image; this will be reflected by values such as T1_3D or WIP T1_3D etc for the Protocol Name associated with that series. Once you have found a candidate series you should view the files to ensure that this series contains T1 images. This can be done by clicking the desired MR series and selecting ‘View→View Selected Images’ from the top menu bar. In order to provide enough image information for later steps we should select a T1 MR series that has adequate content; specifically, one that has numerous images. If you expand a selected MR series, by clicking on the arrow to the left of the series name, you will see the contents of the series. Series with a low number of images (such as ten or twenty) can give less accurate results than series with more images; try to find a T1 series with as many images as possible.

Once a suitable T1 MR series has been identified the extraction procedure is straightforward. First, in the left pane of the main window, click the name of the MR series to be extracted (e.g. MR Series 102, etc). Second, with the MR series name highlighted, choose ‘File→Save’ from the main menu bar. A dialog box entitled ‘Save DICOM files’ will open and options on where, and how, to save the files will be presented. Once the desired write options have been determined, ensure that ‘Write only selected files’ is selected and click

---

1 We remark that the names are chosen by the MR operator and so in principle, it may not be any relation between the name of the protocol and the actual protocol itself that is defined by a number of different parameters.
‘Save’. Following this, we see that the selected images in the MR sequence have been exported from the DICOM database and now reside on disk in a separate directory. The remainder of this chapter assumes that the images have been exported to the directory MRI-Series-T13D; the name of the directory is of no consequence but a fixed name is needed for the discussion.

3.1.1.2 Extracting data using command line tools

The primary advantage of using command line tools in order to extract data from a DICOM database is that the process can be automated via scripting. This is useful, for example, when processing large datasets consisting of a large number of DICOM databases. In this section we illustrate the extraction process on a single database. The extraction process consists of two steps: probing the database for a given tag name and then extracting all data with the given tag from the database.

The probing process is done by utilizing the FreeSurfer command `mri_probedicom`, and we will probe for the tag **Protocol Name**. This will be done by adding the corresponding numeric tag 18 1030 to the command as displayed here

```
Terminal window

$ mri_probedicom -i IM_0001 -t 18 1030
WIP T1_3D
```

**FreeSurfer details**

We use the FreeSurfer command `mri_probedicom` to specifically examine the DICOM meta data. The command can also be used to compare the meta data of different DICOM files with the flag `--compare` followed by two DICOM files. We can also view the selected DICOM file by using the flag `--view`. The complete description of possible options can be view by using the `--help` flag.

We can apply this probing to each file in the DICOM directory, and copying the DICOM files with a specific tag to a new directory. This can be done with the following bash script called `mri_sort.sh`. The script takes three types of input; the DICOM folder, the output directory and an identifier for the protocol name.
#!/bin/bash
# 1st argument $1 = input DICOM folder
# 2nd argument $2 = the output copy directory
# 3rd argument $3 = the identifier
# Find all files in the directory and subdirectories
files=$(find $1 -type f )
for j in ${files}; do
    # probe Protocol name
    name=$(mri_probedicom --i ${j} --t 18 1030)
    # Checks if tag is part of the protocol name.
    if [ "${name}/${3}" != "$name" ];
        # Removes spaces in protocol name
        mkdir -p ${2}/${name/[[:blank:]]}/
        cp ${j} ${2}/${name/[[:blank:]]}/
    fi
done

This script uses the bash command find and the flag -type f to find all files in the input directory and its subdirectories. The script will go through all the files and probe each file for the protocol name, and check if the protocol name contains the identifier argument. Then every file with the identifier in the protocol name will be copied to a folder named after the protocol name in the output directory. This process will also involve the command name/[[:blank:]]/ that will remove all spaces from the protocol name. The removal of spaces is preferred to avoid errors when using FreeSurfer. Thus we can use the script to extract all the scans with T1 string in the Protocol Name;

Terminal window

$ bash mri_sort.sh ./DICOM ./ "T1"

3.1.2 Reconstructing surface information

In this section we will discuss using the FreeSurfer collection of tools to extract a surface of the left-hemisphere of the brain contained in our T1 weighted image sequence. The extracted left-hemisphere surface will then be used in the next section to generate a mesh. Recall that in section 2.5 we discussed
how to download and install FreeSurfer; if this has not yet been done it will be necessary to do so before proceeding.

We assume that the extracted T1 image sequence MRI-Series-T13D has been saved in the MRI directory. In a terminal window, navigate to the MRI-Series-T13D directory and inspect the contents; it should contain numerous images. These images contain header information which links them to one another and, as a result, it will not matter which one we select to start the reconstruction; the reconstruction program has sufficient intelligence to find the files as it needs them.

We will assume that we have selected a file named IM_0001 from the MRI-Series-T13D directory. We now need to decide on a subject ID which will be visible to the FreeSurfer pipeline. For the purposes of this text we will assume that our desired subject ID is ernie although we can select anything we like in practice. We then issue the following command from the terminal window from within the MRI-Series-T13D directory

```
$ recon-all -subjid ernie -i ~/MRI/MRI-Series-T13D/IM_0001 -all
```

The above command conducts the full image segmentation pipeline and surface reconstruction; this is indicated with the ‘-all’ flag at the end of the command. Furthermore, the reconstruction is associated with the subject whose subject-id is ernie and that the path to the first image in the MR sequence to be processed is ~/MRI/MRI-Series-T13D/IM_0001. The recon-all command can take several hours, possibly up to about a day but typically around 5-7 hours depending on your machine, to complete. In the above ~ refers to the home directory or $HOME.

**FreeSurfer details**

The command recon-all is the primary command for FreeSurfer, since it will start the segmentation process. We have already describe the necessary flags for this command, but we will continue the description with some additional options. This description will be on an introductory level. However, the interested reader can use the flag -help, which will detailed information about the entire process and provide references to related articles and texts.
3.1 A basic volume mesh generation pipeline

The command is a sequential process that consists of 34 different stages, which is divided into three different steps\(^2\). Data will be produced throughout the process, and are often required as input for the next stage. We can initiate each step separately by using the following flags instead of `-all`.

- **-autorecon-1**: starts step-1 process that includes stages 1-5, which includes normalization and skull stripping.
- **-autorecon-2**: starts step-2 process that includes stages 6-23, which includes the segmentation and surface generation.
- **-autorecon-2-pial**: starts construction of the surfaces that includes stages 16-23.
- **-autorecon-3**: starts step-3 process that includes stages 24-34, which includes statistical data generation and final parcellation.
- **-autorecon-all**: equivalent to `-all`

These flags can be useful for restarting the command. For instance, if a failure occurred at stage 36, then we can start over from stage 24 and not from the beginning by using the flag `-autorecon-3`. This can also be useful, when it is necessary to re-run the segmentation process after correcting an error. In FreeSurfer there exists two types of errors, known as hard and soft errors. Hard errors will terminate the segmentation process, while soft errors are errors that we find in the produced data. The soft errors are mostly segmentation errors, such as inclusion of the dura in the segmentation and wrong segmentation of white matter. We can edit the segmentation to correct the error, and run the recon-all with the flag `-autorecon-2-pial`. This will create new surfaces based on the corrected segmentation files. The correction of soft errors will not be further covered in this book, but there exists detailed manuals on how to deal with soft errors at the FreeSurfer website.

We continue with the flag `-sd` that can be used to specify the subject directory for the recon-all command. This can be quite useful in order to separate the segmentation data for different cohorts. The segmentation of CSF filled structures, such as the ventricular system, often requires the usage T2 MRI to be good. We can include T2 MRI with the flag `--T2`, and we can also use the flag `-T2-pial` to use the T2 MRI in the construction of the pial surfaces.

The segmentation in FreeSurfer is based on the segmentation atlas of health subjects, therefore the segmentation can often encounter hard errors for patients with abnormal brain anatomy. We can often allow the segmentation

\(^2\) [https://surfer.nmr.mgh.harvard.edu/fswiki/ReconAllTableStableV6.0](https://surfer.nmr.mgh.harvard.edu/fswiki/ReconAllTableStableV6.0)
to finish if we add the flag `-notalairach`, which causes recon-all to skip assertion points in the first step.

We can also include multiple T1-weighted images to the command process, this is done by reusing the input flag `-i`. This is mostly done for motion correction in MRI with motion artifacts.

### 3.1.3 Viewing and extracting a brain surface

We assume that FreeSurfer has been installed in the directory `~/FreeSurfer` and that the recon-all command of the previous section has now finished successfully. The data that were produced by the recon-all command, above, resides in the directory `~/FreeSurfer/subjects/ernie`. If we inspect the contents of this directory we will see several subdirectories; some important subdirectories are:

- `/stats`: contains files providing statistics derived during segmentation.
- `/mri`: contains volume files generated during segmentation
- `/surf`: contains surface files generated during segmentation

For the purpose of this chapter we are concerned with the contents of the `/surf` subdirectory. We want to view the left pial surface which bounds the white and gray matter of the left hemisphere. To do this we will use the `freeview` tool packaged along with the FreeSurfer collection of tools. From the `~/FreeSurfer/subjects/ernie/surf` directory, execute the following command below, in a terminal window, to open freeview.

```
Terminal window
$ freeview &
```

The above command will open the Freeview viewer in the current directory. From the command bar select ‘File→Load Surface’. Select the file titled `lh.pial` (lh means left hemisphere and pial denotes the pial surface, i.e, the surface between the cortical gray matter and the CSF) in the dialog box menu and click ‘Ok’. After a moment the view windows will be populated with 2D surface slices (shown as 1 dimensional curves) in addition to a reconstructed 3D image of the pial surface. An example showing the Freeview interface with left pial surface loaded is shown in Figure 3.1. Now that we have viewed the left pial surface we need to extract a file that we can use to create a mesh of it. We will do this using the freesurfer `mris_convert` command in the subject
directory of freesurfer corresponding to our test subject identifier ernie. From within the current /surf directory issue the command below in a terminal window.

```
Terminal window
$ mris_convert ./lh.pial ernie-pial.stl
```

The above will create a file called lh.ernie-pial.stl in the current directory. The file extension stl is a standard one. As a result, if we would like to visualize the contents of the file, we can open it in several programs such as Paraview or Gmsh.

**FreeSurfer details**

The command `mris_convert` is used handle the conversion between different surface formats, and has the following structure.

```
Terminal window
$ mris_convert [options] input_file output_file
```

The full list of available options can be seen by using the --help flag, and we will cover some basic options. The flag options -c `<scalarfile>` can be used to include scalar data, such as lh.thickness, in the surface conversion.
We can scale the output xyz coordinates with option flag -s, and compute the enclosed volume of the surface with the option flag --volume.

### 3.1.4 Creating a volume mesh from an extracted surface

In the last section we extracted a surface file, `lh.ernie-pial.stl`, of the left pial surface from our MRI data. In this section we use the SVM-Tk library to convert this file into a volume mesh that can be used with the FEniCS finite element software. This process can be involved because while FreeSurfer is made for generating surfaces, it is not made for generating surfaces suitable meshing of 3D structures. Hence, there is a number of problems that be present in the FreeSurfer surfaces. Some of these are:

- the surfaces may have sharp edges unlike biological tissues,
- the surfaces may have triangles with very large aspect ratio,
- the surfaces may have topological defects such as holes,
- the surfaces may have self-intersection or overlap other surfaces.

These defects can cause mesh generation to fail or produce meshes not suitable for computations. Thus some preprocessing steps may be required, and it can also greatly improve the quality of the mesh for use in numerical simulations. In Chapters 3.1.5.1 to 3.1.5.3 we will outline a three-step process: remeshing, smoothing, and volume mesh generation. If no remeshing or smoothing preprocessing of the surface stl file is preferred, the reader can proceed directly to Chapter 3.1.5.3.

### 3.1.5 Preventing surface self-intersections and surface overlap

Creating a good mesh for FEniCS requires that each surface and the combination of surfaces to be suitable. In FreeSurfer, the hemisphere surfaces are generated separately therefore combining surfaces from different hemisphere can create problems. The problems include:

- the hemisphere surfaces overlap, creating bridges in the cortical gray matter.
- the hemisphere surfaces do not overlap, due to smoothing, creating gaps in the commissural nerve tracts connecting the white matter.
3.1 A basic volume mesh generation pipeline

We can consider this a combined problem, since we want to join the hemisphere surface in the nerve tracts, but avoid overlapping surfaces in the cortical gray matter. The module function `separate_surfaces` in SVM-Tk will iteratively separate two surfaces until the surface is farther away than the shortest edge of a vertex. We can also add a third surface to the function, like the union of the white surfaces, so that the function will not move the vertices that are inside this third surface.

The foldings of pial surfaces can produces narrow gaps, and these gaps can often be less than the edge size of the mesh, which may result in bridges instead of foldings in the mesh, see Figure 3.2. In SVM-Tk, the `Surface` class is used to load, repair and deform triangulated surface so that it can be used in creating volume meshes. For instance, the function `separate_gaps` will iteratively separate the close junctures until the edge distance between vertices are less than the distance between unconnected vertices.

Surfaces constructed by using FreeSurfer can have topological defects, like holes, these defects can be repaired by following the guide\(^3\). We can also encounter surfaces that have missing facets, i.e. structural defects, which can be corrected with member function `fill_holes`.

Preprocessing can also lead to anatomical errors, such as white surface outside the pial surface, which may be caused by oversmoothing the pial surface. Therefore, it is recommended that the surface are inspected before meshing.

3.1.5.1 Remeshing a surface file

A primary advantage of remeshing is increasing the mesh quality; this means, for example, reducing the frequency of mesh simplices which may be overly distorted and reducing the density of vertices that have a large number of connected edges with respect to the global mesh connectivity. To increase the quality of the final volume mesh we remesh our surface file, `lh-ernie-pial.stl`, using the SVM-Tk library first discussed in Chapter 2.6.1.

```python
import SVMTK as svm

input_file = 'lh-ernie-pial.stl'
output_file = 'ernie-pial-remesh.stl'
```

\(^3\) https://surfer.nmr.mgh.harvard.edu/fswiki/FsTutorial/TopologicalDefect_freeview
Fig. 3.2 Illustration of the close junctures in a coronal slice of the pial surface created by FreeSurfer.

```bash
# --- Options --- #
number_of_iterations = 3
do_not_move_boundary_edges = False
dge_length_mm = 1.0

#load input file
```
surf = svm.Surface(input_file)

#remesh with SVM-Tk
surf.isotropic_remeshing(edge_length_mm,\    
    number_of_iterations,\    
    do_not_move_boundary_edges)

#save output file
surf.save(output_file)

You can specify more iterations, and produce a finer mesh, of the remeshing algorithm by increasing the integer value of `number_of_iterations`. Likewise the floating point value of `edge_length_mm` tells SVM-Tk what the maximum edge length of a mesh simplex should be. The boolean `allow_moving_boundary_edges` value determines if SVM-Tk is allowed to perturb the boundary facet vertices during the remeshing procedure. Setting this value to `True` requires that the original boundary facet vertices not be moved during the remeshing procedure. However, it is advised to set this value to `False` since requiring the boundary facets to be fixed in space, during remeshing, can generally cause the process to fail. We can then remesh our surface by executing this script from

![Fig. 3.3 Original (left) pial surface and (right) after remeshing with SVM-Tk](image)
the command line, in the same location as `lh-ernie-pial.stl`, by executing the script in a terminal window as follows

```
Terminal window
$ python remesh_surface.py
```

The above will produce the file `ernie-pial-remesh.stl`, in the current directory, which will be used in the next section. The process for executing all of the python scripts in this chapter is similar and will therefore not be repeated. Figure 3.3 shows the result of running the above script with three remeshing iterations on the raw input file `lh-ernie-pial.stl`; the figure has been zoomed in to draw attention to local feature differences.

### 3.1.5.2 Smoothing a surface file

In this section we smooth our remeshed file `ernie-pial-remesh.stl`; the process we describe is applicable for any triangulated surface file with extension stl or off.

```python
import SVMTK as svm

input_file = 'ernie-pial-remesh.stl'
output_file = 'ernie-remesh-smooth.stl'

# --- Options ---
number_of_iterations = 10
preserve_volume = True
smoothing_factor = 1.0

# load input file
surf = svm.Surface(input_file)

if not preserve_volume :
    # Perform Laplacian smoothing
    surf.smooth_laplacian(scalar_factor,\
                         number_of_iterations)
else :
    # Perform Taubin smoothing
    surf.smooth_taubin(number_of_iterations)
```

104
Once more we see that the code above has several options. First, the integer value `number_of_iterations` determines the number of times the smoothing process should occur. Higher values will produce a smoother mesh; however, too high of a value may result in a loss of resolution in brain features such as the sulci and gyri (grooves and bumps) on the brain surface. The boolean variable `preserve_volume` determines whether a volume-preserving Taubin smoothing (True) or not-necessarily volume preserving Laplacian smoothing (False) process should be used. From a conceptual point of view, a Taubin smoothing is essentially a local smoothing iteration followed by a local ‘swelling’ operation which preserves the volume of the original non-smoothed patch whereas the Laplacian smoothing consists only of local smoothing operations and the volume of the original patch may not be preserved [16]. In general, if the number of global smoothing iterations is not overly excessive, then it is suggested to use Taubin smoothing.

