

# Diffusion MRI analysis

## Citation for published version (APA):

Harms, R. L. (2019). *Diffusion MRI analysis: robust and efficient microstructure modeling*. ProefschriftMaken. <https://doi.org/10.26481/dis.20191025rh>

## Document status and date:

Published: 01/01/2019

## DOI:

[10.26481/dis.20191025rh](https://doi.org/10.26481/dis.20191025rh)

## Document Version:

Publisher's PDF, also known as Version of record

## Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

## General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.umlib.nl/taverne-license](http://www.umlib.nl/taverne-license)

## Take down policy

If you believe that this document breaches copyright please contact us at:

[repository@maastrichtuniversity.nl](mailto:repository@maastrichtuniversity.nl)

providing details and we will investigate your claim.

Dissertation

# **Diffusion MRI Analysis**

**Robust and efficient microstructure modeling**

Robbert Harms

2019

© Robbert Harms, Maastricht 2019.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior written permission of the author.

This book was typeset by the author using  $\LaTeX$ .

Cover	Friedemann Vervoort, 2019
Production	ProefschriftMaken, Maastricht
ISBN	978-94-6380-543-8

# **Diffusion MRI Analysis**

## **Robust and efficient microstructure modeling**

Dissertation

To obtain the degree of Doctor at Maastricht University,  
on the authority of the Rector Magnificus, Prof. Dr. Rianne M. Letschert,  
in accordance with the decision of the Board of Deans,  
to be defended in public  
on Friday 25th of Oktober 2019, at 10.00 hours

by

Robbert Leonard Harms



**Promotor**

Prof. Dr. Rainer W. Goebel

**Copromotor**

Dr. Alard Roebroek

**Assessment Committee**

Prof. Dr. Peter de Weerd (Chair)

Prof. Dr. Frank Thuijsman

Dr. Giancarlo Valente

Prof. Dr. Tony Stöcker

*German Center for Neurodegenerative Diseases, Bonn, Germany*

Dr. Alexander L. G. Leemans

*University Medical Center Utrecht, the Netherlands*

*To my parents*



# Contents

1	Introduction	3
1.1	The human brain . . . . .	5
1.2	Diffusion MRI . . . . .	6
1.3	Microstructure modeling . . . . .	9
1.4	This thesis . . . . .	12
2	Robust and fast nonlinear optimization of diffusion MRI microstructure models	15
2.1	Introduction . . . . .	17
2.2	Methods . . . . .	18
2.3	Results . . . . .	31
2.4	Discussion . . . . .	40
3	Robust and fast Markov Chain Monte Carlo sampling of diffusion MRI microstructure models	47
3.1	Introduction . . . . .	49
3.2	Methods . . . . .	51
3.3	Results . . . . .	66
3.4	Discussion . . . . .	77
4	Fast quantification of uncertainty in non-linear diffusion MRI models	81
4.1	Introduction . . . . .	83
4.2	Methods . . . . .	84
4.3	Results . . . . .	92
4.4	Discussion . . . . .	103
5	MDT, the Microstructure Diffusion Toolbox	109
5.1	Introduction . . . . .	111
5.2	Design concepts . . . . .	112
5.3	Interfaces to MDT . . . . .	123
5.4	Application examples . . . . .	130
5.5	Discussion . . . . .	134
6	Summary and Discussion	137
6.1	Summary . . . . .	138
6.2	Discussion . . . . .	140

## Contents

---

### Appendices

A	Appendix to Chapter 2	147
A.1	Parameter transformations . . . . .	147
A.2	Cascaded model initializations . . . . .	149
B	Appendix to Chapter 4	151
B.1	Numerical Hessian . . . . .	151
B.2	Uncertainty propagation . . . . .	152
	Bibliography	155
	Valorisation	169
	Acknowledgments	173
	Research output	179
	About the author	183





# 1

## Introduction



## **Abstract**

Diffusion MRI allows for in vivo investigation of the microstructural organization of the human brain. Clinicians use this to diagnose certain brain pathologies, while scientist use it to further a structural understanding of the human brain. Inferring the microstructure from a diffusion MRI acquisition requires modeling, as the raw MRI images are not directly interpretable. There are multiple microstructure models available, each highlighting different characteristics of brain tissue. To infer information from the diffusion MRI signals, most of these models depend on various parameter estimation methods. In this thesis we analyze those parameter estimation methods.

## 1.1 The human brain

The human brain is one of the most complex systems available for study. This highly detailed assembly of about 100 billion neurons (Herculano-Houzel, 2009) is what allows us to feel, think and act. The relationship between the brain's functioning and its structure is still unclear. On a high level, we can divide the brain in gray matter and in white matter. The gray matter is where most of the information processing occurs and has specialized regions for processes like sensory perception, decision making and muscle control. White matter primarily acts as a relay between gray matter regions, but has also been shown to play an active role in memory and learning (Fields, 2008).

Besides blood vessels and cerebral spinal fluid, brain tissue consists of neurons and glial cells (astrocytes, oligodendrocytes, and microglia), partly illustrated in figure 1.1. Gray matter contains the cell bodies of the neurons and glial cells, as well as neuronal dendrites, short-range intra-cortical axons and the origins of long-range axons extending into the white matter (Alexander, Dyrby, et al., 2017). White matter contains glial cell bodies but consists mostly of densely packed and often myelinated axons which connect different regions of gray matter to each other (Alexander, Dyrby, et al., 2017).

The human brain consists for approximately 73% of water (Mitchel et al., 1945). Most of this water is compartmentalized by axons, glial cells and myelin, sometimes referred to as intra-cellular water. The remaining water, outside these compartments is often referred to as the extra-cellular water. The division of water between the multiple different water pockets are typically referred to as the volume fractions. Using ex vivo brains it is estimated that the water volume fraction of the extra-cellular space in white matter is between 15 and 35%, that of the water inside the axons ~30-40%, myelin contains ~20-30% and glial cell bodies water volume fraction is ~15-20% (Syková and Nicholson, 2008; Perge et al., 2009).

The microstructural organization of white matter is unique to every person and changes over time. Knowing how the brain is wired at any point in time could, in the future, facilitate a synthesis of brain structure with brain functioning (Filley, 2005).

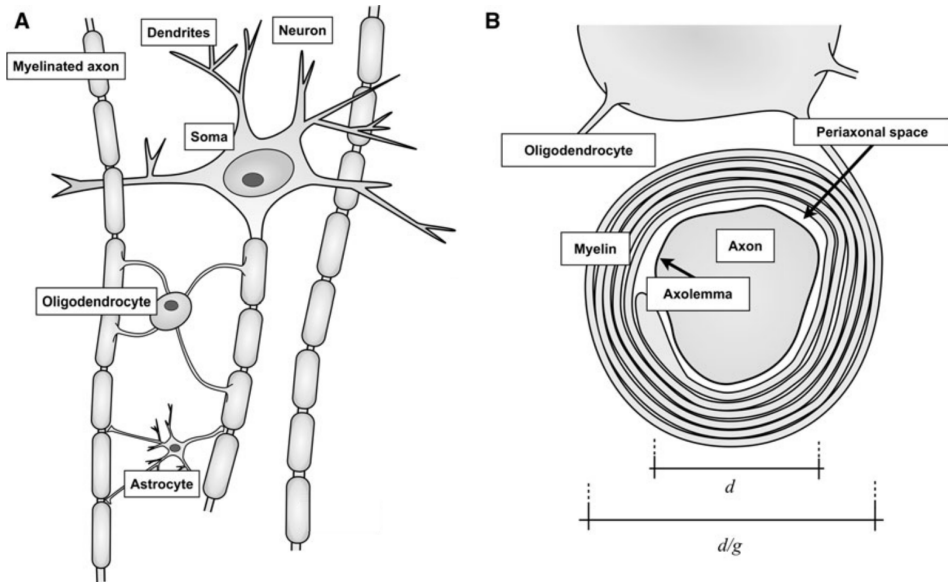


Figure 1.1: Schematic representation of the major cellular elements of the white matter, modified from (Nilsson et al., 2013). In A, the cell body of a neuron, the soma, is typically located in gray matter. Several short dendrites and one long axon originate from the soma. Most of the axons in white matter are wrapped by myelin sheaths. These myelin sheaths are extensions of oligodendrocytes, which form myelin sheaths around several axons. In B, myelinated axon with oligodendrocyte. The small space between the axolemma and the inner part of the myelin sheath is called the periaxonal space and is approximately 15 nm wide and filled with extracellular fluid. The axon diameter is given by  $d$  and the total axon diameter including the myelin is given by  $D = d/g$  with  $g$  the g-ratio.

## 1.2 Diffusion MRI

Diffusion Magnetic Resonance Imaging (dMRI) is one of the preferred approaches for investigating the brain's white matter microstructure *in vivo*. Compared to other imaging modalities like Computerized Tomography (CT) and conventional MRI, dMRI allows for far greater detail in the observation of white matter microstructure. For instance, dMRI was the first non-invasive imaging modality capable of determining fiber tract orientations *in vivo* (Basser, Mattiello, and LeBihan, 1994). The importance of dMRI has also been recognized in the clinical discipline where dMRI was the first neuroimaging modality capable of detecting acute ischemia (lack of oxygen) in the human brain (Gelderen et al., 1994).

Diffusion MRI measures the microscopic displacement of water molecules. All particles in a fluid move randomly due to thermal energy, a property commonly referred to as Brownian motion. In white matter brain tissue, this Brownian motion is hindered and restricted by the tissue's microstructure. For instance, wa-

ter inside axons is restricted by the axolemma, while water outside the axons is hindered by the various cellular structures. The MRI signal intensity is directly influenced by the propagated distance of the water molecules, making diffusion MRI sensitive to restrictions imposed by the microstructure.

A typical diffusion MRI records the brain in voxels of cubic millimeters. Since a single axon has a diameter of about one micrometer, a single voxel typically represents multiple thousands of axons (Walhovd, Johansen-Berg, and Káradóttir, 2014). This makes the signal measured by a diffusion acquisition a statistical average of the signal emanating from at least multiple thousands of axons. In a broader perspective, the signal intensity of a single diffusion measurement is a statistical average over all the water molecules in that voxel, modulated by the restrictions of the tissue microstructure.

Since a single diffusion MRI volume measures water displacement in only one direction over a single time interval, a typical dMRI study consists of multiple diffusion volumes, each sensitive to a different displacement orientation with, optionally, varying diffusion times. Given a constant diffusion time, a series of diffusion volumes with different displacement orientations allow us to construct a three dimensional water displacement distribution, illustrated in figure 1.2. Since this displacement distribution changes with respect to the diffusion time interval, time is typically included in the mathematical description of diffusion. With time included, the four dimensional diffusion displacement distribution is referred to as the diffusion propagator  $P(\mathbf{r}, t)$  which defines the probability of moving distance  $\mathbf{r}$  over time  $t$  (Callaghan, 1993). Since the microstructure affects the signal, this four dimensional distribution is an indirect representation of the microstructure in a voxel.

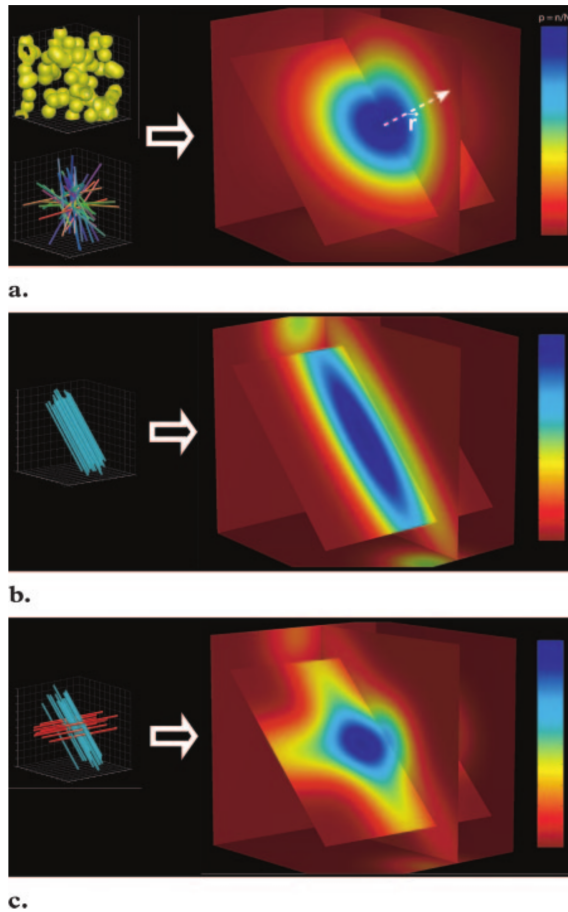


Figure 1.2: From tissue organization to a displacement distribution, illustrated for three different voxels. Figure reproduced from (Hagmann et al., 2006) with permission from the Radiological Society of North America. In each of the three cases (a, b and c) we have a hypothetical tissue organization on the left and a diffusion probability density distribution in the middle. The displacement probabilities are color coded from red (low probability) to blue (high probability). In (a), the 3d diffusion probability density distribution for a voxel that contains either spherical cells (top left) or randomly oriented tubular shapes which intersect, like axons (bottom left). The resulting probability density function is approximately uniform in all directions as there is no preferential direction of diffusion. The center of the image (at the origin of the  $r$  vector) indicate the proportion of water molecules for which the net displacement is zero in the diffusion time interval. In (b), the diffusion probability distribution for a voxel with all axons aligned in a single direction. The resulting probability distribution is cigar shaped and aligned with the axons. In (c), the diffusion probability distribution for a voxel which has two bundles of axons crossing each other. The resulting probability distribution is approximately cross shaped.

## 1.3 Microstructure modeling

The signal obtained from diffusion MRI provides only an indirect view of the underlying cellular structures. This makes modeling essential in order to extract quantitative measures from diffusion MRI data. Microstructure modeling generally refers to the practice of estimating parameters from the data in order to derive useful features about the tissue microstructure. Common features researchers are interested in include fibre orientation vectors, axonal radii and intra-axonal water volume estimates.

Two complementary modeling approaches have been proposed in the literature: signal representations and biophysical models (Jelescu and Budde, 2017; Novikov, Kiselev, and Jespersen, 2018). Both of these approaches are capable of explaining the diffusion MRI signal, but each with their respective advantages and disadvantages.

### 1.3.1 Signal representations

Signal representations aim to describe the diffusion MRI data without prior assumptions on the underlying microstructure. They are often very sensitive to small changes in the signal, but the estimated parameters typically lack specificity regarding the underlying microstructure (Jelescu and Budde, 2017). The assumption free nature makes the signal representations broadly applicable as no precautions have to be made to ensure the model suits the tissue type. Typical features include diffusion orientation estimates and diffusion isotropy and anisotropy quantification.

One of the first diffusion MRI models, the diffusion Tensor (Basser, Mattiello, and LeBihan, 1994), falls in this class of signal representations. The diffusion Tensor is build around a first order Taylor expansion of the diffusion signal and models the signal as if it were a three dimensional Gaussian distribution. That is, the diffusion Tensor defines the diffusion propagator as:

$$P(\mathbf{r}, t) = \frac{1}{\sqrt{4\pi t^3 |\mathbf{D}|}} \exp\left(\frac{-\mathbf{r}^\top \mathbf{D}^{-1} \mathbf{r}}{4t}\right) \quad (1.1)$$

with  $\mathbf{r}$  the measured water displacement vector,  $t$  the known diffusion time and  $\mathbf{D}$  a  $3 \times 3$  positive definite symmetric matrix called the diffusion Tensor. Whereas the terms  $\mathbf{r}$  and  $t$  are known, the diffusion Tensor  $\mathbf{D}$  needs to be estimated. Often derived features from the Tensor are the Mean Diffusivity (MD) and the Fractional Anisotropy (FA) which are respectively the mean of the eigenvalues and the normalized variance of the eigenvalues of the  $\mathbf{D}$  matrix (Basser and Pierpaoli,

1996). Since the diffusion Tensor is widely used, we included it in most of the comparisons in this thesis.

The diffusion Tensor was later extended to the Kurtosis model (Jensen et al., 2005) to include an estimate of the excess kurtosis in the diffusion displacement probability distribution. Other signal representation techniques include using cumulant expansions (Yablonskiy, Bretthorst, and Ackerman, 2003), using stretched exponentials (Bennett et al., 2003), or techniques like Q-Ball imaging (Tuch, 2004), spherical deconvolution (J.-D. Tournier et al., 2004) and diffusion spectrum imaging (Wedeen et al., 2005). We do not use these signal representations in this thesis, except for the Tensor model as it is a wide-spread measure. Instead, we focus on the biophysical models as they promise a higher specificity than the signal representations, and similar to (Novikov, Kiselev, and Jespersen, 2018), we consider specificity a more promising goal of microstructure mapping.

### 1.3.2 Biophysical models

Biophysical models are based on prior assumptions about the expected microstructure. That is, they provide an analytical expression based on a geometrical view of the assumed tissue microstructure. By relating the model parameters directly to relevant microstructural quantities, microstructure models promise greater specificity over the signal representations in describing the white matter tissue (Jelescu and Budde, 2017). However, this increased specificity comes at the cost of applicability since the biophysical models are only valid if the model can accurately describe the data. Typical features modeled by biophysical models include axonal orientation estimates, intra-axonal diffusivity, axonal radii, neurite dispersion estimates and volume fraction estimates.

Biophysical models typically follow the multi-compartment modeling paradigm. In this paradigm, the signal response of a biophysical model is constructed as a weighted sum of one or more compartment models (Behrens et al., 2003; Panagiotaki et al., 2012). These multi-compartment models typically have a single compartment for the extra-axonal water volume (which here includes glial and myelin cellular water as diffusion MRI is generally not sensitive enough to distinguish these), a single compartment for modeling the cerebral spinal fluid (CSF) regions and one or more compartments for the intra-axonal water volume (depending on how many fibre populations are assumed). The weights between the different compartments are often interpreted as water volume fraction estimates. See figure 1.3 for an illustration of a three compartment model.

The biophysical models selected in this thesis fall in the class of the multi-compartment models, the Ball&Stick (Behrens et al., 2003), NODDI (H. Zhang, Schneider, et al., 2012) and CHARMED (Assaf, Freidlin, et al., 2004) models. The Ball&Stick model (Behrens et al., 2003) was the first

multi-compartment model and is often used as local estimator for tractography. The intra-axonal compartment of the Ball&Stick model assumes that axons can be represented as infinitely thin cylinders with fixed length, "Sticks". The extra-axonal fluids are modeled as isotropic "Ball" compartments. The Ball&Stick model is suitable for finding the principle diffusion direction, which is supposed to correlate with the underlying orientation of the axons. The NODDI model extends the Ball&Stick model by assuming not a single axon, but a multitude of axons which can be more or less dispersed. This dispersion allows the NODDI model to capture not only the main axon orientation but also the neurite orientation dispersion. The CHARMED model takes a slightly different approach and models not orientation dispersion but assumes that the cylinders have a radius instead of being infinitely thin.

Other biophysical multi-compartment models include Bingham-NODDI (Tariq et al., 2016), NODDIDA (Jelescu, Veraart, Adisetiyo, et al., 2015), White Matter Tract Integrity model (WMTI) (Fieremans et al., 2013), AxCaliber (Assaf, Blumenfeld-Katzir, et al., 2008), the Minimal Model of White Matter Diffusion (MMWMD in ActiveAx) (Alexander, Hubbard, et al., 2010; H. Zhang, Hubbard, et al., 2011) and diffusion time dependent CHARMED (De Santis, D. K. Jones, and Roebroek, 2016). Although all of these models are included in the software used in this thesis, we considered our selection of Ball&Stick, NODDI and CHARMED a generalizable sample of microstructure models, sufficient for the comparisons in this thesis.

### 1.3.3 Parameter estimation

Both the signal representations and the biophysical models have in common the need to be fitted to the data. Considering modeling specificity a promising goal, we decided to focus primarily on the biophysical models. These models typically require non-linear optimization routines to fit the model parameters, which often faces difficulties in robustness and efficiency. Although we focus primarily on the non-linear analysis methods, we take no fundamental stance in the microstructure modeling debate. Rather, we hope to improve the analysis methods such that the debate can be conducted with more accurate results.



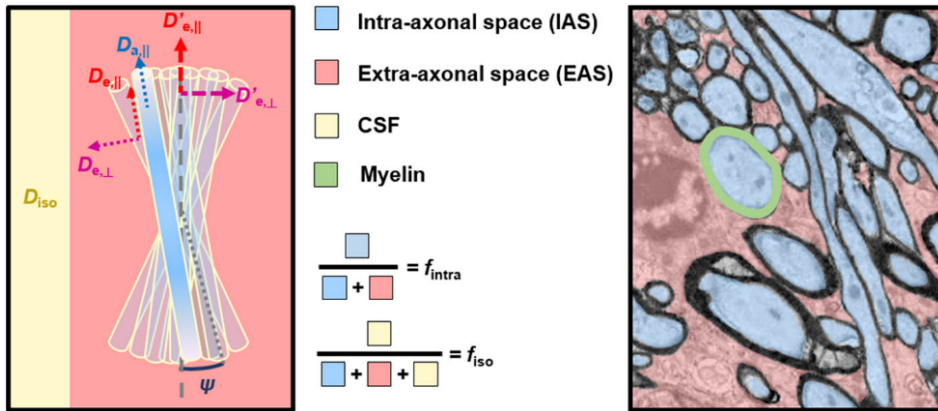


Figure 1.3: Illustration the correspondence between model compartments on the left and tissue compartments on the right. Figure reproduced from (Jelescu and Budde, 2017). Left, a schematic overview of a typical three-compartment model with the relevant parameters  $D_{e,||}$  and  $D_{e,\perp}$  for the local extra-axonal diffusivities,  $D'_{e,||}$  and  $D'_{e,\perp}$  for the apparent extra-axonal diffusivities,  $D_{a,||}$  for the intra-axonal diffusivity and  $D_{iso}$  for the isotropic water diffusivity. Right, a cross-sectional electron microscopy image of a white matter bundle. In microstructure modeling, the diffusivities and the intra-axonal orientation direction  $\psi$  are free parameters which are typically optimized to best match the diffusion signal. Although myelin is present in white matter, it is often not included in a microstructure model since it is difficult to measure in current diffusion MRI acquisitions.

## 1.4 This thesis

The primary aim of this thesis is to improve the robustness and efficiency of the methods used in diffusion MRI microstructure modeling. A secondary aim is to provide the community with software to utilize the findings of this research.

This thesis is structured as follows. **Chapter 2** discusses two major aspects of microstructure model fitting using maximum likelihood estimation, the choice of optimization routine and the choice of initialization strategy. Several optimization routines are compared to each other based on computation run time and their quality of fit, using multiple microstructure models and multiple datasets. Next, multiple initialization strategies are compared to each other based on their effect on the optimization results. The aim of this chapter is to identify a generally applicable optimization strategy with a low run time and a high quality of fit.

As an alternative to maximum likelihood estimation, **Chapter 3** discusses the use of Markov Chain Monte Carlo (MCMC) in microstructure modeling. Similar to the previous chapter, the first part of this chapter compares several MCMC routines, aiming to identify an optimal and generally applicable MCMC method. The second part discusses several MCMC related methodological settings like burn-in, thinning and the number of samples. For the latter we use the concept

of effective sample size to quantify the number of MCMC samples required for the estimate of the posterior to reach a predetermined precision.

**Chapter 4** proposes the Fisher Information Matrix (FIM) as an alternative to MCMC for computing the uncertainties in parameter estimates. After comparing the FIM and MCMC on applicability and processing speed, this chapter investigates several data and model characteristics that can influence the parameter uncertainties. Chapter 4 concludes with potential uses of the uncertainties for artifact detection and increasing the statistical power of group studies.

For our secondary research aim we developed the Microstructure Diffusion Toolbox (MDT). This software package is dedicated to MRI microstructure modeling and includes all models and methods described in this thesis. All routines in MDT have been implemented to run on graphical processors, thereby aiming to reduce the computational run time. **Chapter 5** contains an overview of the design concepts, user interfaces and usage examples of MDT.

This thesis concludes in **Chapter 6** with a discussion of the main contributions, the limitations of this work and an outlook for future research.



# 2

## Robust and fast nonlinear optimization of diffusion MRI microstructure models

---

Adapted from: R. L. Harms, F. J. Fritz, A. Tobisch, R. Goebel, and A. Roebroeck (July 2017). “Robust and fast nonlinear optimization of diffusion MRI microstructure models”. In: *NeuroImage* 155.October 2016, pp. 82–96. DOI: 10.1016/j.neuroimage.2017.04.064.

## Abstract

Advances in biophysical multi-compartment modeling for diffusion MRI (dMRI) have gained popularity because of greater specificity than DTI in relating the dMRI signal to underlying cellular microstructure. A large range of these diffusion microstructure models have been developed and each of the popular models comes with its own, often different, optimization algorithm, noise model and initialization strategy to estimate its parameter maps. Since data fit, accuracy and precision is hard to verify, this creates additional challenges to comparability and generalization of results from diffusion microstructure models. In addition, non-linear optimization is computationally expensive leading to very long run times, which can be prohibitive in large group or population studies. In this technical note we investigate the performance of several optimization algorithms and initialization strategies over a few of the most popular diffusion microstructure models, including NODDI and CHARMED. We evaluate whether a single well performing optimization approach exists that could be applied to many models and would equate both run time and fit aspects. All models, algorithms and strategies were implemented on the Graphics Processing Unit (GPU) to remove run time constraints, with which we achieve whole brain dataset fits in seconds to minutes. We then evaluated fit, accuracy, precision and run time for different models of differing complexity against three common optimization algorithms and three parameter initialization strategies. Variability of the achieved quality of fit in actual data was evaluated on ten subjects of each of two population studies with a different acquisition protocol. We find that optimization algorithms and multi-step optimization approaches have a considerable influence on performance and stability over subjects and over acquisition protocols. The gradient-free Powell conjugate-direction algorithm was found to outperform other common algorithms in terms of run time, fit, accuracy and precision. Parameter initialization approaches were found to be relevant especially for more complex models, such as those involving several fiber orientations per voxel. For these, a fitting cascade initializing or fixing parameter values in a later optimization step from simpler models in an earlier optimization step further improved run time, fit, accuracy and precision compared to a single step fit. This establishes and makes available standards by which robust fit and accuracy can be achieved in shorter run times. This is especially relevant for the use of diffusion microstructure modeling in large group or population studies and in combining microstructure parameter maps with tractography results.

## 2.1 Introduction

Diffusion MRI (dMRI) is a tool for investigating the microstructure of biological tissue by probing the self-diffusion of water (Le Bihan et al., 1986). The conventional method for the analysis of white matter in dMRI imaging is the tensor model in Diffusion Tensor Imaging (DTI) (Basser, Mattiello, and LeBihan, 1994). DTI has shown to be sensitive to microstructural changes due to, for example, development (Pfefferbaum et al., 2000; Neil et al., 2002; Assaf and Pasternak, 2008; Lebel, Caverhill-Godkewitsch, and Beaulieu, 2010) and pathology (Werring et al., 1999; Horsfield and D. K. Jones, 2002; Sotak, 2002; Sundgren et al., 2004). Although sensitive, DTI indices such as Fractional Anisotropy (FA) are also unspecific, since differences in FA can reflect different axonal properties such as axon density, diameter distribution and myelination (Beaulieu, 2002; Assaf, Freidlin, et al., 2004; Assaf, Blumenfeld-Katzir, et al., 2008; D. K. Jones, Knösche, and Turner, 2013; De Santis, Assaf, et al., 2014).

Recently, advances in biophysical multi-compartment modeling have gained popularity because they possess greater specificity than DTI in relating the dMRI signal to the underlying cellular microstructure (Assaf, Freidlin, et al., 2004; Assaf and Basser, 2005; Assaf, Blumenfeld-Katzir, et al., 2008; Alexander, Hubbard, et al., 2010; Panagiotaki et al., 2012; H. Zhang, Schneider, et al., 2012; Assaf, Alexander, et al., 2013; Fieremans et al., 2013; De Santis, Drakesmith, et al., 2014; De Santis, Assaf, et al., 2014; Jelescu, Veraart, Fieremans, et al., 2015). In these diffusion microstructure models the diffusion weighted signal is expressed as a combination of one or more biophysically inspired compartments. Although the compartment models are based on simple geometric shapes, and a strong assumption of no inter-compartment exchange is made (Nilsson et al., 2013; Li et al., 2017), these models can provide specific measures such as fiber density, orientation dispersion, and axonal diameter distributions and have been shown to be sensitive to specific white matter alterations due to development e.g. (Kunz et al., 2014; Jelescu, Veraart, Fieremans, et al., 2015) and pathology (Bergers et al., 2002; Fieremans et al., 2013; Benitez et al., 2014; Timmers et al., 2015; Wen et al., 2015; Kamagata et al., 2016).

There is a large range of diffusion microstructure models, including popular models such as Neurite Orientation Dispersion and Density Imaging model (NODDI) (H. Zhang, Schneider, et al., 2012), the Combined Hindered And Restricted Model of Diffusion (CHARMED) (Assaf and Basser, 2005), the White Matter Tract Integrity model (WMTI) (Fieremans et al., 2013), AxCaliber (Assaf, Blumenfeld-Katzir, et al., 2008), the Minimal Model of White Matter Diffusion (MMWMD in ActiveAx) (Alexander, Hubbard, et al., 2010; H. Zhang, Hubbard, et al., 2011), including further developments of these models like NODDIDA (Jelescu, Veraart, Adisetiyo, et al., 2015), Bingham NODDI (Tariq et al., 2016), diffusion time dependent CHARMED (De Santis, D. K. Jones, and Roebroek,

2016) and fiber-specific T1 CHARMED (Santis, Assaf, Jeurissen, et al., 2016). Each of these models needs to be fitted to the dMRI data to estimate parameter maps, which is commonly accomplished using non-linear optimization. This has two major challenges. First, it is computationally expensive, quickly leading to very long run times. Second, the quality, accuracy and precision of the data fit is often uncertain. To tackle these challenges, each of the popular models comes with its own, mostly different, optimization algorithm, noise model and initialization strategy to estimate its parameter maps. This creates challenges to comparability and generalization of results from diffusion microstructure models, additional to the model formulation itself.

In this technical note we investigate the performance of several optimization algorithms and initialization strategies over a few of the more popular diffusion microstructure models. We investigate whether a single well performing approach exists that could be applied to many models and that equates both run time and fit aspects. To this end, we evaluate the fit, accuracy, precision and run time for different models of differing complexity against three common optimization algorithms and three parameter initialization strategies.

All models, algorithms and strategies were implemented on the GPU to remove run time constraints. Variability of the achieved quality of fit in actual data is evaluated on ten subjects of each of two large group studies with a different acquisition protocol, the MGH-USC part of the Human Connectome Project (Fan et al., 2016) and the Rhineland Study ([www.rheinland-studie.de](http://www.rheinland-studie.de)).

## 2.2 Methods

All biophysical compartment models, noise models / likelihood functions, as well as optimization algorithms were implemented in a python based GPU accelerated toolbox (Maastricht Diffusion Toolbox or MDT, freely available under an open source L-GPL license at <https://github.com/cbclab/MDT>). Its object oriented modular design allows arbitrary combinations of individual single compartment models into composite compartment models, which can then be combined with a chosen likelihood function and optimization algorithm by Python scripting. The complete multi-compartment model, likelihood function and optimization algorithm is automatically compiled into OpenCL code executable on both CPUs and GPUs (or combinations thereof).

### 2.2.1 Single compartment models

Table 2.1 defines the individual compartment models which are combined to construct the multi-compartment models c.f. (Panagiotaki et al., 2012; Ferizi,

Schneider, Tariq, et al., 2013) valid for (singly refocused) Pulsed Gradient Spin Echo (PGSE) (Stepišnik, 1993) acquisitions. The Tensor model was first described in (Basser, Mattiello, and LeBihan, 1994), the Ball and Stick models are defined in (Behrens et al., 2003), NODDI<sub>in</sub> and NODDI<sub>ex</sub> are respectively the intra cellular and extra cellular models in (H. Zhang, Schneider, et al., 2012) and the CHARMED<sub>in</sub> compartment is the restricted compartment defined in (Assaf, Freidlin, et al., 2004).