Finally, the `smoothing_factor` is a floating point value, in the interval [0, 1], which essentially determines the strength of the smoothing operation for each smoothing iteration. For example, in a single iteration, setting this value to
1.0 indicates that the smoothing process should be performed fully whereas a value of 0.0 will not smooth the input mesh at all. Running the above script in the same directory as ernie-pial-remesh.stl will produce a smoothed surface stl named ernie-remesh-smooth.stl. It is advised to check the file visually by opening it directly, using either Paraview or Gmsh, to determine if more or less smoothing is needed; some surface stl files may require more, or less, smoothing iterations than others. Figure 3.4 shows the result of using ten iterations of the Taubin smoothing our remeshed example left pial surface stl file ernie-pial-remesh.stl.

3.1.5.3 Creating a volume mesh from a surface file

In this section we create a volume mesh from our remeshed and smoothed surface file; we then convert this mesh into a final mesh file format that can be used with the FEniCS numerical software.

```python
import SVMTK as svm

input_file = 'ernie-remesh-smooth.stl'
output_file = 'ernie-remesh-smooth.mesh'

# --- Options ---
mesh_resolution = 32.0

# Load input file
surf = svm.Surface(input_file)

# Create the volume mesh
domain = svm.Domain(surf)
domain.create_mesh(mesh_resolution)

# Write output file
domain.save(output_file)
```

Beyond the input and output filenames there is only one option for this script; namely the floating point value of the ratio `mesh_resolution`. In general this floating point value should be of the form $2^n$ for some integer $n$ and determines the maximum size of a tetrahedra in the volume mesh relative to the overall bounding box length for the input surface that the volume mesh is
derived from. The higher the value of \texttt{mesh\_resolution} the smaller the area of each tetrahedra in the resulting volume mesh will be. Roughly speaking, increasing the value of \texttt{mesh\_resolution} from $2^n$ to $2^{n+1}$ will produce individual tetrahedra with approximately half the volume of those produced for the $2^n$ case. Figure 3.5 shows our final volume mesh produced with \texttt{mesh\_resolution} = \texttt{16.0} (on left) and (on right) \texttt{mesh\_resolution} = \texttt{64.0} was used.

The output filename is required to have the extension \texttt{.mesh}, which is the standard format for SVM-Tk. This mesh format is not compatible with FEniCS, therefore we need to convert the mesh to a different file format. For this purpose, we will make use of the python module \texttt{meshio}, which we introduced in Section 2.6.2. We will open the terminal in the directory of the constructed mesh \texttt{ernie.mesh}, and type:

```
Terminal window
$ meshio-convert ernie.mesh ernie.xml
```

This command will convert the mesh to the file \texttt{ernie.xml}, which is one of the file format that is used in FEniCS.
3.2 A simple mathematical model of diffusion in the brain

Now that we have constructed our mesh of the left pial surface this section details the solution of the model problem (3.0.1). Our first model is quite simple; both practically and mathematically. From a practical point of view we are using a mesh, at this point in the text that does not resolve different brain substructures; such as the gray and white matter. From a mathematical point of view, problem (3.0.1) is one of the most well studied problems in science and mathematics. Despite the simplicity of (3.0.1) it is still related to questions of clinical interest. For example, the so-called ‘amyloid-β hypothesis’ [13] suggests a connection between the progression of Alzheimer’s disease and the collection of the toxic protein amyloid-β, also phosphorylated tau, in different brain regions.

To track the movement of such proteins in the brain researchers have used equations similar, in presentation, to (3.0.1). One example is the recent use of the Fisher-Kolmogorov-Petrovsky-Piskunov, or Fisher-KPP, equation [11] coupled to a morphoelastic model of the brain deformation to simulate the affect of protein diffusion and atrophy in Alzheimer’s progression; the Fisher-KPP model is given by

$$\partial_t u - \nabla \cdot (K \nabla u) - \alpha u(1 - u) = 0,$$

where $K$ is, in general, a transversely anisotropic diffusion tensor and $\alpha > 0$ is a growth rate. Now, equation (3.0.1) does not satisfy the $\alpha > 0$ requirement; even in the case of $K$ constant and isotropic. However, the point here is that (3.0.1) is a reasonable ‘first starting point’ based on the current state of the art approaches employed in contemporary modeling practice. Another interesting question, of current clinical debate, is that of how waste proteins generated in the brain, such as amyloid-β, make their way out into the cerebrospinal fluid (CSF) where they may be cleared; either by traditional lymphatic vessels or other mechanisms. In fact the recent ‘glymphatic hypothesis’ [9, 12] suggests a method of CSF circulation in the brain that may act as a kind of ‘interior lymphatic system’; cleaning waste molecules, such as amyloid-β, out of the brain parenchyma.

A recent clinical experiment [14] has aimed to test the communication of the subarachnoid CSF compartment and the extravascular component of human brain tissue; such communication is postulated by the glymphatic hypothesis. The study was conducted using MRI imaging alongside the injection of a tracer
3.2 A simple mathematical model of diffusion in the brain

into the subarachnoid CSF. Following injection, the clinicians then observe how the tracer makes its way through the brain and examine the time-to-clearance of this tracer for both reference patients and a cohort with dementia.

In chapter 3.2.1 we briefly recall a typical numerical strategy for solving the model problem (3.0.1). In chapter 3.2.2 we will adapt a simplified view of the previously mentioned clinical study [14] to our model and discuss the implementation, and solution, of the numerical approximation using FEniCS.

3.2.1 Overview of the numerical formulation

The mathematical model of (3.0.1) is well studied in many mathematics and engineering texts; from heat transfer, to partial differential equations and numerical analysis. The reason for this is that it is representative of physical processes governed by diffusion and is also one of the simplest parabolic problems from the point of view of analysis. Thus, studying this simple problem provides an approachable starting point, for several fields, for a discussion on theoretical and modeling techniques that are useful tools for more challenging equations.

The way in which equation (3.0.1) is written is called a ‘strong formulation’. Often, strong formulations can pose challenges when attempting to solve them either analytically or numerically. To get around this issue one often transforms the strong formulation, of (3.0.1), into a ‘weak form’. From the weak form the idea behind the finite element method is to then discretize in space by using a specific choice of finite dimensional set of ‘function vectors’, called test functions, that are constructed based on the underlying computational mesh of the domain $\Omega$.

Once we have selected our test functions we seek an approximate solution. That is, instead of looking for the exact solution to (3.0.1), the finite element method seeks an approximation to it which is a finite sum of time-dependent coefficients multiplied by functions in this finite dimensional subspace. The result of this viewpoint is a matrix system of ordinary differential equations which can then be discretized in time using one of the numerous methods from the theory of numerical ordinary differential equations. The finite element method has evolved substantially over the last few decades and the interested reader can find several outstanding texts on the subject [3, 4, 5, 6].
A nice account of formulating a fully discretized version of the model problem (3.0.1) can be found in [7, Chp. 3.1] which is freely available online; thus, we will not endeavor a full discussion here. We will, however, include the formulation for completeness. Let $V_h$ denote a choice of finite element space, which in practice will consist of continuous functions whose restriction to each mesh element is a polynomial of fixed degree, and let $0 = t_0, t_1, t_2, \ldots, t_N = T$ denote a uniform partition of the time interval $[0, T]$ for which we want to simulate our model clinical experiment. Then the fully discrete version, i.e. discretized in both space and time, of (3.0.1) that we consider here is to find $u_{h}^0, u_{h}^1, \ldots, u_{h}^N$ in $V_h$ such that for every $v_h \in V_h$ we have:

$$\int_{\Omega} \frac{u_{h}^{n+1} - u_{h}^{n}}{\Delta t} v_{h} \, dx + K \int_{\Omega} \nabla u_{h}^{n+1} \cdot \nabla v_{h} \, dx = \int_{\Omega} f^{n+1} v_{h} \, dx. \quad (3.2.1)$$

$$\int_{\Omega} u_{h}^0 v_{h} \, dx = \int_{\Omega} u_0 v_{h} \, dx, \quad (3.2.2)$$

where (3.2.2) comes from the initial condition $u(0, x) = u_0(x)$, where $u_0(x)$ is a known function, in (3.0.1). The discrete matrix system of equations (3.2.1) corresponds to discretization in space by the finite elements functions in $V_h$ and discretization in time by the implicit Euler method.

### 3.2.2 A clinically motivated simulation using FEniCS

In this section we aim to accomplish three things; (i) adapt a simplified view of a recent clinical experiment to the mathematical model (3.0.1); (ii) implement the finite element discretization scheme (3.2.1)-(3.2.2) using FEniCS; and (iii) investigate the results.

#### 3.2.2.1 Overview and adaptation of a clinical experiment

A recent clinical experiment [14] has used MRI imaging to assess the spreading of a tracer, introduced into the subarachnoid CSF, through the brain. We will go through the process, here, of using the clinical methodology discussed in [14] to determine the different parts, e.g. $K, f, u_d(x,t)$ etc, of our simple

---

model problem (3.0.1). The reader interested in jumping directly into the implementation of the code to simulate the problem can skip forward to 3.2.2.2.

First, we need to decide on what question we would like to investigate. Our model equation is not complex enough to address the full extent of the clinical trial; so we will settle on a simpler question. Namely we could ask what our simplified model, using the left-hemisphere domain and (3.0.1), would predict regarding the patterns of tracer absorption when compared to those of the reference patients during, say, the first nine hours in the study [14, Fig. 1, Fig. 2].

In the study, the MRI contrast agent gadobutrol (600Da) was injected into the subarachnoid CSF at the base of the spine. Since the tracer is injected into the subarachnoid CSF at the start of the experiment this immediately tells us about one aspect of the model (3.0.1); we know that our initial tracer concentration in the brain is

\[ u_0(x) = u(x, 0) = 0, \quad x \in \Omega \setminus \partial \Omega. \]

Moreover, the diffusion coefficient was approximated to the value \( K \approx 12 \times 10^{-7} \text{ cm}^2/\text{s} \). The approximation used the measured diffusion coefficient \( 3.6 \times 10^{-7} \text{ cm}^2/\text{s} \) for the tracer Gd-DPTA (550Da) [1], and the Stokes-Einstein relation, which is used to estimate diffusion coefficient between similar sized molecules.

Since the physical unit of the mesh generated by the SVM-Tk is in mm and the experiment was conducted over several hours we rescale the diffusivity to these units and get \( K = 4.32 \times 10^{-1} \text{ mm}^2/\text{h} \). We note that a more comprehensive modeling approach for simulating this particular experiment would include the subarachnoid CSF compartment surrounding the brain in the computational mesh; we could then solve a flow problem there to determine the tracer distribution. This regional concentration information could then be coupled with the diffusion model for the brain. If we were concerned with modeling this specific experiment more accurately we would, of course, also use a mesh of the entire brain and not just the left hemisphere.

However, our current approach takes the opposite point of view; instead of building a mathematical model suited to the clinical experiment we are trying to fit the clinical experiment to our model. Thus, we instead related the concentration in the subarachnoid CSF compartment to our time-dependent Dirichlet boundary condition in (3.0.1) for the left-hemisphere domain. We can accomplish this by making some simplifying assumptions. First, we assume the tracer starts off as ‘uniformly well mixed’ within the subarachnoid CSF. Second, since we are interested in simulating the first 9 hours for comparison,
we assume that the system is closed; that is, we assume that the total tracer mass is contained in either the subarachnoid CSF or inside the brain tissue domain and that no tracer escapes, via other clearance mechanisms, during this time.

In the clinical experiment 0.5 ml of the 1.0 mmol per ml gadobutrol contrast agent was injected; gadobutrol has a molar mass of approximately 604.71 g per mol. Thus we have a tracer mass of 302.355 mg of injected gadobutrol contrast into the entire subarachnoid CSF. If we estimate the total subarachnoid space CSF volume at 125 ml [15] then the total subarachnoid volume, following tracer solution injection, is 125.5 ml in the full subarachnoid space. Since we are only modeling the left hemisphere, and the tracer is assumed to be uniformly well mixed, we then have approximately 151.178 mg of gadobutrol suspended in 62.75 ml of left hemisphere subarachnoid CSF volume. Since the length scale of our brain mesh is in mm we convert the left subarachnoid CSF volume to mm$^3$ giving $6.275 \times 10^4$ mm$^3$.

Thus, in the left hemisphere, we have a uniform initial tracer concentration of 151.178 mg / $6.275 \times 10^4$ mm$^3$ ≈ $2.4092 \times 10^{-3}$ mg / mm$^3$ in the subarachnoid CSF. The width of the subarachnoid space varies depending on the position inside the skull [10] from about 0.25 mm to around 3 mm. Since the dimensions of the brain are approximately two orders of magnitude larger we will simply assume that the tracer concentration in the subarachnoid space is a good approximation to the concentration on the two dimensional surface of the boundary with the brain. Thus on the boundary we have the initial concentration given approximately by

$$u_0(x) = u(x, 0) = 2.4092 \times 10^{-3}, \quad x \in \partial \Omega,$$

and we now know all of the necessary information for the initial condition of the model (3.0.1). In particular, the full expression of the initial condition is given below.

$$u_0(x) = u(x, 0) = \begin{cases} 0 & x \in \Omega \setminus \partial \Omega \\ 2.4092 \times 10^{-3} & x \in \partial \Omega \end{cases} \quad (3.2.3)$$

As tracer concentration diffuses into the brain there is, of course, a transfer of mass. We define the function $M_b(t)$ to be the mass of the gadobutrol within the brain tissue volume at time $t$. Since $u(x, t)$ represents the tracer concentration at time $t$ we of course have, for any $t_n \in [0, T]$, the relationship

$$M_b(t_n) = \int_{\Omega} u(x, t_n) \, dx, \quad (3.2.4)$$
and it therefore follows from the initial condition above that \( M_b(0) = 0 \). Our second simplifying assumption, that all tracer is either in the left subarachnoid space or inside the left brain hemisphere, yields the conservation of tracer mass relation given by

\[
M_{SAS}(t) + M_b(t) = 151.178 \text{ mg},
\]

where \( M_{SAS} \) represents our mass of tracer in the left hemisphere subarachnoid CSF. Since the tracer is assumed to be uniformly well mixed we can divide both sides of the above equation by the left hemisphere subarachnoid volume, \( 6.275 \times 10^4 \text{ mm}^3 \), to get

\[
C_{SAS}(t_n) \approx 2.4092 \times 10^{-3} - 1.5936 \times 10^{-5} \int_{\Omega} u(x, t_n) \, dx,
\]

where \( C_{SAS}(t_n) \) denotes the concentration in the left subarachnoid space at time \( t_n \in [0, T] \). We have once again used the well-mixed assumption to establish the left-hand side equality; namely that

\[
C_{SAS} = \frac{M_{SAS}}{6.275 \times 10^4}.
\]

We see that, since \( M_b(0) = 0 \), the above equation holds up to time \( t = 0 \). We can then use the relationship between our left-hemisphere subarachnoid CSF volume and Dirichlet boundary condition defined, for every \( t \in [0, T] \), by the spatially constant value

\[
u_d(x, t_n) = C_{SAS}(t_n), \quad x \in \partial\Omega \tag{3.2.5}
\]

The final component of (3.0.1) we need to determine from the clinical experiment is the right-hand side source function \( f(x, t) \). This is straightforward; since no gadobutrol tracer is, at any point, injected into the brain itself we have

\[
f(x, t) = 0 \quad x \in \Omega, \quad t \in [0, T].
\]

### 3.2.2.2 Implementation using FEniCS

From a numerical point of view we will employ a FEniCS implementation of the scheme discussed in 3.2.1. The process begins with an approximation of our initial condition, \( u_h(x, 0) \approx u(x, 0) \), computed using (3.2.2). We will then solve (3.2.1) to obtain our approximate solution \( u_h(x, t_1) \). Afterwards, we will: (i)
compute and approximation to \( M_b(t_1) \) using \( u_h(x, t_1) \) and (3.2.4); (ii) update our Dirichlet boundary condition via (3.2.5); (iii) compute the approximation \( u_h(x, t_2) \) to the tracer concentration in the left hemisphere; and repeat (i)-(iii) until we have finished.