Each compartment has a signal function modeling the signal  $\mathbf{S}$  depending on the unit norm gradient direction vector  $\mathbf{g}$  with scalar  $b = (\Delta - \delta/3)(\gamma\delta G)^2$  and the vector  $\mathbf{q} = \gamma\delta\mathbf{g}G/2\pi$ . Here  $G$  is the gradient amplitude,  $\Delta$  the time between the start of the two gradient pulses,  $\delta$  the duration of the gradient pulse and  $\gamma$  the gyromagnetic ratio for  $^1\text{H}$  in  $\text{rad s}^{-1} \text{T}^{-1}$ . Additionally, the CHARMED<sub>in</sub> compartments depend on the echo time  $\text{TE}$ .

Disregarding fixing of parameter values discussed below, the modelled fiber orientations  $\mathbf{n}$  are optimized as spherical coordinates using the free parameters  $\theta$  and  $\phi$  with scalar  $d$  for the diffusivity. Some models have more than one diffusivity, these are all optimized separately unless specified otherwise in the multi-compartment model. In the Tensor compartment the `rotate` function first rotates the fiber directions  $\mathbf{n}$  by 90 degrees in the  $(x, z)$  plane to make it perpendicular to the principal direction. The Tensor is then rotated around  $\mathbf{n}$  by the angle  $\psi$  resulting in  $\mathbf{n}_{\perp 1}$ , and, after a cross product  $\mathbf{n}_{\perp 2}$ . In the NODDI<sub>in</sub> and NODDI<sub>ex</sub> models, the function  $f(\mathbf{n}, \kappa)d\mathbf{n}$  gives the probability of finding fiber bundles along orientation  $\mathbf{n}$  using a Watson distribution integrated over the unit sphere  $\mathbb{S}^2$ . In the NODDI<sub>ex</sub> model, the diffusion tensor  $D(\mathbf{n})$  is defined as a cylindrically symmetric tensor. For more details on NODDI see (H. Zhang, Schneider, et al., 2012). The CHARMED<sub>in</sub> model is a Gamma Distributed Radii (GDR) Cylinder model (Panagiotaki et al., 2012) where the cylinder distribution is predefined with  $N$  cylinders (we take  $N = 6$  throughout). Our 6 cylinders have radii  $R$  of [1.5, 2.5, 3.5, 4.5, 5.5, 6.5] micrometers and corresponding weights [0.0212, 0.1072, 0.1944 0.2667, 0.2150, 0.1956] taken from a Gamma distribution derived from histological results (Aboitiz et al., 1992).



Compartment	Signal function	Compartment model parameters
Tensor	$\mathbf{S} = \exp(-b( d_{\parallel}(\mathbf{n} \cdot \mathbf{g})^2 + d_{\perp 1}(\mathbf{n}_{\perp 1} \cdot \mathbf{g})^2 + d_{\perp 2}(\mathbf{n}_{\perp 2} \cdot \mathbf{g})^2))$ $\mathbf{n}_{\perp 1} = \text{rotate}(\mathbf{n}, \psi)$ $\mathbf{n}_{\perp 2} = \mathbf{n} \times \mathbf{n}_{\perp 1}$	$d_{\parallel}, d_{\perp 1}, d_{\perp 2}, \theta, \phi, \psi$
Ball	$\mathbf{S} = e^{-bd}$	$d$
Stick	$\mathbf{S} = e^{-bd(\mathbf{n} \cdot \mathbf{g})^2}$	$d, \theta, \phi$
NODDI <sub>lin</sub>	$\mathbf{S} = \int_{\mathbb{S}^2} f(\mathbf{n}, \kappa) e^{-bd(\mathbf{n} \cdot \mathbf{g})^2} d\mathbf{n}$	$d, \theta, \phi, \kappa$
NODDI <sub>ex</sub>	$\mathbf{S} = e^{-b\mathbf{g}^T (\int_{\mathbb{S}^2} f(\mathbf{n}, \kappa) D(\mathbf{n}) d\mathbf{n}) \mathbf{g}}$	$d_{\parallel}, d_{\perp}, \theta, \phi, \kappa$
CHARMED <sub>in</sub>	$\mathbf{S} = \sum_{i=1}^N v_i [ S_{\parallel}(\mathbf{q}, \Delta) \cdot S_{\perp i}(\mathbf{q}, \text{TE}) ]$ $S_{\parallel}(\mathbf{q}, \Delta) = e^{-4\pi^2  \mathbf{q} ^2 (\mathbf{n} \cdot \mathbf{g})^2 (\Delta - \delta/3) d}$ $S_{\perp i}(\mathbf{q}, \text{TE}) = e^{-[4\pi^2  \mathbf{q} ^2 (1 - (\mathbf{n} \cdot \mathbf{g})^2) R_i^4 / (\frac{d \cdot \text{TE}}{2})] \cdot (\frac{7}{96}) \cdot [2 - (\frac{99}{112}) R_i^2 / (\frac{d \cdot \text{TE}}{2})]}$	$d, \theta, \phi$

Table 2.1: The single compartment models, see Table 2.2 for an overview of the optimizable parameters. The primary direction of diffusivity  $\mathbf{n}$ , is parameterized using polar coordinates with angles  $\theta, \phi$  and radius  $d$ . The variables  $b, \mathbf{g}, \mathbf{q}, \Delta, \delta, G$  and  $\text{TE}$  are sequence settings. In the Tensor compartment, the function  $\text{rotate}(n, \psi)$  rotates the Tensor around  $\mathbf{n}$  by the angle  $\psi$ . In the NODDI models, the function  $f(\mathbf{n}, \kappa) d\mathbf{n}$  gives the probability of finding fiber bundles along orientation  $\mathbf{n}$  using a Watson distribution with parameter  $\kappa$  integrated over the unit sphere  $\mathbb{S}^2$ . In the NODDI<sub>ex</sub> model, the diffusion tensor  $D(\mathbf{n})$  is defined as a cylindrically symmetric Tensor (alike the Tensor in this table except for the symmetry). In the CHARMED<sub>in</sub> compartment  $N$  is the number of gamma cylinders used,  $v_i$  is the weight per gamma distributed cylinders and  $R_i$  is the radius per cylinder. In previous work  $|\mathbf{q}|^2 (\mathbf{n} \cdot \mathbf{g})^2$  is sometimes denoted as  $|\mathbf{q}_{\parallel}|^2$  and  $|\mathbf{q}|^2 (1 - (\mathbf{n} \cdot \mathbf{g})^2)$  as  $|\mathbf{q}_{\perp}|^2$ , we inlined these identities here in the CHARMED<sub>in</sub> equation.

Parameter	Compartments	Usage
$d_{\parallel}$ (or $d$ )	Tensor, Ball, Stick, NODDI.in, NODDI.ex, CHARMED.in	Parallel diffusivity along the primary direction of diffusion $\mathbf{n}$
$d_{\perp_1}$	Tensor	Perpendicular diffusivity, perpendicular to both $d_{\parallel}$ and $d_{\perp_2}$ .
$d_{\perp_2}$	Tensor	Perpendicular diffusivity, perpendicular to both $d_{\parallel}$ and $d_{\perp_1}$ .
$\theta$	Tensor, Stick, NODDI.in, NODDI.ex, CHARMED.in	Polar angle used to parameterize $\mathbf{n}$ , the primary direction of diffusion
$\phi$	Tensor, Stick, NODDI.in, NODDI.ex, CHARMED.in	Azimuth angle used to parameterize $\mathbf{n}$ , the primary direction of diffusion
$\psi$	Tensor	Used to rotate the Tensor around its primary axis
$\kappa$	NODDI.in, NODDI.ex	The dispersion index of the Watson distribution

Table 2.2: The parameter descriptions corresponding to Table 2.1.

## 2.2.2 Composite multi-compartment models

The general multi-compartment diffusion microstructure model has the form of a weighted sum of single compartments:

$$S = S_0 \sum_{i=0}^n w_i \mathbf{S}_i \quad (2.1)$$

Where  $S_0$  is the signal for the non-diffusion weighted (or  $b_0$ ) acquisitions,  $w_i$  the volume fractions (signal weights, signal fractions or water fractions) and  $\mathbf{S}_i$  is the signal function for the  $i$ 'th of  $n$  total compartments. For this work we selected the Tensor, Ball&Sticks, NODDI and CHARMED models. Table 2.3 shows these multi-compartment models (henceforth simply 'models'), their constituent compartments and total number of parameters including estimation of  $S_0$ .

Model	Restricted ( <i>intra-cellular</i> ) compartments	Hindered ( <i>extra-cellular</i> ) compartments	Isotropic compartments	Number of parameters	Acquisition requirements
Tensor	-	Tensor	-	7	$b < 1.5 \cdot 10^6 \text{ s/m}^2$
Ball&Sticks_in[ $n$ ]	Stick ( $n$ -times)	-	Ball	$1 + 3n$	-
NODDI	NODDI_in	NODDI_ex	Ball	6	$\geq 2$ b-values / shells
CHARMED_in[ $n$ ]	CHARMED_in ( $n$ -times)	Tensor	-	$7 + 4n$	$\geq 2$ b-values / shells, $b_{\max} \geq 4.0 \cdot 10^6 \text{ s/m}^2$

Table 2.3: The used composite multi-compartment models, their compartments (divided into intra-, extra-axonal and isotropic) and total number of parameters.

Here we take the Stick compartment to be an idealized cylinder with zero radius and classify it as a restricted compartment (cf. (Ferizi, Schneider, Panagiotaki, et al., 2014)). We have added to some of the models the postfix '\_in[ $n$ ]' which is used to identify the number of restricted compartments employed in a model for those models that allow multiple restricted compartments. For example, CHARMED\_in2 indicates a CHARMED model with 2 restricted compartments (and the reg-

ular single hindered compartment) for each of two unique fiber orientations in a voxel. The total number of parameters per model is the sum of the parameters per compartment, the volume fractions (equal to the number of compartments minus one) and the scalar for  $S_0$ . The complete signal equation per model is then given as a volume fraction weighted sum of compartments, scaled by the non-diffusion weighted signal  $S_0$ .

### 2.2.3 Evaluation function and likelihood

Models are optimized by finding the set of free parameter values  $\mathbf{x} \in \mathbb{R}^n$  that minimize the evaluation function or objective function of the modeling errors ( $\mathbf{O} - \mathbf{S}(\mathbf{x})$ ) with  $\mathbf{O}$  the observed data and  $\mathbf{S}(\mathbf{x})$  the model signal estimate. In general, the evaluation function is formulated as the negative log likelihood function which embeds a noise model for the data. Whereas the noise in the complex valued Fourier coefficient images can be described by a Gaussian distribution, the noise in the magnitude reconstructed MR data follows a non-zero mean (Rician or non-central  $\chi$ ) distribution (Gudbjartsson and Patz, 1995). For  $\text{SNR} > 2$  an Offset Gaussian model can be used (Alexander, 2009), with the advantage that it is potentially more stable than the Rician model (Panagiotaki et al., 2012). Therefore, we use the Offset Gaussian likelihood model for all optimizations. Minimizing the negative log likelihood leads to the following objective function (with constant terms dropped):

$$\bar{\mathbf{x}} = \underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} \left( \frac{\sum \left( \mathbf{O} - \sqrt{\mathbf{S}(\mathbf{x})^2 + \sigma^2} \right)^2}{2\sigma^2} \right) \quad (2.2)$$

Here we use  $\sigma$  as the standard deviation of the Gaussian distributed error of the complex valued Fourier images. We estimated this standard deviation  $\sigma$  from the reconstructed magnitude images using the  $\sigma_{\text{mult}}$  method in Dietrich et al. (2007; eq. A6) (Dietrich, Raya, Reeder, Reiser, et al., 2007).

### 2.2.4 Parameter dependencies and global fixes

Some of the models have specific dependencies between their parameters, as well as parameter fixed to a specific value. Defining the dependencies at the level of the multi-compartment model has two advantages. First, this allows the compartments to be defined as general as possible and adapt them for the use in several multi-compartment models. Second, some dependencies relate the parameters of different compartments to each other and therefore must operate at

the level of the multi-compartment model. The list of global fixes and dependencies is given by Table 2.4 where we use the 'dot' notation (e.g. Ball.d for the d parameter of the Ball compartment) to explicitly assign parameters to certain compartments.

Model	Parameter global fixes	Parameter dependencies
Tensor	-	-
Ball&Sticks_in[n]	Ball.d = $3 \cdot 10^{-9} m^2/s$ Stick.d = $1.7 \cdot 10^{-9} m^2/s$	$\sum w_i = 1$
NODDI	NODDI_in.d = $1.7 \cdot 10^{-9} m^2/s$ NODDI_ex.d = $3 \cdot 10^{-9} m^2/s$ Ball.d = $3 \cdot 10^{-9} m^2/s$	1) $\sum w_i = 1$ 2) NODDI_ex.d $_{\perp 1} =$ NODDI_ex.d $_{\parallel} \cdot \frac{w_{ex}}{w_{in} + w_{ex}}$ 3) NODDI_ex. $\{\kappa, \theta, \phi\} =$ NODDI_in. $\{\kappa, \theta, \phi\}$
CHARMED_in[n]	-	$\sum w_i = 1$

Table 2.4: Parameter dependencies and global fixes

In the NODDI model in Table 2.4, the second dependency defines the NODDI tortuosity assumption and the third dependency locks the orientations and dispersion of the NODDI\_in and NODDI\_ex model to each other (the dispersion because the NODDI model adjusts the ratio of parallel and perpendicular extra-axonal diffusivity on the basis of orientation dispersion of the intracellular compartment). Additionally, for all multi-compartment models, the weights (volume fractions) must sum to one. This is accomplished by normalizing the set of  $n - 1$  volume fractions in each optimization iteration by the sum  $s = \sum_i^{n-1} w_i$  if that sum is larger than one. The last compartment weight, not explicitly optimized, is either set to zero, i.e.  $w_n = 0$  or set as  $w_n = 1 - s$  if  $s$  is smaller than one, for signal evaluation in the optimization process.

## 2.2.5 Indices calculated in post-processing

For every model various additional volumetric maps are calculated from the optimized model parameters. These can be model specific and model independent. Table 2.5 highlights the model specific maps. FS (Fraction of Sticks) is the total fraction (sum of weights) of the  $n$  Stick compartments in the Ball&Sticks model. NDI (Neurite Density Index) and ODI (Orientation Dispersion Index) are defined in (H. Zhang, Hubbard, et al., 2011) and FR (Fraction of Restricted) is the total fraction of the  $n$  restricted compartments in the CHARMED model (Assaf, Freidlin, et al., 2004; Santis, Assaf, and D. K. Jones, 2012).

Model	Additional maps
Tensor	-
Ball&Sticks_in[n]	FS = 1 - w <sub>Ball</sub>
NODDI	1) NDI = $\frac{w_{in}}{w_{in} + w_{ex}}$ 2) ODI = $\arctan^2(1, \kappa \cdot 10) \cdot \frac{2}{\pi}$
CHARMED_in[n]	FR = 1 - w <sub>hin</sub>

Table 2.5: Additional volumetric maps per model

In addition to these model specific maps, we add two model independent maps to the result set of every model. The first is the Log Likelihood (LL) map containing the complete (log) likelihood of the model given the data, as given by:

$$LL = -\frac{\sum \left( \mathbf{O} - \sqrt{\mathbf{S}(\bar{\mathbf{x}})^2 + \sigma^2} \right)^2}{2\sigma^2} - m \cdot \log(\sigma\sqrt{2\pi}) \quad (2.3)$$

With  $\mathbf{O}$  the observation,  $\mathbf{S}(\bar{\mathbf{x}})$  the function value given the optimal set of parameters,  $\sigma$  the noise standard deviation and  $m$  the number of volumes in the dataset. This LL map gives an indication of the goodness of fit and a higher LL is preferred. The second model independent map we add is the Bayesian Information Criterion (BIC) map. The BIC (Schwarz, 1978) can be used to compare models with different numbers of parameters on the same dataset (Panagiotaki et al., 2012; Ferizi, Schneider, Panagiotaki, et al., 2014). It is defined as  $BIC = -2 \cdot LL + k \cdot \ln(m)$  where  $LL$  is the log likelihood value,  $k$  is the number of free parameters and  $m$  is the number of observations. The model with the lower BIC is preferred and features the best trade-off between model-fit and complexity (number of parameters).

## 2.2.6 Optimization algorithms

We explicitly separate the models from the optimization algorithms to enable evaluating model/optimization algorithm combinations. We choose three commonly used optimization algorithms, two generations of gradient free algorithms (Nelder-Mead Simplex and Powell’s conjugate-direction) and one gradient-based algorithm (Levenberg-Marquardt). All optimization algorithms in this study were originally programmed in C and converted to OpenCL for the purpose of this work. OpenCL is a cross-vendor language standard and framework that enables highly parallelized calculations on the graphics card (graphical processing unit, GPU), as well as central processor unit (CPU),

potentially allowing for large computational acceleration (c.f. (Hernández et al., 2013)).

The first routine we used is the Nelder-Mead simplex (NMSimplex) method (Nelder and Mead, 1965). This is a general derivative-free nonlinear local optimization algorithm, also used in the MATLAB `fminsearch` function. The NMSimplex routine starts with an  $n$ -dimensional simplex with at each vertex initial values for all  $n$  model parameters. During optimization, the values of the evaluation function at each vertex of the simplex are used to determine the worst corner point. The algorithm then tries to replace this worst point by transforming the simplex about its centroid using reflection, expansion, contraction and shrinking (Rios and Sahinidis, 2013). This algorithm has five coefficients that determine its behavior: the scale of the initial simplex ( $s$ ), and the coefficients for the reflection ( $\alpha$ ), contraction ( $\beta$ ), expansion ( $\gamma$ ), and shrink ( $\delta$ ) of the simplex during each iteration. Preliminary tests using the default coefficients by (Nelder and Mead, 1965), the table of coefficients in (Wang and Shoup, 2011) and the adaptive coefficients in (Gao and Han, 2012) led us to use the adaptive coefficients in the latter, being:  $s = 1$ ,  $\alpha = 1$ ,  $\beta = 0.75 - 1.0/(2 \cdot k)$ ,  $\gamma = 1 + 2.0/k$  and  $\delta = 1 - 1.0/k$  with  $k$  being the number of free parameters in the model. The original implementation followed for translation into OpenCL was programmed in C by Michael F. Hutt (<http://www.mikehutt.com/neldermead.html>).

The second routine studied is the Powell conjugate-direction method (Powell, 1964). This is a general derivative-free nonlinear local optimization algorithm. It starts the search with  $n$  search vectors each containing all  $n$  model parameters. In every iteration, the algorithm tries to update each search vector in turn using a line search across each dimension. Doing so, it will continuously create new vector combinations (using line searches) and replace old combinations until a stop criteria is met. The final result is the search vector with the minimum function value. We used an OpenCL implementation translated from the C implementation described in Numerical Recipes 3<sup>th</sup> edition (Press et al., 2007) while using the Brent algorithm (Brent, 1973) for the line search. We used the default settings (from the literature) with a bracket gold ratio of 1.618034 (default ratio by which successive intervals are magnified in bracketing) and a  $g$ -limit of 100 (the maximum magnification allowed for a parabolic-fit step in function bracketing).

The third routine we studied is the Levenberg-Marquardt (LM) method (Levenberg, 1944). This method is a gradient based least squares optimization routine for which the gradients can either be given analytically or approximated numerically. We used numerical gradients for all models, as this is most commonly applied (Assaf, Blumenfeld-Katzir, et al., 2008; Alexander, Hubbard, et al., 2010; Santis, Assaf, and D. K. Jones, 2012; H. Zhang, Schneider, et al., 2012). Our implementation is an OpenCL version of the `lmfit` (<http://apps.jcns.fz-juelich.de/lmfit>) C implementation of this algorithm.

All three algorithms above have two main stop criteria. Either the difference between two consecutive iterations is below a certain threshold, or the predefined maximum number of iterations is reached. For the first we set the break threshold at thirty times the machine epsilon. For the second we set the so-called patience ( $i = p \cdot (1 + k)$ ) where  $i$  is the number of iterations,  $p$  is the patience and  $k$  is the number of free parameters in the model) to a predefined setting per optimization algorithm. This construction allows fixing the patience while ensuring that models with more parameters are allowed more iterations. We did a preliminary study to determine per algorithm and model the optimal patience given goodness of fit and run time. We used, unless stated otherwise, for NMSimplex a patience of 200, for Powell a patience of 2 and for LM a patience of 100, which gave the optimal trade-off between run time and fit (objective function minimum) for each algorithm. All calculations were performed on a single 2015 AMD Fury X graphics card using single float (32-bit) precision, except for parts where value range (summation of log likelihoods, Euclidian norm in the LM algorithms) or precision (the Watson function in NODDI) are critical, which were evaluated with double precision.

### 2.2.7 Parameter transformations

During model fitting with the NMSimplex, Powell and LM method we use parameter transformations to limit the range of each parameter to biophysical meaningful values and to scale the parameters to a range better suited for optimization (Panagiotaki et al., 2012). These transformations transform the parameters between two sets of parameters spaces: the model space and the optimization space, both in  $\mathbb{R}^n$ . The model space is the space in which the parameters can be used as input to the model. The optimization space is the set of parameters the optimization routine tries to optimize. We define the encoding transformation as transforming the parameters from model to optimization space and the decoding transformation from optimization to model space. Please see the Parameter transformations in Appendix A for a complete list of parameter transformations used in this work.

### 2.2.8 Cascaded model optimization

A common strategy to optimize complex (i.e. high number of parameter) models is initializing subsets of parameters with results from earlier simpler model optimizations (Panagiotaki et al., 2012; H. Zhang, Schneider, et al., 2012; Santis, Assaf, Jeurissen, et al., 2016). We formalize this here as cascaded model optimization. We test three cascading strategies: Cascade S0 (CS), Cascade Initialized (CI) and Cascade Fixed (CF), see Figure 2.1. In the CS cascade we first fit a one parameter  $S_0$  model to the non-diffusion weighted volumes to determine the scale



of the non-diffusion weighted signal. We use this to initialize the  $S_0$  parameter of the model we are interested in. In the CI cascades we initialize a few specific parameters like the volume fractions and the fiber orientation parameters from the results of a simpler model. The CF cascades are similar to the CI cascades except that we now fix some of the parameters, reducing the dimensionality of the last optimization, instead of initializing them. For a complete overview of initialized and fixed parameters for all three cascade variants see the section Cascaded model initializations of Appendix A. In this work, we always use at least CS for all model fitting, as an alternative to dividing away the non-diffusion weighted signal or to fixing it beforehand. Since the range of the  $S_0$  parameter is quite large and arbitrary, initializing with an  $S_0$  estimate improves the performance and run time of the optimization routines.

## 2.2.9 Datasets

For this study we used two groups of ten subjects coming from two studies. The first ten subjects are from the freely available fully preprocessed dMRI data from the USC-Harvard consortium of the Human Connectome project. Data used in the preparation of this work were obtained from the MGH-USC Human Connectome Project (HCP) database (<https://ida.loni.usc.edu/login.jsp>). The HCP project (Principal Investigators: Bruce Rosen, M.D., Ph.D., Martinos Center at Massachusetts General Hospital; Arthur W. Toga, Ph.D., University of California, Los Angeles, Van J. Weeden, MD, Martinos Center at Massachusetts General Hospital) is supported by the National Institute of Dental and Craniofacial Research (NIDCR), the National Institute of Mental Health (NIMH) and the National Institute of Neurological Disorders and Stroke (NINDS). Collectively, the HCP is the result of efforts of co-investigators from the University of California, Los Angeles, Martinos Center for Biomedical Imaging at Massachusetts General Hospital (MGH), Washington University, and the University of Minnesota (<http://www.humanconnectome.org/documentation/MGH-diffusion/>).

The data were acquired on a specialized Siemens Magnetom Connectom with 300mT/m gradient set (Siemens, Erlangen, Germany). These datasets were acquired at a resolution of 1.5mm isotropic with  $\Delta=21.8\text{ms}$ ,  $\delta=12.9\text{ms}$ ,  $TE=57\text{ms}$ ,  $TR=8800\text{ms}$ , Partial Fourier = 6/8, MB factor 1 (i.e. no simultaneous multi-slice), in-plane GRAPPA acceleration factor 3, with 4 shells of  $b=1000, 3000, 5000, 10,000, \text{s/mm}^2$ , with respectively 64, 64, 128, 393 directions to which are added 40  $b_0$  volumes leading to 552 volumes in total per subject, with an acquisition time of 89 minutes. We refer to these datasets as *HCP MGH - 1.5mm -552vol - b10k* and to the multi-shell direction table as the *HCP MGH* table. These four-shell, high number of directions, and very high maximum  $b$ - value datasets allow a wide range of models to be fitted. The second set of ten subjects comes from the diffusion protocol pilot phase of the Rhineland Study ([www.rheinland-studie.de](http://www.rheinland-studie.de)) and was

acquired on a Siemens Magnetom Prisma (Siemens, Erlangen, Germany) with the Center for Magnetic Resonance Research (CMRR) multi-band (MB) diffusion sequence (Moeller et al., 2010; Xu et al., 2013). These datasets had a resolution of 2.0mm isotropic with  $\Delta=45.8\text{ms}$ ,  $\delta=16.3\text{ms}$  and  $\text{TE}=90\text{ms}$ ,  $\text{TR}=4500\text{ms}$  Partial Fourier = 6/8, MB factor 3, no in-plane acceleration with 3 shells of  $b=1000, 2000, 3000 \text{ s/mm}^2$ , with respectively 30, 40 and 50 directions to which are added 14  $b_0$  volumes leading to 134 volumes in total per subject, with an acquisition time of 10 min 21 sec. Additional  $b_0$  volumes were acquired with a reversed phase encoding direction which were used to correct susceptibility related distortion (in

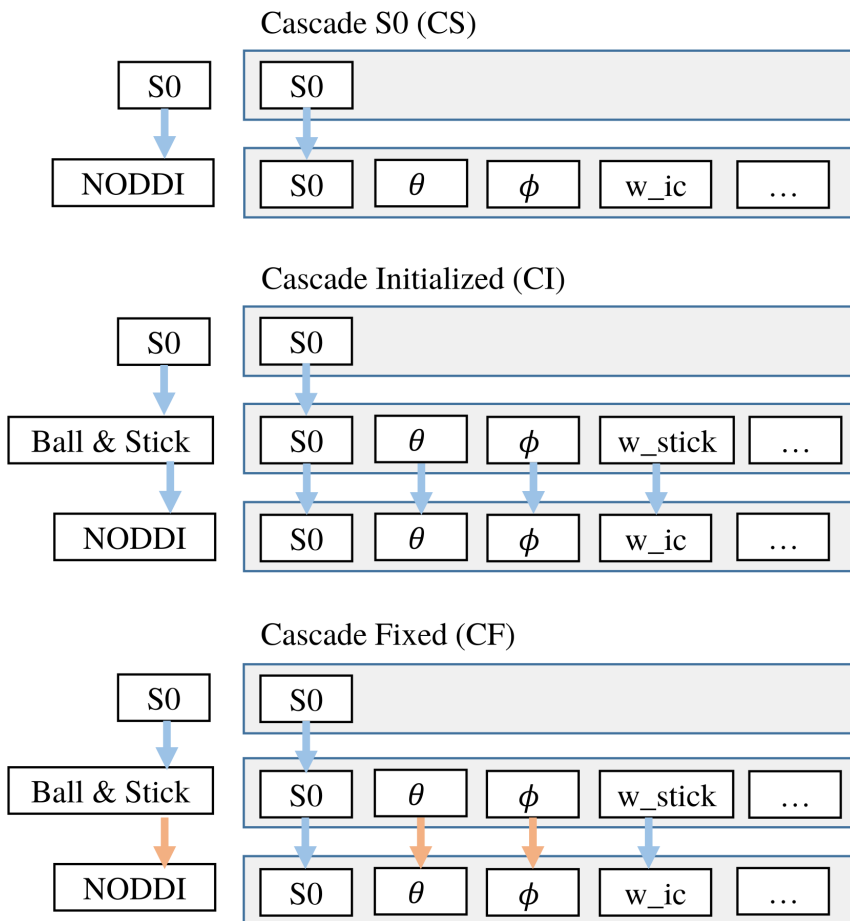


Figure 2.1: Illustration of the three different cascading strategies (for the example of the NODDI model): CS, CI and CF. The blue arrows indicate initialization of a parameter, the orange arrows indicate fixing a parameter.

addition to bulk subject motion) with the topup and eddy tools in FSL version 5.0.9. We refer to these datasets as *RLS - 2mm - 134dir - b3k* and to the multi-shell direction table as the *RLS* table. These three-shell datasets represent a relatively short time acquisition protocol that still allows many models to be fitted.

Since the Tensor model is only valid for b-values up to about  $1200\text{s/mm}^2$ , it is estimated on only the b-value  $1000\text{s/mm}^2$  shell and  $b_0$  volumes. All other models are estimated on all data volumes. For all datasets we created a whole brain mask, using BET from FSL (S. M. Smith, 2002), and a white matter (WM) mask. The whole brain mask is used during the model fitting, whereas averages over the WM mask are used in model or data comparisons. The WM mask was calculated by applying a threshold of 0.3 on the Tensor FA results, followed by a double pass 3D median filter of radius 2 in all directions. The Tensor estimate for this mask generation was calculated using a CI Ball Stick/Tensor cascade optimized with the Powell method.

To compare performance and robustness of different optimization algorithms and cascading strategies the per-subject LL averages over the WM mask were calculated. The mean differences between algorithms or strategies over ten subjects and their standard error of the mean (SEM) were then reported.

## 2.2.10 Ground truth simulations

We performed two sets of ground truth simulations (see Figure 2.2) to quantify accuracy and precision of: 1) the optimization (Figure 2.2A) and 2) the cascading results (Figure 2.2B). First, to assess the accuracy and precision of the different optimization algorithms (Figure 2.2A), we created ten thousand random fiber orientations for each of sixty linearly spaced volume fractions in  $[0.2, 0.8]$ , for the *Ball&Sticks\_in1*, *CHARMED\_in1* and *NODDI* models using both a HCP MGH and a *RLS* multi-shell direction table. For the generated signals we created three copies with a Rician distributed noise realization at SNRs 10, 20 and 40. We compute optimization error as optimized parameter value minus ground truth parameter value for the intra-axonal volume fraction, i.e. fraction of stick (FS) for *Ball&Sticks\_in1*, fraction of restricted (FR) for *CHARMED\_in1* and fraction of restricted FR for *NODDI*. We compute a measure of accuracy as the inverse of the mean optimization error over ten thousand random fiber orientations per volume fraction (each with a unique noise realization). Similarly, as a measure of precision we compute inverse of the standard deviation of the optimization error over ten thousand random fiber orientations per volume fraction. Finally, we aggregated these results over the sixty volume fractions into a mean and standard error for both accuracy and precision.

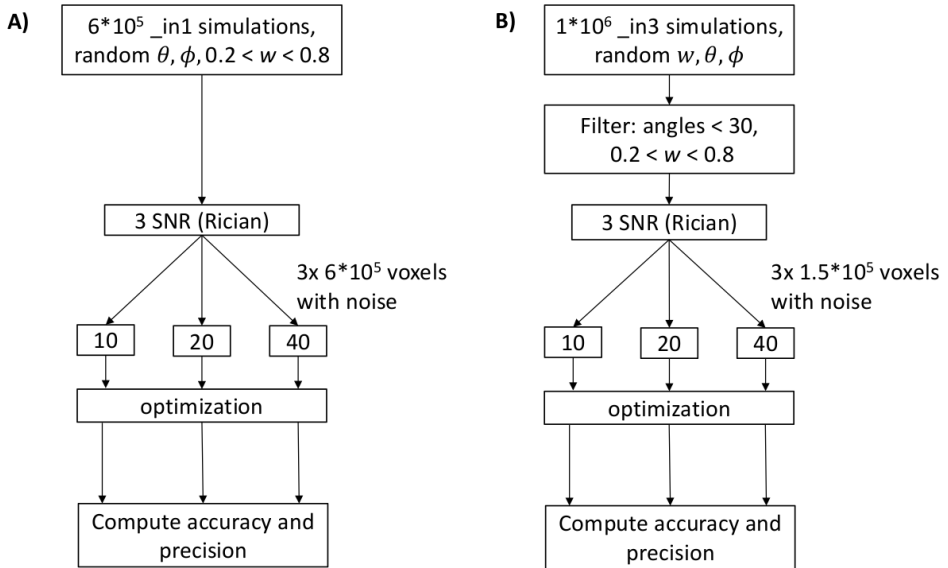


Figure 2.2: Illustration of the ground truth simulation workflow to quantify the accuracy and precision of A) the optimization routine comparisons, B) the cascading results.

## 2.3 Results

We focus first on the effect of optimization algorithms when applied to the entire diffusion model estimation problem. Subsequently, we focus on initialization strategies by cascaded optimization, dividing the estimation problem into sequential lower dimensional problems.

### 2.3.1 Effect of optimization algorithm

Figure 2.3 shows estimated parameter maps for a single HCP MGH subject (HCP MGH - 1.5mm - 552vol - b10k, subject 1003) for four different models (Tensor, Ball&Sticks\_in1, NODDI, CHARMED\_in1) each estimated with three optimization algorithms (NMSimplex, Powell, LM). Note that the Tensor model is mainly shown for comparison and is only estimated on the b-value  $1000s/mm^2$  shell. Table 2.6 shows the corresponding Log Likelihoods (LL), Bayesian Information Criteria (BIC) and run times (RT) for these results, along with the results of a single RLS - 2mm - 134dir - b3k subject. It can be seen that a GPU based implementation gives full model optimization results and estimated whole brain parameter maps in matters of seconds to minutes even for complex models and a large dataset. Powell almost always outperforms other methods in terms of LL and BIC (except

for CHARMED\_in1 for the RLS dataset). The Powell algorithm shows the most consistently high fitting performance over models and datasets, especially on the complex 552 volume HCP MGH dataset. On the simpler 134 volume RLS dataset the fit is sometimes matched (with NMSimplex in Tensor-RLS and with LM in Ball&Sticks\_in1- RLS) or slightly exceeded (by NMSimplex in CHARMED\_in1 - RLS) by one of the other algorithms. In addition, Powell is often the fastest algorithm, and when not the fastest always better fitting than the fastest algorithm. To assess the variability of the achieved fit, Figure 2.4 reports mean differences in fit between algorithms and its standard error over the mean (SEM) over ten HCP MGH and ten RLS subjects. Powell outperforms the other algorithms except for the case of the CHARMED\_in1 model with the RLS data by a very small margin. Error bars show that the variability of the fit differences over subjects is relatively low with the SEM generally at about ten percent of the fit difference.

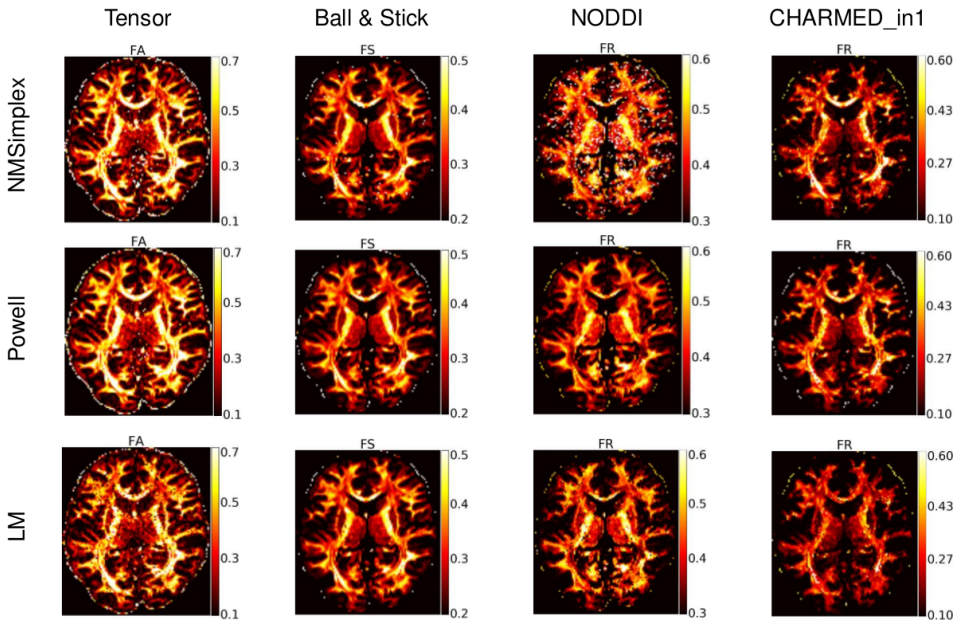


Figure 2.3: Models (in columns) against optimization algorithms (in rows) for a single HCP MGH dataset. FA: Fractional Anisotropy; FS: Fraction of stick compartment; FR: fraction of restricted (fraction of intra-cellular/restricted compartment). Each model parameter map (column) is individually scaled.

	Tensor			Ball&Sticks.in1			NODDI			CHARMED.in1			
	LL	BIC	RT	LL	BIC	RT	LL	BIC	RT	LL	BIC	RT	
HCP MGH	NMSimplex	-528	1099	0:00:41	-3000	6025	0:01:33	-2759	5627	1:07:50	-2698	5466	2:30:53
	Powell	-514	1072	0:01:08	-2985	5995	0:00:50	-2714	5467	0:12:08	-2691	5452	0:21:17
	LM	-519	1082	0:01:30	-2992	6009	0:00:49	-2759	5557	0:14:48	-2708	5486	0:28:10
RLS	NMSimplex	324	682	0:00:07	-1083	2185	0:00:09	-984	1998	0:05:32	962	1977	0:06:36
	Powell	-324	682	0:00:03	-1081	2182	0:00:03	-976	1980	0:01:23	-963	1979	0:01:19
	LM	-328	689	0:00:28	-1081	2182	0:00:04	-997	2023	0:01:31	-972	1998	0:02:59

Table 2.6: Models (in columns) against optimization algorithms (in rows) for a single HCP MGH and RLS dataset. Reported LL and BIC are the mean over a whole brain white matter mask. The run times are over the entire brain mask and are in units of (h:mm:ss). For the BIC, the number of observations is for Ball&Sticks.in1, NODDI and CHARMED.in1 given as (HCP MGH: 552, RLS: 134) and for Tensor given as (HCP MGH: 104, RLS: 44), while the number of free parameters is given as (Tensor: 7, Ball&Sticks.in1: 4, NODDI: 6, CHARMED.in1: 11).

To assess accuracy and precision for different optimization algorithms, Figure 2.5 reports ground truth simulation results over three models (Ball&Sticks\_in1, NODDI and CHARMED\_in1), the two multi-shell direction tables and three simulated SNRs. A general trend of increasing accuracy and precision with SNR can be observed. In general, Powell has a tendency for higher accuracy and precision than the other algorithms over all models, which is often more pronounced at higher SNRs and more complex models. In terms of accuracy, LM can keep up with Powell in Ball&Sticks\_in1 and NMSimplex in the CHARMED\_in1 model, but precision is almost always considerably higher for Powell. When comparing acquisition tables for the Powell method, little difference can be seen for Ball&Sticks\_in1, the simplest model. For NODDI there seems to be a tendency for increased precision when using the extensive HCP MGH direction table instead of the RLS table at the same SNR. For CHARMED\_in1, a clear increase in accuracy and precision can be observed when going to the more complex HCP protocol, which may reflect the presence of high b-values ( $> 3000s/mm^2$ ) as previously reported (De Santis, Assaf, et al., 2014).

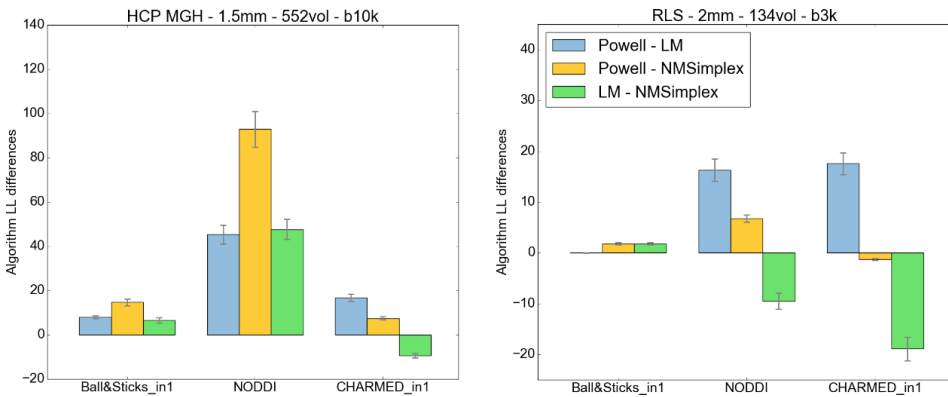


Figure 2.4: Algorithm Log Likelihood differences per model over 10 HCP MGH (left) and 10 RLS subjects (right). Bars depict the mean LL difference and error bars give the standard error of the mean for the difference over subjects.

### 2.3.2 Effects of cascaded optimization

We next focus on the effect of different cascading strategies for a very complex model, CHARMED\_in3 with three different restricted compartments per voxel and a total number of 19 parameters. This is a very difficult model to fit, even on a very large number of volumes (such as the HCP MGH multi-shell direction table), but nonetheless a very useful model for microstructure quantification as it sepa-

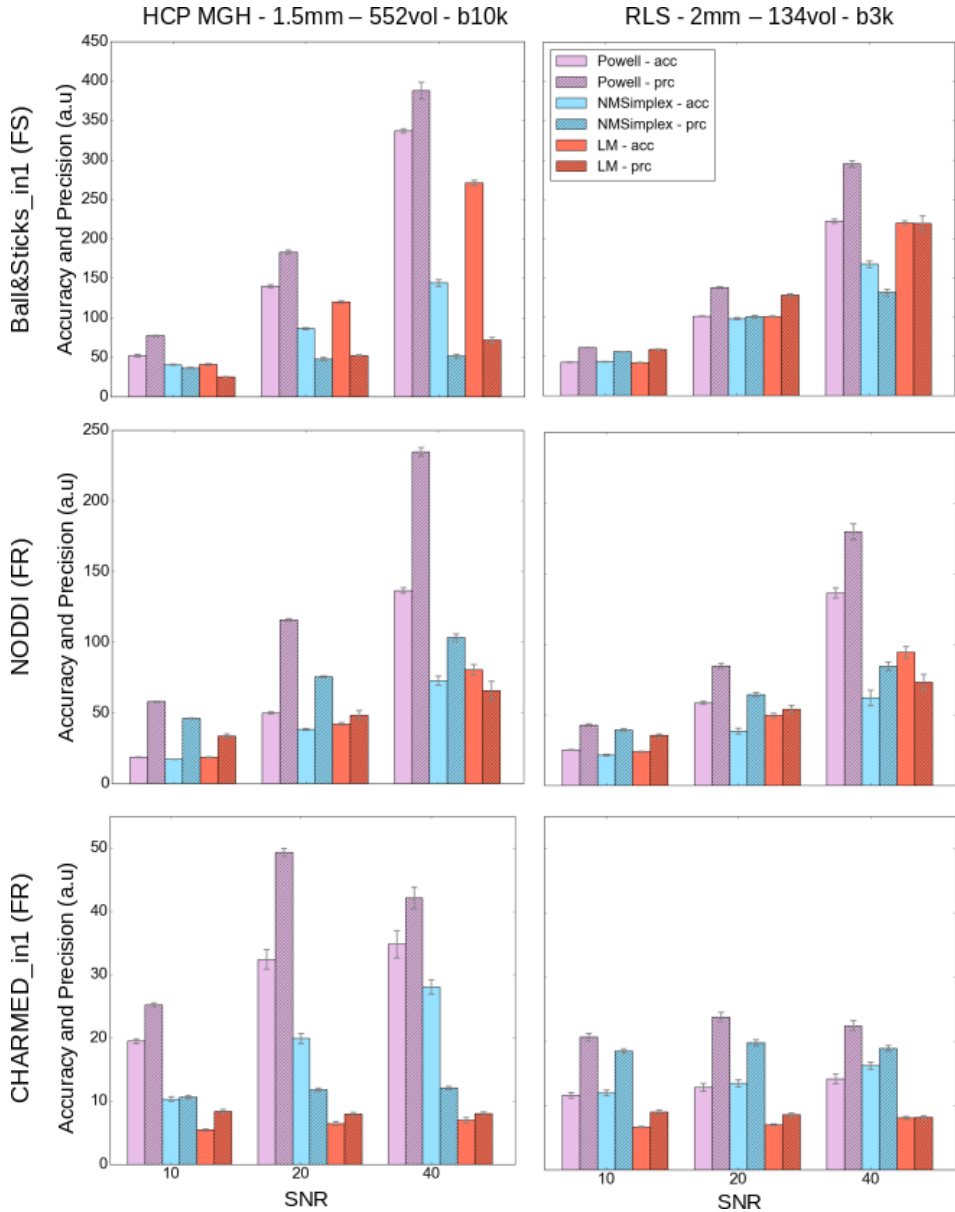


Figure 2.5: Ground truth simulation results for three models (rows) and two multi-shell direction tables (columns) for three SNRs. The unstriped bars represent the accuracy (acc) per optimization algorithm, the striped bars represent the precision (prc), computed over 10 4 realizations per volume fraction. The bars give the mean and the error bars give the standard error computed over seven different simulated volume fractions between 0.2 and 0.8.



rates multiple fiber orientations per voxel (Santis, Assaf, and D. K. Jones, 2012). Figure 2.6 shows FR maps and multiple local restricted compartment orientations for CS (Cascade S0), CI (Cascade Initialized) and CF (Cascade Fixed). CS will optimize the full 19 parameters in the final cascade step; CI will also optimize the full 19 parameters in the final cascade step but the orientations are initialized from an earlier Ball&Sticks\_in3 cascade step; CF will optimize 13 parameters in the final cascade step where the 6 restricted orientation parameters are fixed from an earlier cascade step. Transverse maps and orientation plots are shown at the level of the centrum semiovale where three distinct fiber orientations are expected in white matter. Both FR and orientation estimates appear more structured and smooth in CI and CF, compared to CS. Orientation estimates, in particular, show missing identification of the green A-P orientation (corresponding to the superior longitudinal fasciculus) in some places, for CS and to a lesser degree CI (see white arrow). Table 2.7 reports run times and fit parameters for different models of increasing complexity for a single HCP MGH and RLS dataset. The run times and fit parameters show that quantitatively CI and CF have a better fit than CS. In terms of run time, CF outperforms both CS and CI, especially for the CHARMED\_in3 model by a factor of 3. These results show that a straightforward optimization of a large 19 parameter model as in CS, achieves a suboptimal fit, even when using a large multi-shell direction table. However, with a good initialization of a subset of the parameters, the fit can be improved considerably. Finally, although results are more subtle, fixing (in CF) rather than initializing (in CI) the orientation parameters, provides similarly good results even though the degrees of freedom in the final optimization step have been reduced. For simpler models (NODDI and CHARMED\_in1 and\_in2) with less parameters, the fit advantages of CI and CF strategies are almost absent and the speed advantages less pronounced, although speed-ups of 1.5x to 3x are still achieved.

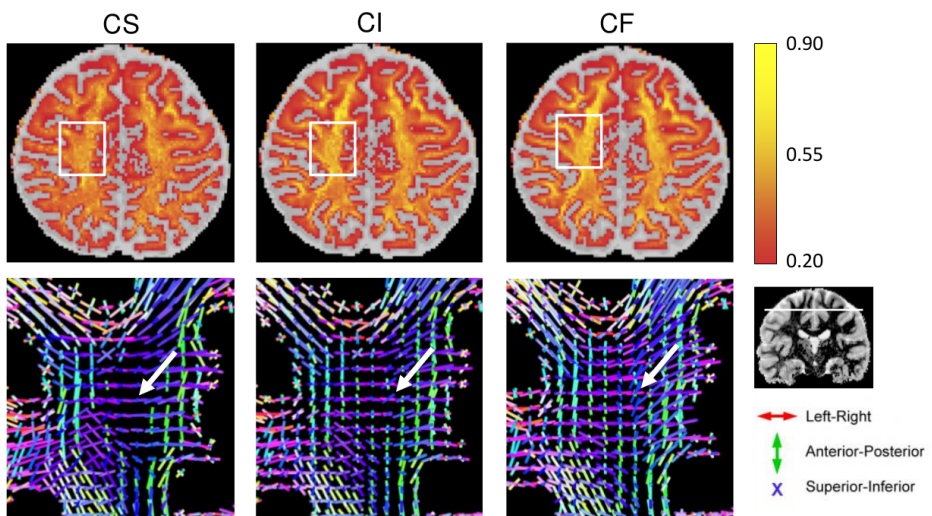


Figure 2.6: Parameter maps (CHARMED.in3, FR map; top row) and local restricted compartment 3D color-coded orientations (bottom row) for different cascading strategies (in columns) for a Y slice 27 single HCP MGH subject (1003). The top row shows a thresholded FR map superimposed on a hindered fraction map in gray-scale.

	NODDI			CHARMED_in1			CHARMED_in2			CHARMED_in3			
	LL	BIC	RT	LL	BIC	RT	LL	BIC	RT	LL	BIC	RT	
HCP MGH	CS	-2714	5467	0:12:08	-2691	5452	0:21:17	-2680	5455	1:03:38	-2677	5474	1:29:34
	CI	-2716	5470	0:11:31	-2689	5447	0:21:02	-2677	5449	1:00:56	-2675	5469	2:07:14
	CF	-2715	5468	0:07:47	-2690	5449	0:13:17	-2678	5451	0:33:44	-2676	5472	1:03:09
RLS	CS	-976	1980	0:01:23	-963	1979	0:01:19	-962	1997	0:03:51	-961	2015	0:11:12
	CI	-974	1977	0:01:09	-961	1977	0:01:42	-959	1991	0:04:57	-958	2009	0:12:31
	CF	-974	1978	0:00:48	-962	1978	0:00:51	-960	1993	0:02:13	-959	2011	0:03:38

Table 2.7: Models (columns) against optimization algorithms (rows) for a single HCP MGH and RLS dataset. Reported LL and BIC are the mean over a whole brain white matter mask. The run times are over the entire brain mask and are in units of (h:mm:ss). For the BIC, the number of observations is given by the number of volumes in the datasets (552 for HCP MGH and 134 for RLS), while the number of free parameters is given as (NODDI: 6, CHARMED\_in1: 11, CHARMED\_in2: 15, CHARMED\_in3: 19).

To assess the robustness of the fit improvements, Figure 2.7 reports mean difference in fit between cascading strategies and its SEM over ten HCP MGH and ten RLS subjects. CI and CF always have a higher LL than CS (with the exception of NODDI for the HCP MGH data). More generally, the fit improvement of CI and CF over CS increases with model complexity (i.e. with CHARMED\_in2, CHARMED\_in3). In addition, the fit of CI is always slightly higher than that of CF, with this fit difference generally being smaller than that separating CS from both of these. Error bars show that the variability of the fit differences over subjects is relatively low.

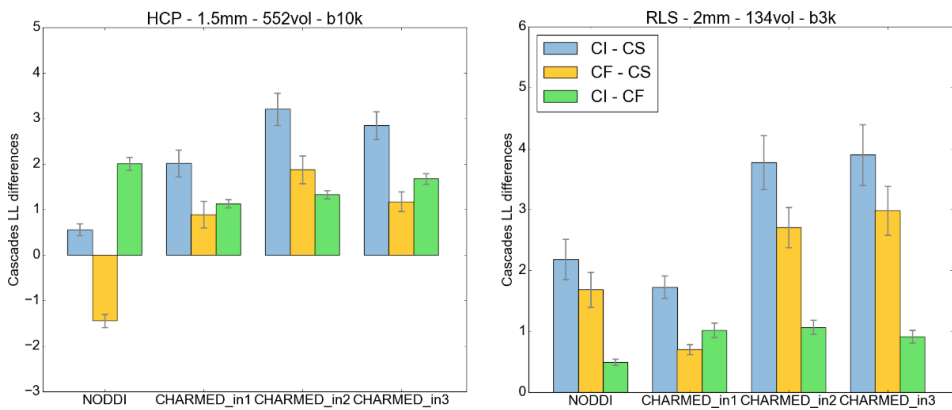


Figure 2.7: Inter-subject differences of the LL for each model and cascading technique over 10 MGH and 10 RLS subjects. The error bars give the standard error of the mean for the inter-subject differences.

To assess accuracy and precision for different cascading strategies, Figure 2.8 reports ground truth simulation results for the two multi-shell direction tables and three SNRs for the most complex CHARMED\_in3 model. The cascading strategies CI and CF outperform CS in terms of accuracy and precision, especially for the higher SNRs of 20 and 40. For the more extensive HCP protocol, accuracy and precision for CI and CF is very similar, irrespective of SNR. For the less extensive RLS protocol, CF has a tendency of outperforming CI, especially for lower SNRs. Also, the accuracy and precision is clearly higher for the more complex HCP MGH multi-shell direction table.

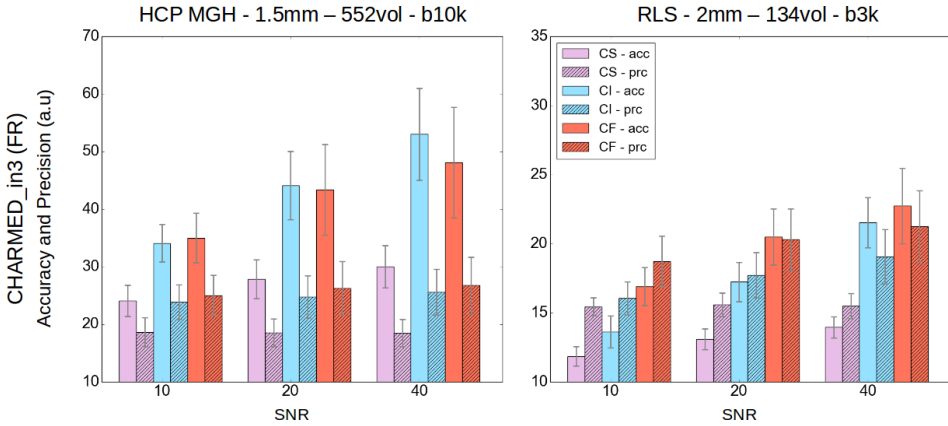


Figure 2.8: Ground truth simulation results for the CHARMED.in3 model for the HCP MGH and RLS multi-shell direction tables for three SNRs. The unstriped bars represent the accuracy (acc) per cascading strategy, the striped bars represent the precision (prc), computed over a total number of  $10^6$  realizations. The bars give the mean and the error bars give the standard error computed over the volume fractions of the three simulated orientations.

## 2.4 Discussion

Using an efficient GPU based implementation, we show that run time can be removed as a fundamental constraint for optimization of multi-compartment models, achieving whole brain dataset fits in seconds to minutes. We find that optimization algorithms and multi-step optimization approaches have a considerable influence on performance and stability over subjects and over acquisition protocols. The gradient-free Powell conjugate-direction algorithm was found to outperform other common algorithms in terms of run time, fit, accuracy and precision. Parameter initialization approaches were found to be relevant especially for more complex models, such as those involving several fiber orientations per voxel.

### 2.4.1 Optimization algorithms

The Powell conjugate-direction algorithm almost always outperforms the other methods in terms of LL and BIC, over multiple models and two different multi-shell acquisition protocols. In addition, Powell is often the fastest algorithm, and when not the fastest it always provides better fitting than a faster algorithm. The variability over ten subjects in the fit differences between algorithms is relatively low with the standard error generally at a third to half of the fitting difference. This shows that the fitting improvement is a significant and robust one likely exceeding the variability of the model parameters in experimental subject groups

and therefore possibly of influence on statistical power. The BIC differences between optimization algorithms are of a similar size as the BIC differences between models. This means that the choice of optimization algorithm could have an influence on later model comparisons and model rankings. Going beyond data fit and considering accuracy and precision using ground truth simulations we find that Powell outperforms the other algorithms, which is even more pronounced at higher SNRs and with more complex models. When comparing accuracy and precision between acquisition protocols for the Powell method, little effect is observed for simpler models such as Ball&Stick.in1 and NODDI, but an increase in accuracy can be observed for CHARMED.in1 when going to the more complex HCP protocol.

### 2.4.2 Cascaded optimization

Different forms of cascaded optimization have already been applied to diffusion microstructure modeling. For instance, in (H. Zhang, Schneider, et al., 2012) and the accompanying NODDI MATLAB toolbox first do an initial grid search over a feasible range of parameter values after which the parameters are optimized twice, once with the fiber orientations and dispersion locked and the second time with all parameters free. In (Santis, Assaf, Jeurissen, et al., 2016), the authors use a two-step optimization approach in which the number of intra-axonal compartments and their orientations are fixed to the orientations found by the CSD method (J. D. Tournier, Calamante, and Connelly, 2007; Jeurissen et al., 2014). We formalize and generalize multi-step optimization here as cascaded model optimization. Results show that CI and CF always have a better fit (higher Log Likelihood) than CS and that the fit improvement of CI and CF over CS increases with model complexity. This shows that with a good initialization of a subset of the parameters, the fit can be improved considerably over one-step optimization (not considering  $S_0$ , as it is common practice to either work on  $S_0$ -normalized signal or to estimate and fix it beforehand). In terms of accuracy and precision CI and CF also outperform CS, especially for the higher SNRs of 20 and 40.

Comparing between CI and CF for the complex CHARMED.in3 model (19 parameters) and for the more extensive HCP protocol, the accuracy and precision for CI and CF is very similar, irrespective of SNR. For the less extensive RLS protocol, CF has a tendency of outperforming CI especially for lower SNRs. These results are interesting given that CF has lower degrees of freedom in its final cascade step than CI. More degrees of freedom potentially allow for a better fit since the model can better adapt to the given data, which is shown in the superior fit performance in CI in the context of high SNR and the rich HCP MGH acquisition protocol. However, in the context of few data points or lower SNR, more degrees of freedom are a disadvantage, as is shown in the superior accuracy of CF in the context of lower SNRs and the RLS acquisition protocol. Hence, overall, CF

has very appealing properties for general use as a cascaded optimization strategy for complex models. Besides providing a good fit, accuracy and precision in many contexts, it is also, by a large margin, invariably, the fastest cascading strategy. Since the Powell algorithm provided the best performance when compared to NMSimplex and LM we focused here on comparing cascading results using Powell only. Comparisons of cascading strategies using the other optimization algorithms (see Supplementary Table 1 of (Harms, Fritz, et al., 2017)) show that CI and CS strategies can provide considerable benefits for NMSimplex and LM as well, but also that the Powell algorithm continues to outperform NMSimplex and LM when combined with CF and CI. Other initialization strategies like random multi-start and initializing with a preliminary grid search have been applied to diffusion microstructure model optimization. Comparisons of these initialization strategies to the cascading strategies investigated here using all optimization algorithms (see Supplementary Table 2 of (Harms, Fritz, et al., 2017)) show that considerable fit improvements can indeed be achieved with multi-start and grid search compared to CS for NMSimplex and LM. However, these results are almost invariably inferior to, or as good as, the CF and CI results, especially for the Powell algorithm. In terms of runtime/performance trade-off the cascading strategies perform vastly better than grid search and multi-start.

An additional argument for the use of CF is that it allows combining microstructure parameter maps with tractography results in a structured way (e.g.; (Bells et al., 2011)). One can use the fiber orientations from a Ball&Sticks\_in[n] fit for tractography and subsequently use CF with the same orientations for a CHARMED\_in[n] model to ensure that microstructure parameters mapped onto these tracks agree with the local track orientation. Alternatively, when using CI or when fitting the tractography orientations and the microstructure models separately, one would not have this assurance. Future research could focus on more ways of initializing or fixing parameters over cascading steps, such as limiting the extra-axonal Tensor compartment in CHARMED to the orientations of the intra axonal orientations. Additionally, one could also use other methods for orientation initialization, such as CSD (J. D. Tournier, Calamante, and Connelly, 2007; Jeurissen et al., 2014). Additionally, one can think of cascades where the final model is initialized with the results of more than one model, for example, initializing the orientations with a Ball&Sticks model and initializing a (possible) T1 or T2 compartment from a different model. These extensions fit within the framework proposed here.

### 2.4.3 Run times

All run times reported here were achieved with a single 2015 AMD Fury X graphics card with a theoretical peak performance of 8.6 teraflops single precision. Higher performing hardware, as well as use of multiple devices in parallel should

deliver proportional run time gains. In practice, OpenCL driver performance, number of parallel streaming processors and streaming processor architecture, device memory and memory bandwidth (here 4GB and 512GBytes/sec, respectively) also determine computational performance.

#### 2.4.4 Limitations

In this work we compared three optimization algorithms: Nelder-Mead Simplex, Powell's conjugate-direction and Levenberg-Marquardt, which covers a set of generally used algorithms. Other optimization algorithms could be considered. Since all the tested optimization routines are local optimizers, it would be interesting to also use global optimization algorithms such as the Self-Organizing Migrating Algorithm (SOMA) (Zelinka, 2004) to evaluate their capacity of avoiding local minima. Also Markov Chain Monte Carlo sampling of the full Bayesian posterior would be an interesting extension since it can avoid local minima and would provide precision or uncertainty estimates for the obtained parameters. However MCMC sampling is a different type of algorithm than the optimization algorithms covered in this work, in two ways. First, it has a different goal: it aims to sample the full posterior distribution of parameters, rather than provide a (maximum likelihood) point estimate of the optimal parameter value. Generating the full posterior inevitably also takes more runtime, which is often only justified if more than just the maximum of the posterior is used (e.g. the standard deviation of the posterior). Second, it requires more information. All algorithms in this work are local optimization routines that only require an objective (likelihood) function to perform the minimization. MCMC on the other hand additionally requires the user to define a prior probability function and a proposal function for each of the model parameters. Other optimization approaches avoid non-linear optimization altogether by linearizing the optimization problem (Daducci et al., 2015; Canales-Rodríguez et al., 2015; Novikov, Jelescu, and Fieremans, 2015). For instance, AMICO (Daducci et al., 2015; Canales-Rodríguez et al., 2015) constructs a convex formulation with a single global minimum by choosing a finite dictionary of atoms and has been successfully applied to the NODDI, ActiveAx (Daducci et al., 2015) and VERDICT (Bonet-Carne, Daducci, et al., 2016) models. This different approach comes with its own trade-offs. For instance, atom dictionary size can increase exponentially in more complex models when sampling interdependent parameters sufficiently dense (Yap, Y. Zhang, and Shen, 2015) and it is limited to a Gaussian noise model, which embeds assumptions almost never met in standard multichannel acquisitions (Gudbjartsson and Patz, 1995; Dietrich, Raya, Reeder, Ingrisch, et al., 2008; Daducci et al., 2015).



### 2.4.5 Software implementation note

All biophysical compartment models, noise models / likelihood functions, as well as optimization algorithms were modularly implemented in a python based GPU accelerated toolbox (Microstructure Diffusion Toolbox or MDT) freely available under an open source L-GPL license at <https://github.com/cbclab/MDT>. It comes with a graphical user interface for easy usage as an analysis tool. MDT can be used to perform all model optimizations reported in this work and the modular design of the toolbox allows easy extension with new single compartment models and composite multi-compartment models.

### 2.4.6 Conclusions and recommendations

When the aim is optimization of a point estimate, rather than sampling a full parameter posterior, (e.g. with MCMC), we would advise the use of the Powell algorithm for microstructure modeling in dMRI. For low complexity models up to about 8 parameters, including NODDI and Ball&Sticks.in1 a simple CS strategy can be used, although a CF strategy can yield run time gains. For intermediate complexity models, such as Ball&Sticks.in3, we recommend using a successive CI cascade on an increasing number of fiber orientations. Finally, CF is recommended for the higher complexity models such as CHARMED.in3 yielding run time, fit, accuracy and precision advantages as well as advantages for combined tractography and microstructure.





# 3

## Robust and fast Markov Chain Monte Carlo sampling of diffusion MRI microstructure models

---

Adapted from: R. L. Harms and A. Roebroeck (Dec. 2018). “Robust and Fast Markov Chain Monte Carlo Sampling of Diffusion MRI Microstructure Models”. In: *Frontiers in Neuroinformatics* 12.December, pp. 1–18. DOI: 10.3389/fninf.2018.00097.

## Abstract

In diffusion MRI analysis, advances in biophysical multi-compartment modeling have gained popularity over the conventional Diffusion Tensor Imaging (DTI), because they possess greater specificity in relating the dMRI signal to underlying cellular microstructure. Biophysical multi-compartment models require parameter estimation, typically performed using either Maximum Likelihood Estimation (MLE) or using Markov Chain Monte Carlo (MCMC) sampling. Whereas MLE provides only a point estimate of the fitted model parameters, MCMC recovers the entire posterior distribution of the model parameters given the data, providing additional information such as parameter uncertainty and correlations. MCMC sampling is currently not routinely applied in dMRI microstructure modeling because it requires adjustments and tuning specific to each model, particularly in the choice of proposal distributions, burn-in length, thinning and the number of samples to store. In addition, sampling often takes at least an order of magnitude more time than non-linear optimization. Here we investigate the performance of MCMC algorithm variations over multiple popular diffusion microstructure models to see whether a single well performing variation could be applied efficiently and robustly to many models. Using an efficient GPU-based implementation, we show that run times can be removed as a prohibitive constraint for sampling of diffusion multi-compartment models. Using this implementation, we investigated the effectiveness of different adaptive MCMC algorithms, burn-in, initialization and thinning. Finally we apply the theory of Effective Sample Size to diffusion multi-compartment models as a way of determining a relatively general target for the number of samples needed to characterize parameter distributions for different models and datasets. We conclude that adaptive Metropolis methods increase MCMC performance and choose the Adaptive Metropolis-Within-Gibbs (AMWG) algorithm as primary method. We further advise initializing the sampling with an MLE point estimate, in which case 100 to 200 samples are sufficient as a burn-in. Finally we make a point against thinning for most use-cases. As a relatively general target for the number of samples, we recommend a multivariate Effective Sample Size of 2200.