<table>
<thead>
<tr>
<th>Model variable</th>
<th>Value used in the simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K )</td>
<td>( 4.32 \times 10^{-1} )</td>
</tr>
<tr>
<td>( f(x, t) )</td>
<td>0</td>
</tr>
</tbody>
</table>
| \( u_0(x) \)   | \[
\begin{cases}
2.4092 \times 10^{-3} & x \in \partial \Omega \\
0 & x \in \Omega \setminus \partial \Omega
\end{cases}
\]|
| \( u_d(x, t_n) \) | \( 2.4092 \times 10^{-3} - 1.5936 \times 10^{-5} \int_\Omega u(x, t_{n-1}) \, dx \) |

**Table 3.1** Simplified clinical model specification for (3.0.1)

To accomplish this we will implement the FEniCS code to solve (3.2.1)-(3.2.2) using the values listed in table 3.1. The most popular, and approachable, way of implementing simple numerical codes using FEniCS is via the Python scripting interface. Thus, we adopt that methodology here. First we import the relevant libraries and specify some of the easier parts of our model; this is done in the following snippet.

```python
from fenics import *
import time

# time units: Hours
# time step: every 1 min
time.final = 9.0
time.steps = 540

# Initial concentration of
# tracer inside the brain
u.0 = Constant(0.0)
```

114
Next we define the time dependent Dirichlet boundary condition by using a FEniCS expression that we will update in the time loop. Consider the following snippet.

```
# Initial concentration at
# the boundary (mg / mm3)
Mb = Constant(0.0)
u_d_str = '(2.4092e−3 − (1.5936e−5)*mb)'
u_d = Expression(u_d_str, mb=Mb, degree=2)
```

We see that the above snippet implements (3.2.5). We note that FEniCS sees that the expression $mb$ defined in the string $u_d$ depends on the value of the variable $Mb$. If we change the value of $Mb$ during the course of our numerical solve iterations then FEniCS will take this into account, automatically, when evaluating the expression object $u_d$. True to its name, $u_d$ will be used, momentarily, to define the Dirichlet boundary condition. Next, we load our brain mesh, produced using SVM-Tk in Section 3.1.4, define our time increment and specify where the output should be saved.

```
# Load the SVM−Tk mesh
mesh = Mesh('ernie.xml')

# Save output here
vtkfile = File('chp3−diffusion/solution.pvd')

# Set time increment
dt = time_final / time_steps
```

The next snippet specifies that we will use continuous finite elements whose restriction to each tetrahedra in our brain mesh is a first degree (linear) polynomial in the spatial variables $(x, y, z)$. We then specify that, on the boundary, the solution should agree with the function $u_d$ we defined above. We then define the initial iterate $u_n$ and apply the boundary conditions to it; in the time loop $u_n$ will contain the previous solution at each solve iteration.
# boundary conditions
# specified by u_d
V = FunctionSpace(mesh, 'Lagrange', 1)
bcs = DirichletBC(V, u_d, 'on_boundary')

# initial value with
# boundary conditions
u_n = interpolate(u_0, V)
bcs.apply(u_n.vector())

We note that, at this point, the variable $u_n$ above now correctly implements the initial data (3.2.3). The next step is to describe the variational (weak) problem, given by (3.2.1), to FEniCS.

```python
u = TrialFunction(V)
v = TestFunction(V)

# Formulation of the model problem
F = u*v*dx + dt*K*dot(grad(u), grad(v))*dx - 
  (u_n + dt*f)*v*dx

# Store the left-hand side matrix, a, and right-hand side vector L
a, L = lhs(F), rhs(F)
```

Next, we prepare for time-stepping by defining a variable $u$ to hold the result of our solve and set the initial time $t = 0$. Since the left-hand side of our equation does not need to be updated every time step we can assemble it before starting the time stepping loop.

```python
# Setup for time stepping
u = Function(V)
t = 0

# Assemble left-hand side
# (static) matrix
A = assemble(a)
```

Finally, we put everything together and solve our problem for each discrete time $t_0 = 0, t_1, t_2, \ldots, t_N = T$ and output the solution for visualization. Note that at the end of each time loop we compute the current concentration and
update the variable \( C_b \) used in the definition of the Dirichlet boundary condition. The FEniCS library takes care of updating \( u_d \), and the boundary condition on the function space \( V \), behind the scenes.

```python
for n in range(time_steps):
    # Update current time
    t += dt

    # Assemble the (time-dependent) right hand side
    b = assemble(L)

    # Apply the boundary conditions to the system
    bc.apply(A, b)

    # Compute solution
    solve(A, u.vector(), b)

    # Save solution for visualization
    vtkfile << (u, t)

    # Update previous solution
    u_n.assign(u)

    # estimate the mass of gadobutrol in the brain
    # and updated boundary condition variable \( M_b \)
    M = u*dx
    uint = assemble(M)
    M_b.assign(uint)
```

The full source code file now follows, without comments, for convenience.

```python
from fenics import *
import time

time_final = 9.0
time_steps = 540
```
u₀ = Constant(0.0)
K = Constant(4.32e−1)
f = Constant(0.0)
Mb = Constant(0.0)

ud_str = '2.4092e-3 - (1.5936e-5)*mb'
u_d = Expression(ud_str, mb=Mb, degree=2)

mesh = Mesh('ernie.xml')
vtkfile = File('chp3-diffusion-mritracer/solution.pvd')

V = FunctionSpace(mesh, 'Lagrange', 1)
bc = DirichletBC(V, u_d, 'on_boundary')

u_n = interpolate(u₀, V)
bc.apply(u_n.vector())

M = u_n*dx
tvtkfile << u_n
print (assemble(M))

u = TrialFunction(V)
v = TestFunction(V)

F = u*v*dx + dt*K*dot(grad(u), grad(v))*dx - \
  (u_n + dt*f)*v*dx
a, L = lhs(F), rhs(F)

t = 0

u = Function(V)
A = assemble(a)

for n in range(time_steps):
  t += dt
  b = assemble(L)
  bc.apply(A,b)
As we have seen previously you can execute this program by opening and terminal window and typing the following command.

```
Terminal window

$ python ./chp3-diffusion.py
```

### 3.2.2.3 Visualization of results and comparison remarks

We will use the open source Paraview visualization program to look at the results. In a terminal window navigate to the directory where the visualization files are stored and type the command below to launch Paraview and open the main file for the visualization.

```
Terminal window

$ paraview ./solution.pvd
```

When Paraview has opened you will see, on the left pane, that ‘solution.pvd’ is already there. Click the button labeled ”Apply” to activate it in the viewing window. You can rotate the image by clicking in the viewing window and while turning your mouse while holding the left mouse button down. Since the tracer concentration was constant on the boundary you will notice that the entire left hemisphere is the same color. To see the variation in tracer concentration inside the brain we need to ‘clip’ the mesh open. To do this click ‘Filters→Clip’ in the menu bar. You will see the left pane change to display the ‘Plane Parameters’ that determine where the clip will occur. You can modify these parameters if you wish and you will see the red ‘clipping plane’ change accordingly; c.f. Figure 3.6. When you are satisfied with the location of the red clipping plane, click apply to slice the brain open for visualization of the tracer concentration within it.
Fig. 3.6 The clip plane operation in Paraview

Fig. 3.7 Approximate distribution of tracer concentration (red is higher, blue is lower) from $t = 2h$ to $t = 72h$ post injection into subarachnoid CSF
3.2 A simple mathematical model of diffusion in the brain

After you have clipped your mesh you will want to rescale the displayed value, which represents the concentration of the tracer, over all of the time steps in the full series. To do this, first select ‘solution.pvd’, by clicking on the name, in the top left pane. Then, you will see options appear in the left pane below the top-left window. One of these options is titled ‘Coloring’. Moving and holding your mouse cursor over the buttons in this section will display text telling you their functionality. Select the button that shows ‘Rescale to data range over all timesteps’. Clicking this button will start a process that can take several minutes to complete. After this completes you can view the result of the computation by clicking the green ‘play button’ icon at the top of the graphical interface just below the menu bar. Figure 3.7 shows the simple model predictions for the distribution of tracer concentration from 2 hours post injection to 72 hours post injection. In figure 3.7 higher concentrations, near the maximum of $3.984 \times 10^{-3}$, are red while lower concentrations, near or at zero, are blue.

![Fig. 3.8 Prediction of relative subarachnoid gadobutrol tracer remaining](image)

If we modify the FEniCS script `chp3-diffusion.py` to print the concentration in the brain, $C_b$, at different time points we can evaluate the model prediction for the concentration remaining in the subarachnoid CSF via $3.294 \times 10^{-3} - C_b$. We can then plot the percentage of tracer concentration in the subarachnoid CSF relative to the initial concentration to visualize the uptake of the tracer into the brain; this is plotted in figure 3.8.

From figure 3.8 we can see that the model predicts that in 72 hours approximately 55% of the gadobutrol tracer remains in the subarachnoid CSF so
3 A simple diffusion problem in the brain

that 45% has been diffusively absorbed by the brain tissue. If we compare the images in figure 3.7 to those of the actual tracer study for reference, or dementia, patients [14] we see that they do not match. In particular, the tracer MRI intensity distributions of [14] show much more complex patterns than those of figure 3.7. This is not bad news; it implies that the brain is more complex than a simple sponge and that there is an exciting modeling depth to be found in brain phenomena; even for simple clinical experiments.

### 3.2.3 SVM-Tk Implementation Details

We implemented SVM-Tk for the purpose of constructing a volume meshes using a closed surface. The process can be divided into two steps, the processing of the surface and the construction of the mesh. In general, surfaces can often have different types of defects, which needs to be repaired so that the surface can be used for constructing meshes.

The idea of a volume mesh with subdomains caused by overlapping surface prompted the implementation of the `Surface` class. This means that each surface can be examined and repaired as needed, and subsequently used for constructing a mesh. The `Surface` class was implemented with basic utilities, such as boolean operations, but also specific repair algorithms, like `isotropic_remeshing`. We will go in more details about the implementation of core member functions, which includes functions that we have used earlier in this chapter.

The core mesh construction process was implemented with the `Domain` class, which required the declaration of CGAL trait classes for domain and mesh. The domain class contains the information required to construct the mesh, while the mesh class contains the data structure of mesh, i.e. cells, facets and vertices, of the resulting mesh. In CGAL, the domain classes have the copy constructor and assignment operator disabled, so we utilized unique pointer to avoid the common pointer pitfalls in C++. Furthermore, the disabled operations made iterative additions, such as `push_back` and `add`, of surfaces impossible. Therefore, all surfaces was required as input for `Domain` class constructor. The storing of the volume mesh as member variable was due to the limited number of implemented functions for the volume mesh, such as `save` and `remove_subdomain`, and did not necessitate the implementation of a new class. However, if new functions are added, then the implementation of a Mesh class can be sought in the future.
The structure of SVM-Tk included the CGAL headers when needed, and we declared the relevant CGAL trait classes with typedefs within each class. This implementation structure was used to avoid compilation errors due to hanging declarations while writing the code, and to avoid long variable names in the function implementation. We should note that the SVM-Tk is an active module, thus it can be subjected to additions, bug fixes and improvements, hence the implemented presented here may differ from the current implementation in the code repository. Additionally, not all functions are exposed to the python interface, and can therefore not be used with python.

We will now go through the implementation of SVM-Tk in detail, focusing on the classes and functions that we have used. First, we have the code implementation of `Surface`.

```cpp
class Surface
{
public:
    typedef CGAL::Exact_predicates_inexact_constructions_kernel Kernel;
    typedef Kernel::Point_3 Point_3;
    typedef CGAL::Surface_mesh<Point_3> Mesh;
    typedef CGAL::Polyhedron_3<Kernel> Polyhedron;
    typedef CGAL::Side_of_triangle_mesh<Mesh,Kernel> Inside;

    // ---- Auxiliary ----
    typedef std::vector<Point_3> Polyline_3;
    typedef std::vector<Polyline_3> Polylines;
    typedef std::vector<std::size_t> Face;

    // --- Constructors ---
    Surface(){}
    Surface(std::vector<Point_3>& points, std::vector<Face>& faces);
    Surface(const std::string filename);

    // --- Boolean ---
    void surface_intersection( CGALSurface& other);
    void surface_difference( CGALSurface& other);
    void surface_union( CGALSurface& other);

    // --- Repair/Deformation ---
};
```
First, we specified the CGAL kernel that we used for the implementation, which was the exact predicates inexact construction kernel. This kernel was determined to be best kernel in CGAL for the purpose of handling surfaces and constructing meshes. The kernel provides exact geometric predicates, i.e. inquiries about orientation, position and other methods that provide boolean or enum return values. However, the geometric construction may be inexact, due to round-off errors when constructing new points. The inexact kernel operations are subsequently faster than the exact construction kernels that also exists in CGAL, more details are found in [2].

We implemented the class with two different declarations of surface mesh, namely Polyhedron and Mesh. Both classes was used to handle triangulated surface mesh, but the Polyhedron class used pointers and Mesh used indices to structure the surface points. In most cases these meshes are interchangeable, and all functions in SVM-Tk are implemented to convert between the classes if needed.

The function Inside was frequently used in the class functions to determine whether a point was inside another Surface. For instance, this function was used to evaluate if two surfaces overlap and needs to be separated. We have mentioned the function separate_narrow_gaps in Sec. 3.1.4 has a method to avoid bridges in the cortical gray matter. This function has no arguments, as it will iteratively separate non-neighbor vertices until the distance between them was larger than the shortest connected edge.

The boolean operations listed in the header of Surface use functions implemented in CGAL\(^5\). These function may fail if the user specified a non-
functional operation, like union of non-overlapping surfaces. We will use the union operation in Chapter 4.1.3, to create a union of the white matter surfaces.

We used the function *isotropic_remeshing* to reconstruct the loaded surface so that each edge have similar length. The function has three arguments; the edge length, the number of iterations and protection of constrained edges. The function was implemented as a combination of two CGAL functions, *split_edges* and *isotropic_remeshing*. The function *isotropic_remeshing* was implemented in CGAL using an incremental triangle-based isotropic remeshing algorithm. We decided to use the function *split_edges* to handle the splitting of long protected constrained edges. This was due to the fact that if a protected edge was too long it can cause remeshing to fail by create an infinite loop of edge splits in the incident faces.

We presented the function *smooth_laplacian* in Section 3.4. The implementation was based on the python module mshr, 6, which can be considered as a precursor to SVM-Tk. The function was implemented to determine the new position of a vertex by summation of all the connected edges as vectors. The resulting vector was then multiplied with a smoothing factor, the function argument.

The function *smooth_taubin* was covered in Section 3.4 as a volume preserving smoothing algorithm. It was implemented by utilizing the *smooth_laplacian* with two different parameters $\mu$ and $\lambda$. These parameters are selected so that $0 < \mu < -\lambda$, further details are in [16]. The arguments for taubin smoothing were set $\mu = 0.8$ and $\lambda = -0.805$. For other parameters, the user can use the *smooth_laplacian* with specific parameters.

We have discussed the defects, such as holes, in Section 3.1.4. These defects can be corrected by FreeSurfer, but there can also be holes in the surfaces due to missing facets. The function *fill_holes* followed the example provided in CGAL to add the missing facets.

The *save* function was implemented to write the surface mesh to the file that was given as an argument. This can be used to inspect the processed surface, and determine if it can be used for mesh construction. Currently, we have two supported file formats, namely the default surface mesh in CGAL off and stl.

The mesh construction used the concept of mesh domains, which represents the information required for the object to be discretized, that is boundaries, subdomains and 0-1 dimensions features. The Domain was implemented to

\[\text{6} \text{https://bitbucket.org/fenics-project/mshr/src/master/}\]
automatically wrap the input surface to a function, this function was used to evaluate a point position relative to the surface, i.e., inside, outside or at the boundary. We will detail the main components of `Domain` header in the following code excerpt.

```cpp
class Domain{
    public :
        typedef CGAL::Exact_predicates_inexact_constructions_kernel K;
        typedef CGAL::Mesh_polyhedron_3<K>::type Polyhedron;
        typedef CGAL::Polyhedral_mesh_domain_with_features_3<K,
            Polyhedron> Polyhedral_mesh_domain_3;
        typedef CGAL::Polyhedral_vector_to_labeled_function_wrapper<
            Polyhedral_mesh_domain_3, K > Function_wrapper;

        typedef Function_wrapper::Function_vector Function_vector; //
        typedef CGAL::Labeled_mesh_domain_3<K> Labeled_Mesh_Domain;

        typedef CGAL::Mesh_domain_with_polyline_features_3<
            Labeled_Mesh_Domain> Mesh_domain;

        typedef CGAL::Mesh_triangulation_3<Mesh_domain>::type Tr;

        typedef CGAL::Mesh_complex_3_in_triangulation_3< Tr> C3t3;

    Domain(Surface& surface);
    Domain(std::vector<Surface> surfaces);
    Domain(std::vector<CGALSurface> surfaces, AbstractMap& map);

    void remove_subdomain(std::vector<int> tags);
    void remove_subdomain(int tag);
    void save(std::string OutPath);
    void create_mesh(const double mesh_resolution );

    private :
        std::unique_ptr<Mesh_domain> domain_ptr;
        Minimum_sphere<K> min_sphere;
        C3t3 c3t3;
}
```
The class **Domain** starts by declaring the CGAL kernel and classes that was used in the implementation. The **Domain** class uses the exact predicates inexact construction kernel, which was covered for the case of **Surface**.

The next batch of declarations concerns the wrapping of polyhedron surfaces to labeled mesh function, which allows for the construction meshes with indexed domains. First, we declared the triangulated surface structure with the class **Mesh polyhedron 3**, and used it together with kernel to define the mesh domain class for polyhedron **Polyhedral mesh domain**. The labeled domain class was defined with **Labeled Mesh Domain**, and the wrapping of polyhedron to a labeled function with **Function wrapper**. We declared **Function vector**, so that multiple surfaces can be used to construct meshes.

The wrapping of surfaces to a labeled mesh domain was done, under the hood, in the constructor. In the constructor implementation, all surfaces were loaded into a function vector, which was used to declare the mesh domain pointer **domain_ptr**. We implemented **AbstractMap** as a virtual base class for the mapping of the different subdomain, which was created by overlapping surfaces. We will come back to this in Chapter 4.1.1, with the class **SubdomainMap**.