## 3.1 Introduction

Advances in microstructure modeling of diffusion Magnetic Resonance Imaging (dMRI) data have recently gained popularity since they possess greater specificity than Diffusion Tensor Imaging (DTI) in relating the dMRI signal to the underlying cellular microstructure, such as axonal density, orientation dispersion or diameter distributions. Typically, dMRI models are fitted to the data using non-linear optimization (Assaf, Freidlin, et al., 2004; Assaf and Basser, 2005; Assaf, Blumenfeld-Katzir, et al., 2008; Panagiotaki et al., 2012; H. Zhang, Schneider, et al., 2012; Assaf, Alexander, et al., 2013; Fieremans et al., 2013; De Santis, Drake-smith, et al., 2014; De Santis, Assaf, et al., 2014; Jelescu, Veraart, Fieremans, et al., 2015; Harms, Fritz, et al., 2017), linear convex optimization (Daducci et al., 2015), stochastic optimization (Farooq et al., 2016) or analytical (Novikov, Veraart, et al., 2018) methods to obtain a parameter point estimate per voxel. These point estimates provide scalar maps over the brain of micro-structural parameters, such as the fraction of restricted diffusion as a proxy for fiber density. These point estimates however do not provide the entire posterior distribution, which can be useful for, for example, probabilistic tractography and to quantify the uncertainty and interdependency of parameters.

The gold standard of obtaining the posterior distribution is by using Markov Chain Monte Carlo (MCMC) sampling, as for example in (Behrens et al., 2003; Alexander, 2008; Alexander, Hubbard, et al., 2010; Sotiropoulos et al., 2013). MCMC generates, per voxel, a multi-dimensional chain of samples, the stationary distribution of which is the posterior distribution, i.e. the probability density of the model parameters given the data. Per voxel, these samples capture parameter dependencies, multimodality and the width of peaks around optimal parameter values. For instance, summarizing the chain under Gaussian assumptions with a sample covariance matrix, would provide mean parameter estimates and corresponding uncertainties (the standard deviation), as well as inter-parameter correlations (figure 3.1).

Despite the advantages of providing the full posterior information, MCMC sampling is currently not routinely applied in dMRI microstructure modeling since it often requires adjustments and tuning specific to each model, particularly in the choice of proposals, burn-in length, thinning and the number of samples to store. In addition, sampling often takes at least an order of magnitude more time than non-linear optimization, even when using GPU's to accelerate the computations by one or two orders of magnitude (Hernández et al., 2013).

The main purpose of this paper is to provide an effective MCMC sampling strategy combined with an efficient GPU accelerated implementation. To this end, we investigate the performance of a few variants of Random Walk Metropolis MCMC algorithms over multiple popular diffusion microstructure models to see

whether a single well performing variation could be applied efficiently and robustly to many models. Furthermore, we discuss the use of burn-in and thinning in dMRI modeling, and apply the concept of effective sample sizes to determine a lower bound on the number of samples needed. To reduce run time constraints we provide an efficient parallel GPU implementation of all models and MCMC algorithms in the open source Microstructure Diffusion Toolbox (MDT; <https://github.com/cbclab/MDT>).

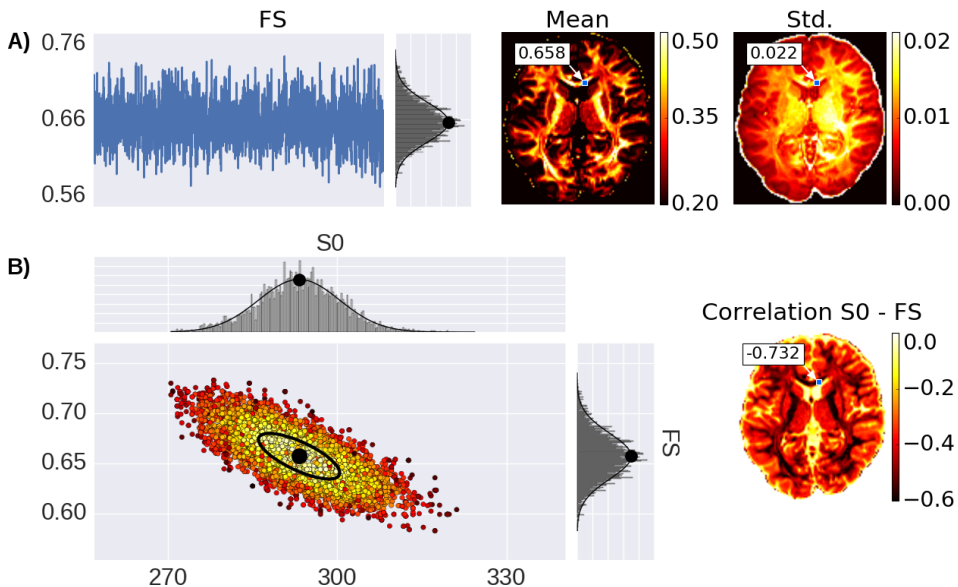


Figure 3.1: Illustration of parameter uncertainty and correlation for the Ball&Stick model using MCMC sampling, with the Fraction of Stick (FS) and the non-diffusion weighted signal intensity (S0). **A)** On the left, a single FS sampling trace and its corresponding histogram for the highlighted voxel with a Gaussian distribution function fitted to the samples with its mean indicated by a black dot. On the right, the mean and standard deviation (std.) maps generated from the independent voxel chains per voxel. **B)** On the left, the scatter-plot for two parameters (FS and S0) with the corresponding marginal histograms for the voxel highlighted in the maps. On the right, the S0-FS correlation map.

## 3.2 Methods

The biophysical (multi-)compartment models and the Markov Chain Monte Carlo (MCMC) algorithms used in this study are implemented in a Python based GPU accelerated toolbox (the Microstructure Diffusion Toolbox, MDT, freely available under an open source L-GPL license at <https://github.com/cbclab/MDT>). Its modular design allows arbitrary combinations of models with likelihood and prior distributions. The MCMC implementations are voxel-wise parallelized using the OpenCL framework, allowing parallel computations on multi-core CPU and/or Graphics Processing Units (GPUs).

We use the models and MCMC routine as implemented in MDT version 0.15.0. Unless stated otherwise, we initialize the MCMC sampling with a Maximum Likelihood Estimator (MLE) obtained from non-linear parameter optimization using the Powell routine with cascaded model initialization and patience 2 (Harms, Fritz, et al., 2017). Scripts for reproducing the results in this article can be found at [https://github.com/robbert-harms/sampling\\_paper](https://github.com/robbert-harms/sampling_paper).

First, we define and review posteriors, likelihoods and priors relevant to diffusion multi-compartment models. We next define the Metropolis-Hastings as the general type of Markov Chain Monte Carlo algorithms used in this work. Then, under assumptions of symmetric and current position centered proposals, updated one dimension at a time, we derive the Metropolis-Within-Gibbs algorithm. The Metropolis-Within-Gibbs algorithm is then explained with and without the use of adaptive proposals. We subsequently define burn-in, thinning, effective sample size and number of samples as the targets of investigation for diffusion microstructure models.

### 3.2.1 Diffusion microstructure models

The general multi-compartment diffusion microstructure model has the form of a weighted sum of single compartments:

$$\mathbf{S} = S_0 \sum_{i=0}^n w_i S_i \quad (3.1)$$

Where  $S_0$  is the signal for the non-diffusion weighted (or  $b_0$ ) acquisitions,  $w_i$  the volume fractions (signal weights, signal fractions or water fractions) and  $S_i$  is the signal function for the  $i$ 'th of  $n$  total compartments. For this work we selected the Tensor (Basser, Mattiello, and LeBihan, 1994), Ball&Sticks (Behrens et al., 2003), NODDI (H. Zhang, Schneider, et al., 2012) and CHARMED (Assaf, Freidlin, et al., 2004) models. Table 3.1 shows these multi-compartment models (henceforth simply 'models'), their constituent compartments and total number of parameters



including estimation of  $S_0$ . For signal model naming we use the postfix ‘\_in[n]’ to identify the number of restricted compartments employed in models which allow multiple intra-axonal compartments. For example, CHARMED\_in2 indicates a CHARMED model with 2 intra-axonal compartments (and the regular single extra-axonal compartment), for each of two unique fiber orientations in a voxel. Table 3.2 lists the compartments referenced to in table 3.1, with the corresponding optimizable parameters listed in table 3.3. See (Harms, Fritz, et al., 2017) for implementation details of these compartments and multi-compartment models.

Model	Restricted ( <i>intra-</i> cellular) compart- ments	Hindered ( <i>extra-</i> cellular) compart- ments	Isotropic compart- ments	Number of parameters	Acquisition require- ments
Tensor	-	Tensor	-	7	$b < 1.5 \cdot 10^6 \text{ s/m}^2$
Ball&Sticks _in[n]	Stick ( <i>n-</i> times)	-	Ball	$1 + 3n$	-
NODDI	NODDI_in	NODDI_ex	Ball	6	$\geq 2$ b-values / shells
CHARMED _in[n]	CHARMED _in ( <i>n-</i> times)	Tensor	-	$7 + 4n$	$\geq 2$ b-values / shells, $b_{\max} \geq 4.0 \cdot 10^6 \text{ s/m}^2$

Table 3.1: The used composite multi-compartment models, their compartments (divided into intra-axonal, extra-axonal and isotropic) and total number of parameters. For an overview of the individual compartments, please see table 3.2.

Compartment	Signal function	Compartment model parameters
Tensor	$\mathbf{S} = \exp(-b( d_{\parallel}(\mathbf{n} \cdot \mathbf{g})^2 + d_{\perp 1}(\mathbf{n}_{\perp 1} \cdot \mathbf{g})^2 + d_{\perp 2}(\mathbf{n}_{\perp 2} \cdot \mathbf{g})^2))$ $\mathbf{n}_{\perp 1} = \text{rotate}(\mathbf{n}, \psi)$ $\mathbf{n}_{\perp 2} = \mathbf{n} \times \mathbf{n}_{\perp 1}$	$d_{\parallel}, d_{\perp 1}, d_{\perp 2}, \theta, \phi, \psi$
Ball	$\mathbf{S} = e^{-bd}$	$d$
Stick	$\mathbf{S} = e^{-bd(\mathbf{n} \cdot \mathbf{g})^2}$	$d, \theta, \phi$
NODDI <sub>in</sub>	$\mathbf{S} = \int_{\mathbb{S}^2} f(\mathbf{n}, \kappa) e^{-bd(\mathbf{n} \cdot \mathbf{g})^2} d\mathbf{n}$	$d, \theta, \phi, \kappa$
NODDI <sub>ex</sub>	$\mathbf{S} = e^{-b\mathbf{g}^T (\int_{\mathbb{S}^2} f(\mathbf{n}, \kappa) D(\mathbf{n}) d\mathbf{n}) \mathbf{g}}$	$d_{\parallel}, d_{\perp}, \theta, \phi, \kappa$
CHARMED <sub>in</sub>	$\mathbf{S} = \sum_{i=1}^N v_i [ S_{\parallel}(\mathbf{q}, \Delta) \cdot S_{\perp i}(\mathbf{q}, \text{TE}) ]$ $S_{\parallel}(\mathbf{q}, \Delta) = e^{-4\pi^2  \mathbf{q} ^2 (\mathbf{n} \cdot \mathbf{g})^2 (\Delta - \delta/3) d}$ $S_{\perp i}(\mathbf{q}, \text{TE}) = e^{-[4\pi^2  \mathbf{q} ^2 (1 - (\mathbf{n} \cdot \mathbf{g})^2) R_i^4 / (\frac{d \cdot \text{TE}}{2})]} \cdot (\frac{7}{96}) \cdot [2 - (\frac{99}{112}) R_i^2 / (\frac{d \cdot \text{TE}}{2})]}$	$d, \theta, \phi$

Table 3.2: The single compartment models, see Table 3.3 for an overview of the optimizable parameters. The primary direction of diffusivity  $\mathbf{n}$ , is parameterized using polar coordinates with angles  $\theta, \phi$  and radius  $d$ . The variables  $b, \mathbf{g}, \mathbf{q}, \Delta, \delta, G$  and  $\text{TE}$  are sequence settings. In the Tensor compartment, the function  $\text{rotate}(\mathbf{n}, \psi)$  rotates the Tensor around  $\mathbf{n}$  by the angle  $\psi$ . In the NODDI models, the function  $f(\mathbf{n}, \kappa) d\mathbf{n}$  gives the probability of finding fiber bundles along orientation  $\mathbf{n}$  using a Watson distribution with parameter  $\kappa$  integrated over the unit sphere  $\mathbb{S}^2$ . In the NODDI<sub>ex</sub> model, the diffusion tensor  $D(\mathbf{n})$  is defined as a cylindrically symmetric Tensor (alike the Tensor in this table except for the symmetry). In the CHARMED<sub>in</sub> compartment  $N$  is the number of gamma cylinders used,  $v_i$  is the weight per gamma distributed cylinders and  $R_i$  is the radius per cylinder. In previous work  $|\mathbf{q}|^2 (\mathbf{n} \cdot \mathbf{g})^2$  is sometimes denoted as  $|\mathbf{q}_{\parallel}|^2$  and  $|\mathbf{q}|^2 (1 - (\mathbf{n} \cdot \mathbf{g})^2)$  as  $|\mathbf{q}_{\perp}|^2$ , we inlined these identities here in the CHARMED<sub>in</sub> equation.

Parameter	Compartments	Usage
$d_{\parallel}$ (or $d$ )	Tensor, Ball, Stick, NODDI.in, NODDI.ex, CHARMED.in	Parallel diffusivity along the primary direction of diffusion $\mathbf{n}$
$d_{\perp 1}$	Tensor	Perpendicular diffusivity, perpendicular to both $d_{\parallel}$ and $d_{\perp 2}$ .
$d_{\perp 2}$	Tensor	Perpendicular diffusivity, perpendicular to both $d_{\parallel}$ and $d_{\perp 1}$ .
$\theta$	Tensor, Stick, NODDI.in, NODDI.ex, CHARMED.in	Polar angle used to parameterize $\mathbf{n}$ , the primary direction of diffusion
$\phi$	Tensor, Stick, NODDI.in, NODDI.ex, CHARMED.in	Azimuth angle used to parameterize $\mathbf{n}$ , the primary direction of diffusion
$\psi$	Tensor	Used to rotate the Tensor around its primary axis
$\kappa$	NODDI.in, NODDI.ex	The dispersion index of the Watson distribution

Table 3.3: The parameter descriptions corresponding to the diffusion compartment models in table 3.2.

### 3.2.2 Posterior, likelihoods and priors

Given observations  $\mathcal{O}$  and a model with parameters  $\mathbf{x} \in \mathbb{R}^n$ , we can construct a posterior distribution  $p(\mathbf{x}|\mathcal{O})$  from a log-likelihood distribution  $l(\mathcal{O}|\mathbf{x})$  and prior distribution  $p(\mathbf{x})$ , as:

$$\ln p(\mathbf{x}|\mathcal{O}) \propto l(\mathcal{O}|\mathbf{x}) + \ln p(\mathbf{x}) \quad (3.2)$$

In this work we are interested in approximating the posterior density of  $p(\mathbf{x}|\mathcal{O})$  using MCMC sampling.

#### Likelihood distribution

The likelihood distribution  $l(\mathcal{O}|\mathbf{x})$  contains a signal model, embedding the diffusion microstructure modeling assumptions combined with a noise model. As discussed in previous work (Harms, Fritz, et al., 2017; Panagiotaki et al., 2012; Alexander, 2009), we use the Offset Gaussian model as likelihood distribution:

$$l(\mathcal{O}|\mathbf{x}) = -\frac{\sum \left( \mathcal{O} - \sqrt{\mathbf{S}(\mathbf{x})^2 + \sigma^2} \right)^2}{2\sigma^2} - m \cdot \log(\sigma\sqrt{2\pi}) \quad (3.3)$$

with  $l(\mathcal{O}|\mathbf{x})$  the log-likelihood function,  $\mathbf{x}$  the parameter vector,  $\mathcal{O}$  the observations (the data volumes),  $\mathbf{S}(\mathbf{x})$  the signal model,  $\sigma$  the standard deviation of the Gaussian distributed noise (of the complex valued data, i.e. before calculation of magnitude data) and  $m$  the number of volumes in the dataset (number of observations). We estimated  $\sigma$  a priori from the reconstructed magnitude images using the  $\sigma_{\text{mult}}$  method in (Dietrich, Raya, Reeder, Reiser, et al., 2007, eq. A6).

### Priors

The prior distribution  $p(\mathbf{x})$  describes the a priori knowledge we have about the model and its parameters. We construct a complete model prior as a product of priors per parameter,  $p_i(\mathbf{x}_i)$  (see table 3.4), with one or more model specific priors over multiple parameters,  $p_j(\mathbf{x}|M)$ , for model prior  $j$  of model  $M$  (see table 3.5):

$$p(\mathbf{x}) = \prod p_i(\mathbf{x}_i) \cdot \prod p_j(\mathbf{x}|M) \quad (3.4)$$

Assuming no further a priori knowledge than logical or biologically plausible ranges, we use uniform priors for each parameter,  $p_i(\mathbf{x}_i) \sim U(a, b)$ . Additionally, for multi-compartment models with volume fraction weighted compartments (i.e. Ball&Stick\_in[ $n$ ], NODDI and CHARMED\_in[ $n$ ]) we add a prior on the  $n - 1$  volume fractions  $w_k$  to ensure  $\sum_{k=0}^{n-1} w_k \leq 1$  to ensure proper volume fraction weighting. Note that the last volume fraction is not sampled but is set to one minus the sum of the others,  $w_n = 1 - \sum_{k=0}^{n-1} w_k$ . To the Tensor compartment (used in the Tensor and CHARMED\_in1 model), we add a prior to ensure strictly decreasing diffusivities ( $d > d_{\perp 0} > d_{\perp 1}$ ), this prevents parameter aliasing of the Tensor orientation parameters (see (Gelman, Carlin, et al., 2013) on aliasing).

### 3.2.3 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) is a class of numerical approximation algorithms for sampling from the probability density function  $\pi(\cdot)$  of a target random variate, by generating a Markov chain  $\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \dots$  with stationary distribution  $\pi(\cdot)$ . There are a large number of MCMC algorithms, including Metropolis-Within-Gibbs (a.k.a Metropolis) (Metropolis et al., 1953), Metropolis-Hastings (Hastings, 1970), Gibbs (Turchin, 1971; S. Geman and D. Geman, 1984), Component-wise Hit-And-Run Metropolis (Turchin, 1971; R. L. Smith, 1984), Random Walk Metropolis (Muller, 1994), Multiple-Try

Parameter	Prior
$S_0$	$U(0, 1 \cdot 10^{10})$
$w_i$	$U(0, 1)$
$d_{\parallel}$ (or $d$ )	$U(3 \cdot 10^{-11}, 1 \cdot 10^{-8})$
$d_{\perp_1}, d_{\perp_2}$	$U(0, 1 \cdot 10^{-8})$
$\theta, \phi$	$U(0, \pi)$
$\psi$	$U(0, \pi)$
$\kappa$	$U(0, 2\pi)$

Table 3.4: The priors  $p_i(\mathbf{x}_i)$  per model parameter. These priors are combined with the model specific parameters in table 3.5 to form the complete model priors. For parameter usage and specification see table 3.3.

Model ( $M$ )	Prior
BallStick_in[ $n$ ], CHARMED_in[ $n$ ]	NODDI, $p_j(\mathbf{x}, M) = 1$ if $\sum_{k=0}^{n-1} w_k \leq 1$ , else $p_j(\mathbf{x}, M) = 0$
Tensor, CHARMED_in[ $n$ ] (extra ax- onal compartment)	$p_j(\mathbf{x}, M) = 1$ if $d > d_{\perp_0} > d_{\perp_1}$ , else $p_j(\mathbf{x}, M) = 0$

Table 3.5: The model priors  $p_j(\mathbf{x}|M)$  for model  $M$ . Each of these priors should be interpreted as a boolean, that is, they return a value of 1 if the condition is fulfilled, else they return 0. These priors are combined with the parameter specific priors in table 3.4 to form the complete model priors.

Metropolis (J. S. Liu, Liang, and Wong, 2000), No-U-Turn sampler (Hoffman and Gelman, 2011) and many more. All of these algorithms are known as special cases of the Metropolis-Hastings algorithm and differ only in the proposal distributions they employ (A. A. Johnson, G. L. Jones, and Neath, 2013; Chib and Greenberg, 1995).

The general Metropolis-Hastings algorithm works as follows. Given a current position  $\mathbf{X}^{(t)}$  at step  $t$  on a  $p$ -dimensional Markov chain, a new position  $\mathbf{X}^{(t+1)}$  is obtained by generating a candidate position  $\mathbf{Y}$  from the proposal density  $q(\mathbf{X}^{(t)}|\cdot)$ , which is then either accepted with probability  $\alpha$ , or rejected with probability  $1-\alpha$ . If the candidate position is accepted,  $\mathbf{X}^{(t+1)} = \mathbf{Y}$ , else,  $\mathbf{X}^{(t+1)} = \mathbf{X}^{(t)}$ . The acceptance criteria  $\alpha$  is a function given by (Hastings, 1970):

$$\alpha(\mathbf{X}^{(t)}, \mathbf{Y}) = \min \left( 1, \frac{\pi(\mathbf{Y})}{\pi(\mathbf{X}^{(t)})} \frac{q(\mathbf{X}^{(t)}|\mathbf{Y})}{q(\mathbf{Y}|\mathbf{X}^{(t)})} \right) \quad (3.5)$$

where  $\pi(\cdot)$  is our target density, generally given by our posterior distribution function  $p(\mathbf{X}|\cdot)$ . The subsequent collection of points  $\{\mathbf{X}^{(0)}, \dots, \mathbf{X}^{(s)}\}$  for a sample size  $s$  is called the chain and is the algorithm's output. The *ergodic* property of this algorithm guarantees that this chain converges (in the long run) to a stationary distribution which approximates the target density function  $\pi(\cdot)$  (Metropolis et al., 1953; Hastings, 1970).

In this work we use a component wise updating scheme in which a new position  $\mathbf{X}^{(t+1)}$  is proposed one component (i.e. one dimension) at a time, in contrast to updating all  $p$  dimensions at once. Combined with a symmetric proposal distribution centered around the current sampling position (for every component), this scheme is typically referred to as Metropolis-Within-Gibbs (MWG; Ravenzwaaij, Cassey, and Brown, 2016; Robert, 2015; Sherlock, Fearnhead, and G. O. Roberts, 2010). Let  $\mathbf{X}^{(t)} = (\mathbf{X}_0^{(t)}, \dots, \mathbf{X}_p^{(t)})$  define the components  $\mathbf{X}_i^{(t)}$  of  $\mathbf{X}^{(t)}$ , then we can define

$$\mathbf{Y}_i = (\mathbf{X}_0^{(t+1)}, \dots, \mathbf{X}_{i-1}^{(t+1)}, \mathbf{Y}_i^*, \mathbf{X}_{i+1}^{(t)}, \dots, \mathbf{X}_p^{(t)})$$

as the candidate position for component  $i$ , and

$$\mathbf{X}^{(t+1)*} = (\mathbf{X}_0^{(t+1)}, \dots, \mathbf{X}_{i-1}^{(t+1)}, \mathbf{X}_i^{(t)}, \mathbf{X}_{i+1}^{(t)}, \dots, \mathbf{X}_p^{(t)})$$

as the temporary position in the chain while component  $i$  is being updated. The proposals  $\mathbf{Y}_i^*$  are generated using the symmetric proposal  $q_i(\mathbf{X}^{(t+1)*}|\cdot)$  which updates the  $i$ th component dependent on the components already updated. One iteration of the MWG algorithm cycles through all  $i$  components, where each proposal  $\mathbf{Y}_i$  is accepted or rejected using probability  $\alpha(\mathbf{X}^{(t+1)*}, \mathbf{Y}_i)$ .

### 3.2.4 Proposal distributions

As symmetric proposal distributions for our MWG algorithm we use centered Normal distributions, i.e.  $q_i(\mathbf{X}^{(t+1)*}|\cdot) \sim \mathcal{N}(\mathbf{X}_i^{(t)}, \sigma_i)$ , where  $\sigma_i$  is the proposal standard deviation of the  $i$ th component (not to be confused with the  $\sigma$  used in the likelihood distribution above). For the orientation parameter  $\psi$  we use a circular Normal modulus  $\pi$ , i.e.  $q_i(\mathbf{X}^{(t+1)*}|\cdot) \sim \mathcal{N}(\mathbf{X}_i^{(t)}, \sigma_i) \bmod \pi$ . The orientation parameters  $\theta$  and  $\phi$  are proposed using a standard Normal distribution, but are immediately transformed together such that the corresponding vector lies in the right hemisphere of the unit circle. See appendix table 3.6 for an overview of the default proposal distributions used per parameter.

Parameter	Proposal distribution
$S_0$	$\mathcal{N}(\mathbf{X}_i^{(t)}, 10)$
$w_i$	$\mathcal{N}(\mathbf{X}_i^{(t)}, 0.01)$
$d, d_{\parallel}, d_{\perp_1}, d_{\perp_2}$	$\mathcal{N}(\mathbf{X}_i^{(t)}, 1 \cdot 10^{-10})$
$\psi$	$\mathcal{N}(\mathbf{X}_i^{(t)}, 0.1) \bmod \pi$
$\theta, \phi$	$\mathcal{N}(\mathbf{X}_i^{(t)}, 0.1)$
$\kappa$	$\mathcal{N}(\mathbf{X}_i^{(t)}, 0.01)$

Table 3.6: The proposal distributions  $q_i(\mathbf{X}^{(t+1)} \mid \cdot)$  per model parameter with their default proposal standard deviations. For parameter usage and specification see table 3.3.

### 3.2.5 Adaptive Metropolis

While in the traditional Metropolis-Within-Gibbs algorithm each  $\sigma_i$  in the proposal distribution is fixed, variations of this algorithm exist that auto-tune each  $\sigma_i$  to improve the information content of the Markov chain. While technically each of these variations is a distinct MCMC algorithm, we consider and compare three of these variations here as proposal updating strategies for the MWG algorithm.

The first adaptation strategy compared is the Single Component Adaptive Metropolis (SCAM) algorithm (Haario, Saksman, and Tamminen, 2005), which works by adapting the proposal standard deviation to the empirical standard deviation of the component's marginal distribution. That is, the standard deviation  $\sigma_i^{(t)}$  for the proposal distribution of the  $i$ th component at time  $t$  is given by:

$$\sigma_i^{(t)} = \begin{cases} \sigma_i^{(0)}, & t \leq t_s \\ 2.4 * (\sqrt{\text{Var}(\mathbf{X}_i^{(0)}, \dots, \mathbf{X}_i^{(t-1)})} + \epsilon_i), & t > t_s \end{cases} \quad (3.6)$$

where  $t_s$  denotes the iteration after which the adaptation starts (we use  $t_s = 100$ ). A small constant  $\epsilon_i$  is necessary to prevent the standard deviation from shrinking to zero (we use  $\epsilon_i = 10^{-5} \cdot \sigma_i^{(0)}$ ). The SCAM algorithm has been proven to retain ergodicity, meaning it is guaranteed to converge to the right stationary distribution in the limit of infinite samples (Haario, Saksman, and Tamminen, 2005).

The other two methods work by adapting the acceptance rate of the generated proposals. The acceptance rate is the ratio of accepted to generated proposals and

is typically updated batch-wise. In general, by decreasing the proposal standard deviation the acceptance rate increases and vice versa. Theoretically, for single component updating schemes (like in this work), the optimal target acceptance rate is 0.44 (Gelman, G. Roberts, and Gilks, 1996).

The first of the two acceptance rate scaling strategies is from the FSL BedpostX software package. This strategy, which we refer to as the *FSL* strategy, tunes the acceptance rate to 0.5 (Behrens et al., 2003). It works by multiplying the proposal variance by the ratio of accepted to rejected samples, i.e. it multiplies the standard deviation  $\sigma_i$  by  $\sqrt{(a+1)/(b-a+1)}$  after every batch of size  $b$  with  $a$  accepted samples. We update the proposals after every batch of size 50 ( $b = 50$ ) (Behrens et al., 2003). Since this method never ceases the adaptation of the standard deviations, it theoretically loses ergodicity of the chain (G. O. Roberts and Rosenthal, 2009; G. O. Roberts and Rosenthal, 2007).

The last method, the Adaptive Metropolis-Within-Gibbs (*AMWG*) method (G. O. Roberts and Rosenthal, 2009) uses the current acceptance rate over batches to tune the acceptance rate to 0.44. After the  $n$ th batch of 50 iterations (G. O. Roberts and Rosenthal, 2009), this method updates the logarithm of  $\sigma_i$  by adding or subtracting an adoption amount  $\delta(n) = \sqrt{1/n}$  depending on the acceptance rate of that batch. That is, after every batch,  $\sigma_i$  is updated by:

$$\sigma_i^{(t)} = \begin{cases} \sigma_i^{(t-n)} \cdot \exp(\delta(n)), & \text{ar}_{batch} > \text{ar}_{target} \\ \sigma_i^{(t-n)} / \exp(\delta(n)), & \text{ar}_{batch} \leq \text{ar}_{target} \end{cases} \quad (3.7)$$

where  $\text{ar}_{batch}$  is the acceptance rate of the current batch and  $\text{ar}_{target}$  is the target acceptance rate (0.44). Since this method features diminishing adaptation, the chain remains ergodic (G. O. Roberts and Rosenthal, 2009).

We compare all three strategies and the default, no adaptation, on the number of effective samples they generate (see later section) and on accuracy and precision using ground truth simulation data. We sample all models with 20000 samples, without thinning and using the point optimized Maximum Likelihood Estimator (MLE) as starting point. We report statistics over the first 10000 samples in the article, considering it the common number of samples in MCMC sampling. Estimates of the standard error of the mean (SEM) are obtained by averaging the statistics over 10 independent MCMC runs.

### 3.2.6 Burn-in

Burn-in is the process of discarding the first  $z$  samples from the chain and using only the remaining samples in subsequent analysis. The idea is that if the starting point had a low probability then the limited number of early samples may over



sample low probability regions. By discarding the first  $z$  samples as a burn-in, the hope is that, by then, the chain has converged to its stationary distribution and that all further samples are directly from the stationary distribution (Robert, 2015). Theoretically, burn-in is unnecessary since any empirical average

$$\hat{\mu}_T(g) = \frac{1}{T} \sum_{t=0}^T g(\mathbf{X}^{(t)}) \quad (3.8)$$

for any function  $g$  will convert to  $\mu(g)$  given a large enough sample size and given that the chain is ergodic (Robert, 2015). Additionally, since it can not be predicted how long it will take for the chain to reach convergence, the required burn-in can only be estimated post-hoc. In practice, discarding the first few thousand samples as a burn-in often works and is less time-consuming than generating a lot of samples to average out the effects of a low probability starting position.

An alternative to burn-in, or, to reduce the need for burn-in, is to use a Maximum Likelihood Estimator as starting point for the MCMC sampling (Ravenzwaaij, Cassey, and Brown, 2016). If the optimization routine did its work well, the MLE should be part of the stationary distribution of the Markov chain, removing the need for burn-in altogether. We compare initialization using a MLE obtained using the Powell routine (Harms, Fritz, et al., 2017), with a initialization from a default a priori value (see table 3.7). For most models the MLE optimization results can be used directly, for the Tensor model we occasionally need to sort the diffusivities and reorient the  $\theta$ ,  $\phi$  and  $\psi$  angles to ensure decreasing diffusivities. To evaluate the effect of burn-in and initialization single-slice datas was sampled using the NODDI model with the default starting point and with MLE. For selected single voxels the NODDI model was also sampled using the MLE starting point and two random volume fractions as a starting point. We compare these starting points on moving mean and moving standard deviation, as well as on autocorrelation (the correlation of a chain with itself) given by:

$$R(s, t) = \frac{\mathbb{E}[(X_t - \mu_t)(X_s - \mu_s)]}{\sigma_t \sigma_s} \quad (3.9)$$

with mean  $\mu_t$ , variance  $\sigma_t^2$  at time  $t$ , this computes the autocorrelation  $R(s, t)$  between times  $s$  and  $t$ .

### 3.2.7 Thinning

Thinning is the process of using only every  $k$ th step of the chain for analysis, while all other steps are discarded, with as goal reducing autocorrelation and

Parameter	Default starting value
$S_0$	$1 \cdot 10^4$
$w_i$	0.5
$d_{\parallel}$ (or $d$ )	$1.7 \cdot 10^{-9}$
$d_{\perp_1}$	$1.7 \cdot 10^{-10}$
$d_{\perp_2}$	$1.7 \cdot 10^{-11}$
$\theta, \phi, \psi$	$\pi/2$
$\kappa$	1

Table 3.7: The default starting points for the MCMC sampler, used only when the sampler is not initialized with a maximum likelihood estimator.

obtaining relatively independent samples. Several authors have recommended against the use of thinning, stating that it is often unnecessary, always inefficient and reduces the precision of the posterior estimates (Link and Eaton, 2012; MacEachern and Berliner, 1994; Jackman, 2009; Geyer, 1991; Christensen et al., 2010).

The only valid reason for thinning is to avoid bias in the standard error estimate of posterior mean, when that mean estimate was computed over all (non thinned) samples (Link and Eaton, 2012; MacEachern and Berliner, 1994). In general, thinning is only considered worthwhile if there are storage limitations, or when the cost of processing the output outweighs the benefits of reduced variance of the estimator (Geyer, 1991; MacEachern and Berliner, 1994; Link and Eaton, 2012).

To evaluate the effect of thinning we sampled a single voxel with 20000 samples and compared the effect of using all samples in computing the posterior mean and posterior standard deviation of a volume fraction against using only a thinned amount of samples. We compare the effect of taking  $n$  samples with a thinning of  $k$  (the *thinning* method) against just using all  $n \cdot k$  samples (the *more samples* method).

### 3.2.8 Effective Sample Size

The Effective Sample Size (ESS) in the context of MCMC, measures the information content, or effectiveness of a sample chain. For example, 1000 samples with an ESS of 200 have a higher information content than 2000 samples with an ESS of 100. The ESS can be defined as the minimum size of a set of posterior samples (taken directly from the posterior), which have the same efficiency (measure of quality) in the posterior density estimation as a given chain of samples obtained

from MCMC sampling (Martino, Elvira, and Louzada, 2017). Conversely, ESS theory can quantify how many samples should be taken in a chain to reach a given quality of posterior estimates. We use the ESS theory to comparing proposal adaptation strategies and to estimating the minimum number of samples necessary for adequate sampling of diffusion microstructure models.

Multivariate ESS theory (Vats, Flegal, and G. L. Jones, 2015) is an extension of univariate ESS theory (Gong and Flegal, 2016; J. S. Liu, 2004; Robert and Casella, 2004; Kass et al., 1998) and computes the empirical ESS as:

$$\widehat{\text{ESS}} = s \left( \frac{|\Lambda_s|}{|\Sigma_s|} \right)^{1/p} \quad (3.10)$$

with  $s$  is the number of obtained samples,  $p$  the number of parameters,  $\Lambda_s$  the covariance matrix of the samples and  $\Sigma_s$  an estimate of the Monte Carlo standard error (the error in the chain caused by the MCMC sampling process), here calculated using a batch means algorithm (Vats, Flegal, and G. L. Jones, 2015).

### 3.2.9 Number of samples

The multivariate ESS theory dictates that one can terminate the sampling when the empirical number of effective samples,  $\widehat{\text{ESS}}$ , satisfies:

$$\widehat{\text{ESS}} \geq W(p, \alpha, \epsilon) \quad (3.11)$$

where  $W(p, \alpha, \epsilon)$  gives a theoretical lower bound with  $p$  the number of parameters in the model,  $\alpha$  the level of confidence of a desired confidence region and  $\epsilon$  a desired relative precision (the relative contribution of Monte Carlo error to the variability in the target distribution).  $W(p, \alpha, \epsilon)$  can be determined a priori and is defined as:

$$W(p, \alpha, \epsilon) = \frac{2^{2/p} \pi}{(p\Gamma(p/2))^{2/p}} \frac{\chi_{1-\alpha, p}^2}{\epsilon^2} \quad (3.12)$$

with  $\chi^2$  the chi-square function and  $\Gamma(\cdot)$  the Gamma function (Vats, Flegal, and G. L. Jones, 2015). Figure 3.2 shows the effect of  $\alpha$  and  $\epsilon$  on  $W(p, \alpha, \epsilon)$ . Given the exponential increase in the number of samples need for very high confidence and precision, we aim for a 95% confidence region ( $\alpha = 0.05$ ) with a 90% precision ( $\epsilon = 0.1$ ) in this work.

Since online monitoring of the ESS (during MCMC sampling) is an expensive operation, and terminating on ESS will yield different sample sizes for different voxels, we instead use the ESS theory to estimate a fixed minimum number of samples needed to reach a desired ESS when averaged over a white matter mask. We sampled with the BallStick\_in1, BallStick\_in2, BallStick\_in3, Tensor, NODDI, CHARMED\_in1, CHARMED\_in2 and CHARMED\_in3 models, using respectively 15000, 20000, 25000, 20000, 20000, 30000, 40000 and 50000 samples and computed from those samples the average ESS over three slices in the white matter masks, namely the center slice of the volume, one slice five slices below the center and one slice five slices above. For  $\alpha = 0.05$  and  $\epsilon = 0.1$  we computed per model the theoretical minimum required effective sample size  $W(p, \alpha, \epsilon)$ . We compared those theoretical numbers to the obtained average effective sample size and estimated a minimum required number of samples  $\hat{s}$  using the ratio:

$$\hat{s} = s + \frac{W(p, \alpha, \epsilon) - \widehat{\text{ESS}}}{\widehat{\text{ESS}}/s} \quad (3.13)$$

where  $s$  is the number of samples we started out with,  $W(p, \alpha, \epsilon)$  the theoretical ESS requirements and  $\widehat{\text{ESS}}$  the estimated number of effective samples in our chain when averaged over the white matter mask. As an estimate of computation times, we record runtime statistics for sampling the recommended number of samples using an AMD Fury X graphics card and an Intel Xeon e3 18 core CPU.

### 3.2.10 Datasets

For this study we used two groups of ten subjects coming from two studies, each with a different acquisition protocol. The first ten subjects are from the freely available fully preprocessed dMRI data from the USC-Harvard consortium of the Human Connectome project. Data used in the preparation of this work were obtained from the MGH-USC Human Connectome Project (HCP) database (<https://ida.loni.usc.edu/login.jsp>). The data were acquired on a specialized Siemens Magnetom Connectom with 300mT/m gradient set (Siemens, Erlangen, Germany). These datasets were acquired at a resolution of 1.5mm isotropic with  $\Delta=21.8\text{ms}$ ,  $\delta=12.9\text{ms}$ ,  $\text{TE}=57\text{ms}$ ,  $\text{TR}=8800\text{ms}$ , Partial Fourier = 6/8, MB factor 1 (i.e. no simultaneous multi-slice), in-plane GRAPPA acceleration factor 3, with 4 shells of  $b=1000, 3000, 5000, 10,000 \text{ s/mm}^2$ , with respectively 64, 64, 128, 393 directions to which are added 40 interleaved  $b_0$  volumes leading to 552 volumes in total per subject, with an acquisition time of 89 minutes. We refer to these datasets as *HCP MGH - 1.5mm -552vol - b10k* and to the multi-shell direction table as the *HCP MGH* table. These four-shell, high number of directions, and very high maximum  $b$ - value datasets allow a wide range of models to be fitted.

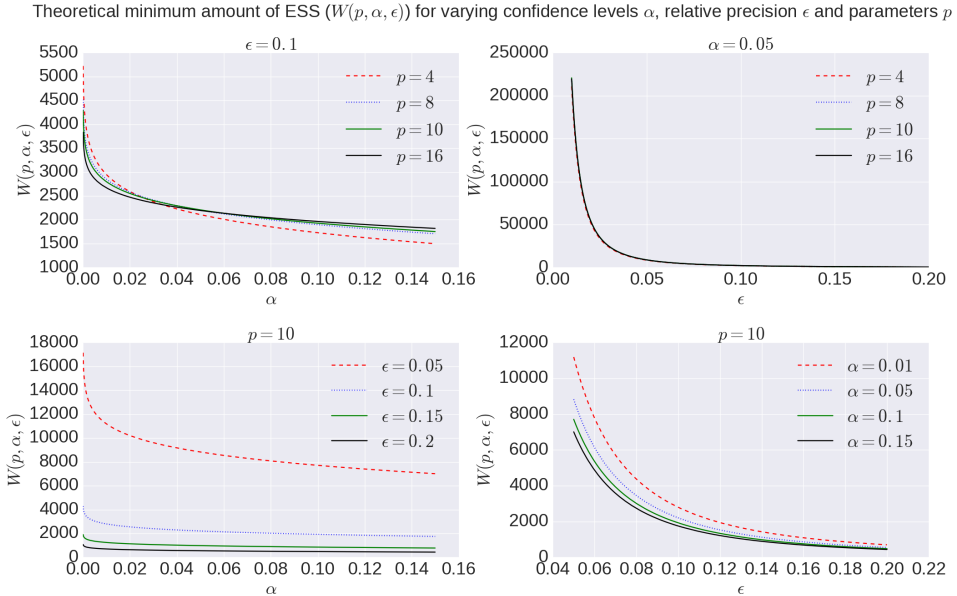


Figure 3.2: Overview of theoretical minimum ESS,  $W(p, \alpha, \epsilon)$ , to reach a specific confidence level  $\alpha$  with a desired relative precision  $\epsilon$  for a model with number of parameters  $p$ .

The second set of ten subjects comes from the diffusion protocol pilot phase of the Rhineland Study ([www.rheinland-studie.de](http://www.rheinland-studie.de)) and was acquired on a Siemens Magnetom Prisma (Siemens, Erlangen, Germany) with the Center for Magnetic Resonance Research (CMRR) multi-band (MB) diffusion sequence (Moeller et al., 2010; Xu et al., 2013). These datasets had a resolution of 2.0mm isotropic with  $\Delta=45.8\text{ms}$ ,  $\delta=16.3\text{ms}$  and  $\text{TE}=90\text{ms}$ ,  $\text{TR}=4500\text{ms}$  Partial Fourier = 6/8, MB factor 3, no in-plane acceleration with 3 shells of  $b=1000, 2000, 3000 \text{ s/mm}^2$ , with respectively 30, 40 and 50 directions to which are added 14 interleaved  $b_0$  volumes leading to 134 volumes in total per subject, with an acquisition time of 10 min 21 sec. Additional  $b_0$  volumes were acquired with a reversed phase encoding direction which were used to correct susceptibility related distortion (in addition to bulk subject motion) with the topup and eddy tools in FSL version 5.0.9. We refer to these datasets as *RLS-pilot - 2mm - 134dir - b3k* and to the multi-shell direction table as the *RLS-pilot* table. These three-shell datasets represent a relatively short time acquisition protocol that still allows many models to be fitted.

Since the CHARMED<sub>in</sub>[ $n$ ] models require relatively high  $b$ -values ( $\sim 10000$ ), which are not present in the RLS-pilot dataset, we will only use the HCP-MGH dataset when showing CHARMED<sub>in</sub>[ $n$ ] results. Additionally, since the Tensor model is only valid for  $b$ -values up to about  $1200\text{s/mm}^2$ , we only use the  $b$ -value

1000s/mm<sup>2</sup> shell and b0 volumes during model optimization and sampling. All other models are estimated on all data volumes. For all datasets we created a white matter (WM) mask and, using BET from FSL (S. M. Smith, 2002), a whole brain mask. The whole brain mask is used during sampling, whereas averages over the WM mask are used in model or data comparisons. The WM mask was calculated by applying a lower threshold of 0.3 on the Tensor FA results, followed by a double pass 3D median filter of radius 2 in all directions. The Tensor estimate for this mask generation was calculated using a CI Ball Stick/Tensor cascade optimized with the Powell method (Harms, Fritz, et al., 2017).

### 3.2.11 Ground truth simulations

We performed ground truth simulations to illustrate the effects of the adaptive proposals on effective sample sizes and on accuracy and precision of parameter estimation. For all models in the study, we simulated 10000 repeats with random volume fractions, diffusivities and orientations, using both a HCP MGH and a RLS-pilot multi-shell direction table with Rician noise of an SNR of 30. For the Tensor model we only use the b-value 1000s/mm<sup>2</sup> shell and b0 volumes of the acquisition tables. To ensure Gaussianity of the sampled parameter distributions, we generate the parameters with a smaller range than the support of the sampling priors (table 3.8). To allow a uniform SNR of 30 we fix  $S_0$  to  $1 \cdot 10^4$ .

Parameter	Lower bound	Upper bound
$w_i$	0.2	0.8
$d_{\parallel}, d_{\perp_1}, d_{\perp_2}$	$5 \cdot 10^{-11}$	$5 \cdot 10^{-9}$
$\theta, \phi, \psi$	0	$\pi$
$\kappa$	0.1	60

Table 3.8: The simulation ranges per model parameters. We generate uniformly distributed parameter values using the upper and lower bounds presented.

Analogous to (Harms, Fritz, et al., 2017), we compute estimation error as the mean of the (marginal) posterior minus ground truth parameter value for the intra-axonal volume fraction, i.e. fraction of stick (FS) for Ball&Sticks\_in1, fraction of restricted (FR) for CHARMED\_in1 and fraction of restricted (FR) for NODDI. We compute a measure of accuracy as the inverse of the mean of the average estimate error over ten thousand random repeats and a measure of precision as the inverse of the standard deviation of the average estimates. Finally, we aggregate these results per model and per experiment over 10 independent ground truth

simulation trials into a mean and standard error of the mean (SEM) for both accuracy and precision. When reported, the effective sample size (ESS) is computed using the multivariate ESS theory, averaged over the 10000 voxels with again an SEM over 10 trials.

## 3.3 Results

We begin by comparing the four different proposal strategies for sampling the different microstructure compartment models: Tensor, Ball&Sticks.in1, CHARMED.in1 and NODDI. We then present burn-in and thinning given an effective proposal strategy, and end with ESS estimates on the minimum number of samples needed for adequate characterization of the posterior distribution.

### 3.3.1 Adaptive proposal strategies

We compare three different adaptive proposal strategies, the Single Component Adaptive Metropolis (SCAM), the FSL acceptance rate scaling (FSL) and the Adaptive Metropolis-Within-Gibbs (AMWG), against the default of no adaptive proposals (None). Comparisons are based on multivariate Effective Sample Size, and accuracy and precision using ground truth simulations. Figure 3.3 illustrates the effect of using MCMC algorithms with adaptive proposal strategies using the Ball&Stick.in1 model, the HCP MGH dataset, an initial standard deviation of 0.25, after a burn-in of 1000 steps. The illustration clearly shows that without adaptive proposals the chain can get stuck in the same position for quite some time, while all adaptive proposal methods can adapt the standard deviations to better cover the support of the posterior distribution.

The empirical ESS (eq. 3.10) measures the information content or effectiveness of a sample chain. As such, comparing the ESS for an equal number of actual samples for different proposal strategies evaluates how effectively each strategy generates useful information about the posterior distribution. Figure 3.4 shows that all adaptive methods clearly outperform the default, None, by generating at least 2~3 times more effective samples for equal length chains. The AMWG method generates the largest ESS in all cases, with a considerable margin with respect to SCAM and with a small margin compared to FSL, with margins increasing somewhat for more complex models and the larger HCP MGH protocol. Compared on accuracy and precision in ground truth simulations (figure 3.5), the SCAM strategy performs slightly better (highest accuracy and precision) than the other adaptive methods for the lower number of parameter models (Tensor, NODDI) while the AMWG and FSL methods perform considerably better in the higher

number of parameter crossing fiber models (NODDI, CHARMED\_in1). Repeating, with double number of samples, the simulations of empirical ESS (Supplementary Figure 1) and accuracy and precision (Supplementary Figure 2), reproduces these results. Given the all-round efficiency, accuracy and precision, and the maintained ergodicity of the chain in the AMWG method, we selected this method to generate chains in the rest of this work.

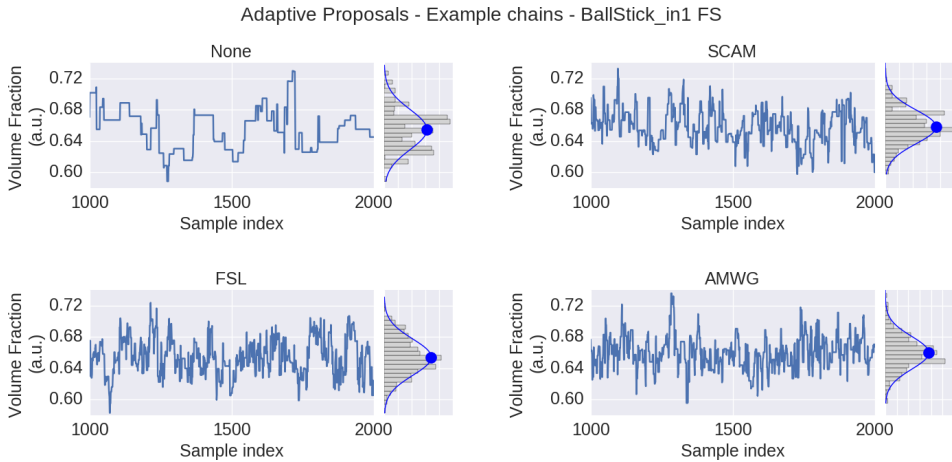


Figure 3.3: MCMC sample traces for the voxel indicated in figure 3.1, using Ball&Stick\_in1 Fraction of Stick (FS), for no adaptive metropolis (None), the Single Component Adaptive Metropolis (SCAM), the FSL acceptance rate scaling (FSL) and Adaptive Metropolis-Within-Gibbs (AMWG) adaptive proposal methods. Results were computed with an initial proposal standard deviation of 0.25. A Gaussian distribution function was fitted to the samples, superimposed in blue on the sample histograms, with its mean indicated by the blue dot.



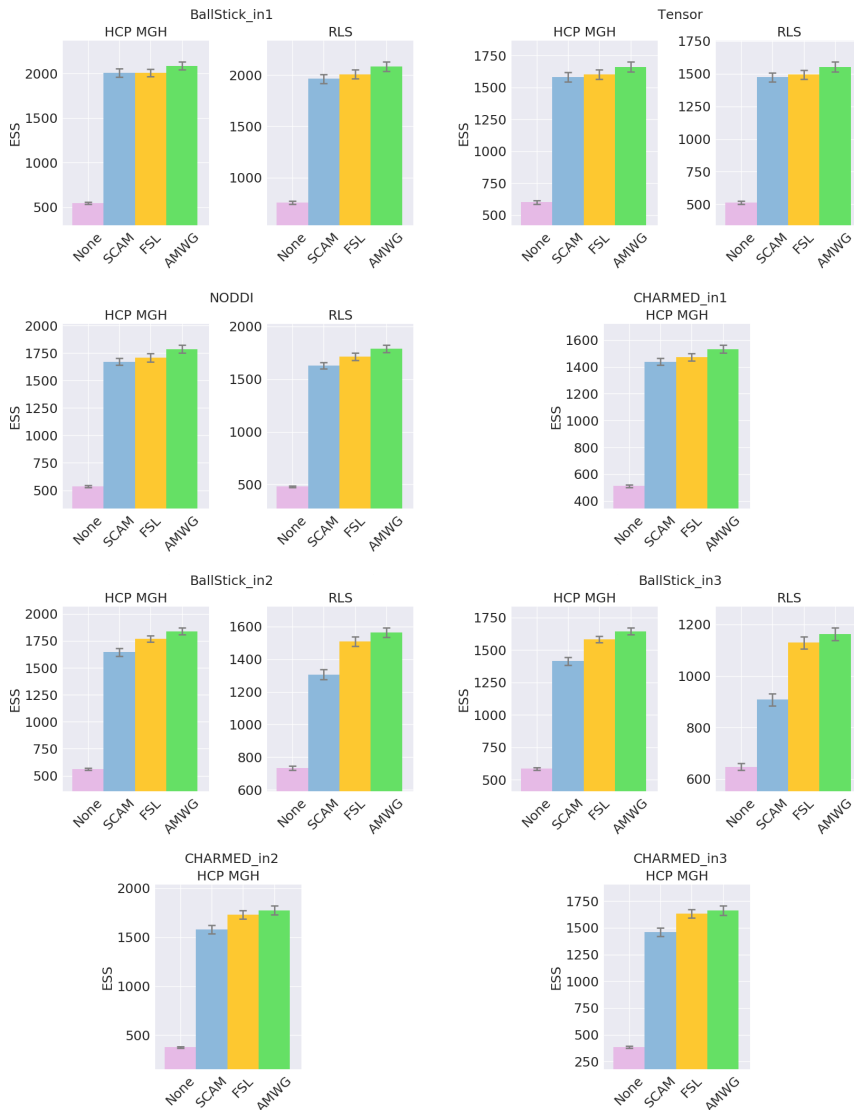


Figure 3.4: Estimated multivariate Effective Sample Size (ESS), for no adaptive metropolis (None), the Single Component Adaptive Metropolis (SCAM), the FSL acceptance rate scaling (FSL) and Adaptive Metropolis-Within-Gibbs (AMWG) adaptive proposal methods. Whiskers show the standard error of the mean computed over 10 repeats. Results are over 10000 samples, without burn-in and thinning.

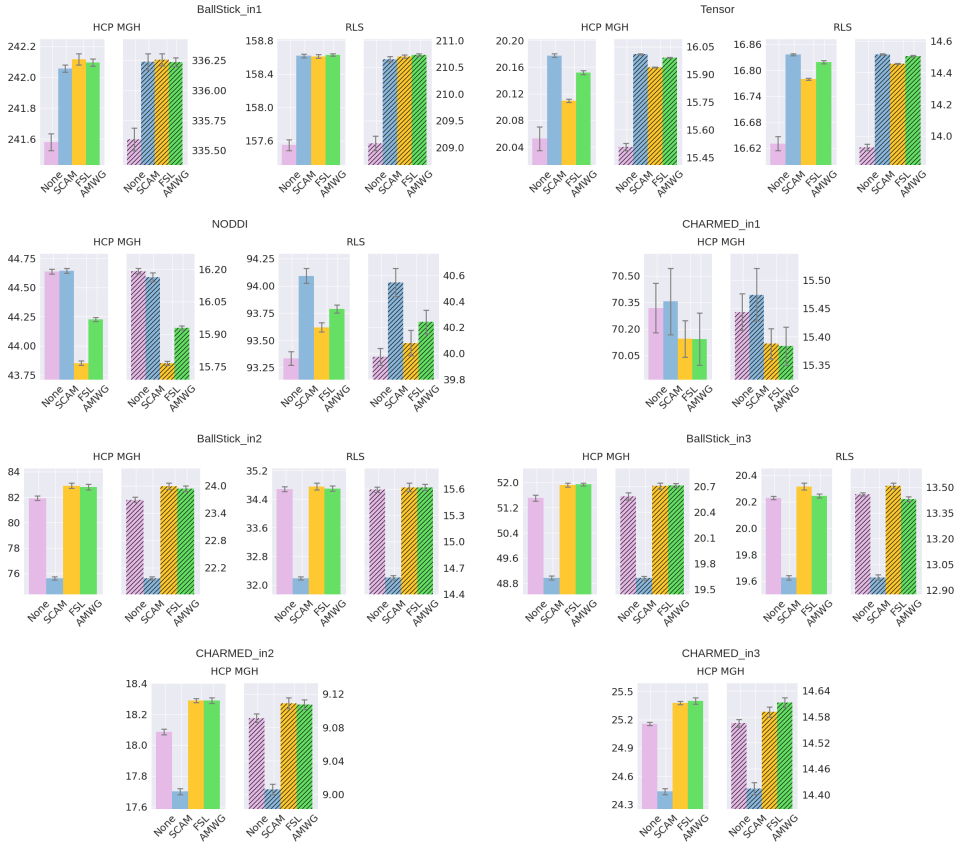


Figure 3.5: Estimated accuracy (left plots) and precision (right, shaded, plots), for no adaptive metropolis (None), the Single Component Adaptive Metropolis (SCAM), the FSL acceptance rate scaling (FSL) and Adaptive Metropolis-Within-Gibbs (AMWG) adaptive proposal methods. The results are averaged over 10000 voxels and 10 trials, the whiskers show the standard error of the mean computed over the 10 trials. Results are over 10000 samples, without burn-in and thinning. The reported offsets need to be added to the y-axis for absolute results.

### 3.3.2 Burn-in

Figure 3.6 shows a comparison of mean and standard deviation estimates over 10,000 samples (no thinning), between sampling started from the default starting point (table 3.7) and from a Maximum Likelihood Estimator starting point, over an increasing length of burn-in. When started from a default starting point, the chains of most voxels will have converged to their stationary distribution after a burn-in of about 3000 samples. When started from an MLE starting point, the chain starts from a point in the stationary distribution and no burn-in is necessary. Starting from an MLE starting point has the additional advantage of removing salt- and pepper-like noise from the mean and std. maps. For example, even after a burn-in of 3000 samples, there are still some voxels in the default starting point maps that have not converged yet. Burn-in also seems to have a greater impact on the standard deviation estimates than it does on the mean estimates. After a burn-in of 1000 samples, the means of the default starting point maps seem to have converged, while the many of the standard deviations clearly have not. In contrast, stable standard deviation estimates are obtained from the MLE initialized chain even without burn-in.

To illustrate this on a single chain basis, in figure 3.7 we plot the first 1000 samples of an MCMC run of the Ball&Stick\_in1 and NODDI model, for the voxel indicated in figure 3.6, using the MLE starting point and two random volume fractions as a starting point, with the sampling traces in the top row showing how the sampler moves through the parameter space before converging to the stationary distribution. The second row shows the effect of discarding the first  $z$  samples when computing the posterior mean and standard deviation (with statistics over 1000 samples, after the burn-in  $z$ ), and finally in the third row autocorrelation plots for the chains of each starting point method. Interestingly, the default initialized points first seem to move toward an intra-axonal volume fraction of zero, before moving up again. This is probably caused by a misalignment of the model orientation with the data's diffusion orientation, making the intra-axonal volume less likely. Only after a correct orientation of the model, the volume fraction can go up again. The moving mean and moving standard deviation plots in the second row show the convergence of the mean and standard deviation with an increased burn-in length. These plots again show that, when started from the MLE, no burn-in is needed, while starting from another position some burn-in is required for the chains to converge. The autocorrelation also confirm that the MLE initialized chain starts from a converged state, whereas the default initialized chains are far from convergence.

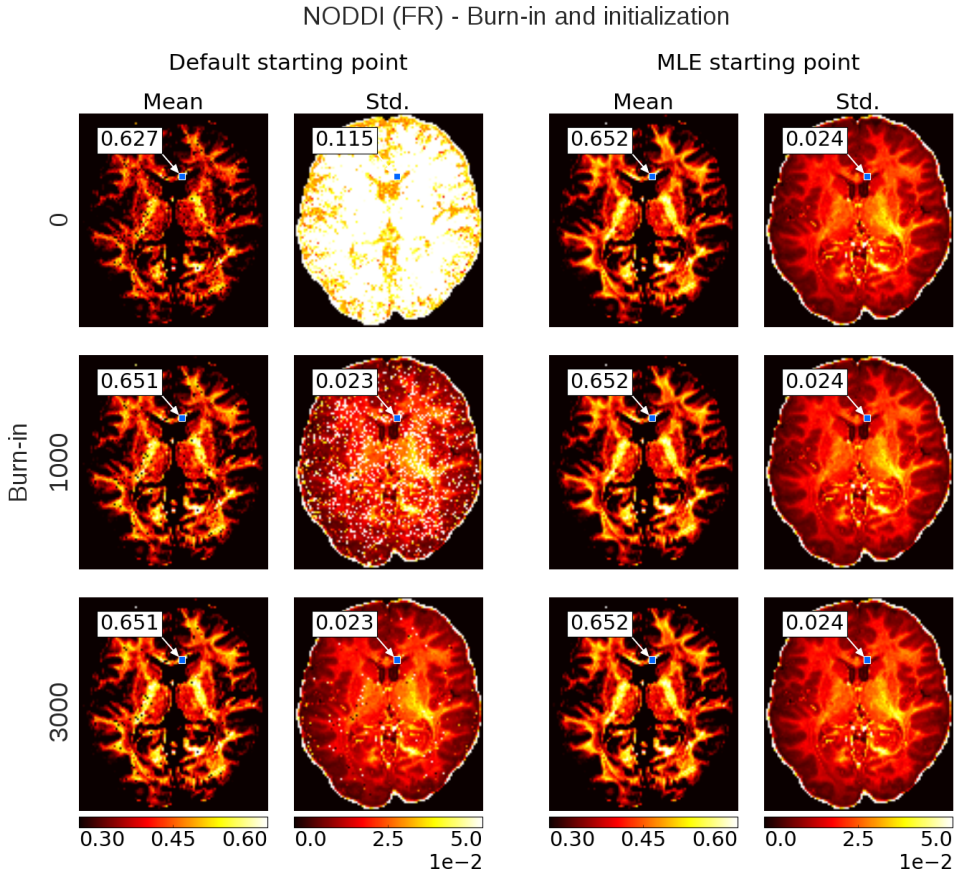


Figure 3.6: Burn-in demonstration and chain initialization using NODDI Fraction of Restricted (FR). On the left, the posterior mean and standard deviation (std.) maps when sampling NODDI from the MDT default starting point, on the right the mean and std. maps when sampling NODDI using a Maximum Likelihood Estimator (MLE) as starting point. The rows show the effect of discarding the first  $z \in \{0, 1000, 3000\}$  samples as burn-in before the mean and std. estimation. Statistics are without thinning and over 10,000 samples after  $z$ . The value insets show the mean and standard deviation value from a Gaussian fit to the sampling chain for the indicated voxel.

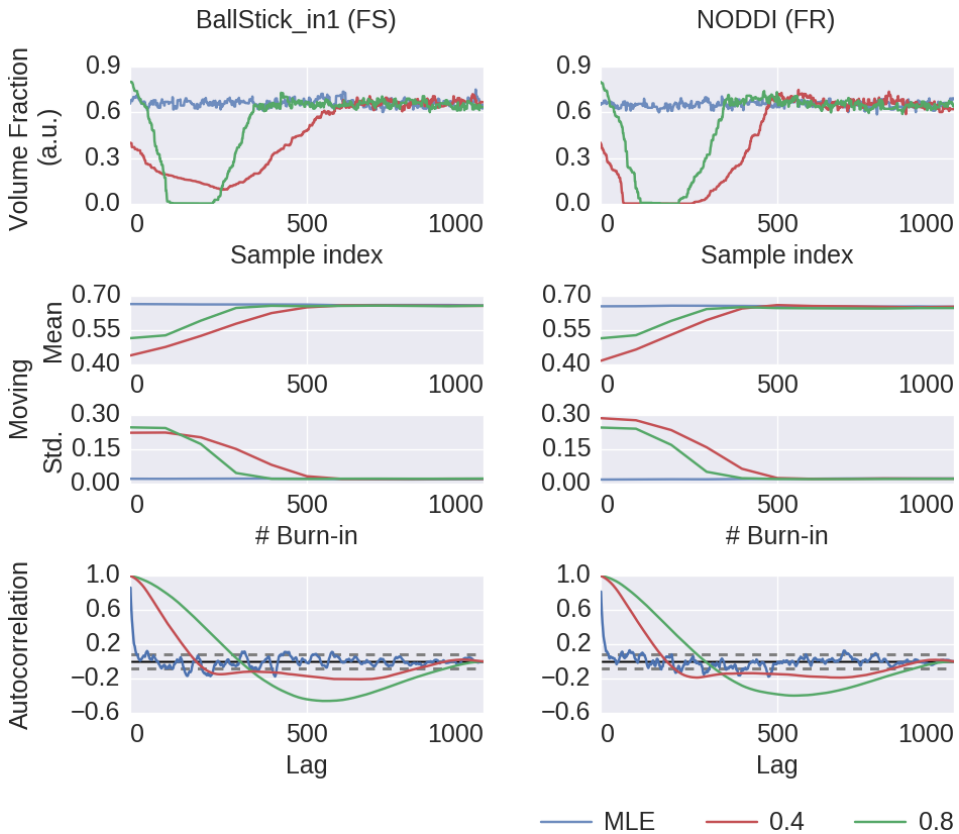


Figure 3.7: MCMC chains and burn-in results of a single voxel (the voxel indicated in figure 3.6) for the BallStick\_in1 Fraction of Stick (FS) and the NODDI Fraction Restricted (FR) model parameters. In the first row, the sampling trace when starting at the MLE or at two default points with (only) a varying volume fraction. In the second row, moving mean and moving standard deviations computed over 1000 samples with increasing burn-in. In the bottom row, autocorrelation plots computed over 1000 samples, with the 99% confidence interval in dashed gray.