The mesh construction required that we defined, the mesh domain, mesh criteria, the triangulation and the cell structure. We declared the volume mesh triangulation class with **Mesh triangulation 3**, and the cell structure with **Mesh complex 3 in triangulation 3**. The mesh was stored using the cell structure variable **c3t3**, which was also used for the triangulation. We would like to note an important distinction between the triangulation and the cell structure (mesh), as the triangulation can have vertices not connected to any cell in the mesh. The construction of isolated vertices can be minimized by sufficient preprocessing, but to ensure no isolated vertices, we implemented a function to remove all isolated vertices from the mesh.

We used the function **create mesh** with the input of the **mesh resolution** to determine the mesh criteria that was needed for CGAL. This was done with the struct **Minimum sphere**, which computes the minimum sphere radius needed to enclose all points in the polyhedron surfaces. We used the radius with the **mesh resolution** input to set the maximum size of edges in the volume mesh. If the triangulated surfaces had smaller edges, then the smaller edges will be preserved in the constructed mesh.

The function **remove subdomain** was implemented to remove the subdomain from a constructed mesh. This requires that the user specifies a subdomain ‘tag’, and it will remove all cells with the same ‘tag’. We can also remove multiple subdomains by adding the multiple ‘tags’ to a vector and use the vector as the argument. This function and usage will be further detailed in Chapter 4.1.2.1.
We can write the constructed mesh to file with the `save` function. This function will write the volume mesh to the provided file. There is currently only one supported file format, which is the medit file format. This file format is the default format for CGAL, and has the extension `.mesh.`
References

Chapter 4
Simple diffusion in separate brain regions

In this chapter, the primary goal is to extend our ability to handle slightly more complicated physiological models by introducing different brain regions characterized by their own parameters. That is, we simply focus on including region-specific diffusion parameters within our mesh construction.

From a practical point of view this implies a focus on differentiating regions of our computational mesh; including identifying a gray matter region and a white matter region. We will do this in both the left hemisphere domain of Chapter 3, to start, as well as the whole brain. We will then solve the simple model problem given by

\[
\partial_t u - \text{div} (K(x) \nabla u) = f, \tag{4.0.1}
\]

\[
u = u_d(t,x) \text{ on } (0,T] \times \partial \Omega, \quad u(0,x) = u_0(x) \text{ on } \Omega,
\]

\[
K(x) = \begin{cases} 
K_{\text{gray}} & x \in \Omega_{\text{gray}}, \\
K_{\text{white}} & x \in \Omega_{\text{white}},
\end{cases}
\]

where \(K_{\text{gray}}\) and \(K_{\text{white}}\) are two real constants and \(\Omega = \Omega_{\text{gray}} \cup \Omega_{\text{white}}\) signifies the gray and white matter regions, respectively.

In practice the flow of interstitial fluids through the brain, which is crucial for a healthy metabolism; i.e., the timely delivery of nutrition and oxygen as well as the corresponding removal of metabolic waste and carbon dioxide, is often studied from the view point of either diffusion or porous media models. Diffusion models were established in the seminal works of Nicolson & Syková [3] where well crafted experiments with molecules of different sizes demonstrated that transport governed by extracellular diffusion is fundamental in the brain’s metabolism. Later, several studies have examined to what extent
convection may play a role, in particular for larger molecules. For instance, convection enhanced delivery by enforcing transmantle pressure gradients has been proposed as method to accelerate the transport of for instance drugs [2] by pressurizing a region by infusion. Corresponding computational modeling studies are e.g. [5, 4]. It is interesting to note that in these studies the variations in permeability are larger than that of diffusivity. Recently, it has been proposed that the CSF plays an important role in the clearance of metabolic waste. The intimate coupling between the CSF and the interstitial fluid of the extracellular space has been named the lymphatic system [12, 11, 10] and it is crucial here that this system facilitate convection. While there has been many studies of this system on the micro-level [13, 14] the macroscopic behaviour of the system has not received that much attention. Part of the reason for this, we assume, is the lack of utilities to create patient-specific computational models easily. While mathematical models for porous flow, such as Biot’s equations [6] or the equations of generalized poroelasticity [7, 8] are readily available, a crucial challenge is to create meshes with plausible realism in a reasonable time-frame.

In some cases it is beneficial to remove the ventricles from the computational brain mesh. Hence, in this chapter, we will also discuss how one can remove a region from a computational domains based on a surface stl files; we will use this approach to demonstrate the removal of the ventricular volume from our computational domain.

### 4.1 Combining regional data from separate surface files

This section focuses on combining data, from separate surface files generated by FreeSurfer, to create computational meshes of varying complexity. Section 4.1.1 covers the basics of tagging gray and white matter regions using SVM-Tk; section 4.1.2 demonstrates how to use SVM-Tk to remove one surface from another in order to remove the ventricles; finally, section 4.1.1 combines these approaches to create a comprehensive mesh, of the left hemisphere, with differentiated cortical gray and white matter in which the ventricular volume has been removed.
4.1 Combining regional data from separate surface files

4.1.1 The left hemisphere with differentiated gray and white matter

To introduce the basic concepts of differentiating brain regions using segmented FreeSurfer MRI data alongside the SVM-Tk we will first create a computational mesh of the left hemisphere, as in Chapter 3, but where the gray and white matter regions are differentiated with ‘tags’.

The concept of a ‘tag’ can be given a mathematical definition as a simplicial function; that is, a function whose argument is a mesh element, i.e. simplex or sub-simplex, and returning a positive integer. In simpler language a ‘tag’ is an associated integer value for each simplex such that all of the simplices sharing the same value are considered to be part of the same ‘region’ of the mesh. Within FEniCS, MeshFunctions are used to specify these tags. The MeshFunctions may be defined in terms of cells, facets, or vertices. A comprehensive description of this can be found in [9].

Our task, then, is to create a mesh, $\Omega_h$, for which we have a tag that differentiates the gray and white matter. In order to do this, we first need to extract a bounding (outer) surface for the gray matter and a bounding surface for the white matter. When we used FreeSurfer to segment our MRI data one thing that it did was create a collection of bounding surfaces for several substructures present in the brain. In fact, the bounding surface for the gray matter is the pial surface we worked with in Chapter 3; a bounding surface for the white matter is also created during a standard segmentation.

Navigate to the FreeSurfer subjects directory; e.g. for our current subject ernie this is `~/freesurfer/subjects/ernie`. Then convert the pial surface file `lh.pial` and the white matter surface `lh.white` to a surface stl file using the following commands.

```
Terminal window
$ mris_convert ./lh.pial lh-ernie-pial.stl
$ mris_convert ./lh.white lh-ernie-white.stl
```

If desired we could carry out the preprocessing procedures discussed in Chapters 3.1.5.1-3.1.5.2 on the above surface stl files to increase their mesh quality. However, since our purpose here is only to demonstrate tagging, we will work with the raw surface stl files; the exact same procedure described here will, of course, work with preprocessed files.

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

#!/usr/bin/python
import SVMTK as svm

# --- Options ---
gray_matter_filename = 'lh-ernie-pial.stl'
white_matter_filename = 'lh-ernie-white.stl'

output_mesh_filename = 'ernie-graywhite.mesh'

# resolution of final mesh
mesh_resolution = 32.0

# Load the surfaces into SVM-Tk
pial = svm.Surface(gray_matter_filename)
white = svm.Surface(white_matter_filename)

# Create a list of surfaces
surfaces = [pial, white]

# Create a surface domain map
smap = svm.SubdomainMap()
smap.add("10", 1)
smap.add("11", 2)

# Create a tagged domain from the surfaces and surface domain map
domain = svm.Domain(surfaces, smap)

# Create a volume mesh from the domain
domain.create_mesh(mesh_resolution)

# Save the mesh
domain.save(output_mesh_filename)

In the above code we see some familiar aspects. Namely, we see the input and output filenames in addition to the same mesh_resolution parameter we encountered in Chapter 3.1.4. Recall that the higher the floating point value used for mesh_resolution the smaller the tetrahedra will be in the final output volume mesh.

However, we are now choosing to use two different surface as input to the svm. Domain object. This will create two functions, which are combined to an array.
In the mesh construction, each function will iteratively evaluate if a point is inside or outside the wrapped surface, producing strings consistent of "0" and "1", i.e. a binary string. If no further mapping is specified, then each string will be converted to a 10 base integer, creating a unique tag for each enclosed surface region.

**Subdomain Maps in SVM-Tk**

The python code of `two-domain-tagged.py`, above, uses a feature of the SVM-Tk that we have not encountered yet; this feature is the `SubdomainMap`. A `SubdomainMap` is a specification on how the SVM-Tk will handle the generated bit-strings, $D : \mathcal{T}_h \rightarrow \mathbb{N}$, we discussed at the beginning of the section. Let us analyze the following code snippet in more detail:

```python
surfaces = [pial, white]
smap = svm.SubdomainMap()
smap.add("10",1)
smap.add("11",2)
```

We note that our surface list, `surfaces`, has length two, which will be wrapped to query functions inside `svm.Domain`. This means that there four possible evaluation of a point, namely "00", "10", "11" and "01", with "1" means 'inside' and "0" means 'outside'. If we had three surfaces in our list then the bit-string would have a combination of three ones and zeroes; such as "100" or "101" etc. So for the example above the line

```python
smap.add("10",1)
```
reads 'mark the volume (e.g. the resulting mesh tetrahedra) that lie inside the pial surface and outside the white surface with a 1'. The line

```python
smap.add("11",2)
```
reads 'mark the volume inside the pial surface and inside the white surface with a 2'. Note that the SVM-Tk considers the volume contained inside the white surface as 'volume inside the pial surface and inside the white surface' and not simply the 'volume inside the white surface'. Thus the `Subdomain` class takes a global approach to volume specification; that is to say a volume is defined relative to all surfaces present in the `surfaces` list and not simply one of those surfaces. As a result, some prior knowledge of what your surface stl files look like, e.g. through visualization, can be quite helpful when utilizing the `Subdomain` class.
As a second example of a subdomain map suppose that we have four surfaces; a left pial surface, a right pial surface, the white matter surface and the ventricle surface. We would then have the following example code.

```python
leftpial = svm.Surface(leftp_filename)
rightpial = svm.Surface(rightp_filename)
white = svm.Surface(white_filename)
ventricles = svm.Surface(ventricle_filename)
surfaces = [leftpial, rightpial, white, ventricles]
smap = svm.SubdomainMap()
```

Now suppose that we want to mark all tetrahedra in the ventricles which are also in the left pial surface with a numerical tag of ‘6’. We will begin by considering a simplification, where each surface is represented by a square, creating a structure of figure 4.1. The upper left panel in figure 4.1 shows four different colors, and each color represents the enclosed area of a surface. When we combine these surfaces, we get a surface structure like the upper right panel, where each color represents a different string consistent of “0” and “1”. Thus the string ”1000” will give you the left hemisphere cortical gray matter, and ”1010” will give you the left hemisphere white matter.

In our example code displayed earlier, the surfaces list have the following order; left pial surface, right pial surface, white matter surface and the ventricle surface. Then the appropriate call to `smap.add(...)` is as below.

```python
smap.add("1011",6)
```

Likewise if we wanted to mark the volume in the ventricles lying in the right pial surface we would add the line below.

```python
smap.add("0111",6)
```

Finally, if we wanted to mark the ventricular volume which lies at the pial intersection we would do this by adding the line indicated next.

```python
smap.add("1111",6)
```

Since there are a total of four surfaces in our example surface list then tagging the entire ventricular volume with the SVM-Tk requires that we add all three of the lines above.

**Creating and visualizing the two-domain mesh**

The procedure to create the mesh starts by opening a terminal window inside
4.1 Combining regional data from separate surface files

The upper left panel shows colored squares, each representing a surface. That is, the left hemisphere is blue, the right hemisphere is red, the white matter is purple and the ventricles are yellow. The upper right panel shows the combination of each colored square. The bottom panel shows four different subdomains and its corresponding bit-string. The left image ”1000” denotes the volume that is within the left hemisphere, but not within the right hemisphere, white matter or ventricles; hence the gray matter of the left hemisphere. The ”0100” is completely analogous. Then ”1011” means the surface that is within the left hemisphere and within both white and ventricles; hence the left ventricle. Finally, for ”0110” is used to indicate the white matter of the right hemisphere.

The directory containing the surfaces files. Then we can run the `two-domain-tagged.py` python program by, as we have seen before, executing the python script as follows.

```
Terminal window
$ python two-domain-tagged.py
```

The SVM-Tk creates a volume mesh from the surface stl files and the resulting mesh file saved to disk is `ernie-graywhite.mesh`. To visualize this file we first convert it to a format that is readable by Paraview. In a terminal window issue the following command from within the same directory. We will use `meshio` module to convert the mesh to the vtu file format. This file format is used for unstructured meshes in The Visualization Toolkit (VTK).
Start Paraview and load \texttt{ernie-graywhite.vtu}; in order to see the tags we clip the domain as in figure 3.6. Note that, from the drop down menu at the top or the left-pane menu on the side, you may need to select the ‘medit:ref’ cell data in order to see the tag values. Figure 4.2 shows the tags on the sliced domain. On the left we see the gray matter tetrahedra have been tagged with a value of 1 and the white matter has been tagged with a value of 2; a color scheme clearly showing ‘gray’ and ‘white’ has been selected. The right image in figure 4.2 shows that the ventricles have been tagged as ‘white matter’. This is due to the fact that they live inside the white matter boundary; removing the ventricles from our computational mesh is the topic of section 4.1.2.

\subsection*{4.1.2 Removing a ventricular volume}

In this section we perform two tasks. First, we use FreeSurfer command line tools to extract and postprocess a ventricle surface stl file; second, we use this extracted surface to remove the ventricles using the \texttt{SubdomainMap} object discussed in section 4.1.1.

\subsubsection*{4.1.2.1 Extracting a ventricular surface from MRI data}

To begin, we first need to extract the ventricle surface from our MRI data using FreeSurfer command line tools. In a terminal window, navigate to
the FreeSurfer subject directory. We have previously been working with files located in the `surf` subdirectory; we will now work with a file in the `mri` subdirectory. For our example here, the full path to this directory is 

```
~/.freesurfer/subjects/ernie/mri
```

This directory consists of volume based data, like T1 weighted images, segmentations and parcellations. These volume files have the extension `mgz`, which is the default format in FreeSurfer. The segmentations and parcellations can be identified by substring in the filename. For instance we have the file `aseg.mgz`, which stands for automatic segmentation, and the file `wmparc.mgz` that stands for white matter parcellation. The parcellation will split the segmentation into finer regions, like the cortical gray matter will be divided into 35 regions for each hemisphere. We can use the segmentation files or the parcellation files to construct the surface of the ventricular volume with the following script.

```bash
#!/bin/bash

# --- Options --- #
input="wmparc.mgz"
output="ernie-ventricles.stl"

# Basic
smoothing_iterations=3

# Advanced
closing_iterations=2
minvolume_size=100

postprocess="false"

# Attempt to keep the
# fourth ventricle and
# the aqueduct
keep_fourth_and_aqueduct="false"

# Set the match string
# based on options above
matchstring='1'

if [ "$keep_fourth_and_aqueduct" = true ]; then
    matchstring='15'

```

```
fi

if [ "$postprocess" = false ] ; then
  # Basic ventricle extraction
  # without postprocessing
  mri_binarize --i $input --ventricles \n  --match $matchstring \n  --surf-smooth $smoothing_iterations \n  --surf $output
else
  # Ventricle extraction
  # with postprocessing
  mri_binarize --i $input --ventricles \n  --o "tmp.mgz"

  mri_volcluster --in "tmp.mgz" \n  --thmin 1 \n  --minsize $minvolume_size \n  --ocn "tmp-ocn.mgz"

  mri_binarize --i "tmp-ocn.mgz" \n  --match $matchstring \n  --o "tmp.mgz"

  mri_morphology "tmp.mgz" \n  close $closing_iterations "tmp.mgz"

  mri_binarize --i "tmp.mgz" \n  --match $matchstring \n  --surf-smooth $smoothing_iterations \n  --surf $output

  rm tmp.mgz
  rm tmp-ocn.mgz
fi
Freesurfer details

The FreeSurfer command `mri_binarize` is used to extract voxels that contain a certain type of information. For example, the information can be a range of signal values, or a collection of segmentation tags. The command requires two flags to function, a volume input `--i` and volume output `--o`, and there are about 40 optional flags that can be used. We will only go through some of these flags, but they are described using the flag `--help` and also online. The selection of voxels can be done using several optional flags, but we will focus on those used in relation to segmented volume files. First, we have the flag `--match` that is followed by one or more integers. This will mark all voxels that have the same integer values has the ones listed. In case of specific regions, there are optional flags that can be used instead, and we will present some of these optional flags:

- `--ventricles` marks voxels in the 3rd, lateral ventricles and the choroid plexus.
- `--ctx-wm` marks voxels in the cerebral white matter
- `--gm` marks voxels in the gray matter
- `--subcort-gm` marks voxels in the subcortical gray matter, including the gray matter in the cerebellum and brainstem.

These optional flags can also be combined with the match flag, for example:

```
Terminal window
$ mri_binarizes –i aseg.mgz –ventricles
–match 15 –o ventsys.mgz
```

This example will combine the ventricle flag and the voxels with segmentation value 15, which is the 4th ventricle. The segmentation values used in FreeSurfer are listed in the file `~/freesurfer/FreeSurferColorLUT.txt`. It is also possible to visualize and navigate through the different segmentation tags by using Freeview to open a segmentation or parcellation file with the lookup table colormap, like:

```
Terminal window
$ freeview –colormap lut –v aseg.mgz
```

---

1 https://surfer.nmr.mgh.harvard.edu/fswiki/mri_binarize
In the Freeview window, there will be a list on the left hand side. This list will contain all the segmentation tags that FreeSurfer uses. We can also, as mentioned, navigate through the volume file. This is done by letting the mouse hover over a region, which will cause the segmentation value and the name will appear in the bottom right corner.

In the output file, all the marked voxels will have the value 1 and the rest will be set 0. This can be changed by specifying the output binvalue with the optional flag `--bin` followed by an integer. The marked voxels will now have the selected bin value in the output. In some cases we only want to mark voxels within a specific region, thus can be accomplished using the flag `--mask` and a volume file. The volume file should contain non-zero voxels values in the region we wish to mark. We can also create surface using `mri_binarize` with the optional flag `--surf` followed by the filename of the output surface. It is recommended to use the .stl extension on the output surface. We can also use the surface flag instead or together with the volume output. The surface flag is often used together with the flag `--surf-smooth` followed by an integer determining the number of smoothing iterations on the output surface.

The FreeSurfer command `mri_volcluster` is used to find clusters in a volume file, and we define a cluster as a set of continuous voxels that satisfies a threshold criteria. This function has four required flags, and have about 40 optional flags. Therefore we will only cover some of the optional flags, but all flags are described with the flag `--help` and also online ². We will start with the four required flag, and the first one is `--in`, which is used for the input volume. Then we have the flag `--thmin` followed by a non-zero value, which means that the voxels must exceeds this value to be considered a part of the cluster. We continue with the flag `--minsize`, which is combined with a selected volume in mm³. Only cluster with more volume than the chosen value will be considered a cluster. The last required flag is the output flag, but there are different types of output possibilities. For example, we have

- `--out` used to save the output volume file
- `--ocn` used to save the output volume file with sorted clusters, all voxels of largest cluster will be have the tag 1, second largest would have the tag 2 and so on.
- `--sumfile` used to save a text summary of clusters, including size, max signal and coordinates.

---

² https://surfer.nmr.mgh.harvard.edu/fswiki/mri_volcluster
We are mostly interested in largest cluster, and we can extract this cluster by using the output file with \texttt{mri\_binarize} and the flag \texttt{--match 1}.

The FreeSurfer command \texttt{mri\_morphology} is used to perform certain operations on volume files. There are 9 different operations, which includes open, close, dilate, erode and fill holes. This function uses positional arguments, and the structure is

<table>
<thead>
<tr>
<th>Terminal window</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ \texttt{mri_morphology [options] input_volume operation iterations output_volume}$</td>
</tr>
</tbody>
</table>

Here the positional argument of iterations determine the number of times the function will perform the operations on the input volume. We can add one type of option that can be used with the function and that is the flag \texttt{-1} followed by a non-zero value. This will cause the operation only to be performed on voxels that have the specified non-zero value, otherwise it will perform the operation on all non-zero voxels.

The ventricle extraction script, above, uses several FreeSurfer commands and has two modes of operation. The primary modes are extraction without postprocessing and extraction with postprocessing; in addition, we have the option to attempt to keep the fourth ventricle and aqueduct or discard it. In practice, the decision to keep or discard the fourth ventricle and aqueduct is data specific. The aqueduct may not be well resolved in the MRI data on a patient-by-patient basis; if the aqueduct is not visible in the data then keeping the fourth ventricle leads to a ventricle system that is not connected. Moreover, if the \texttt{keep\_fourth\_and\_aqueduct} is set to \texttt{True} then one should take care with how much smoothing is applied; more on smoothing will be discussed momentarily.

From a larger scale, the decision in the script above is about extraction with or without postprocessing. For extraction without postprocessing, only the \texttt{mri\_binarize} command is used; extraction with postprocessing uses \texttt{mri\_binarize} and \texttt{mri\_morphology}. To change between the two, one can set the \texttt{postprocess} variable to either \texttt{postprocess=false}, yielding the basic extraction process, or \texttt{postprocess=true}.

The script variable \texttt{smoothing\_iterations} specifies the number of times that \texttt{mri\_binarize} should go over the smoothing process to smooth the resulting output. Likewise the script variable \texttt{closing\_iterations} specifies the number of times that \texttt{mri\_morphology} should iterate over the surface mesh and try to close holes. It is suggested to set \texttt{smoothing\_iterations} to an integer value between 1 and 5. Figure 4.3 shows an example of the ventricle
Fig. 4.3  Extracted ventricles (left) no smoothing and (right) five smoothing iterations

surface extracted without postprocessing and keep_fourth_and_aqueduct set to ‘True’; demonstrating the lack of MRI data fidelity in resolving the aqueduct for this specific patient. The left figure was extracted with no smoothing iterations, i.e. smoothing_iterations=0 in the above code, whereas the right-hand surface is the result of extraction with five smoothing iterations; that is, smoothing_iterations=5 in the script above.

If the above script is run with postprocess=true then the ventricle extraction process includes a few more commands and operations on intermediate files. Documentation for FreeSurfer commands is available online\(^3\); however we give a high-level overview, here, of their usage. The mri_volcluster command attempts to find clusters of volumes; the purpose here is to eliminate ‘floating regions’ that may have been produced by the segmentation but that are not desirable to keep. It is advised to set the minvolume_size variable to around 100; the units of this variable is mm\(^3\) and the average adult volume of cerebrospinal fluid in the ventricles is about 150 mm\(^3\).

The mri_morphology command is used to fill any holes that might be present in the surface by using a number of iterations of a closing algorithm; the number of iterations to be performed is indicated by the script variable closing_iterations. It is advised to set the number of closing iterations rel-

\(^3\) https://surfer.nmr.mgh.harvard.edu/fswiki/FreeSurferCommands
4.1 Combining regional data from separate surface files

Fig. 4.4 Ventricle (left) raw surface and (right) after smoothing and five closing iterations

atively low; for example closing_iterations=1 or closing_iterations=2. Setting the number of closing iterations too high can cause connections in the ventricle surface that are not physical; an illustration of this is shown in figure 4.4. The left-side image in figure 4.4 shows the raw, unsmoothed ventricle surface with zero closing iterations; the right-side image shows the extracted ventricle surface with three smoothing iterations and five closing iterations. Notice that in the raw surface the third ventricle does not connect to the lateral ventricle; whereas the postprocessed surface exhibits this non-physical connection as a result of excessive closing iterations.

If we compare figure 4.3 to figure 4.4 we can also see the effect of the mri_volcluster command directly; namely, figure 4.3 has a part of the fourth ventricle present whereas figure 4.4 does not; this is due to the fact that the fourth ventricle connected component is smaller than the 100 mm$^3$ value specified by minvolume_size. It is desirable to have a completely connected surface when removing areas from the mesh. Thus, even though the presence of the fourth ventricle is a physical truth, it is beneficial to remove them for the purposes of removing the ventricles from our computational domain. One can, if desired, use advanced FreeSurfer techniques to manually edit the fourth ventricle surface in Freeview in order to ensure that the raw extracted ventricles, i.e. the left image in figure 4.3, have a connected fourth ventricle surface; however, we do not cover such techniques here.
For our purposes, moving forward, we will use the script as it appears above. That is, we will use `postprocess=true` along with `smoothing_iterations=3`, and `closing_iterations=2`, with `minvolume_size=100` in order to extract and postprocess the ventricle surface. Figure 4.5 shows the surface stl file, `lh.ernie-ventricles.stl`, generated by the above code and viewed in Paraview.

The described process on how to obtain the surfaces of the ventricles can also be applied for cerebral subcortical gray matter. However, it is not recommend to use the flag `--subcort-gm`, since it will also include the gray matter in both the cerebellum and the brain stem. Furthermore, the structure of the subcortical gray matter is more complex with more narrow gaps between different components. Thus it requires some additional processing to be suitable for mesh construction. In the case of the segmentation of the hippocampus, then FreeSurfer offers the option of further segmentation. We will not cover this subject in the book, and the interested reader should look at [1].

---

4 Segmentation of hippocampal subfields
4.1.2.2 Using the SVM-Tk to remove a volume from a mesh

In this section we demonstrate how to use the SVM-Tk to remove an enclosed volume. Though we will focus, here, on removing the volume enclosed by the ventricle surface, extracted in the previous section, the general process will also work for any volume defined by a closed surface stl file. The basic idea is to use the SVM-Tk to generate a tag for all volumes in the domain and then simply delete the volume corresponding to the desired tag.

Tagging three left hemisphere domains with the SVM-Tk

We again assume that the extracted and postprocessed ventricle volume filename, from the previous section, is named lh.ernie-ventricles.stl.

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

import SVMTK as svm

# Options
gray_matter_filename = 'lh-ernie-pial.stl'
white_matter_filename = 'lh-ernie-white.stl'
ventricle_filename = 'lh.ernie-ventricles.stl'

output_mesh_filename = 'ernie-lh-novent.mesh'

# resolution of final mesh
mesh_resolution = 32.0

# Domain tag values
pialtag = 1
whitetag = 2
ventricletag = 3

# Removal flag
keep_ventricles = False

# Load the surfaces into SVM-Tk
pial = svm.Surface(gray_matter_filename)
white = svm.Surface(white_matter_filename)
ventricles = svm.Surface(ventricle_filename)
```
# Create a list of surfaces
surfaces = [pial, white, ventricles]

# Create a surface domain map
smap = svm.SubdomainMap()
smap.add("100", pialtag)
smap.add("110", whitetag)
smap.add("111", ventricletag)

# Create a tagged domain from the surfaces and surface domain map
domain = svm.Domain(surfaces, smap)

# Create a volume mesh from the domain
domain.create_mesh(mesh_resolution)

# Removes ventricle subdomain check
if keep_ventricles == False):
    domain.remove_subdomain(ventricletag)

# Save the mesh
domain.save(output_mesh_filename)

Consider the following code snippet where pialtag, whitetag and ventricletag are integer values.

surfaces = [pial, white, ventricles]
smap = svm.SubdomainMap()
smap.add("100", pialtag)
smap.add("110", whitetag)
smap.add("111", ventricletag)

Inspecting the above snippet we see that the domains are marked using `smap.add(...)` as discussed in the previous section. Moreover, the string of ones and zeros used to specify the subdomains contain three values; this is due to our surfaces list having three entries. In addition, we see again that the ordering of the ones and zeros in the strings passed to `smap.add(...)` are relative to the ordering of the surfaces in the `surfaces` list. If, for example, we changed the ordering in the surfaces list to be the line given below.
surfaces = [white, pial, ventricles]

The line marking the pial volume would change from its current form

smap.add("100", pialtag)

to the form reflecting the ordering in 'surfaces'; the required modification, in
this hypothetical example, is then:

smap.add("010", pialtag)

We mention again that the SVM-Tk sees subdomains relative to the global
surface list; therefore, to mark the ventricle subdomain in the left hemisphere
we specify ”111”, and not for instance ”001”, since the ventricle volume lies
within the surface enclosed by both the pial and white matter surfaces. Now,
consider the next snippet below.

domain = svm.Domain(surfaces ,smap)
domain.create_mesh(mesh_resolution)

if (keep_ventricles == False):
    domain.remove_subdomain(ventricletag)

domain.save(output_mesh_filename)

The important aspect of the above snippet is that the call to domain.create_mesh(...)
comes before any calls to domain.remove_subdomain(...). The reason for this
is that no physical mesh exists before domain.create_mesh(...) is called; after
the call to domain.create_mesh(...) the physical, labeled tetrahedra for each
region exist and can then be removed with domain.remove_subdomain(...) by
specifying the corresponding tag value. The function can also handle the
removal of multiple subdomains by adding the corresponding tags to a vector
and using the vector as input.

Figure 4.6 shows the left hemisphere mesh, generated by running ‘three-
domain-tagged.py from the command line, clipped in two different regions;
the same clipping plane was used for (a) and (c) and a separate plane was
used for (b) and (d). Images (a) and (b) of figure 4.6 correspond to the value
keep_ventricles set to False in the script while images (c) and (d) corre-
spond to keep_ventricles being set to True. The color scheme is chosen to
highlight the ventricles; the gray matter is an off-white, the white matter is
colored orange, while the ventricles in (c) and (d) are colored black. Compar-
ing Fig 4.6(a) and (b) to Fig 4.6(c), and (d) we can clearly see that Fig 4.6(a)
and (b) have had the ventricle volume removed.
In closing, we note that when constructing a computational mesh for use in numerical simulations, as in chapter 4.2, it is advised to first preprocess the individual surface files, as described in chapter 3.1.5.1-3.1.5.2, before constructing the final volume mesh using the `three-domain-tagged.py` script. We will briefly describe the preprocessing steps of the surface stl files in section 4.2 alongside the computation of the model problem (4.0.1).

### 4.1.3 Combining the left and right hemisphere domains with differentiated gray and white matter

In this section our aim is to create a mesh that includes both the left and right hemispheres, with gray and white regions tagged, and ventricular volume removed. As such, we will combine the approaches of the previous section alongside the introduction of the surface ‘union’ operation with the SVM-Tk.

We assume that the left pial, left white, right pial and right white surface stl files have been converted, using the process described in 4.1.1, and that the ventricle surface has been extracted; we assume the files are named as indicated in the code below. Furthermore, we assume that all of these files have been postprocessed to a satisfactory degree.
4.1 Combining regional data from separate surface files

```python
#!/usr/bin/python
import SVMTK as svm

# --- Options ---
pial_left_file = 'lh-ernie-pial.stl'
pial_right_file = 'rh-ernie-pial.stl'
white_left_file = 'lh-ernie-white.stl'
white_right_file = 'rh-ernie-white.stl'
ventricles_file = 'ernie-ventricles.stl'

#output
mesh_resolution = 32.0
output_file = 'ernie-fullbrain%d-novent.mesh' % int(mesh_resolution)

#keep (True) or discard (False) ventricles
keep_ventricles = False

tags
pialtag = 1
whitetag = 2
ventricletag = 3

# --- ---

#load surfaces
white_left = svm.Surface(white_left_file)
white_right = svm.Surface(white_right_file)
pial_left = svm.Surface(pial_left_file)
pial_right = svm.Surface(pial_right_file)
ventricles = svm.Surface(ventricles_file)

# union the white matter and rename
white_left.union(white_right)
fullwhite = white_left

# --- Subdomain Tagging ---
surfaces = [pial_left,pial_right,\n    fullwhite,ventricles]
```
smap = svm.SubdomainMap()

# left and right pial surface labels
smap.add("1000", pialtag)
smap.add("0100", pialtag)

# white matter on the left pial side
smap.add("1010", whitetag)

# white matter on the right pial side
smap.add("0110", whitetag)

# white matter at the intersection
# of left and right pial sides
smap.add("1110", whitetag)

# ventricles (left pial side)
smap.add("1011", ventricletag)

# ventricles (right pial side)
smap.add("0111", ventricletag)

# ventricles (at pial
# intersection)
smap.add("1111", ventricletag)

# Mesh generation
domain = svm.Domain(surfaces, smap)
domain.create_mesh(mesh_resolution)

if (keep_ventricles == False):
    domain.remove_subdomain(ventricletag)

domain.save(output_file)

We see that the code above uses familiar routines. One primary difference
is the use of the union(...) function of the SVM-Tk Surface object; consider
the following code snippet.
white_left = svm.Surface(white_left_file)
white_right= svm.Surface(white_right_file)

white_left.union(white_right)
fullwhite = white_left

Here we see that the union function of the white_left surface object is called to join the right white matter surface to the left white matter surface. The variable is then renamed, in the last line, to reflect this change. This approach has the advantage of making the upcoming calls to smap.add(...) a bit easier since it reduces the number of surfaces by one.

It is natural to ask whether we can do the same with the pial matter; thus further reducing the global surface number. Taking the union of the left and right pial surfaces is possible. However, in general, whether or not one can successfully mesh the joined surface, without further postprocessing, is patient specific. More specifically, there is a higher chance that the FreeSurfer segmentation process of the left and right pial MRI surface data can lead to non-physical intersections; thus producing left and right pial surface files that overlap and self-intersect when unioned together. Self intersections within a surface can then cause the SVM-Tk meshing process to fail. Certainly, one can try; the authors have seen cases where the pial surfaces can, indeed, be successfully unioned together just as we have done for the white matter, above. Likewise, one could simply work with all five surfaces separately; this would lead to a more complicated tagging process with the SubdomainMap but alleviate difficulties that may arise when computing the union two surface objects.