### 3.3.3 Thinning

Thinning of sampler chains has theoretically been shown to reduce the accuracy of posterior analyses (Geyer, 1991; MacEachern and Berliner, 1994; Link and Eaton, 2012), and empirical evidence has been provided for the limited usefulness of thinning (Link and Eaton, 2012). Here we will show some empirical results of thinning applied to diffusion MRI modeling. Figure 3.8 shows the effect of thinning on the variability of the returned sampling trace, on the estimates of the mean and standard deviation and in terms of autocorrelation. The sampling trace shows that the chains produce roughly the same distribution, while with increased thinning many more samples are required ( $k$  times more samples, for a thinning of  $k$ ). Comparing the effect of thinning on the mean and standard deviation shows that, as predicted by theory, there is less or equal variance in the estimates when using more samples as compared to thinning the samples. Results also show that 1000 samples without thinning may not be enough for a stable estimates and more samples are required. Yet in accordance with theory, instead of thinning the chain, results indicate that just using more samples (e.g. all  $1000 \cdot k$  samples instead a thinning of  $k$ ) is preferred. Therefore, while autocorrelations are reduced as expected for thinned chains, mean and standard deviation estimates are robust against autocorrelation and the larger number of samples without thinning is preferred over thinned samples with reduced autocorrelation.

### 3.3.4 Minimum number of samples

Using multivariate ESS theory we determined, a priori, per model, the number of actual samples needed to generate a sufficient number of effective samples (the effective sample size or ESS) to approximate the underlying posterior density within a 95% confidence region and with a 90% relative precision. Figure 3.9 shows an estimate on the number of actual samples needed to reach this desired ESS, on average for a large number of voxels.

For lower order models (Ball&Stick\_in1, Tensor, NODDI) the sampling requirements do not depend on the acquisition table, with similar numbers of samples needed for the HCP MGH and RLS-pilot protocols. For multi-directional models (Ball&Stick\_in2, Ball&Stick\_in3) more samples are needed for the RLS-pilot protocol than for the HCP MGH protocol where we need 1.5x to 2x the amount of samples for the RLS-pilot protocol. Since the RLS-pilot dataset is not suitable for the CHARMED models, no RLS-pilot results are shown for the CHARMED models. Two and three fiber versions of the same models (Ball&Stick and CHARMED) require almost linearly increasing number of samples.

Table 3.9 summarizes the estimated sample requirements, as the one standard error above the mean point (upper whisker in figure 3.9), together with the required

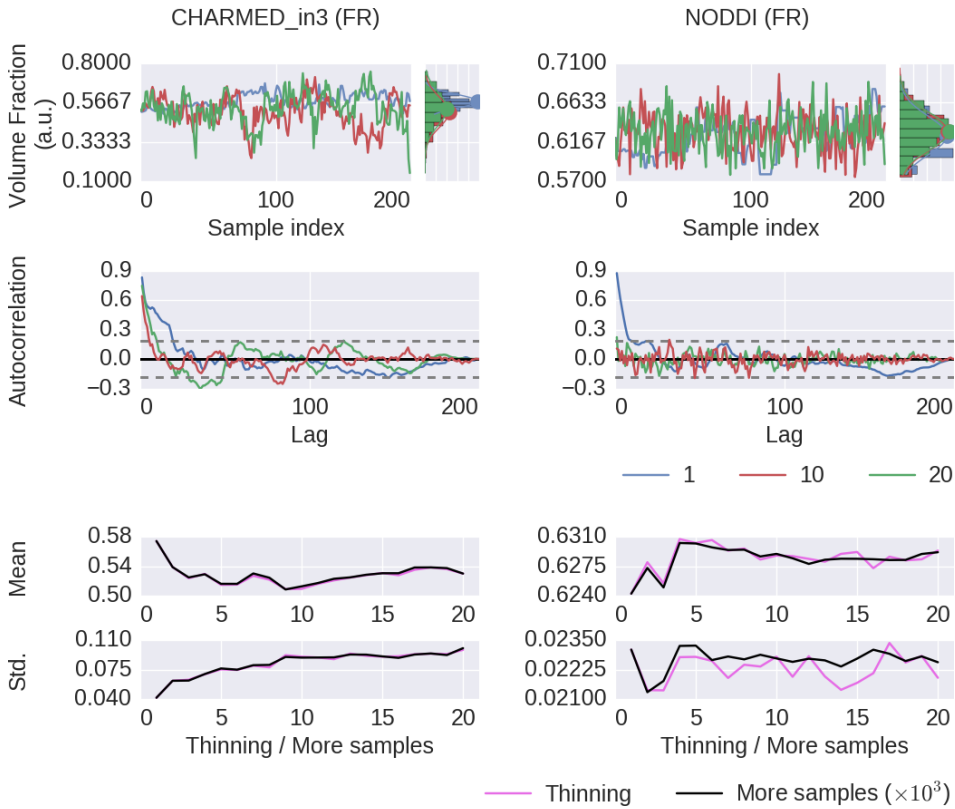


Figure 3.8: Thinning results of a single voxel (the voxel indicated in figure 3.6) for the CHARMED\_in3 Fraction of Restricted (FR) and the NODDI FR model parameters. In the first row, sample traces for the returned samples after a thinning of 1 (no thinning), 10 and 20, with their corresponding histograms. In the second row, an autocorrelation plot computed over 200 samples, with the 99% confidence interval in dashed gray. In the bottom row, a comparison of the posterior mean and standard deviation when thinning the chain or when using more samples. When thinning,  $1000 \cdot k$  samples are generated of which only every  $k$ th sample is used (so, always 1000 samples are used). When using more samples, all  $1000 \cdot k$  samples are used, without thinning. Results are without burn-in and started from a maximum likelihood estimator.

ESS and the number of estimated parameters in each model. In general, models with more parameters need more actual samples to reach the same confidence and precision, although the Tensor model with seven parameters requires less samples than the NODDI model with six parameters. This is probably related to the higher complexity (nonlinear parameter inter-dependencies) of the NODDI model compared to the Tensor model.

As can be seen in figure 3.2 (upper left panel), the required ESS to reach the desired 95% confidence region with a 90% relative precision is relatively invariant to the number of parameters (at about 2200), although the numbers of actual samples needed to realize this are different for every model, as seen in figure 3.9.

As an illustration of computation times, table 3.10 shows runtime statistics for sampling the recommended number of samples for HCP MGH dataset and a RLS-pilot dataset using an AMD Fury X graphics card and an Intel Xeon e3 18 core processor. For most models the GPU outperforms the CPU by about 20x, except for the more complex models (CHARMED\_in2, CHARMED\_in3) where the GPU is only about 6x faster. In general, although 4 to 7 hours are needed to sample the single fiber CHARMED\_in1 and NODDI models on the very large HCP MGH dataset (with 552 volumes), GPU-accelerated implementation can provide full posterior sampling of diffusion microstructure models over whole brain datasets in reasonable time on a standard graphics card. On the more clinically feasible RLS-pilot protocol (134 volumes) whole brain sampling of Tensor and Ball&Stick models can be performed within 20 minutes and NODDI within an hour.

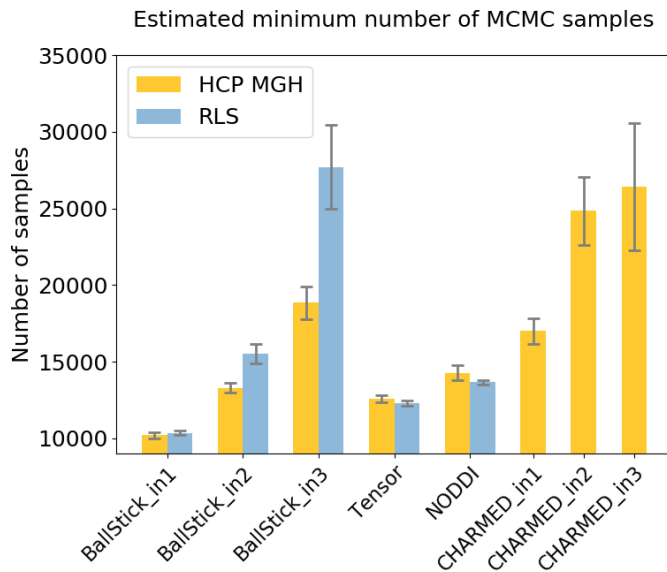


Figure 3.9: Estimates on the number of samples needed per model, to reach, averaged over the white matter, a 95% confidence region with a 90% relative precision. Results are shown for both an HCP MGH and RLS-pilot acquisition table. Whiskers show the standard error of the mean over 10 subjects.



Model	Number of parameters	Required ESS	Required nr. of samples
BallStick_r1	4	2108	11000
BallStick_r2	7	2192	15000
BallStick_r3	10	2208	25000
NODDI	6	2177	15000
Tensor	7	2192	13000
CHARMED_r1	11	2208	17000
CHARMED_r2	15	2198	25000
CHARMED_r3	19	2183	30000

Table 3.9: Estimates on the number of samples needed per model, to reach, when averaged over the white matter, a 95% confidence region with a 90% relative precision. While the required ESS can be determined a priori (see figure 3.9), the inherent model complexity determines how many samples are needed to reach that ESS.

Model	Number of samples	HCP MGH - GPU	HCP MGH - CPU	RLS-pilot - GPU	RLS-pilot - CPU
Ball&Stick_in1	11000	1h	36h	0.25h	4.5h
Ball&Stick_in2	15000	2h	48h	0.5h	15h
Ball&Stick_in3	25000	6h	72h	1.5h	45h
NODDI	15000	4h	80h	1h	20h
Tensor	13000	1h	22h	0.5h	4h
CHARMED_in1	17000	6h	54h	n/a	n/a
CHARMED_in2	25000	24h	139h	n/a	n/a
CHARMED_in3	25000	47h	200h	n/a	n/a

Table 3.10: Runtime statistics in hours (h) for MCMC sampling the estimated minimum number of samples (no burn-in, no-thinning) of various models, using a single HCP MGH (552 volumes) and single RLS-pilot (134 volumes) dataset. Statistics are for a whole brain mask of 410,000 voxels for HCP MGH and 204,993 voxels for RLS-pilot. Results are computed using an AMD Fury X GPU and an Intel Xeon e3 18 core CPU.

## 3.4 Discussion

Using an efficient GPU based implementation, we show that run times can be removed as a prohibitive constraint for sampling of diffusion multi-compartment models, achieving whole brain sampling in under an hour for typical datasets and most common dMRI models. Newer generations of graphics cards are likely to reduce these times even further. Using this implementation, we investigated the use of adaptive MCMC algorithms, burn-in, initialization and thinning. We finally applied the theory of Effective Sample Size to diffusion multi-compartment models as a way of determining a sufficient number of samples for a given model and dataset.

### 3.4.1 Adaptive MCMC

The use of adaptive MCMC algorithms increases the effectiveness of the sampling process by generating more effective samples for the same amount of MCMC samples. Adaptive methods generally have higher multivariate Effective Sample Size (ESS) than MCMC without adaptation. Although adaptive methods generally score close to each other in generated ESS, the FSL and AMWG methods have better sampling efficiency for complex crossing fiber models, such as CHARMED.in3 and low #volumes/#parameters situations, with AMWG slightly outperforming FSL. In accuracy and precision, AMWG and FSL perform well overall, whereas performance for the fixed proposal method (None) and SCAM is more inconsistent over all models. The ESS performance of the fixed proposal method could, in theory, be increased to the same levels as the adaptive methods by manual calibration, but since this is model and data (voxel) dependent, manual tuning could be very burdensome and unpractical.

This work covers only variations of the Metropolis-Within-Gibbs method, which has the advantage of high efficiency sampling with relatively general model-unspecific proposals. Future work could focus on MCMC algorithms which allow for block-updates of correlated parameters, or could investigate different proposal schemes altogether such as Component-wise Hit-And-Run Metropolis (Turchin, 1971; R. L. Smith, 1984), Multiple-Try Metropolis (J. S. Liu, Liang, and Wong, 2000) and/or No-U-Turn sampler (Hoffman and Gelman, 2011).

### 3.4.2 Burn-in

When starting from an arbitrary position, burn-in is advisable to reduce possible bias due to (possibly) low probability starting positions. Burn-in should ideally be considered post-sampling, since it is difficult to know a priori the time needed for the chain to converge and, due to randomness, past convergence rates provide

no guarantee for the future. This is why common practice dictates a relatively large number of burn-in samples which guarantees convergence in most cases.

While not harmful, burn-in is generally unnecessary and inefficient if the starting point is part of the stationary distribution of the Markov chain, which can, for example, be achieved by taking a Maximum Likelihood Estimator (MLE) as starting point. Even when starting from an MLE, a small burn-in of about 100 to 200 samples could be considered to remove correlations with the starting position. Additionally, when using adaptive proposal methods, a small burn-in could be considered to let the adaptation algorithm adapt the proposal distribution before sampling, slightly increasing the effective sample size of the chain.

### 3.4.3 Thinning

Already on theoretical grounds, thinning is not recommended and considered as often unnecessary, always inefficient and reducing the precision of posterior estimates (Link and Eaton, 2012; MacEachern and Berliner, 1994; Jackman, 2009; Geyer, 1991; Christensen et al., 2010). Illustrations based on the the Ball&Stick.in1 and NODDI model show that, with or without thinning, the posterior distribution is approximated about equally, while thinning needs  $k$  times more samples (for a thinning of  $k$ ). Results did show a convergence of mean and standard deviation estimates with an increased thinning, but these results are easily duplicated by incorporating not only the thinned samples but also the non-thinned samples in the statistical estimates (the 'more samples' strategy). Furthermore, using more samples instead of thinning provides estimates with a higher precision, as illustrated by the higher variability of the thinned estimates compared to the estimates with more samples (figure 3.8, right).

One legitimate reason for thinning is that, with independent samples, one can approximate the precision of an MCMC approximation (Link and Eaton, 2012). That is, it allows for more accurate assessment of the standard error of an MCMC estimate like the posterior mean. However, even in that case, thinning must be applied post-hoc, otherwise the precision of the mean itself will be reduced if computed from only the thinned samples. Furthermore, we are often more interested in the variability of the posterior distribution (which can be provided by e.g. the standard deviation) than in the precision of the posterior mean estimate. Another legitimate reason for considering thinning is hardware limitations, such as sampling post-processing time and storage space. However, barring such limitations, avoiding thinning of chains is far more efficient in providing high precision in posterior estimates.

### 3.4.4 Number of samples

The issue of the number of samples needed in a chain is often somewhat enigmatic and arbitrary. A common perception is that the number should be 'high', rather too high than too low. Multivariate Effective Sample Size (ESS) theory provides a theoretical lower bound on the number of effective samples needed to approximate the posterior, based on a desired confidence level and precision. How many MCMC samples are required to reach that target effective sample size is then dependent on the data, the model and the MCMC algorithm. We show that there is some dependency on the data in terms of sampling requirements for diffusion microstructure models, considering the increasing discrepancy between required sample numbers for `Ball&Stick_in1`, `Ball&Stick_in2` and `Ball&Stick_in3`. The dependency of sampling requirements on the model is higher, showing that more complex models seem to need more actual samples to reach the target ESS. As can be seen in figure 3.2 (upper left panel), the required ESS to reach the 95% confidence region with a 90% relative precision is relatively invariant to the number of parameters (at about 2200), although the numbers of actual samples needed to realize this are different for every model, as seen in figure 3.9. This sets an informed relatively general target for the amount of samples required in sampling diffusion micro-structural models, which scales the number of actual samples with the complexity of the model, data and the performance of the MCMC algorithm. This also means that MCMC algorithms which can generate effective samples more efficiently (such as the AMWG) can reduce the number of samples needed to reach the same confidence levels, reducing run time.

### 3.4.5 Conclusions and recommendations

Considering the theoretical soundness and its general robust performance, we advise the Adaptive Metropolis-Within-Gibbs (AMWG) algorithm for efficient and robust sampling of diffusion MRI models. We further recommend initializing the sampler with a maximum likelihood estimator obtained from, for example, non-linear optimization, in which case 100 to 200 samples are sufficient as a burn-in. Thinning is unnecessary unless there are memory or hard disk constraints or a strong reliance on posterior estimates that require uncorrelated samples. As a relatively general target for the number of samples, we recommend 2200 multivariate effective samples, which reaches 95% confidence and 90% relative precision, invariant of the number of parameters. The amount of actual MCMC samples required to achieve this is algorithm and model dependent and can be investigated in a pre-study, with numbers for common dMRI models reported here as an indication.



# 4

## Fast quantification of uncertainty in non-linear diffusion MRI models

---

Adapted from: R.L. Harms, F.J. Fritz, S. Schoenmakers and A. Roebroeck (2019). "Fast quantification of uncertainty in non-linear diffusion MRI models for artifact detection and more power in group studies". Preprint available on *bioRxiv*, p. 651547. DOI: 10.1101/651547.

## Abstract

Diffusion MRI (dMRI) allows for non-invasive investigation of brain tissue microstructure. By fitting a model to the dMRI signal, various quantitative measures can be derived from the data, such as fractional anisotropy, neurite density and axonal radii maps. The uncertainty in these dMRI measures is often ignored, while previous work in functional MRI has shown that incorporating uncertainty estimates can lead to group statistics with a higher statistical power. We propose the Fisher Information Matrix (FIM) as a generally applicable method for quantifying the parameter uncertainties in non-linear diffusion MRI models. In direct comparison with Markov Chain Monte Carlo sampling, the FIM produces similar uncertainty estimates at lower computational cost. Using acquired and simulated data, we then list several characteristics that influence the parameter variances, like data complexity and signal-to-noise ratio. In individual subjects, the parameter standard deviations can help in detecting white matter artifacts as patches of relatively large standard deviations. In group statistics, we recommend using the parameter standard deviations by means of variance weighted averaging. Doing so can reduce the overall variance in group statistics and reduce the effect of data artifacts without discarding data from the analysis. Both these effects can lead to a higher statistical power in group studies.

## 4.1 Introduction

Diffusion Magnetic Resonance Imaging (dMRI) allows for non-invasive investigation of brain tissue microstructure. By fitting a dMRI model to each voxel, various quantitative measures can be derived from the data, such as fractional anisotropy (Basser, Mattiello, and LeBihan, 1994), neurite density (H. Zhang, Schneider, et al., 2012) and axonal radii maps (Assaf and Pasternak, 2008; Alexander, Hubbard, et al., 2010). These quantitative measures can be used in statistical group analysis. For example, tract-based spatial statistics (TBSS) is a popular approach to group analysis of fractional anisotropy measures (S. M. Smith et al., 2006). More often than not, these approaches (including TBSS) ignore the uncertainty in the quantitative measures. In functional magnetic resonance imaging, previous work has shown that incorporating uncertainty estimates can lead to group statistics with a higher statistical power (Chen et al., 2012; Woolrich et al., 2004). For linear diffusion models, a method for computing and using uncertainty estimates has been shown before (Sjölund et al., 2018), but this has not yet been generalized to non-linear diffusion models like NODDI (H. Zhang, Schneider, et al., 2012) and CHARMED (Assaf and Basser, 2005).

Previous work in quantifying the parameter uncertainties include Markov Chain Monte Carlo (MCMC) (Behrens et al., 2003; Wegmann, Eklund, and Villani, 2017; Gu et al., 2017) and bootstrapping (D. K. Jones, 2003; Chung, Lu, and Henry, 2006; Whitcher et al., 2008) methods. Of these two techniques, bootstrapping is often not applicable as it is either model specific (Whitcher et al., 2008) or requires very specific additional MRI measurements (D. K. Jones, 2003; Chung, Lu, and Henry, 2006) which are often not acquired in diffusion MRI datasets. MCMC on the other hand can readily be extended to all microstructure models, but often requires long computation times, even with parallel processing on graphical processing units (Harms and Roebroek, 2018).

We propose the Fisher Information Matrix (FIM) as a generally applicable method for quantifying the parameter uncertainties in non-linear diffusion MRI models. The FIM allows for estimating the local variances around the maximum likelihood point estimate, which is the point estimate typically used in group statistics. Computing the FIM is a relatively fast operation, requiring only a few additional model evaluations. In other fields, like for example astrophysics, the Fisher Information Matrix is already recognized as a useful tool for quantifying the uncertainty in parameter estimates (Vallisneri, 2008; Rodriguez et al., 2013). In diffusion MRI, the FIM has been applied before, but only specific to the multi-Tensor model (Versteeg et al., 2018) and has not yet been generalized to all non-linear microstructure models.

The Fisher Information Matrix can additionally be used to compute the Cramér Rao Lower Bound (CRLB; Rao, 1945; Cramer, 1946), if and only if the true pa-



parameters are known (Kay, 1993). For example, in simulation studies the CRLB can function as a ground truth lower bound on the estimable variances, thereby indirectly evaluating the performance of the maximum likelihood routines (Kay, 1993). Although in brain data the FIM can be interpreted as an approximation to the CRLB, we follow the results in astrophysics and only interpret the FIM as a measure of uncertainty around the estimated parameters (Vallisneri, 2008).

We first compare the uncertainty estimates from the Fisher Information Matrix to those of MCMC, using multiple datasets and multiple dMRI microstructure models. We then investigate several data and model characteristics that can influence the parameter variances, like data complexity and Signal-to-Noise Ratio (SNR). In the end, we discuss the use of uncertainty estimates in white matter artifact detection (e.g. detecting fat saturation) and show how weighted averaging could lead to an increase in power in group studies.

## 4.2 Methods

### 4.2.1 Parameter distribution estimates

We compare two different methods for summarizing the parameter posterior distributions of a single voxel, a frequentist method using Maximum Likelihood Estimation (MLE) and the Fisher Information Matrix (FIM) and a Bayesian method using Markov Chain Monte Carlo (MCMC) (see figure 4.1 for a schematic overview). With both methods we summarize the voxel-wise posteriors as a point estimate with a corresponding standard deviation (std.).

In the first method we use the Powell optimization routine (Powell, 1964; Harms, Fritz, et al., 2017) to get an MLE parameter point estimates. We estimate the standard deviations around those point estimates using the theory of the FIM. Standard deviations in derived parameter maps (e.g. Tensor Fractional Anisotropy) can be obtained by propagating the uncertainty of the model parameters. We refer to this method as MLE+FIM.

The second methodology uses MCMC sampling to approximate the full posterior distribution, using the Adaptive Metropolis Within Gibbs routine as discussed in (Harms and Roebroek, 2018). From these samples we summarize the posterior distribution using a mean and standard deviation, as done before in before in dMRI modeling (Behrens et al., 2003; Sotiropoulos et al., 2013; Wegmann, Eklund, and Villani, 2017). Uncertainties in derived parameter maps can be obtained by computing the derived parameter maps at every sampled point and summarizing the result. We refer to this method as MCMC.

The MLE+FIM provides a local variance around a mode while MCMC provides a global variance around the mean. As such, these methods are only comparable

if the posterior is unimodally Gaussian distributed, since then the mean equals the mode. As in previous work (Behrens et al., 2003; Sotiropoulos et al., 2013; Wegmann, Eklund, and Villani, 2017), we assume the posteriors to be unimodally Gaussian distributed.

This assumption may not necessarily hold. For example, multi-modality could arise when fitting a single fiber model to a crossing fiber voxel. In such cases, different post-processing would be required on the MCMC samples to correctly reflect the parameter variances. The FIM would be less sensitive to this issue since the FIM provides only local variances estimates. That is, the MLE would choose one mode of the distribution and the FIM would provide a local variance estimate around the chosen mode. This issue could also be circumvented by applying appropriate model selection to every voxel.

Non-Gaussian distributions can happen near parameter boundaries. For instance, very low (close to zero) or very high (close to one) compartment volume fractions can lead to a truncated posterior. In such cases the FIM no longer applies. For MCMC different post-processing would be required, like fitting a truncated normal distribution to the posterior. This could again be solved by appropriate model selection. We take no special precautions for these boundary effects and assume these to not be present in white matter.

Nevertheless, we expect most posteriors to be unimodally Gaussian distributed. This assumption is also supported by two theoretical arguments. First, if the model is suitable to describe the data (e.g. if model selection was successfully applied), the posterior asymptotically approaches a Gaussian distribution (Gelman, Carlin, et al., 2013). Second, according to the central limit theorem, each parameter's marginal distribution will asymptotically tend to a Gaussian as the number of model parameters increases (Gelman, Carlin, et al., 2013).

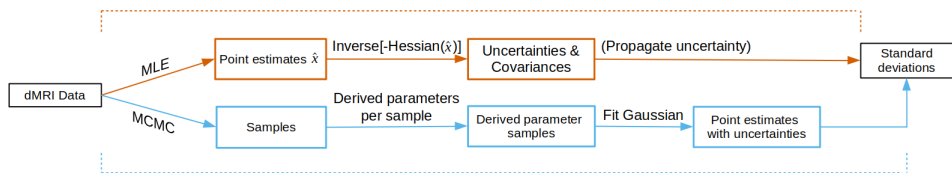


Figure 4.1: The uncertainty computation methods for both the Maximum Likelihood Estimation (MLE) and Markov Chain Monte Carlo (MCMC) methods.

### Fisher Information Matrix

The observed Fisher Information Matrix is defined as the negative Hessian of the log-likelihood function when evaluated at the maximum likelihood estimate

(Pawitan, 2013; Gelman, Carlin, et al., 2013). The inverse of the FIM is an asymptotic estimator of the covariance matrix (Pawitan, 2013; Gelman, Carlin, et al., 2013). Formally, let  $l(\mathbf{x})$  be a log-likelihood function with maximum likelihood estimate  $\hat{\mathbf{x}}$ . A second order Taylor approximation of  $l(\mathbf{x})$  centered at  $\hat{\mathbf{x}}$  is then given by:

$$l(\mathbf{x}) = l(\hat{\mathbf{x}}) + \frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}})^T \frac{\partial^2}{\partial \mathbf{x}^2} l(\hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}}) \quad (4.1)$$

ignoring the higher terms and having dropped the linear term since the first derivative of a function is zero at the mode. Considering the first term,  $l(\hat{\mathbf{x}})$ , a constant and the second term,  $\frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}})^T \frac{\partial^2}{\partial \mathbf{x}^2} l(\hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})$ , proportional to the logarithm of a normal density, we get the approximation:

$$l(\mathbf{x}) \approx \mathcal{N}(\hat{\mathbf{x}}, [\mathbf{I}(\hat{\mathbf{x}})]^{-1}) \quad (4.2)$$

where  $\mathbf{I}(\hat{\mathbf{x}})$  is the observed Fisher Information Matrix:

$$\mathbf{I}(\hat{\mathbf{x}}) = -\mathbf{H}(\hat{\mathbf{x}}) = -\frac{\partial^2}{\partial \hat{\mathbf{x}}^2} l(\hat{\mathbf{x}}) \quad (4.3)$$

For the Hessian to be positive definite, this theory requires  $\hat{\mathbf{x}}$  to lie within the boundaries of the parameter space (Gelman, Carlin, et al., 2013). We compute the Hessian numerically (see Appendix B.1) and its inverse using a direct inverse where possible with a fallback on the (Moore-Penrose) pseudo-inverse for ill-conditioned Hessians. Ill-conditioned Hessian can for example arise with parameter estimates lying at a predefined parameter boundary (Gelman, Carlin, et al., 2013).

### Uncertainty propagation

Given a function  $\mathbf{y} = f(\boldsymbol{\theta})$  where  $f(\cdot)$  is a known function, uncertainty propagation provides the probability distribution of  $\mathbf{y}$  given the probability distribution of  $\boldsymbol{\theta}$ . For example, we can use this to estimate the standard deviation of a Tensor Fractional Anisotropy (FA) estimate, by propagating the standard deviation estimates of the Tensor diffusivities. We use a first order Taylor expansion linear approximation (Arras, 1998), which states that if  $\boldsymbol{\theta}$  is normally distributed with mean  $\boldsymbol{\mu}_\theta$  and covariance matrix  $\Sigma_\theta$ , the distribution of  $\mathbf{y}$  can be approximated as:

$$\mathbf{y} \approx \mathcal{N}(\boldsymbol{\mu}_\mathbf{y}, \Sigma_\mathbf{y}) = \mathcal{N}(f(\boldsymbol{\theta}), \mathbf{J}_f(\boldsymbol{\theta})\Sigma_\theta\mathbf{J}_f(\boldsymbol{\theta})^T) \quad (4.4)$$

with  $\mathbf{J}_f$  the Jacobian matrix of  $f$ . More succinctly, the covariance matrix of  $\mathbf{y} = f(\boldsymbol{\theta})$  is given by:

$$\Sigma_{\mathbf{y}} = \mathbf{J}_f \Sigma_{\boldsymbol{\theta}} \mathbf{J}_f^{\top} \quad (4.5)$$

which holds as a generally applicable formula for linear propagation of covariances (Arras, 1998). In the case of an univariate output  $y = f(\boldsymbol{\theta})$ , the Jacobian can be formulated as a gradient vector  $\nabla_f$ , leading to the following expression for the variance in  $y$ :

$$\sigma_y^2 = \nabla_f \Sigma_{\boldsymbol{\theta}} \nabla_f^{\top} \quad (4.6)$$

This error propagation technique uses both the variances and the co-variances of all the propagated parameters. Additionally, this technique takes into account the functional form of the propagated function, i.e. if the function is linear or non-linear. The Jacobian or gradient can be computed numerically using finite-differences or can be evaluated at an analytical derivative. We use analytical expressions for all uncertainty propagations. See Appendix B.2 for error propagation examples of the Tensor FA and Ball&Stick Fraction of Stick parameters.

## 4.2.2 Variance weighted average

Variance weighted averaging makes it possible to include the variances of the data points when computing a mean and standard deviation. For example, the voxel-wise variances discussed earlier can be used in averages of white matter regions within a subject, or in voxel-wise averages over multiple subjects. First, given  $n$  data points  $z_i$ , we define the regular mean as:

$$\bar{\mu}_{\text{regular}} = \frac{1}{n} \sum_i^n z_i \quad (4.7)$$

and regular standard deviation as:

$$\bar{\sigma}_{\text{regular}} = \sqrt{\frac{\sum_i^n (z_i - \bar{\mu}_{\text{regular}})^2}{n}} \quad (4.8)$$

If each data point  $z_i$  has a corresponding weight  $w_i$ , we can compute a weighted mean as:

$$\bar{\mu}_{\text{weighted}} = \frac{\sum_i^n w_i z_i}{\sum_i^n w_i} \quad (4.9)$$

and a weighted standard deviation as:

$$\bar{\sigma}_{\text{weighted}} = \sqrt{\frac{\sum_i^n w_i (z_i - \bar{\mu}_{\text{weighted}})^2}{\frac{(m-1)}{m} \sum_i^n w_i}} \quad (4.10)$$

with  $m$  for the number of non-zero weights, included here to allow for non-normalized weights. It has been shown that the weights that minimize the variance of the weighted average are the reciprocals of the variances of each of the data points  $z_i$  (Shahar, 2017). That is, given the variances  $\sigma_i^2$  for each  $z_i$ , the weights that minimize  $\text{Var}(\sum_i w_i z_i)$  is given by:

$$w_i = \frac{1}{\sigma_i^2} \quad (4.11)$$

Incidentally, these weights are also the maximum likelihood estimator of the weighted mean and variance under the assumption that the data points  $z_i$  are independent and normally distributed with the same mean (Cochran, 1937).

### 4.2.3 Diffusion microstructure models

To capture the variety of microstructure models in diffusion MRI we chose four different models, the Tensor (Basser, Mattiello, and LeBihan, 1994), Ball&Stick (Behrens et al., 2003), Bingham-NODDI (Tariq et al., 2016) and CHARMED (Assaf, Freidlin, et al., 2004) models. The Tensor model is the oldest diffusion MRI model and still sees widespread usage in the literature. From the Tensor we derive the Fractional Anisotropy (FA) quantity. The Ball&Stick model (Behrens et al., 2003) is the first multi-compartment model and is often used as local estimator for tractography. To delineate multiple fiber orientations, the Ball&Stick model can feature multiple Stick compartments, but always with a single Ball compartment. To differentiate between the Ball&Stick models with one or more Stick compartments, we denote the specific Ball&Stick model as "BallStick\_in1", "BallStick\_in2" and "BallStick\_in3" for respectively one, two or three Stick compartments. This is a general naming scheme to denote models that can have one or more intra neuronal compartments relative to the other compartments. From the Ball&Stick model we derive the Fraction of Stick (FS) quantity, which is the sum of the volume fractions of the Stick compartments.

More recent, biologically inspired, models include Bingham-NODDI and CHARMED. The Bingham-NODDI model assumes that white matter consists of restricted intra-cellular and hindered extra-cellular water compartments, with the intra-cellular compartment capturing neurite orientation dispersion. From the Bingham-NODDI model we use the Fraction of Restricted (FR) quantity, the volume fraction of the restricted intra-cellular compartment. The CHARMED model assumes a tissue model of restricted intra-neuronal and hindered extra-neuronal water compartments, with the intra-neuronal compartment assuming a bundle of axons. Since CHARMED can be used with multiple intra-neuronal compartments we again denote these with the 'in' suffix. Here, we only use CHARMED with one intra-neuronal compartment, denoted as "CHARMED\_in1". From the CHARMED model we use the Fraction of Restricted (FR) quantity, the volume fraction of the restricted intra-neuronal compartment. For implementation notes of these models see (Harms, Fritz, et al., 2017).

#### 4.2.4 Software

All models and routines used in this study are implemented in a Python based GPU (graphical processing unit) accelerated toolbox, the Microstructure Diffusion Toolbox, MDT, which is freely available under an open source license at <https://github.com/cbclab/MDT>. We used the models and MCMC routine as implemented in MDT version 0.18.3. From this version onward, MDT automatically computes the FIM after every maximum likelihood estimation operation and writes out the variances and covariances alongside the parameter estimates. Scripts for reproducing the results in this article can be found at [https://github.com/robbert-harms/uncertainty\\_paper](https://github.com/robbert-harms/uncertainty_paper). All computations for this paper were performed on a single AMD Fury X graphics card.

#### 4.2.5 Datasets

In this study we used simulated data and imaging data from two population studies. To illustrate the methods on a dataset with a clinically feasible, fast to acquire, acquisition scheme, we used data from the diffusion protocol pilot phase of the Rhineland Study ([www.rheinland-studie.de](http://www.rheinland-studie.de)). We refer to these datasets and acquisition schemes as *RLS-pilot*. To illustrate the methods on a dataset with a high-end, long acquisition time, acquisition scheme, we used data from the Human Connectome Project MGH-USC Young Adult study. We refer to these datasets and acquisition schemes as *HCP MGH*. For simulated data we used a single representative acquisition scheme from both the RLS-pilot and HCP MGH studies.

The RLS-pilot datasets were acquired on a Siemens Magnetom Prisma (Siemens, Erlangen, Germany) with the Center for Magnetic Resonance Research (CMRR) multi-band (MB) diffusion sequence (Moeller et al., 2010; Xu et al., 2013). These datasets had a resolution of 2.0 mm isotropic with diffusion parameters  $\Delta = 45.8$  ms,  $\delta = 16.3$  ms, TE = 90 ms and TR = 4500 ms, and with Partial Fourier = 6/8, MB factor 3, no in-plane acceleration with 3 shells of  $b = 1000, 2000, 3000$  s/mm<sup>2</sup>, with respectively 30, 40 and 50 directions to which are added 14 interleaved b0 volumes leading to 134 volumes in total per subject. Additional b0 volumes were acquired with a reversed phase encoding direction which were used to correct susceptibility related distortion (in addition to bulk subject motion) with the topup and eddy tools in FSL version 5.0.9 (J. L. Andersson and Sotiropoulos, 2016). The total acquisition time is 10 min 21 sec. These three-shell datasets represent a relatively short time acquisition protocol that still allows many models to be fitted. From this dataset we used a single representative subject (v3a\_1\_data.ms20).

The HCP MGH datasets come from the freely available fully preprocessed dMRI data from the USC-Harvard consortium of the Human Connectome project. Data used in the preparation of this work were obtained from the MGH-USC Human Connectome Project (HCP) database (<https://ida.loni.usc.edu/login.jsp>). The data were acquired on a specialized Siemens Magnetom Connectom with 300 mT/m gradient set (Siemens, Erlangen, Germany). These datasets were acquired at a resolution of 1.5 mm isotropic with diffusion parameters  $\Delta = 21.8$  ms,  $\delta = 12.9$  ms, TE = 57 ms, TR = 8800 ms, Partial Fourier = 6/8, MB factor 1 (i.e. no simultaneous multi-slice), in-plane GRAPPA acceleration factor 3, with 4 shells of  $b = 1000, 3000, 5000, 10,000$  s/mm<sup>2</sup>, with respectively 64, 64, 128, 393 directions to which are added 40 interleaved b0 volumes leading to 552 volumes in total per subject, with an acquisition time of 89 minutes. These four-shell, high number of directions, and very high maximum b-value datasets allow a wide range of models to be fitted. From these datasets we used a single representative subject (hcp\_1003) in single subject illustrations and we used all 35 subjects in the group comparisons.

Since the CHARMED\_in1 model requires relatively high b-values ( $\geq \sim 6000$  s/mm<sup>2</sup>), which are not present in the RLS-pilot datasets, we will only use the HCP MGH dataset when showing CHARMED\_in1 results. Additionally, since the Tensor model is only valid for b-values up to about 1200 s/mm<sup>2</sup>, we only use the b-value 1000 s/mm<sup>2</sup> shell and b0 volumes in maximum likelihood estimation and MCMC sampling of the Tensor model. All other models use all the data volumes.

For all datasets we created a white matter (WM) mask from the Tensor FA estimates and, using BET from FSL (S. M. Smith, 2002), a whole brain mask. The whole brain mask is used for MLE and MCMC sampling, whereas averages over

the WM mask are used in model or data comparisons. For each dataset, voxel-wise SNR is estimated using only the unweighted ( $b_0$ ) volumes, by dividing the mean of the unweighted volumes by the standard deviation.

### Ground truth simulations

We additionally created simulated data to illustrate the effects of the signal-to-noise ratio (SNR) on the variance of the estimated parameters. We used a single representative acquisition scheme from both the RLS-pilot and HCP MGH datasets (the acquisition schemes of subject `v3a_1_data_ms20` and `hcp_1003`), and simulated data for each model. For each acquisition scheme and each model, we simulate 10000 voxels with random volume fractions in  $[0.2, 0.8]$ , diffusivities in  $[5e-11, 5e-9]$   $\text{mm}^2/\text{s}$ , and orientations in  $[0, \pi]$ . From these, we created multiple copies with Rician noise (Gudbjartsson and Patz, 1995) of SNRs 5, 10, 20, 30, 40 and 50. We then fit and sample each model ten times to these simulated datasets and estimate the standard deviation using both the FIM and MCMC approach as described above. Per SNR we summarize the results of these ten trials as a mean standard deviation and its corresponding standard error of the mean.

### Group statistics

For the group statistics we computed Tensor FA and Bingham-NODDI FR and FR standard deviation maps on all 35 subjects using the MLE+FIM method. To be able to compare the subjects, we first registered the Tensor FA estimates to the `FMRIB58_FA_1mm` template using FLIRT and FNIRT from FSL (J. Andersson, Jenkinson, and S. Smith, 2010). Next, we used those registration templates to co-register the Bingham-NODDI FR and FR standard deviation maps.

With uncertainty maps available there are three methods to compute group statistics that are robust against subject-level artifacts. Method one, apply variance weighted averaging using the uncertainty estimates to downweight voxels with a high standard deviation. This would automatically remove artifacts if these artifacts lead to high parameter uncertainties. Method two, exclude outlier subjects from the group statistic. Outlier subjects could be detected using the point estimates or using the uncertainty maps. Method three, use a combination of method one and two, i.e. computing weighted group estimates after removal of outliers.

To illustrate these three artifact reduction methods, we first computed a baseline statistic using a simple mean and standard deviation over all 35 subjects. We then used artifact reduction method one and used the FR standard deviation maps as weights in the variance weighted averaging. To apply artifact reduction method two and three, we created a new subgroup with only 30 subjects, where



we manually removed five subjects (mgh\_1008, mgh\_1009, mgh\_1013, mgh\_1017 and mgh\_1032) that had a large white matter artifact over the corpus callosum. We then applied regular averaging and weighted averaging over these remaining 30 subjects.

As a comparison method between regular and weighted averaging we computed  $(\mu_{\text{weighted}} - \mu_{\text{regular}})/\mu_{\text{regular}}$  and  $(\sigma_{\text{weighted}} - \sigma_{\text{regular}})/\sigma_{\text{regular}}$  as difference measure for the mean and standard deviation estimates between regular and weighted averaging.

## 4.3 Results

We begin by comparing the parameter estimates and parameter uncertainty estimates of MLE+FIM to the corresponding estimates from MCMC. Next, we investigate the effect of SNR on the parameter standard deviations using both simulated and imaging data. We end with a comparison of regular versus weighted averaging in group statistics.

### 4.3.1 Parameter distribution estimates

Figure 4.2 visually compares the results of MLE+FIM to those of MCMC, using the Bingham-NODDI Fraction of Restricted (FR) parameter, on a single subject from the RLS-pilot dataset. Comparing results of a single transverse slice shows high qualitative correspondence between the MLE and MCMC methods (figure 4.2A), with both the point estimates and corresponding standard deviations (stds.) in close resemblance. A single voxel illustration of the estimated Gaussian distributions (figure 4.2B) again shows a high degree of similarity, with both Gaussian fits capturing the characteristics of the MCMC sample distribution to a large degree.

To further quantify the correspondence between the MLE and MCMC methodologies, we created scatter plots between the MLE and MCMC estimates of both the point estimate and standard deviation estimate. This was performed over a white matter mask for a single subject from both the HCP MGH and RLS-pilot datasets. Figure 4.3 shows Bingham-NODDI FR mean and standard deviation scatter plots. The FR point estimates are very tightly confined to the identity line, illustrating a high degree of correspondence in the point estimates from MCMC and MLE. The variation of point estimates along the diagonal corresponds to variation of FR values over the white matter mask, ranging between roughly 0.3 and 0.7. The std. estimates between the MLE and MCMC methodologies again show a high correspondence, although the off-diagonal spread in the std. plot is visibly larger than that in the point estimate plot. There is also some clipping visible in

the std. plot, with MLE estimating a zero std. while MCMC provides a range of values. This is mostly due to very low point estimates (near zero), at which point the FIM is no longer applicable. The blue-green-yellow-red coded points in both plots account for 97-99.5% of the voxels and the purple points account for the remaining fraction of outliers. The std. estimates for the HCP MGH data are clearly lower than for the RLS-pilot data, confirming an expected higher precision (lower uncertainty) of point estimates based on more dMRI data-points.

To investigate the correspondence in MCMC and MLE uncertainty estimates for a larger number of models, figure 4.4 shows scatter plots for multiple microstructure models. Parameter point estimate comparisons are not shown here, but are generally in correspondence to a very high degree. Across all models and data, except for the CHARMED\_in1 model fit on RLS-pilot data, MCMC and MLE uncertainty estimates are in high correspondence and located close to the identity diagonal. A relatively large off-diagonal variance in standard deviation estimates is visible in the CHARMED\_in1 FR parameter on the RLS-pilot data. This is expected as the RLS-pilot dataset is not well suited for the CHARMED\_in1 model due to too low b-values (the CHARMED\_in1 model requires b-values  $\leq 6000\text{s/mm}^2$ ). Standard deviation estimates for CHARMED\_in1 on the HCP MGH data are not only much more tightly confined to the identity diagonal, the std. estimates themselves are also about a factor two lower. A large spread to the right is also visible in the Ball&Stick\_in3 results. This might be related to MLE choosing a different mode and is perhaps solved using model selection. There is also again some clipping visible, with MLE providing a zero std. with voxels with a very low point estimate.

Irrespective of the method (MCMC or MLE+FIM), the std. estimates on the RLS-pilot data are always higher than the corresponding estimates on the HCP MGH data, once again confirming the expected higher precision on datasets with a larger number of direction. Conversely, one would expect higher complexity models (i.e. models with more compartments and more parameters to fit) to have higher uncertainty when fitted on the same data. This is indeed illustrated by the Ball&Stick\_in{1,2,3} results, where we see an increasing estimated standard deviation for an increasing number of Sticks, within each of the HCP MGH and RLS-pilot datasets. Finally, Tensor FA standard deviations are about a factor two higher than those of the other models. This is probably related to Tensor FA being a compound parameter.

To quantify correspondence in the MCMC and MLE std. estimates in the scatter plots, table 4.1 shows the percentage of voxels for which the difference between the MLE and MCMC variances is less than two standard deviations from the mean difference. We note an average similarity of  $\sim 98.7\%$  across six models and two datasets, even including the 97.9% similarity for the CHARMED\_in1 model fit on RLS-pilot data. Table 4.2 compares runtimes between the MLE with the

FIM and the MCMC methodologies, measuring the time between loading the data and writing the results. Averaged over six models and two subjects, the GPU-optimized MLE and FIM together compute approximately 38 times faster than GPU-optimized MCMC.

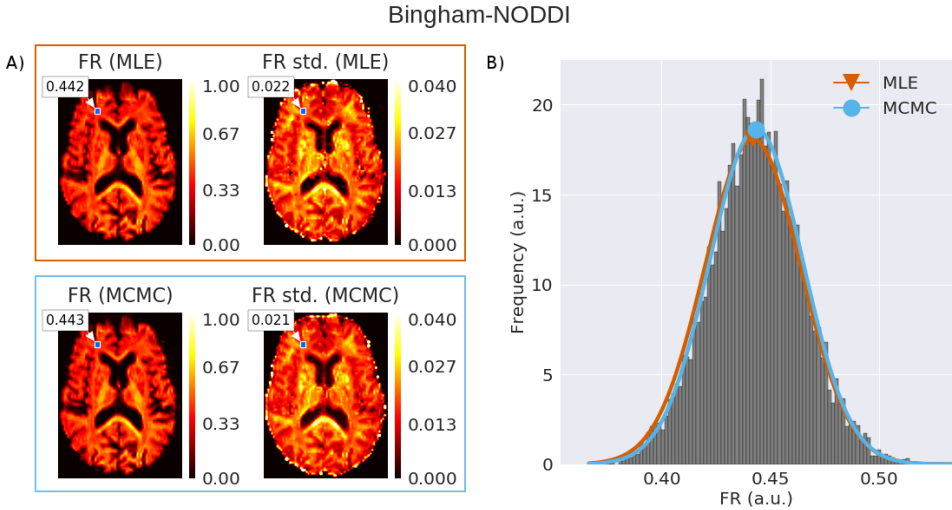


Figure 4.2: A) Visual comparison of parameter and standard deviation uncertainty maps between the Maximum Likelihood Estimation (MLE) and Markov Chain Monte Carlo (MCMC) methodologies for the Bingham-NODDI Fraction of Restricted (FR) on an RLS-pilot dataset. B) Histogram of the 20 thousand MCMC samples of the highlighted voxel in figure A, with in red and blue the fitted Gaussian distributions of, respectively, the MLE and MCMC methodologies.

	HCP MGH	RLS-pilot
Ball&Stick_in1	99.5%	98.8%
Ball&Stick_in2	99.9%	99.4%
Ball&Stick_in3	98.8%	97.5%
Bingham-NODDI	98.9%	98.7%
CHARMED_in1	98.6%	96.9%
Tensor	99.0%	97.9%

Table 4.1: For each model and dataset the percentage of voxels where the difference between the parameter stds. from the FIM and of MCMC are within two standard deviations from the mean difference. These percentages correspond to the red/yellow high densities in figure 4.4.

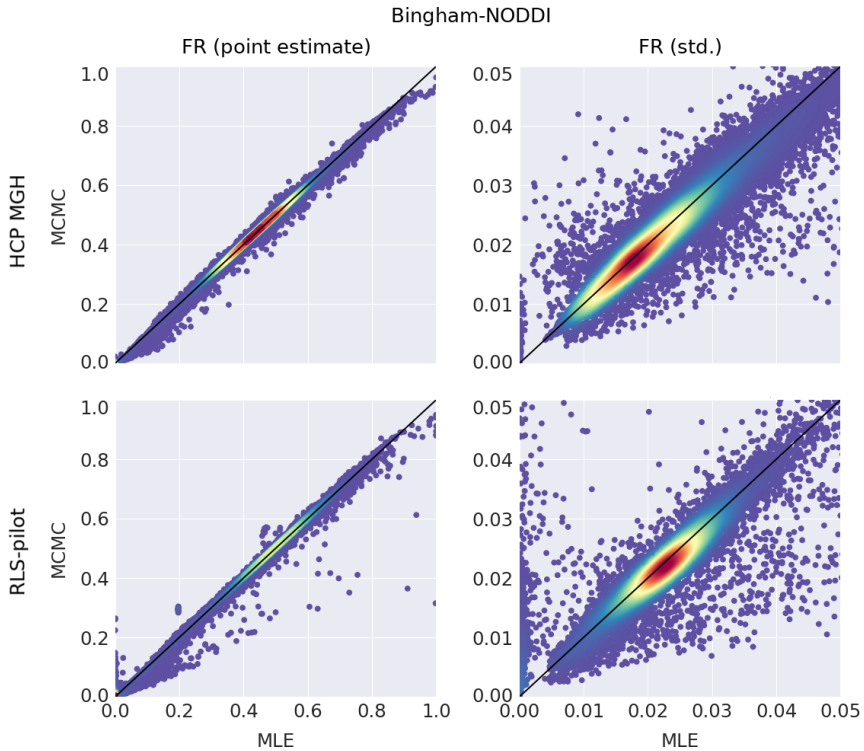


Figure 4.3: Scatter plots comparing Maximum Likelihood Estimation (MLE) and Markov Chain Monte Carlo (MCMC) point estimates (left column) and standard deviations (right column) for the Bingham-NODDI Fraction of Restricted (FR) values over a white matter mask for both a complex, long acquisition time HCP MGH dataset and a clinically feasible RLS-pilot dataset. Plots are color coded using a kernel density estimate (a.u) from purple (low density) to red (high density). Purple points correspond to a small percentage (0.5-3%) of the data (c.f. Table 4.1).

	HCP MGH			RLS-pilot		
	MLE + FIM	MCMC	rel.	MLE + FIM	MCMC	rel.
Ball&Stick_in1	00:01:49	01:21:55	45x	00:00:30	00:20:49	42x
Ball&Stick_in2	00:04:32	02:33:18	34x	00:01:08	00:42:36	38x
Ball&Stick_in3	00:13:01	07:00:51	32x	00:03:19	01:53:33	34x
Bingham-NODDI	02:06:19	111:32:52	53x	00:28:19	26:11:47	56x
CHARMED_in1	02:09:49	53:34:47	25x	00:21:53	07:49:55	21x
Tensor	00:02:41	01:59:07	44x	00:02:18	01:02:11	27x

Table 4.2: Runtime comparison between the two methodologies for computing parameter statistics, Maximum Likelihood Estimation (MLE) with the Fisher Information Matrix (FIM) and Markov Chain Monte Carlo (MCMC) sampling, for six different models and using a single representative subject from both the HCP MGH and the RLS-pilot datasets. Reported run times are over the entire brain mask and are in units of (h:m:s), with next to it the relative speed advantage of the MLE + FIM over MCMC.

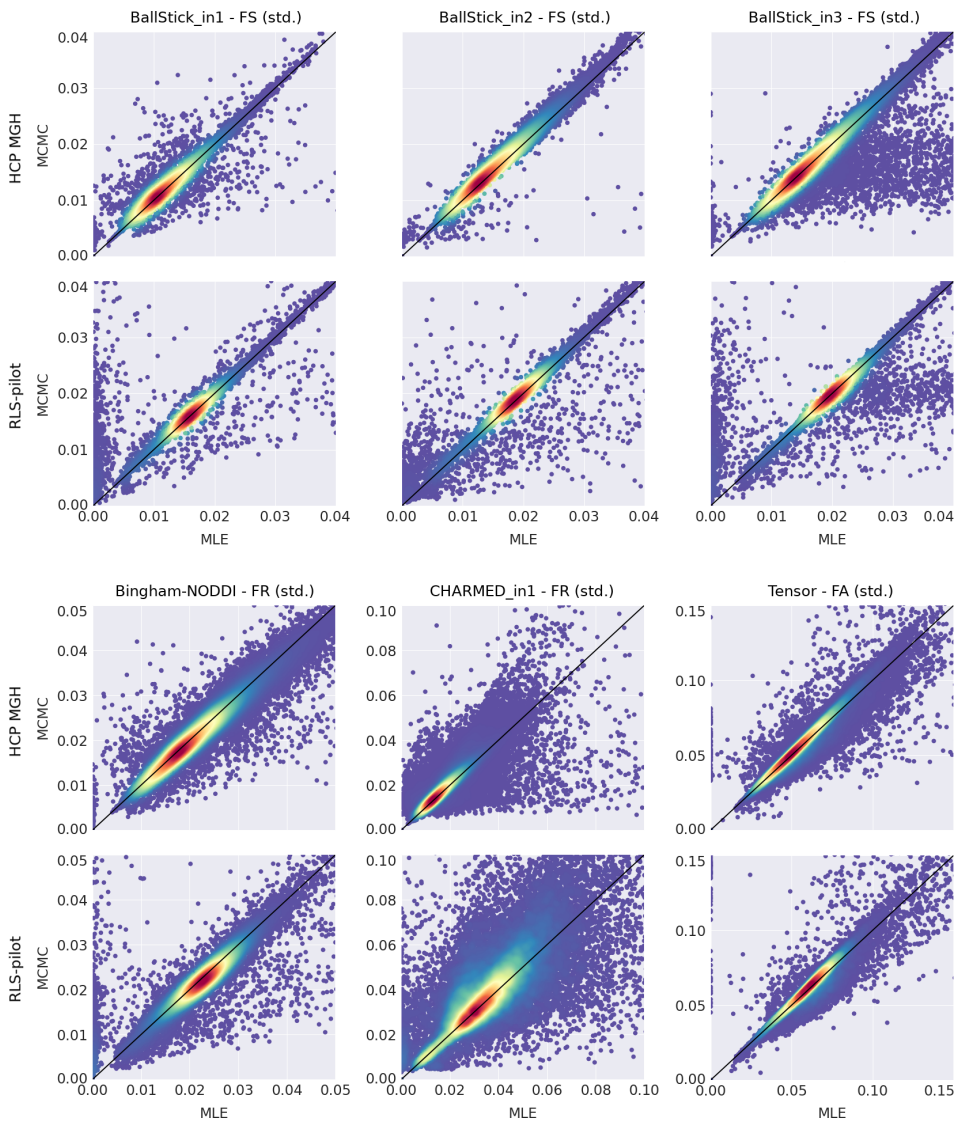


Figure 4.4: Scatter plots comparing Maximum Likelihood Estimation (MLE) and Markov Chain Monte Carlo (MCMC) standard deviations for multiple models over a white matter mask for both an HCP MGH and an RLS-pilot dataset. Acronyms are Fraction of Stick (FS), Fraction of Restricted (FR) and Fractional Anisotropy (FA). Plots are color coded using a kernel density estimate (a.u) from purple (low density) to red (high density). Purple points correspond to a small percentage (0.5-3%) of the data (c.f. Table 4.1).

### 4.3.2 Effect of SNR on parameter variances

Lower SNR per data point (i.e. single diffusion volume) is expected to lead to higher uncertainty in fitted parameter estimates. This issue is of extra importance in brain dMRI by the fact that SNR is non-uniform over the brain, especially in modern high number-of-channel phased array RF-coils. In order to assess the effect of SNR on parameter variances, figure 4.5 compares an estimate of SNR, its reciprocal, and the parameter standard deviation estimates of multiple white matter models on a single HCP MGH dataset. We observe a decreased SNR in the center of the brain and an increase of SNR towards the periphery. A very similar gradient can be observed in the standard deviation maps, with a decrease in parameter standard deviations towards the periphery. As in the previous results, we observe an increase in standard deviations for an increased number of Sticks in the Ball&Stick\_in{1,2,3} models, and Tensor FA standard deviations are about a factor two higher than the other standard deviation estimates.

To further compare SNR and standard deviation estimates, figure 4.6 plots SNR versus parameter standard deviations, for both simulated data and imaging data. In general, we observe an inverse relationship between SNR and standard deviation, where an increase in SNR leads to a decrease in parameter std. estimates. Standard deviations on RLS-pilot data are always higher than corresponding estimates on HCP MGH data, except for the imaged data analysis at an SNR of 5, where the RLS-pilot dataset has a lower standard deviation. For lower SNR ( $< 10$ ), MLE std. estimates are slightly higher than the MCMC estimates. For higher SNR ( $> 10$ ), the MLE and MCMC standard deviation estimates quickly converge, except for Ball&Stick\_in2, Ball&Stick\_in3 and Tensor estimates on the RLS-pilot dataset, where MLE standard deviations stay higher than those from MCMC. For the HCP MGH dataset, results are consistent between simulated and imaging data, with differences within the Standard Error of the Mean (SEM). Results on the RLS-pilot dataset are generally also consistent, except for an SNR of 5, where imaging data results are lower than those on simulated data. We finally observe that the standard error of the mean is generally higher for the simulated data compared to the imaging data, especially for lower SNR.

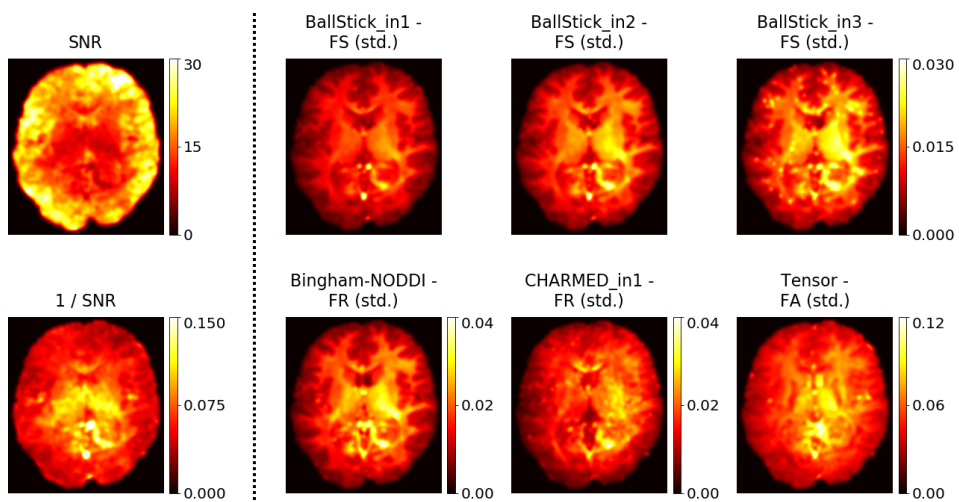


Figure 4.5: Illustration of the effect of Signal to Noise ratio (SNR) on parameter standard deviation estimates (using the MLE methodology), for a single HCP MGH subject (subject 1003). Maps are slightly smoothed with a 3d Gaussian filter ( $\sigma = 1\text{ voxel}$ ). Parameter acronyms are Fraction of Stick (FS), Fraction of Restricted (FR) and Fractional Anisotropy (FA).

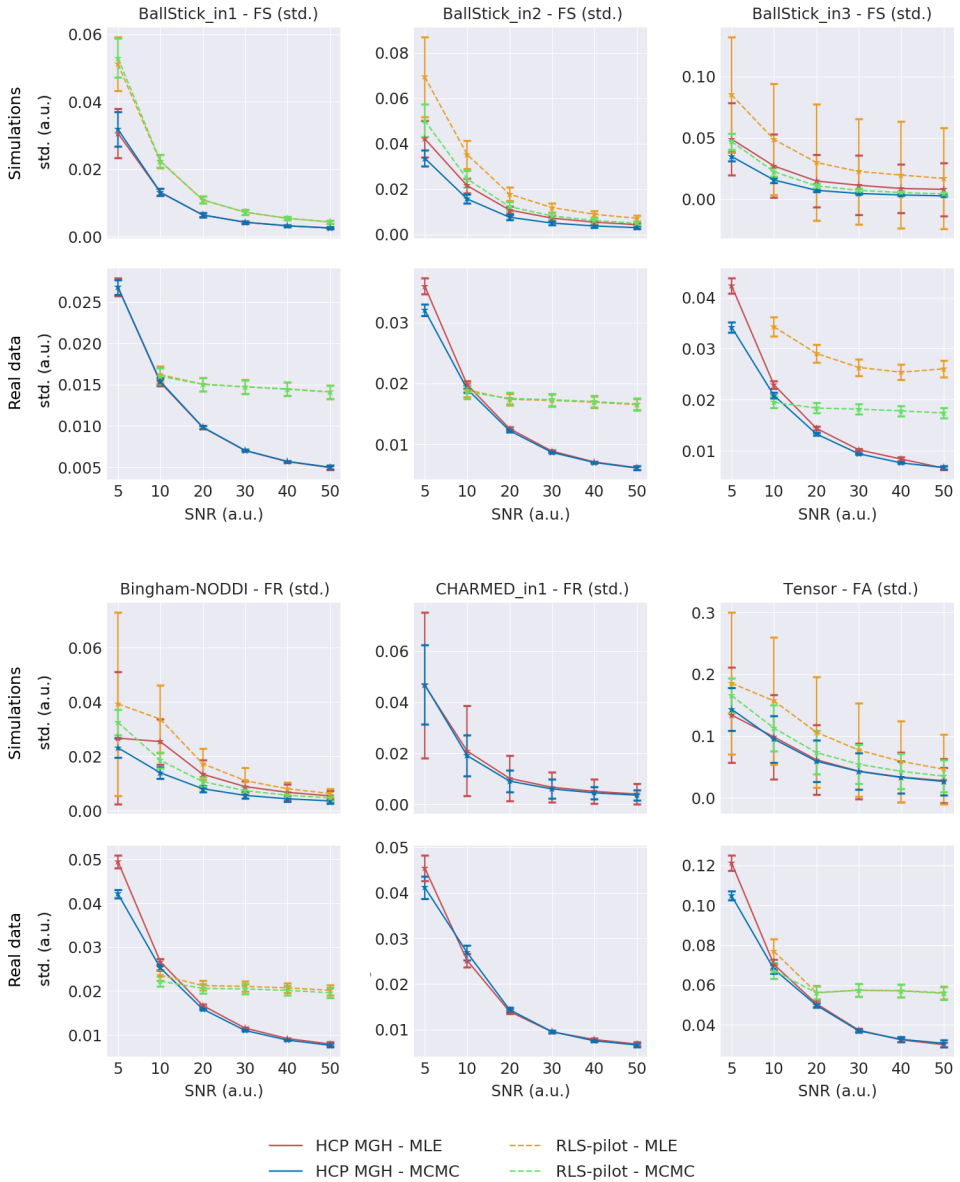


Figure 4.6: Effect of SNR on parameter standard deviations for simulated data and imaging data. Simulation results are over 10000 simulated voxels per SNR with a standard error of the mean (SEM) as error bar over 10 optimization and sampling trials. Real data results are for 10 subjects of the HCP MGH and 10 subjects of the RLS-pilot datasets, with SNR estimated as  $\text{mean}(b_0\text{volumes})/\text{std}(b_0\text{volumes})$ .



### 4.3.3 Group statistics

Figure 4.7 shows Bingham-NODDI FR results of three subjects of the HCP MGH dataset after co-registration, to illustrate the behavior of standard deviations in regions of white matter acquisition artifacts. The first subject (top row) has a clear artifact across the genu of the corpus callosum, perhaps due to incomplete fat saturation. This artifact is visible in both the mean parameter estimates and the standard deviation estimates. The second subject (middle row) shows a patch of relatively large standard deviations in and near the splenium of the corpus callosum, without an easily detectable alteration in the mean parameter map. For comparison, we show a third subject (bottom row) at the same contrast scaling, with no visible artifacts or alterations in either the mean or standard deviation estimates. This figure illustrates that parameter std. maps can play a role in detecting biased estimates resulting from imaging artifacts. In particular, artifacts which may not always be detectable in the parameter maps themselves.