Finally, we see that the tagging procedure for the left pial, right pial, white matter and ventricle surfaces, taken together, requires several more calls to smap.add(...). The following lines, then, tag the entire gray matter volume.

smap.add("1000",pialtag)
smap.add("0100",pialtag)

To tag the white matter volume we must now distinguish between white matter on the left pial side, right pial side and boundary region. To do this we have the following tags.

smap.add("1010",whitetag)
smap.add("0110",whitetag)
smap.add("1110",whitetag)
4.2 A simple model of diffusion in separate brain regions

The goal of this section is to solve the model problem (4.0.1) using FEniCS on a computational domain where the gray and white matter have been tagged and the ventricles have been removed. We will do this by extracting a ventricle surface stl file, as discussed in chapter 4.1.2.1, and preprocessing the pial, white and ventricle surface stl files according to the techniques in chapter 3.1.5.1-3.1.5.2. We will then create a final volume mesh using the SVM-Tk functionality in ‘three-domain-tagged.py’, of the previous section, and then extend the simple diffusion code, originally given in chapter 3.2.2, to take into account the different constant diffusion values, $K_{\text{gray}}$ and $K_{\text{white}}$, in the gray and white matter domains of our model problem (4.0.1).
4.2 A simple model of diffusion in separate brain regions

4.2.1 Volume mesh generation with preprocessing

In this section we briefly overview the combined preprocessing procedure using the scripts of chapter 3.1.5.1-3.1.5.2. We demonstrate the process on the left hemisphere and comment that the same approach can be used on the right pial matter and right white matter surfaces to create a combined two-hemisphere mesh of the brain as in 4.1.3.

In chapter 4.1.1 we extracted the left pial surface, saved as lh-ernie-pial.stl, the left white matter surface, saved as lh-ernie-white.stl, and in chapter 4.1.2.1 we extracted a ventricular surface, saved as lh.ernie-ventricles.stl, from our MRI subject data. We assume that the ventricular surface stl file has been created using the postprocessed, smoothing and closure, iterations turned on as described in section 4.1.2.1.

We begin with the extracted pial surface, named lh-ernie-pial.stl, and use the python script remesh_surface.py, first described in chapter 3.1.5.1, with the following options set for the variables.

```python
## file: remesh_surface.py
input_file = 'lh-ernie-pial.stl'
output_file = 'ernie-pial-remesh.stl'

# --- Options --- #
number_of_iterations = 2
do_not_move_boundary_edges = True
edge_length_mm = 1.0
```

We note that since we are using several surface files together to create the final mesh we would like to maintain the original position of the surfaces, as closely as possible, in order to prevent overlap; overlap can lead to failure in the generation of the volume mesh. For this reason we set do_not_move_boundary_edges to True in the options above.

We then run the remesh_surface.py script again for the file lh-ernie-white.stl by changing the input and output files to

```python
## file: remesh_surface.py
input_file = 'lh-ernie-white.stl'
output_file = 'ernie-white-remesh.stl'
```

and once more for the ventricles with

```python
## file: remesh_surface.py
```
Since we are assuming that the ventricular surface was created using the smoothing operations described in chapter 4.1.2.1 we need only smooth the remeshed pial and white matter surfaces. To accomplish this we use the script smooth_surface.py created in chapter 3.1.5.2 with the following options; beginning with the pial surface.

```python
## file: smooth_surface.py
input_file = 'ernie-pial-remesh.stl'
output_file = 'ernie-gray-matter.stl'

# --- Options ---
number_of_iterations = 3
preserve_volume = True
smoothing_factor = 1.0
```

We run the smooth_surface.py script with the options above to create the smoothed gray matter surface. For the white matter we need to change the input and output file options so that the file looks like this

```python
## file: smooth_surface.py
input_file = 'ernie-white-remesh.stl'
output_file = 'ernie-white-matter.stl'
```

, and run the script again from the command line. Now we have obtained three surface files: ernie-gray-matter.stl, ernie-white-matter.stl and ernie-ventricles-remesh.stl. We can now create a mesh with the post-processed surface files with the script three-domain-tagged.py, from chapter 4.1.2.2, with the following options.

```python
## file: three-domain-tagged.py
# --- Options ---
gray_matter_filename = 'ernie-gray-matter.stl'
white_matter_filename = 'ernie-white-matter.stl'
ventricle_filename = 'ernie-ventricles-remesh.stl'

output.mesh.filename = 'ernie-chp4.mesh'

mesh.resolution = 64.0
```
Likewise, if we apply the above procedures to the right pial surface stl file and right white matter surface stl file then we can run the script `fullbrain-five-domain.py` to create a two-hemisphere computational mesh from the postprocessed files as in 4.1.3. As we have done previously we can convert the above file for visualization in paraview with the following command issued in a terminal window.

```
Terminal window
$ meshio-convert ernie-chp4.mesh ernie-chp4.vtu
```

![Fig. 4.8](image)

Fig. 4.8 Left hemisphere mesh with differentiated gray and white matter regions and without ventricles. Constructed from preprocessed surface stl files using the SVM-Tk

Two different clipping-plane views, sagittal (left) and coronal (right), of the final postprocessed mesh are shown in figure 4.8. We can clearly see that the preprocessed surface files have been used to successfully generate a mesh with tagged gray and white matter surfaces and with the ventricles removed. In general, we can follow the same postprocessing steps of this section on the right pial matter and right white matter surfaces to create a combined two-hemisphere mesh of the brain as in 4.1.3.
4.3 Subdomains in FEniCS

4.3.1 Converting Mesh Data to FEniCS

Constructing a mesh with multiple subdomains will generate boundaries between different ‘tags’, including "0" and each boundary will have an unique facet tag associated with it. In FEniCS, the facet tag can be used to set boundary conditions, like pressure, to simulate increased amounts of CSF inside the lateral ventricles. However the xml format can only handle either cell or facet data, but not both. Therefore, we will in this section go through the method of importing both cell data and facet data to FEniCS using meshio. We will need to use another mesh format compatible with FEniCS known as xdmf, which is one of the supported mesh formats in meshio. The mesh data can be converted to FEniCS with the following script.

```python
import meshio

msh = meshio.read("ernie-chp4.mesh")

meshio.write("ernie-chp4.xdmf",
             meshio.Mesh(points=msh.points,
                         cells={"tetra": msh.cells["tetra"]}))

meshio.write("ernie-chp4-cell.xdmf",
             meshio.Mesh( points=msh.points,
                         cells={"triangle": msh.cells["triangle"]},
                         cell_data={"triangle": {"patches":
                                        msh.cell_data["triangle"]['medit:ref']}}))

meshio.write("ernie-chp4-facet.xdmf",
             meshio.Mesh(points=msh.points,
                         cells={"tetra": msh.cells["tetra"]},
                         cell_data={"tetra": {"subdomains":
                                             msh.cell_data["tetra"]['medit:ref']}}))
```

In this script, we load the medit mesh data, and we chose to write the mesh to the xdmf format. This require that we specify the mesh points and cell structure, and if we want to add cell or facet data, we need to use the keyword argument cell_data. This script will produces three xdmf files, named
4.3 Subdomains in FEniCS

mesh.xdmf, mf.xdmf and cf.xdmf, which can be loaded in to FEniCS with the following script

```python
from dolfin import *

mesh = Mesh()

with XDMFFile("ernie–chap4–mesh.xdmf") as infile:
    infile.read(mesh)

cf = MeshFunction("size_t", mesh, mesh.topology().dim(), 0)

with XDMFFile("ernie–chap4–subdomains.xdmf") as infile:
    infile.read(cf, "subdomains")

mf = MeshFunction("size_t", mesh, mesh.topology().dim() - 1, 0)

with XDMFFile("ernie–chap4–boundaries.xdmf") as infile:
    infile.read(mf, "patches")

mf.array()[mf.array() > 100] = 0
```

It should be noted that the internal facet in the mesh are not marked in the medit mesh file. This will cause the internal facet to be tagged with a default value when written to the xdmf file format. This default value is significantly higher than 100, so we easily set these internal facet to zero with `mf.array()[mf.array() > 100] = 0`. 
References


13. Holter, Karl Erik, Kehlet, Benjamin, Devor, Anna, Sejnowski, Terrence J., Dale, Anders M., Omholt, Stig W., Ottersen, Ole Petter, Nagelhus, Erlend Arnulf,

Can diffusion alone explain brain-wide distribution of CSF tracers within 24 hours?

Can diffusion alone explain brain-wide distribution of a CSF tracer within 24 hours?

Lars Magnus Valnes¹, Sebastian K. Mitusch², Geir Ringstad³, Per Kristian Eide⁴⁵, Simon W. Funke², and Kent-Andre Mardal¹²*¹

¹Department of Mathematics, University of Oslo, Norway
²Center for Biomedical Computing, Simula Research Laboratory, Lysaker, Norway
³Division of Radiology and Nuclear Medicine, Department of Radiology - Rikshospitalet, Oslo, Norway
⁴Institute of Clinical Medicine, Faculty of Medicine, University of Oslo, Oslo, Norway
⁵Department of Neurosurgery, Oslo University Hospital – Rikshospitalet, Oslo, Norway
*kent-and@math.uio.no

ABSTRACT

The recently proposed glymphatic system suggests that bulk flow is important for clearing waste from the brain, and as such may underlie the development of e.g. Alzheimer’s disease. The glymphatic hypothesis is still controversial and several biomechanical modeling studies at the micro-level have at least partially dismissed the system and its assumptions. In contrast, at the macro-level, there are many experimental findings in support of bulk flow. Here, we will investigate to what extent the CSF tracer distributions seen in novel magnetic resonance imaging (MRI) investigations over hours and days are suggestive of bulk flow or diffusion. In order to include the complex geometry of the brain, the heterogeneous CSF flow around the brain, and the transport over the time-scale of days, we employed the methods of partial differential constrained optimization to identify the apparent diffusion coefficient (ADC) that would correspond best to the MRI findings. We found that the computed ADC in grey and white matter was respectively 23% and 82% larger than the ADC estimated with DTI, which suggests that diffusion may not be the only mechanism governing transport.

Introduction

Most types of dementia are associated with accumulation of metabolic by-products within the brain. In contrast to the rest of the body, the brain lacks a lymphatic system to clear these by-products. In 2012, a new pathway, called the paravascular pathway, was proposed¹, which enables efficient brain-wide circulation and clearance. The network of paravascular pathways in the brain was named the glymphatic system as it resembles the lymphatic system in the rest of the body, while the ‘g’ in glymphatic highlights the importance of the supportive glia cells in the brain. The paravascular pathways consists of cerebrospinal fluid flowing in parallel with the vasculature in paravascular spaces. These pathways have the potential to facilitate exchange between the cerebrospinal fluid (CSF) and the extracellular fluid deep within the brain.

To what extent and at what scale the glymphatic system accelerates transport compared to extracellular diffusion is still controversial, and several computational modeling studies have dismissed parts of the system at micro-scale. For example, the previous studies²³ suggest that diffusion dominates in the interstitium. Furthermore,⁴–⁶ have found that dispersion in the paravascular spaces adds less than a factor two to diffusion for solute transportation. However, multiple experimental and imaging findings at the micro-level point towards transport being different and faster than diffusion¹⁷⁸.¹

Investigation of the paravascular transport at macro-scale was proposed and tested in a rat’s brain⁹. The procedure involved injecting MRI contrast agent into the CSF and subsequently imaging the transportation of the MRI contrast agent at multiple time-points during a few hours after the injection. The MRI contrast agent worked as a CSF tracer, and was brain-wide in the rat after a few hours. The procedure was tested in humans for the first time in 2017¹⁰ with acquisition of MRI images repeatedly during 48 hours after the injection and later quantified in a region-specific manner in 2018¹¹ in individuals with dementia and controls. Overall, the MRI contrast agent transportation showed a centripetal pattern in all participants, but the MRI contrast agent was more protracted in individuals with dementia compared with controls¹¹. It was also noted that the CSF-tracer transport appeared faster than what can be expected from diffusion in simplified planar geometries.

On this background, our purpose in this paper is to explore whether the CSF tracer distribution seen in¹¹ can be explained by diffusion alone, as predicted by the seminal work of Syková and Nicholson¹². We will investigate this hypothesis with finite element simulations of the diffusion process combined with a parameter identification procedure for the apparent diffusion coefficients (ADCs). Thus, we aim to investigate whether we can assess ADC on long time-scales (hours or days), by fitting a diffusion model to the MRI data obtained at multiple time-points when the CSF tracer is propagating through the brain.
This includes taking into account the complexity of the folding brain surface by constructing a patient-specific geometry. We propose that if the fit between images and our model is good and we identify ADC values that are in line with those predicted by diffusion tensor imaging (DTI) then enhanced solute transport of the MRI contrast agent in question can be ignored. Our approach for the parameter identification is to solve an optimization problem constrained by a diffusion equation with unknown coefficients, where the optimization targets the observed CSF tracer concentrations at the 10 available acquisitions during 24 hours after CSF tracer injection.

An outline of the paper is as follows: In Section 1, we present the methodology of the paper. We start in Section 1.3 with a detailed description of the medical imaging methods relevant for this study. Section 1.4 describes the mathematical models, and the computational methodology for this paper. In Section 2, we will present the results of the study, starting with the MRI analysis in Section 2.1. We continue in Section 2.2 with a synthetic test case, which involves finding robust regularization parameters with a uniform distributed noise added to the images. The construction of the synthetic test case and the concentration estimation can be found in the Supplementary. While in Section 2.3, we present the computed ADC using the MRI images, and compare the values with ADC estimated with DTI. In Section 2.3.2, we present the results of different method to decrease the boundary noise. The results will facilitate the general discussion in Section 3.

1 Methods

1.1 Simulation workflow

An overview of the simulation workflow for this paper is outlined in Fig. 1. We obtained MRI data for one patient, which included MRI images with MRI contrast agent at different times (Box A in Fig. 1). The first MRI image was segmented and used to construct a patient specific mesh (Box B in Fig. 1). MRI images were used for estimating the CSF tracer concentration for the different times, and were subsequently sampled onto the patient specific mesh (Box C in Fig. 1). The sampled concentrations at the different times were then used with the mathematical model (Box D in Fig. 1). Values for numerical and regularization parameters were inputs for the computation (Box E in Fig. 1). The simulations produced the optimal ADC for grey and white matter to explain the observations for different input parameters (Box F in Fig. 1).

1.2 Approvals and MRI Acquisition

The approval for MRI observations was retrieved by the Regional Committee for Medical and Health Research Ethics (REK) of Health Region South-East, Norway (2015/96) and the Institutional Review Board of Oslo University Hospital (2015/1868) and the National Medicines Agency (15/04932-7). The study participants were included after written and oral informed consent. The MRI images included 3D T1-weighted volume, sagittal 3D FLAIR, DTI and T1 map for the same patients. All methods were performed in accordance with the relevant guidelines and regulations.

The contrast observations were obtained using a 3 Tesla Philips Ingenia MRI scanner (Philips Medical Systems) with the same imaging protocol settings at all time points to acquire sagittal 3D T1-weighted volume scans. The imaging parameters were as follows: repetition time, “shortest” (typically 5.1 ms); echo time, “shortest” (typically 2.3 ms); flip angle, 8 degrees; field of view, 256 × 256 cm; and matrix, 256 × 256 pixels (reconstructed 512 × 512). We also obtained observation a sagittal 3D FLAIR volume sequence of the same patient, that was taken before the injection of contrast. The main imaging parameters were: repetition time = 4,800 ms; echo time 318 ms; inversion recovery time, 1,650 ms; field of view, 250 × 250 mm; and matrix, 250 × 250 pixels (reconstructed 512 × 512). The T1 map was obtained with a MOLLIS(3)13 sequence with the following imaging parameters; repetition time 2.3 ms; echo time 1.0 ms; flip angle 20; field of view 257 × 257; and matrix 240 × 240 pixels. The DTI acquisition was done with the parameters; repetition time 12171 ms; echo time 60.0 ms; flip angle 90; field of view 240 × 240; and matrix 96 × 96 pixels.

1.3 MRI Analysis

A time sequence of T1-weighted MRI images showing the CSF- and brain enhancement in a patient diagnosed with NPH during 48 hours after intrathecal administration of Gadobutrol was obtained from a previous study11. The software FreeSurfer14–17 was used to segment and align each of the observations, which made it possible to estimate voxelwise signal increase. Figure 2 shows the distribution of MRI contrast agent in a selected region, as a percentage change in MRI signal unit ratios. The full data set used in this study (not all shown) consists of 10 MRI images, including a baseline MRI image taken before the contrast agent was injected. The MRI scans were obtained at different times distributed over 5 scans within 1-2 hours after injection, a single MRI scan at 2 hours, 6 hours, 24 hours and 48 hours. We segmented the baseline image with FreeSurfer and aligned the other images to the baseline. The exponential relation between the MRI signal values and the CSF tracer concentration, and the estimation of the concentration for each voxel is documented in the Supplementary 4.3. The estimation of the concentration produced images similar to the MRI images, but the values have the unit millimolar (mM). Therefore, we will denote the concentration images as observations to distinguish the concentrations from the MRI image intensities.
The segmentation process also produced polyhedral surfaces of the white and cortical grey matter that were used for mesh construction. We used the Computational Geometry Algorithms Library (CGAL)\textsuperscript{18} to combine the surfaces and construct the mesh with different subdomains. The computational requirement for the resulting mesh was significant, therefore two submeshes were also constructed, see Fig. 3. The three domain mesh (white matter, grey matter and CSF compartment), shown in Fig. 3 A, consists of 244,318 tetrahedral cells and 22,057 vertices, while the two domain mesh (without the CSF compartment), shown in Fig. 3 B, consists of 183,138 tetrahedral cells and 42,514 vertices.

1.4 Mathematical Model
The macroscopic extracellular diffusion in the brain can be considered a hindered diffusion with an apparent diffusion coefficient (ADC) depending on the structure of the extracellular space\textsuperscript{12}. The relation between the apparent and free diffusion coefficients is defined as

\[ \lambda = \sqrt{D/D_{ADC}} \]  \hspace{1cm} (1)

with \( \lambda \) denoting the tortuosity of the extracellular space. In order to estimate the ADC involved in the contrast transportation, shown in Fig. 2, we assume that the process can be modeled by a diffusion equation. Then, we constructed an optimization

\[ \text{Mathematical model} \]

\[ \frac{\partial \psi}{\partial t} - \nabla \cdot (\psi \nabla D \psi) = 0 \text{ in } \Omega \\
\psi = 0 \text{ on } \partial \Omega, \psi = 0 \text{ on } \Omega_{a} \text{ and } \psi = 0 \text{ on } \Omega_{b} \]

\[ \text{Observations} \]

\[ \text{Simulation Outcomes} \]

\[ \text{Grey matter} \]

\[ \text{ADC} \]

\[ \text{White matter} \]
Figure 2. The image shows the percentage change in T1 signal unit ratios from baseline at different observation times in the slice (marked red in the left panel) used in the subsequent analysis. The color-bar was restricted to the range (0, 100). The upper row shows the axial slice, and the bottom row shows the coronal slice.

Figure 3. The leftmost image A) shows the mesh created from the baseline MRI image with three domains, while the rightmost image B) shows the mesh created from the baseline MRI image with two domains. The blue domain corresponds to CSF domain, $\Omega_{CSF}$, the purple domain corresponds to grey matter, $\Omega_{GM}$, and the white domain corresponds to white matter, $\Omega_{WM}$. 

4/20
problem with the aim of minimize the difference between the observed and the modeled contrast distribution by optimizing the boundary conditions and the apparent diffusion coefficient. Thus enhanced transportation because of effects such as dispersion would result in an ADC larger than that predicted by DTI. The optimization problem was defined as

\[
\min_{\beta} \sum_{i=1}^{n} \int_{\Omega} [u(t_i) - u_{obs}(t_i)]^2 d\Omega + \int_0^T \int_{\partial\Omega_D} \left( \frac{\alpha}{2} |g|^2 + \frac{\beta}{2} \left| \frac{\partial g}{\partial t} \right|^2 + \frac{\gamma}{2} |\nabla g|^2 \right) d\Omega dt
\]

subject to

\[
\frac{\partial u}{\partial t} = D \nabla^2 u \quad \text{in} \quad \Omega \times (0, T) \\
u = g \quad \text{on} \quad \partial\Omega_D \times (0, T) \\
\frac{\partial u}{\partial n} = 0 \quad \text{on} \quad \partial\Omega_N \times (0, T)
\]

Here, \(u\) [mM] is the simulated, time-varying CSF tracer distribution, \(D\) [mm\(^2\)/s] is the ADC, \(g\) [mM] is the boundary condition, \(\Omega\) is the domain, and \(T\) [h] is the final simulation time. We assume that the domain \(\Omega\) consists of three sub domains, each with a different ADC. We denote the CSF (subarachnoid and lateral ventricle) domain as \(\Omega_{\text{CSF}}\), the grey matter as \(\Omega_{\text{GM}}\) and the white matter as \(\Omega_{\text{WM}}\). The ADC was assumed to be constant within the CSF, grey and white matter but each region may have different values. The \(\alpha\), \(\beta\) and \(\gamma\) parameters are non-negative regularization parameters and \(u_{obs}\) [mM] are the concentration distribution at time-points \(t_i\) [h]. Spacial regularization parameter \(\alpha\) enforces smoothness on the boundary by minimizing the concentration, i.e. high value of \(\alpha\) will give less concentration in the optimal solution. Temporal regularization parameter \(\beta\) enforces smoothness in time on the boundary, i.e. high value of \(\beta\) will give a smoother concentration curve in time. Gradient regularization parameter \(\gamma\) enforces continuity between adjacent concentrations at the boundary, i.e. high values of \(\gamma\) will give smoother concentration values at the boundary.

### 1.4.1 Boundary conditions.

For the three domain geometry (with grey and white matter, and CSF compartment), the Dirichlet boundary condition \(\Omega_D\) was only applied on the outward facing boundary of the CSF domain, \(\partial\Omega_{\text{CSF}}\). Homogeneous Neumann conditions \(\Omega_N\) were applied on the remaining boundaries.

The implementation of the gradient regularization \(\gamma\) for the case containing only grey and white matter required that the outward facing boundary was decomposed in different regions to avoid the boundary values being continuous at the interface between CSF, grey and white matter. We decomposed the boundary as seen in Fig. 4, with the red and blue boundary adjacent to the CSF. We defined the red boundary \(\partial\Omega_r\) and the blue boundary \(\partial\Omega_b\) as Dirichlet boundaries \(\Omega_D\), while the green and yellow were Neumann boundaries \(\Omega_N\). The regularization parameter \(\gamma\) was subsequently set to be non-zero in (2). We initially tested gradient regularization with the same parameter \(\gamma\) on both boundaries, but the different distribution of tracers on the boundaries made it difficult to find an adequate value for \(\gamma\). This may be attributed to the fact that the concentration in the lateral ventricles were more uniform than that in the SAS, so we defined \(\gamma\) as

\[
\gamma = \begin{cases} 
0.01 \hat{\gamma} & \text{in} \quad \partial\Omega_b \\
\hat{\gamma} & \text{in} \quad \partial\Omega_r
\end{cases}
\]

with \(\hat{\gamma}\) as the referenced, i.e. mentioned in the text, regularization parameter.

### 1.4.2 Synthetic test case with a manufactured solution.

In order to assess the robustness and accuracy of the methodology of ADC estimation via PDE constrained optimization we constructed a synthetic test case with a known, manufactured solution. The setup for the numerical tests can be found in the Supplementary 4.1. In the case of three domains we varied \(\alpha \in (10^{-6}, 10^{-2})\) and \(\beta \in (10^{-6}, 10^{2})\). In the case of two domains \(\alpha \in (10^{-6}, 1), \beta \in (10^{-4}, 10^{2})\) and \(\gamma \in (10^{-4}, 1.0)\).

We tested the noise susceptibility with a uniform distributed of noise. This was done by adding noise in the range of \((-\text{namp}, \text{namp})\) to the observation after loading, i.e. each vertex. We tested the noise with \(\text{namp} \in (0.03, 0.15, 0.30, 0.475)\), compared to the maximum values for the manufactured solution being in the range (0.3,1.3) for all times. We used the observation with the added noise to compute and confirm the expected SNR, since there was a finite number of vertices. The variation in the number of observations was tested with 5,10 and 20 evenly spaced observations over the course of 24 hours combined with 10 times steps, 20 time steps and 40 time steps.
1.5 CSF tracer distribution reconstruction from MRI data
The MRI data consisted of scans at times $t_i$ that were distributed in the 48 hours time frame, with the first observation with tracer occurring 1-2 hours after the injection. This was followed by 4 observations within the first hour, and we observed no visible change in the tracer distribution for these observations. Therefore, we used observation times listed in Tab. 1 for the computation of the ADC values.

1.5.1 Comparison with data obtained from DTI analysis
We compared the ADC computed by solving (2)-(3) with ADC values obtained from the DTI image of the same patient. We remark that a direct comparison with DTI is not possible because DTI measures the ADC of water. Therefore, we used the DTI to estimate the tortuosity in grey and white matter, which together with the free diffusion coefficient of Gadobutrol and (1) can be used to approximate ADC for Gadobutrol, details and references are found in the Supplementary 4.4.

1.5.2 Preprocessing of concentrations of Gadobutrol at ventricular and subarachnoid surfaces
High frequency concentration changes was observed at the boundaries of our mesh, see top row of column A in Fig. 6 and Fig. 7, which can be interpreted as sampling errors. Such errors may be caused by noise in the MRI data, errors in the segmentation, the segmented polyhedral surfaces which typically cut voxels, miss-alignment between different observations, and by the inaccuracy of sampling discontinuous voxel data. Therefore, we investigated two approaches to reduce high frequency components at mesh boundaries:

- A projection of the segmented CSF Gadobutrol concentrations directly at the ventricle and subarachnoid surfaces (CP),
- A Gaussian smoothing (GS) procedure.

In detail, the CP method was implemented by finding the voxel corresponding to each boundary vertex on $\Omega_b$ and $\Omega_e$ using an affine transformation matrix in FreeSurfer. Then, for each voxel, we computed the average of all surrounding voxels with CSF segmentation mark in a $7 \times 7 \times 7$ matrix. The average values were then used at the corresponding boundary vertex in the computations. In order to estimate the CSF concentration, we assumed that the T1 value corresponding to CSF was 3000 ms.

Finally, for the GS method we used the Gaussian smoothing function found in the python-module scipy$^{19}$, and applied the smoothing to all voxel in the observations estimated from MRI images. The standard deviation of the Gaussian distribution was set to 1.5 mm² compared to MRI voxel length of 1.0 mm. In addition to the CP and GS methods, we also use the raw data without any preprocessing. This method is referred to as RAW.

We performed the simulations with the regularization parameters $\alpha \in (10^{-6}, 10^{-5}), \beta \in (1.0, 10), \gamma \in (0.0, 0.01, 1.0)$. Additionally, we ran simulation with $\beta = 100$ and $\gamma = 100$ to obtain simulations that clearly showed differences in the concentration distributions on the boundary.
Table 1. The table shows an overview of the parameters used for boundary constrained optimization with MRI data.

<table>
<thead>
<tr>
<th>Description</th>
<th>Model variable</th>
<th>Value used in the simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of timesteps</td>
<td>$k$</td>
<td>24, 48</td>
</tr>
<tr>
<td>Observational time points</td>
<td>$\tau$</td>
<td>${0.0, 2.1, 6.1, 24.1, 47.8}$</td>
</tr>
<tr>
<td>Timestep</td>
<td>$dt$</td>
<td>1.2 hours</td>
</tr>
<tr>
<td>Spacial Regularization</td>
<td>$\alpha$</td>
<td>${10^{-6}, 10^{-2}}$</td>
</tr>
<tr>
<td>Time Regularization</td>
<td>$\beta$</td>
<td>${1.0, 10.0}$</td>
</tr>
<tr>
<td>Gradient Regularization</td>
<td>$\gamma$</td>
<td>${0.0, 0.01, 1.0}$</td>
</tr>
</tbody>
</table>

2 Results

2.1 MRI Analysis

Statistical analysis was done on the observations to determine the signal to noise ratio (SNR) for each observation. The first 5 observations had SNR in the range 0.6–0.8 in grey matter and 0.2–1.3 in white matter. For the remaining observations the SNR ranged 1.4–1.9 in grey matter and 1.3–2.6 in white matter.

2.2 Assessment of accuracy and robustness on a synthetic test case

For the first geometry, involving CSF, grey and white matter, we ran a series of 448 tests with different regularization parameters $\alpha$ and $\beta$ using the manufactured solutions every 2.4 hours over the course of 24 hours. It was found that for $\alpha \in (10^{-6}, 10^{-2})$ and $\beta \in (10^{-6}, 1.0)$, the error in the ADC for CSF, grey and white matter was less than 5%. For combinations with $\beta = 10^2$, the smallest error was 63.6% in the CSF, 10.8% in grey matter and 3.5% in white matter. While for $\alpha = 1.0$, the smallest error was 160.2% in the CSF, 23.2% in grey matter and 8.9% in white matter. Second, the robustness of the parameter identification process with respect to noise in the data was investigated. The noise values were randomly obtained from a uniform distribution in the range (-0.3,0.3), where 0.3 equaled the maximum initial value of the manufactured solution and was 23% of the manufactured solution at its max. Again, for $\alpha \in (10^{-6},10^{-2})$ and $\beta \in (10^{-6}, 1.0)$ the error in the ADC in CSF was less than 23% and less than 9.7% in grey and white matter.

For the second geometry, including only grey and white matter, we ran a selection of 186 tests to ensure that the results were consistent. The boundary conditions were applied to the boundaries of the SAS and lateral ventricle. In the range $\alpha \in (10^{-6}, 10^{-2})$ and $\beta \in (10^{-4},1.0)$, the error was less than 4.2% for the ADC in both grey and white matter.

The noise susceptibility was tested with the addition of the regularization parameter $\gamma \in (0.0, 10^{-2}, 1.0)$, which enforces smoothness at the boundary. We computed the error on the SAS boundary and on the lateral ventricle boundary, and it was observed that $\gamma$ did not contribute to a lower error in the ADC, but decreased the boundary error with a few percentage on average. In the range $\alpha \in (10^{-6}, 10^{-4})$ and $\beta \in (10^{-4}, 1.0)$, we had maximum error of 7.0% in the grey matter and 3.1% in the white matter with noise randomly obtained from the uniform distribution range of (-0.3,0.3). The computed the SNR of 0.74 for first time step, and increased to the maximum of 6.3.

The synthetic test case revealed that the second geometry, involving only grey and white matter was the most accurate method. Hence, in the following, only the second geometry was used.

2.3 CSF tracer distribution reconstruction from MRI data

In the synthetic case we managed to reproduce the ADC within 7% error for a wide range of parameters, $\alpha \in (10^{-6}, 10^{-2})$, $\beta \in (10^{-4},1.0)$, and $\gamma \in (0.0, 10^{-2}, 1.0)$. The same parameters were then used to compute the optimal ADC for the CSF tracer estimated from MRI images. A few of the reconstructions are shown in Fig. 5. It is clear that our model assumption of an underlying diffusion equation is adequate, that is; from a visual point of view the observed data in Fig 5, row A, is reconstructed accurately for the various different regularization parameters Fig 5, row B-D. We computed the average ADC over the range of regularization parameters to be $0.57 \pm 0.05 \text{mm}^2/\text{h}$ in grey matter and $0.72 \pm 0.02 \text{mm}^2/\text{h}$ in the white matter, which respectively corresponds to $1.6 \pm 0.2 \times 10^{-4} \text{mm}^2/\text{s}$ and $2.0 \pm 0.1 \times 10^{-4} \text{mm}^2/\text{s}$, shown in Fig. 8.

2.3.1 Comparison with data obtained from DTI analysis

The median ADC for water in the DTI was estimated to be $1.0 \pm 0.4 \times 10^{-3} \text{mm}^2/\text{s}$ in grey matter and $0.9 \pm 0.3 \times 10^{-3} \text{mm}^2/\text{s}$ in white matter resulting in a tortuosity of 1.73 and 1.85 based on (1). The free diffusion coefficient for Gadobutrol was approximated to be $3.8 \times 10^{-4} \text{mm}^2/\text{s}$ in the Supplementary 4.4. This gives an estimate median Gadobutrol ADC to be $1.3 \pm 0.5 \times 10^{-4} \text{mm}^2/\text{s}$ in the grey and $1.1 \pm 0.4 \times 10^{-4} \text{mm}^2/\text{s}$ in the white matter. This estimation assumed that the tortuosity was independent for molecules with mass lower than 1kDa. For Gadobutrol, this gives an average of 23 ±15% larger ADC in grey matter and 82±10% larger ADC in white matter compared to the associated ADC values based on DTI.
Figure 5. The image displays observations from MRI images and computational reconstruction of the same observations at observational time-points, i.e. 2 hours, 6 hours, 24 hours and 48 hours after the administration of CSF tracer. Row A) shows the observations estimated from MRI images. Row B) shows the reconstructed observations with $\alpha = 0.0001$, $\beta = 10.0$, $\gamma = 1.0$ and 48 time steps. Row C) shows reconstructed observations with $\alpha = 10^{-6}$, $\beta = 1.0$, $\gamma = 0.01$ and 48 time steps. Row D) shows reconstructed observations with $\alpha = 10^{-6}$, $\beta = 10.0$, $\gamma = 1.0$ and 48 time steps. The color-bar was restricted to the range (0.0-0.8).
2.3.2 Preprocessing of concentrations of Gadobutrol at ventricular and subarachnoid surfaces

The three different reconstructions with RAW, CP and GS methods are illustrated in Fig. 6 and Fig. 7 for some different regularization parameters. As can be seen in RAW column in Fig. 6 and Fig. 7 the boundary gradient regularization $\gamma$ caused the concentration to be more uniform on the boundary and in particular for the high value $\gamma = 100$. The CP method results in similar values for most of the ventricular boundary. The resulting ADC Gadobutrol values for RAW, CP and GS are shown in Fig. 8 together with the Gadobutrol ADC value estimated from DTI. We computed the percentage difference for each computed ADC and compared with ADC estimated from DTI. For the CP method, the average difference was $37 \pm 8\%$ in grey matter, $-5 \pm 11\%$ in white matter and for the GS method, the average difference was $250 \pm 22\%$ in grey matter, $68 \pm 18\%$ in white matter.

3 Discussion

The glymphatic system proposes that the paravascular network facilitates brain-wide transportation by bulk flow. Several mathematical modeling studies have been performed at the micro-level, but to our knowledge, our current study is the first study to investigate this process based on human imaging data at the macro-level. Our investigations are based on the distribution of Gadobutrol in CSF and brain tissue up to 48 hours after intrathecal injection. Our results here are supportive of the glymphatic system in the sense that the ADC estimated from the Gadobutrol distribution is higher than corresponding numbers obtained from DTI. Hence, our study suggests that at longer time scales, the slow glymphatic system has an impact.

We assumed that the white matter was isotropic, however, it is well-known that the white matter is anisotropic. The main reason for our choice was that introducing anisotropy would result in a model with many more free variables, requiring substantial parameter tuning, and consequently result in less predictability. Furthermore, it can be seen in Fig. 5 that the reconstruction is quite good. Demanding that the transport in the CSF was governed by a diffusion process yielded a much larger error in the synthetic test case and this assumption should probably be avoided.

In our approach we have employed the FreeSurfer toolkit to segment and register grey and white matter surfaces. FreeSurfer provides segmentations with sub-voxel accuracy, which means that at the boundaries towards the ventricles and subarachnoid space, the surface boundary is usually in the interior of a voxel rather than at the voxel boundaries. The consequence is that the raw data appear to have noisy image intensities at the boundaries which cut voxels. For this reason, we investigated two different approaches to interpret the data at the boundaries, in addition to using the raw data itself. We observed in Fig. 8 that the GS method increased the computed ADC with approximate $250 \pm 22\%$ in grey matter, while the CP method corresponded best with $-5 \pm 11\%$ difference to the ADC estimated with DTI in white matter. However, it should be noted that the CP method imposes
Figure 7. The images shows the ventricular wall after 12 hours. The upper row shows the observations with following preprocessing left to right: Raw observations, projection of CSF value onto the boundary, Gaussian smoothing. The middle row shows the corresponding states with the regularization parameters $\alpha, \beta, \gamma = (10^{-6}, 1.0, 1.0)$. The bottom row shows the corresponding states with the regularization parameters $\alpha, \beta, \gamma = (10^{-6}, 100.0, 100)$. 

Figure 8. The images shows the average ADC estimated with DTI and the computed ADC in grey and white matter for different sampling methods and regularization parameters $\alpha \in (10^{-6}, 10^{-4}), \beta \in (1.0, 10.0), \gamma \in (0.0, 0.01, 1.0)$ and number of time steps $k \in (24, 48)$. The error bars show the standard deviation.
concentration values in the CSF onto the tissue boundaries, not accounting for the transverse propagation of MRI contrast agent through a membrane, like the pia mater or the endothelial layer of the ventricles. Hence, potentially, an unnaturally large concentration gradient at the boundary that is caused by a partial barrier rather than the white matter itself may have resulted in the low ADC seen when using this method. Which of the methods that best depicted the actual boundary concentration is unknown and would need to be determined by phantom studies. Another limitation is that we have not yet been able to assess the diffusivity of Gadobutrol and have have relied on literature values for the free diffusion coefficient of a similar mass molecule as a substitute for Gadobutrol. However, it should be noted that even though the concentration values appear noisy at the boundaries, Fig. 6, the interior reconstructions appear accurate, Fig. 5.

Concerning the imaging, a current limitation is that standard T1 weighted MRI images have higher resolution and SNR than corresponding T1-maps. In detail, the intrathecal contrast enhanced T1 weighted volume scans had 1.0 mm slice thickness, while the T1-map slice thickness was 4.0 mm. This means that the calculation of the concentration at boundaries can suffer from mismatch of tissue and CSF. Furthermore, the T1-map sequence is designed to estimate the T1 times in tissue, and therefore does not give accurate values for the CSF. The average T1 relaxation time for the CSF in left lateral ventricle was 181 ± 349 ms, compared to the value 1000-5500 ms that can be found in the literature and we used the literature values to compute the concentration in the CSF. We also estimated the T1 relaxation time for grey and white matter to be 1200±271 and 819±180 ms, which compares better with the literature values ranging 1470-1800 ms in grey and 1084-1110 ms in white matter.

The ADC of water in this study was measured with DTI to be 1.0 ± 0.4 × 10⁻³ mm²/s in grey matter and 0.9 ± 0.3 × 10⁻³ mm²/s in white matter. It has been reported that ADC of extracellular water in young and healthy subjects was 0.78 – 1.09 × 10⁻³ in the cortical grey matter and 0.7 – 0.9 × 10⁻³ mm²/s in white matter. Although our values match with the DTI values of this study, the white matter seem to be on the upper threshold. This can be explained by the fact that subjects with dementia typically have higher ADC values, less anisotropy and greater variation in the white matter. Thus, the DTI values found in this study seems to correspond with the literature values. Based on the raw data, our estimated Gadobutrol ADC was 23 ± 10% and 82 ± 4% larger in the grey and white matter, respectively, than what the DTI data corresponds to for Gadobutrol.

Previous mathematical modeling studies at the micro-level suggest that diffusion dominates in the interstitium. However, diffusion depends on molecular size as described by the Stokes-Einstein equations and large molecules are transported slower than small molecules. Convective flow of solutes, on the other hand, is independent of the molecular size. Furthermore, transport has been reported to be independent of molecular size, a fact that suggests convective transport. In fact, convective velocities of 0.8 – 4 × 10⁻³ mm/s have been demonstrated or estimated, indicating that the solute transport would be dominated by convection for large molecules, whereas similar to diffusion for smaller molecules, such as water. Gadobutrol is in this context a molecule of moderate size, i.e., 604 Da, and hence not ideally suited for the study of bulk flow for larger molecules such as Aβ (4.5 kDa) or CSF-τ (45 kDa). In fact, Gadobutrol is predicted to have a Peclet number less than one which from these micro-level studies would imply that the distribution of Gadobutrol is governed mainly by diffusion.

Studies have found that dispersion in the paravascular spaces adds less than a factor two to diffusion for solute transportation. However, all these studies were done with modeling that was on the micro-scale over shorter time periods. To the authors’ knowledge, the only other study that has considered macroscopic modeling on the time-scales of hours and days, where uncertainties representing both variations in ADC and paravascular velocities where modelled with extensive testing using Monte Carlo methods. They found that, in particular, the CSF tracer distribution within the deep white matter found in could not be explained by diffusion alone. Recent high-resolution MRI imaging of paravascular spaces in a rat’s brain points towards significant contributions from white matter paravascular spaces connected to the ventricles. It should also be mentioned that the permeability of the white matter is several orders of magnitude higher than the grey matter, and may as such be more susceptible to bulk flow than grey matter.

In conclusion, we computed that the ADC in grey and white matter and found that in both cases the ADC was somewhat larger than estimates based on DTI alone. Thus, indicating that there is potential for enhanced solute transportation in the brain over a longer time period. There are, however, a number of uncertainties that needs to be taken into account, for instance the resolution of DTI and T1 mapping.

Author contributions statement
L.M.V, P.K.E, G.R., KAM conceived the experiments. L.M.V., S.K.M, S.F. implemented the simulators. L.M.V. conducted the experiments and made the figures. All authors discussed and analyzed the results. L.M.V, K.A.M wrote the first draft. All authors revised the manuscript and approved the final manuscript.

Additional information
Competing interests The authors declare no competing interest.
Acknowledgement

The simulations were run on the Abel Cluster (Project NN9279 K), owned by the University of Oslo and the Norwegian Metacenter for High-Performance Computing and operated by the Department for Research Computing at University Center for Information Technology, the University of Oslo Technical Department, www.hpc.uio.no/.

Corresponding Author

The corresponding author’s e-mail is kent-and@math.uio.no.

Data Availability

The datasets analyzed in the current study are available from the corresponding author upon request.

References

5. Sharp, M. K., Carare, R. & Martin, B. Dispersion in porous media in oscillatory flow between flat plates: Applications to intrathecal, periarterial and paraarterial solute transport in the central nervous system. J. Fluid Mech. (Accepted)).


4 Supplementary

4.1 Constructing the synthetic solutions to assess accuracy and robustness.

Below we will discuss the parameter identification and its sensitivity with respect to the regularization parameters, noise, number of observations and time-resolution of the forward model. The manufactured observations used in (2) were obtained by forward computation of (3) with the Dirichlet boundary condition defined as
\[
g(t) = 0.3 + 0.167t - 0.007t^2 \quad \text{ for } 0 \leq t \leq 24.
\]

The initial condition was set to 0 everywhere, the time step was \( dt = 0.24 \), and the diffusion coefficients were selected to be
\[
D_{\Omega_1} = 1000.0, \quad D_{\Omega_2} = 4.0, \quad D_{\Omega_3} = 8.0
\]

4.2 Additional results: Assessment of accuracy and robustness on a synthetic test case

We will in this section present additional results that was not presented in the main text, Sec. 2.2. The convergence was monitored for a wide range of parameters, some shown in Figure 13, and we saw the similar convergence for ADC and boundary conditions for \( \alpha \in (10^{-6}, 10^{-4}) \) and \( \beta \in (10^{-4}, 1) \). Furthermore, the method was found robust with respect to the noise as illustrated in the Figure 9.

The variation in the number of observations was tested with 5, 10 and 20 evenly spaced observations over the course of 24 hours. This was done together with 10 time steps, 20 time steps and 40 time steps. Illustrative results are shown in Figure 12, which shows that a high number of time steps compared to observations causes oscillations (e.g. purple line) at the boundary for \( \alpha \gg \beta \). This is counteracted by selecting high values of the temporal regularization \( \beta \) relatively to \( \alpha \) hereby enforcing smoothness (e.g. blue line). For example, in the geometry with grey and white matter, it was observed that the ADC error increased by a factor \( \approx 4 \) for \( \alpha \sim \beta \) relative to the case where \( \beta \gg \alpha \) for the case with 20 time steps and 10 observations.

4.3 MRI contrast agent concentration - image signal relation

Below, we briefly describe the relationship between the imaging signal seen in Figure 2 and the underlying MRI contrast agent concentration. We remark that we use notations that are common in medical literature, which includes the use of two letter symbols. Hence, we will also use two letter symbols, such as \( T_E \) and \( T_R \), to keep the notation consistent with the presentation in\[33, 34\]. The MRI contrast agent concentration \( c \) causes the longitudinal (spin-lattice) relaxation time \( T_1 \) to shorten with the following relation
\[
\frac{1}{T_1^c} = \frac{1}{T_1^0} + r_1 c.
\]

The superscripts indicate relaxation time with MRI contrast agent \( T_1^c \) and without MRI contrast agent \( T_1^0 \), while \( r_1 \) is the relaxivity constant for the MRI contrast agent in a medium. The MRI observations were obtained using a MRI sequence known as Magnetization Prepared Rapid Acquisition Gradient Echo (MPRAGE) with an inversion prepared magnetization. The relation between the signal and the relaxation time is non-linear, and is expressed with the following equations. The MRI signal value \( S \) for this sequence can be expressed as
\[
S = M_n \sin \theta e^{-TE/T_2^*},
\]

with \( T_E \) and \( \theta \) respectively denoting the echo time and the flip angle, and \( M_n \) is the magnetization for the n-echo that we described below. Also \( T_2^* \) is the transverse magnetization caused by a combination of spin-spin relaxation and magnetic field inhomogeneity, defined as
\[
\frac{1}{T_2^*} = \frac{1}{T_2} + \gamma \Delta B_{in}.
\]

Here \( T_2 \) is the transverse (spin-spin) relaxation time, \( \gamma \) is the gyromagnetic ratio and \( \Delta B_{in} \) is the magnetic field inhomogeneity across a voxel. The expression (8) can be simplified by neglecting the exponential term, since \( T_E \ll T_2^* \) is a general trait for this MRI sequence. Thus, (8) becomes
\[
S = M_n \sin \theta.
\]

Magnetization for the n-echo \( M_n \) is defined as\[33\]:
\[
M_n = M_0 \left[ (1 - \beta) \frac{1 - (\alpha \beta)^{n-1}}{1 - \alpha \beta} + (\alpha \beta)^{n-1} (1 - \gamma) + \gamma (\alpha \beta)^{n-1} \frac{M_e}{M_0} \right].
\]
with
\[
\frac{M_e}{M_0} = -\left[ 1 - \delta + \alpha \delta (1 - \beta) \frac{1 - \alpha^m}{1 + \rho \alpha^m} + \alpha \delta (\alpha \beta)^{m-1} - \alpha^m \rho \right].
\] (12)

Using the following definitions
\[
\begin{align*}
\alpha &= \cos(\theta) \\
\beta &= e^{-T_b/T_i} \\
\delta &= e^{-T_a/T_i} \\
\gamma &= e^{-T_w/T_i} \\
\rho &= e^{-T_R/T_i} \\
T_w &= T_R - T_a - T_b(m - 1).
\end{align*}
\] (13)

Here \(T_b\) is known as the echo spacing time, \(T_a\) is the inversion time, \(T_w\) the time delay, \(T_R\) as the repetition time, \(m\) is the number of echoes and \(M_0\) is a calibration constant for the magnetization. The center echo denoted as \(n = m/2\) will be the signal that we will consider when estimating concentration of the MRI contrast agent. Given (10), the relative signal increase can be written as
\[
\frac{S_c}{S_0} = \frac{M_e n \sin(\theta)}{M_0 n \sin(\theta)}.\] (14)

We define that
\[
f(T_1) = \frac{M_e n}{M_0}.
\] (15)

Figure 10 shows \(f(T_1)\) in CSF, grey and white matter. This gives the following relation
\[
\frac{f(T_1^c)}{f(T_1^0)} = \frac{S^c}{S^0}.
\] (16)

The signal difference between observation times were adjusted in\(^\text{11}\). Thus we can express the change in \(T_1\) due to MRI contrast agent as
\[
f(T_1) = \frac{S^c}{S^0} f(T_1^0)
\] (17)

and then estimate the concentration using (7). The \(T_1^0\) values were obtained by T1 mapping of the brain using a MRI sequence known as MOLLI5(3)\(^\text{13}\). This takes into account patient specific characteristic, such as tissue damage. Tissue damage can be observed in the MRI due to a lower signal in the white matter compared to healthy white matter tissue, thus damaged tissue have different \(T_1\) relaxation time. The MRI contrast agent concentration was estimated in a preprocessing step, using the parameters obtained from the T1-map, MPRAGE MRI protocol\(^\text{11}\) and the value for \(r_1\) found in\(^\text{35}\). The values of the function (15) was computed for \(T_1 \in (200, 4000)\) creating a lookup table. The lookup table was utilized with the baseline signal increase to estimate \(T_1^c\), and then the concentration was computed using (7).

4.4 Diffusion tensor imaging

DTI images provide apparent diffusion coefficients (ADC) for water molecules (18Da) on short time-scales. A processed DTI image is shown in Figure 11 with the largest ADC (shown in red in the middle figure) to be around 1.3e–3mm\(^2\)/s.

In order to compare the computed diffusion coefficient, we need to estimate the ADC for Gadovist (604 Da)\(^\text{36}\) by utilizing (1) together with the tortuosity and the free diffusion coefficient for Gadovist. The tortuosity in the white and grey matter was estimated using (1) with the ADC obtained from DTI and the self-diffusion of water which is \(3.0 \times 10^{-3}\)mm\(^2\)/s at 37\(^o\)C\(^\text{37}\). We used the free diffusion coefficient for Gd-DPTA (550 Da)\(^\text{38}\), which was reported in\(^\text{39}\) to be \(3.8 \times 10^{-4}\)mm\(^2\)/s, as a surrogate for Gadovist. This is based on the Stokes-Einstein equation, which states that molecules with similar weight will have similar
diffusion coefficients\textsuperscript{24}. Furthermore, we need to assume that the tortuosity is independent for molecules with mass lower than 1kDa, i.e. not large proteins. The fractional anisotropy is defined as

\[ FA^2 = \frac{3}{2} \left( \frac{(\lambda_1 - MD)^2 + (\lambda_2 - MD)^2 + (\lambda_3 - MD)^2}{\lambda_1^2 + \lambda_2^2 + \lambda_3^2} \right), \]  

(18)

with the mean diffusivity \( MD \), closely related to ADC, defined as

\[ MD = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3}. \]  

(19)

In these equations \( \lambda_i \) denotes the eigenvalues of the diffusion tensor. \( FA \) is a common measure of anisotropy and shown in the right-most image in Figure 11 for our patient.
Figure 9. The upper row shows the manufactured observation. A) Shows the manufactured observation at time-point 24 with no noise added. B) Shows the manufactured observation after 12 hours with noise range of (-0.15,0.15). C) Shows the manufactured observation after 12 hours with noise range of (-0.3,0.3). The lower row shows the results with optimized parameter obtained with $\alpha = 0.0001$, $\beta = 1.0$ and $k = 20$. D) Shows the resulting state given the observation in A. E) Shows the resulting state given the observation in B. F) Shows the resulting state given the observation in C.
Figure 10. The image shows the function defined in (15) where the white region indicates $T_1$ values for white matter, the grey region indicates $T_1$ values for grey matter, the blue region indicates $T_1$ values for CSF.
Figure 11. The left panel shows the anatomical map. The middle panel shows the apparent diffusion coefficients (ADC) obtained from DTI. The right panel shows the computed fractional anisotropy (FA) from the DTI at a color coded scale.

Figure 12. The image displays plots over time for a selection of points at the boundary of $\Omega_1$ with different regularization parameters and number of time steps $k$. The left panel shows the legend for the plot over time, together with the selection of points. The middle panel shows the average boundary value $g$ for different regularization parameter with $k = 10$. The right panel shows the average boundary value $g$ for different regularization parameter with $k = 40$. 
Figure 13. Convergence plots of the diffusion coefficients, boundary conditions and functional (2) with respect to different $\alpha$ and $\beta$ values.