Figure 4.8 shows four group statistic estimates, a regular (baseline) and three statistics using the three mentioned artifact reduction methods using the parameter variances. To reiterate, these were method one, a weighted average on all 35 subjects, method two, remove outlier subjects and apply regular averaging and method three, a weighted average with outlier subjects removed. Between regular and weighted averaging we computed a percentile difference map over a white matter mask to highlight the differences in estimates of both the group mean and group standard deviations.

For both the all-subjects and outliers-removed subject groups, the variance weighted mean is approximately lower across the artifact above the corpus callosum and, to a lesser degree, over the left internal and external capsules. For both groups, standard deviation estimates vary more between regular and weighted averaging, with a lower weighted average across the white matter artifact, equal values in most of the white matter and higher estimates near the border with gray matter. Group statistics with a few outlier subjects removed give lower averages and lower standard deviations for both weighted and regular averaging. Removing the outlier subjects brings the regular and weighted averages closer to each other, with percentile differences dropping by at least half.

The white matter artifact is most present in the regular average over all subjects (baseline), followed by regular averaging over the reduced group (artifact reduction method two), then by weighted averaging over all subjects (artifact reduction method one), and the artifact is least present in weighted averaging over the reduced group (artifact reduction method three).

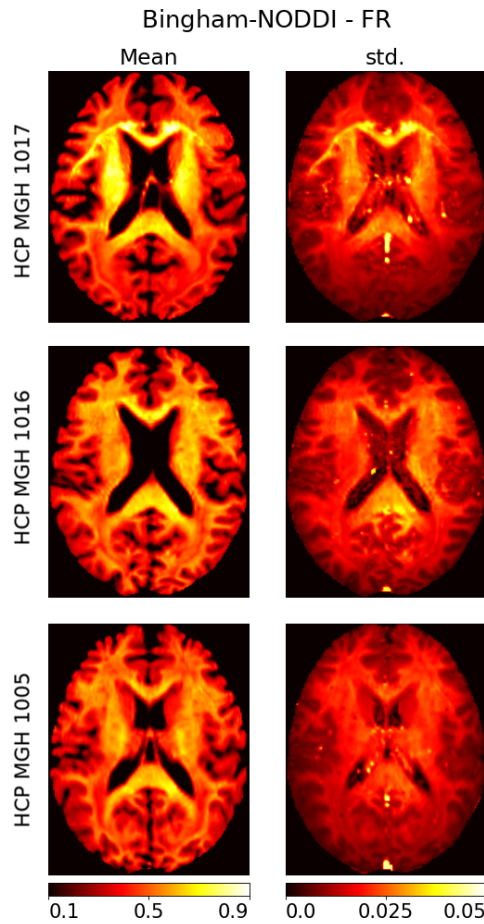


Figure 4.7: Illustration of artifacts in the HCP MGH datasets using the Bingham-NODDI Fraction of Restricted (FR) mean and standard deviation (std.) estimates from the MLE methodology. In the top row, estimates for HCP MGH subject 1017, with an artifact across the corpus callosum. In the middle row, estimates for HCP MGH subject 1016 with increased standard deviation estimates near a ventricle. In the bottom row, estimates for HCP MGH subject 1016 with no artifacts visible in the mean or standard deviation map.

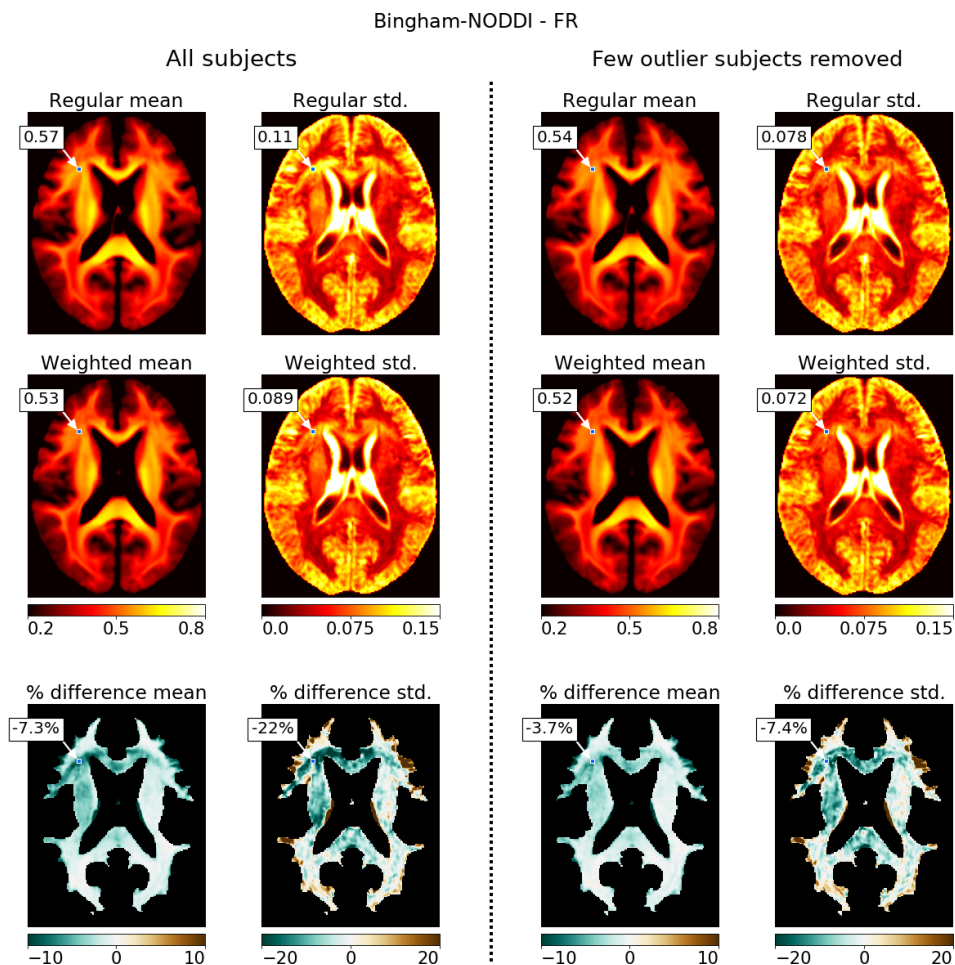


Figure 4.8: Group averages of Bingham-NODDI Fraction of Restricted (FR) estimates using the HCP MGH data, once over all 35 subjects (left two columns) and once over only 30 subjects where 5 outlier subjects have been removed (right two columns). First row, the regular mean and standard deviation, second row, the variance weighted mean and standard deviations, final row, percentage difference between regular and weighted averages. Point estimates and variances were computed using the MLE methodology.

## 4.4 Discussion

We evaluated parameter variance estimates as a quantification of parameter uncertainties. We compared standard deviation estimates from Maximum Likelihood Estimation (MLE) plus the Fisher Information Matrix (FIM) to those of Markov Chain Monte Carlo (MCMC) sampling and showed that both results are identical in  $\sim 98.7\%$  of the voxels. In terms of computer processing time, the estimates of MLE+FIM computed about 38x faster than those of MCMC. We then showed how data complexity and the signal-to-noise ratio can affect the parameter variances. Finally, we illustrated how the parameter variances can be applied in group studies to identify and downweight the effect of outliers, thereby decreasing the variance in group estimates, leading to an increase in statistical power of group studies.

### 4.4.1 Comparison of the FIM and MCMC

In general, we noted a close correspondence between the parameter distribution estimates from the FIM and those from MCMC sampling, with an average similarity of  $\sim 98.7\%$  across six models and two datasets. Compared on runtime, computing MLE+FIM is about 38x faster than the use of MCMC, for comparable results.

We made the explicit assumption that the parameter posterior distributions would follow a Gaussian distribution with a single mode. Theoretically, only a symmetrical distribution with a single mode would have an equal mode and mean. Therefore, if the MLE point estimate, which attempts to find the mode of the posterior, and the MCMC point estimate, which was computed here as the mean of the sample distribution are equal, then this is evidence towards symmetric single mode posteriors. Since our results from the FIM and MCMC were highly comparable (i.e. up to 98.7% of points estimates were indeed nearly equal), the Gaussian assumption is often confirmed. In the remaining 1.3% of the voxels, it could either be that the parameter posteriors were not fully Gaussian distributed, or that the posterior distributions were multi-modal. In the case of a multi-modal distribution, the FIM will give variance estimates around a single mode only, the mode found by the maximum likelihood routine. Our current MCMC methodology would provide an average and variance over all modes. To properly deal with multi-modal distributions when using MCMC, would require fitting a multi-modal normal distribution to the MCMC samples. If the parameters are not normally distributed, like for example near parameter boundaries or with skewed posteriors, the FIM no longer applies and MCMC would require different post-processing of the samples.

Compared on signal-to-noise (SNR), we note that the FIM provides higher standard deviation estimates at low SNR ( $< 10$ ) when compared to MCMC. These

differences are small and quickly vanish for  $\text{SNR} \geq 10$ . This follows results from astrophysics, where they recommend a minimum SNR of 10 to compute variances using the FIM, in gravitational wave assessments (Rodriguez et al., 2013).

In general, both the FIM and MCMC give comparable answers and both can be used for computing parameter standard deviations estimates to compute uncertainty. The only essential difference is one of computation time, computing a maximum likelihood point estimate together with the FIM is about 38x faster than using MCMC. This was expected, MCMC is generally known to be a time-consuming process, even when run on a GPU (Harms and Roebroek, 2018). MLE on the other hand can be applied very efficiently using a GPU (Harms, Fritz, et al., 2017) and computing the FIM requires only a few extra function evaluations (dependent on the number of parameters, see Appendix B.1).

#### 4.4.2 Effects on estimates of the standard deviations

There are several model and data characteristics that can affect standard deviation estimates, like data complexity, derived parameter maps and the signal-to-noise ratio. In general, these effects apply equally to both the FIM and MCMC.

Concerning data dependency, as expected, standard deviation estimates on the RLS-pilot dataset are generally higher than those on the HCP MGH dataset, reflecting a decrease in point estimate uncertainties with more data points. The same holds for the relatively large standard deviations in the Tensor Fractional Anisotropy (FA) estimates, since for the Tensor model we used only the data volumes with a low b-value.

A higher variance can additionally be observed for parameter maps which are not estimated directly but derived from the estimated parameters. This makes the variance of such derived parameters maps also a function of multiple variances, often leading to a higher total variance. This can for example be observed in the Tensor FA measure. The same compound effect could apply to the variance of the Fraction of Stick (FS) of the Ball&Stick models. For an increasing number of Sticks, the variance in FS is also a function of multiple volume fractions, which could increase the total variance.

For all models, parameter standard deviations are influenced by the signal-to-noise (SNR) ratio of the data, with a low SNR ( $< 10$ ) leading to a large increase in standard deviations. Both shown in real and simulated data, the effect of SNR on the standard deviation estimates seems to be more gradual after an  $\text{SNR} \geq 20$ .

### 4.4.3 Artifact detection

The computed parameter standard deviations (either from the FIM or MCMC) could be used as a tool for detecting acquisition artifacts. In one provided example (figure 4.7 top row), an artifact in the white matter was visible in both the parameter estimate and in the standard deviation as a patch of high intensity voxels. In another example (figure 4.7 middle row), a patch of high intensity voxels was visible in the standard deviation estimate but not in the parameter estimate itself. As such, standard deviation maps have the potential to be more sensitive in detecting white matter artifacts than point estimate maps themselves. A promising future development could be to include these standard deviation maps into quality control frameworks (Bastiani et al., 2019; Z. Liu et al., 2010; Oguz et al., 2014).

### 4.4.4 Increasing power in group studies

By weighing down voxels with a high standard deviation, weighted averaging can reduce the effect of white matter artifacts, approach lower and more accurate estimates of group variances and increase power of group statistics. In theory, if the within group datapoints are distributed with the same mean, variance weighted averaging promises the lowest possible variance in the group mean. We observe this in large parts of the white matter where weighted averaging lowers the variance in the group average as expected, thereby indirectly increasing power in group comparisons.

We have shown that some white matter artifacts are visible in the parameter standard deviation maps as patches of relatively large standard deviations. Since variance weighted averaging automatically reduces the effects of outliers whenever they have a large variance, variance weighted averaging automatically reduces the presence of artifacts. Even after removing a few subjects with a similar artifact, white matter averaging still reduces the presence of what appears to be a lower-expressed artifact in the remaining subjects. Due to this mechanism, subjects no longer need to be excluded from analysis, thereby improving the power of one's study.

Near the gray-white matter border we noticed some voxels where weighted averaging provides a higher variance than regular averaging. Theoretically, weighted averaging only predicts lower standard deviations if the points are distributed with the same mean. Misalignment between subjects can cause a single voxel to contain white matter for one subject and gray matter for another subject. Parameter estimates on such voxels will then be distributed with a different mean, leading to a higher group standard deviation when applying weighted averaging. This could be considered to be desirable, since such misalignment should not lead to high certainty group results and is therefore downweighted by the weighted

averaging. In other words, the weighted group standard deviation could diagnose alignment errors in group studies.

We note that although weighted averaging is shown here over subjects, weighted averaging can also be applied within subjects. For example, when averaging voxels over a white matter tract. In essence, weighted averaging can be applied in all cases where variances of an estimate are available. In the future this could be applied to tract based microstructure or tractometry studies (Bells et al., 2011), for tract based summary statistics with a lower variance.

#### 4.4.5 Conclusions and recommendations

Considering the advantages in processing time and close correspondence to Markov Chain Monte Carlo estimates, we recommend the use of the Fisher Information Matrix theory to quantify the uncertainties in parameter estimates. In individual subjects, the parameter standard deviations can help in detecting white matter artifacts as patches of relatively large standard deviations. In group statistics, we recommend using the parameter standard deviations by means of variance weighted averaging. Doing so can reduce the overall variance in group statistics and reduce the effect of data artifacts without discarding data from the analysis. Both these effects can lead to a higher statistical power in group studies.







# 5

## MDT, the Microstructure Diffusion Toolbox

---

Adapted from: R.L. Harms and A. Roebroek (2019). "GPU accelerated MRI analysis with the Microstructure Diffusion Toolbox (MDT)", *In prep.*.

## Abstract

We present the Microstructure Diffusion Toolbox (MDT), a framework and library for microstructure and diffusion modeling of Magnetic Resonance Imaging (MRI) data. We designed MDT for robust and efficient model fitting, modularity of models and optimization routines, and easy extensibility. This software package focuses on parametric microstructure models, formulated as biophysical multi-compartment models. Since these parametric models often share similar structures, we designed a modeling framework to maximize the reuse of code. By parallelizing computations on graphical processing units, this software greatly reduces computation times. Since the available models and optimization algorithms have been discussed in previous works, this work focuses on the design concepts, user interfaces and application examples. MDT is free, open source and well documented with installation steps for all major platforms and a step by step user guide, available at: <https://github.com/cbclab/MDT>.

## 5.1 Introduction

The Microstructure Diffusion Toolbox, MDT, is a software package dedicated to microstructure and diffusion modeling of Magnetic Resonance Imaging (MRI) data. We designed MDT for robust (i.e. lowest bias and variance), efficient (i.e. fast), and flexible (i.e. applicable to many models) microstructure model fitting. We provide a common platform for microstructure modeling with a multitude of models already provided. For comparable results, all models can be processed using the same noise models, non-linear optimization routines and MCMC sampling routines. For maximum performance, all models and algorithms were implemented with parallel processing capabilities, drastically reducing the runtime of the computations. MDT combines flexible modeling with fast processing, targeting both model developers and data analysts.

MDT focuses on parametric diffusion MRI models, formulated as biophysical multi-compartment models. These models subdivide the MRI signal into a weighted sum of compartments, inspired by different water pockets in biological tissue like intra-axonal fluid, extra-cellular fluid, myelin pockets, glial cells and cerebrospinal fluid (Assaf, Freidlin, et al., 2004; Assaf and Basser, 2005; Assaf, Blumenfeld-Katzir, et al., 2008; Alexander, Hubbard, et al., 2010; Panagiotaki et al., 2012; H. Zhang, Schneider, et al., 2012; Assaf, Alexander, et al., 2013; Fieremans et al., 2013; De Santis, Drakesmith, et al., 2014; De Santis, Assaf, et al., 2014; Jelescu, Veraart, Fieremans, et al., 2015). These models are typically of the form:

$$S = S_0 \sum_{i=0}^n w_i \mathbf{S}_i \quad (5.1)$$

Where  $S_0$  is the signal for the non-diffusion weighted (or  $b_0$ ) acquisitions,  $w_i$  the volume fractions (signal weights, signal fractions or water fractions) and  $\mathbf{S}_i$  is the signal function for the  $i$ 'th of  $n$  total compartments (Harms, Fritz, et al., 2017). Although there are many microstructure models available, these models often share similar compartments. To be able to reuse previously implemented compartments, we designed a modeling framework specific for microstructure modeling of diffusion MRI data. This modeling framework defines compartments as separate identities which can then be combined into microstructure models using a high level modeling language.

For maximum performance, all microstructure models and all analysis methods were implemented for processing on graphical processing units (GPUs). Previous work has shown that GPUs can greatly reduce computation times by parallelizing the computations over many processing cores (Harms, Fritz, et al., 2017; Harms and Roebroek, 2018; Hernández et al., 2013; Hernandez-Fernandez et

al., 2018). Since most microstructure models consider the voxels to be independent of each other, all voxels can be processed in parallel. In addition, intra-voxel parallelization is possible wherever computations can be parallelized over the MRI volumes. By making use of both these levels of parallelization, MDT greatly reduces the computation times from days and hours to minutes and seconds (Harms, Fritz, et al., 2017; Harms and Roebroek, 2018). This work focuses on the design concepts, user interfaces and application examples of MDT.

## 5.2 Design concepts

The software is split into two packages, the Microstructure Diffusion Toolbox (MDT) and the Multi-threaded Optimization Toolbox (MOT), as illustrated in the software overview in figure 5.1. The MDT package contains the microstructure modeling framework while the MOT package contains the GPU accelerated optimization and sampling algorithms. This division hides the complexity of the numerical routines, allowing MDT to focus on user friendly microstructure model formulation.

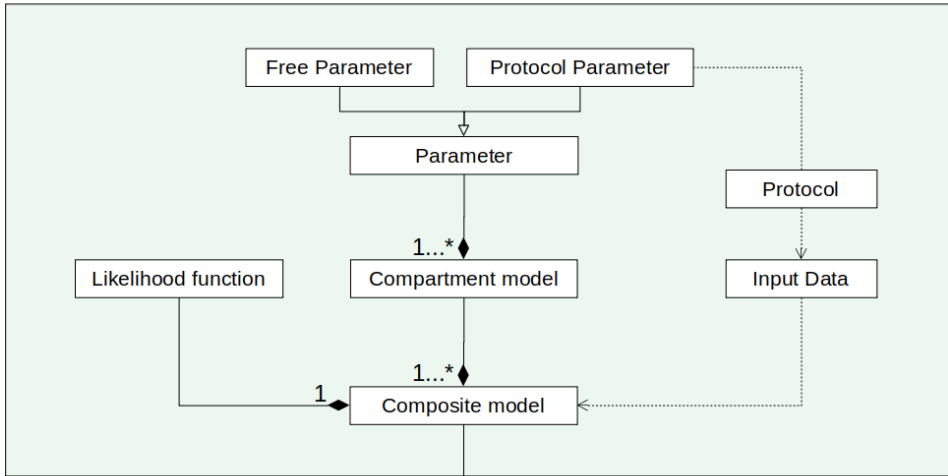
Both software packages are written in the programming languages Python and OpenCL. The optimization and sampling routines in the MOT library are implemented in OpenCL, a programming language specifically designed for high performance computations. OpenCL can leverage both multi-core central processing units (CPU's) as well as graphical processing units (GPU's) (Stone, Gohara, and Shi, 2010). GPU's can process thousands of data points simultaneously, often allowing for large speed-gains compared to the traditional CPU's. Although OpenCL can be a very efficient language, it is very technical and deals with multi-core hardware at a detailed level. To lower the technical expertise required for programming models in MDT, we wrapped the OpenCL functions in Python code. Python is generally recognized as an easy to use programming language and is widely used for scientific analysis, as for example in the neuroimaging software packages Dipy (Garyfallidis et al., 2014) and Nipype (Gorgolewski et al., 2011).

We first summarize the MOT package as it provides the numerical foundation for the MDT package. All remaining subsections discuss various aspects of the modeling framework in MDT.

### 5.2.1 The Multi-threaded Optimization Toolbox

The Multi-threaded Optimization Toolbox (MOT) contains all the numerical routines necessary for optimizing and sampling the models built in MDT. The MOT package is a domain agnostic, general purpose numerical optimization library,

## MDT



## MOT



Figure 5.1: Unified Modeling Language (UML) diagram of the MDT and MOT software packages, in which diamond arrow are aggregation relations (has-a), open arrows are dependency relations (depends-on) and closed arrows are usage relations (uses-a). Composite models are composed out of one or more compartments and combined with a likelihood function. Compartment models are composed out of two types of parameters, free and protocol parameters. Protocol parameters are implicitly linked to the Input Data, an object containing all relevant data needed for model fitting and sampling. Composite models can be optimized or sampled using the GPU accelerated routines from the MOT package.

specialized in parallel processing. In contrast, MDT is a domain specific microstructure modeling toolbox and is invariant to the numerical library used for processing the data. Although MDT can work with any numerical optimization library, we primarily use MOT for the optimization and MCMC sampling due to its parallel processing capabilities.

The advantage of providing MOT as a separate software package is that it opens the possibility of using the developed parallel routines for purposes other than that of MRI microstructure analysis. For example, in (Kashyap et al., 2018) the authors use the MOT software package for combining the multiple channels of a multi-channel functional MRI acquisition.

All routines in the MOT library are implemented in OpenCL. In addition to the

choice of language (OpenCL), MOT puts strong emphasis on efficient processing. For example, the numerical routine used to compute the Fisher Information Matrix has been specifically adapted to only use the upper triangular elements of the Hessian matrix. By recognizing the Hessian to be symmetric and subsequently adapting the code we effectively halved the run time and storage space. Other optimizations are caching common computations and using approximations to costly functions wherever it can be done without sacrificing computational accuracy.

For maximum likelihood estimation, we (re-)implemented four different optimization algorithms (Harms, Fritz, et al., 2017), the Powell (Powell, 1964), Levenberg-Marquardt (Levenberg, 1944), Nelder-Mead simplex (Nelder and Mead, 1965) and Subplex (Rowan, 1990) algorithms. For MCMC sampling (Harms and Roebroek, 2018), we implemented the Adaptive Metropolis Within Gibbs (AMWG) (G. O. Roberts and Rosenthal, 2009), Single Component Adaptive Metropolis (SCAM) (Haario, Saksman, and Tamminen, 2005), t-walk sampler (Christen and Foxy, 2010) and Metropolis Within Gibbs (MWG) (Ravenzwaaij, Cassey, and Brown, 2016). By default, MDT uses the Powell routine for maximum likelihood estimations and the AMWG routine for MCMC sampling (Harms, Fritz, et al., 2017; Harms and Roebroek, 2018).

## 5.2.2 Providing the input data

All input data relevant for maximum likelihood estimation and MCMC sampling are stored in a single data structure, called the `InputData` structure. This structure should at least contain the MRI volumes, the volume-by-volume acquisition details, and a mask for marking the region of interest. It can optionally contain a gradient deviations matrix, a noise standard deviation estimate and volume weights.

The volume-by-volume acquisition details are primarily provided by the user in a Protocol file, a tabular representation of the acquisition details with for every measured volume a row and for every acquisition parameter a column. For example, a simple protocol file containing only b-vectors and b-values looks like:

```
#gx, gy, gz, b
0.000e+00  0.000e+00  0.000e+00  0.000e+00
5.572e-01  6.731e-01  -4.860e-01  1.000e+09
4.110e-01  -5.254e-01  -7.449e-01  2.000e+09
...
```

A protocol file additionally containing diffusion times can be defined as:

```
#gx, gy, gz, Delta, delta, b
-0.00e+0  0.00e+0  0.00e+0  2.18e-2  1.29e-2  0.00e+0
2.92e-1   1.71e-1 -9.41e-1  2.18e-2  1.29e-2  3.00e+9
-9.87e-1 -8.54e-3 -1.60e-1  2.18e-2  1.29e-2  5.00e+9
...
```

The first row of the protocol denotes the name of each of the columns. All other rows contain the tabularized acquisition settings with values in the International System of Units (SI) units. By matching column names with model parameter names (see modeling section), columns in the protocol file are automatically linked to model parameter, enabling flexibility in modeling and allowing MDT process a variety of MRI modalities with a variety of changing acquisition parameters (such as echo time TE, repetition time TR, inversion time TI, mixing time TM and flip angle alpha). The columns in the protocol are unordered and there can be as many columns as desired, only the name of the columns, are significant.

Since every cell in the protocol file can have unique values, this format enables the use of a variety of diffusion MRI sampling schemes like cartesian q-space sampling, single-shell or multi-shell sampling and sparse sampling schemes, such as compressed sensing diffusion imaging. In most of these acquisition schemes, the diffusion times and b-values are not constant over acquisitions, meaning that MDT's flexibility is required for analyzing multi-shell or compressed sensing data.

For applications that demand a specific acquisition setting per voxel, MDT also allows loading 3d or 4d volumes to the input data. These volumes can hold for every voxel (and optionally every volume) a distinct protocol setting, enabling the model to account for voxel-wise deviations in the acquisition settings, such as those caused by gradient non-linearities or flip angle inhomogeneities. To save memory, gradient non-linearities can also be represented as a gradient deviations matrix, a 3d or 4d matrix with for every voxel, or optionally every volume a 3x3 matrix, with an affine transformation of the b-vectors and b-values.

The input data can additionally contain a noise standard deviation estimate and volume weights. The estimate of the noise standard deviation is either given as a scalar or as a voxel-wise volume. The noise standard deviations are used in the likelihood model during model fitting and sampling. The volume weights can be provided as a 4d volume, with for every voxel and volume a normalized scalar which weights the objective function values per observation. These volume weights can be used for voxel-wise compensation of artifacts or outlier volumes. (Sairanen, Leemans, and Tax, 2018).



### 5.2.3 Adding a microstructure model to MDT

Microstructure models in MDT are structured in three layers: parameters, compartment models and composite models, as shown in figure 5.1. The composite models correspond to complete microstructure models from the literature, such as NODDI (H. Zhang, Schneider, et al., 2012), CHARMED (Assaf and Basser, 2005) and Ball&Stick (Behrens et al., 2003). These composite models are build using compartment models, like the NODDI intra-cellular compartment (H. Zhang, Schneider, et al., 2012) or the Ball and Stick compartments in the Ball&Stick model (Behrens et al., 2003). Each compartment has its own set of parameters. These parameters are divided between free parameters and protocol parameters, where free parameters indicate that the parameter needs to be estimated from the data and protocol parameters represent known values such as acquisition parameters.

#### Composite models

To illustrate modeling in MDT, we show how the Ball&Stick model can be implemented in MDT. For one diffusion volume, the unidirectional Ball&Stick model is mathematically defined as (Behrens et al., 2003):

$$S = S_0 * \left[ w_{\text{Ball}} * e^{-bd_{\text{Ball}}} + w_{\text{Stick}} * e^{-bd_{\text{Stick}}(\mathbf{n} \cdot \mathbf{g})^2} \right] \quad (5.2)$$

Here,  $S_0$  is the level of the unweighted signal or  $b_0$  volume, the  $w$  are volume fractions that should sum to one, the diffusion b-value is given by  $b$ , and the  $d$  parameters represent diffusivities. The diffusion gradient vector is given by  $\mathbf{g}$ . The expected direction of diffusion is given by  $\mathbf{n}$  and is typically parameterized as the spherical coordinates  $\theta$  and  $\phi$ .  $S_0$ ,  $w$ ,  $d$ , and  $\mathbf{n}$  are free parameters. The  $b$  and  $\mathbf{g}$  are protocol parameters (i.e. acquisition settings).

If we would abstract away the parameters in equation 5.2, we can identify the following functional elements:

$$S = S_0 * [\text{Weight}_{\text{Ball}} * \text{Ball} + \text{Weight}_{\text{Stick}} * \text{Stick}] \quad (5.3)$$

With the mapping  $S_0 := S_0$ ,  $\text{Weight}_{\text{Ball}} := w_{\text{Ball}}$ ,  $\text{Ball} := e^{-bd_{\text{Ball}}}$ ,  $\text{Weight}_{\text{Stick}} := w_{\text{Stick}}$  and  $\text{Stick} := e^{-bd_{\text{Stick}}(\mathbf{n} \cdot \mathbf{g})^2}$ . In terms of multi-compartment modeling, we identified the compartments Ball and Stick, each individually modulated by a weight in a weighted-sum signal model. (Behrens et al., 2003). If we assume for a moment that each of  $S_0$ ,  $\text{Weight}$ ,  $\text{Stick}$  and  $\text{Ball}$  are defined in MDT, we could implement equation 5.3 as:

```

class BallStick(CompositeModelTemplate):
    model_expression = '''
        S0 * ( (Weight(w_ball) * Ball) +
              (Weight(w_stick) * Stick) )
    '''

```

This example defines the Ball&Stick model as a composite model by inheriting from `CompositeModelTemplate`, using the `model_expression` attribute. MDT considers each of `S0`, `Weight`, `Ball` and `Stick` as a compartment, with its own signal equation, and the model expression allows arbitrary compositions of compartments using the algebraic operators  $\{*, /, +, -\}$ . To ensure unique compartment names, the syntax `Weight(w_ball)` and `Weight(w_stick)` defines two separate `Weight` compartments, with one `Weight` nicknamed `w_ball` and the other `w_stick`. The normalization rule for the weights (Behrens et al., 2003), i.e. all model weights must sum to unity, is automatically handled by MDT.

### Compartment models

The compartment models represent distinct parts of a model equation (Panagiotaki et al., 2012). In the Ball&Stick example we recognized the compartments `S0`, `Weight`, `Ball` and `Stick`. Each of these compartments need to be defined in MDT before being used in a composite model. As an example, we show how the `Stick` compartment can be implemented in MDT:

```

class Stick(CompartmentTemplate):
    parameters = ('g', 'b', 'd', 'theta', 'phi')
    dependencies = ['SphericalToCartesian']
    cl_code = '''
        float4 n = SphericalToCartesian(theta, phi);
        return exp(-b * d * pown(dot(n, g), 2));
    '''

```

This defines `Stick` as a compartment model by inheriting from `CompartmentTemplate`. The list of parameters are references to parameter objects. The dependencies attribute list functions this compartment model depends on. The `cl_code` attribute contains the OpenCL code for the compartment model, which in this case implements the equation  $e^{-bd(\mathbf{n}\cdot\mathbf{g})^2}$ .

### Parameters

Each compartment model depends on one or more parameters. This can either be free parameters or protocol parameters depending on their usage. Free parameters can be defined as follows:

```
class theta (FreeParameterTemplate):
    init_value = PI / 2.0
    lower_bound = 0
    upper_bound = PI
```

This defines `theta` as a free parameter by inheriting from `FreeParameterTemplate`. Each free parameter must define an initialization value and a lower and upper bound. Defining a parameter as a free parameter indicates that this parameter is meant to be estimated from the data.

Protocol parameters are defined as follows:

```
class b (ProtocolParameterTemplate):
    pass
```

This defines `b` as a protocol parameter by inheriting from `ProtocolParameterTemplate`. No further definition is required for protocol parameters. Defining a parameter as a protocol parameter indicates that this parameter depends on provided acquisition data. Parameters of this type are automatically coupled to the columns in the protocol file (see the section on providing the input data).

### Extending the Ball&Stick model to multiple Sticks

Once defined, each of these components can be reused for new model definitions. For example, after having defined all parameters and compartments for a Ball&Stick model with a single Stick, extending this to a two Stick model can be as simple as:

```
class BallStick_r2 (CompositeModelTemplate):
    model_expression = '''
        S0 * ( (Weight(w_ball) * Ball) +
              (Weight(w_stick0) * Stick(Stick0)) +
              (Weight(w_stick1) * Stick(Stick1)) )
    '''
```

By default, MDT ships with a large library of parameter definitions, compartment models and composite models, allowing the user to quickly get started with implementing a new model and easily compose new models using existing components.

### 5.2.4 Parameter dependencies

Some diffusion MRI microstructure models have dependencies between the parameters of different compartments. These dependencies can be shared parameters, or represent complex combinations of multiple compartment parameters. To keep the compartment models reusable, MDT supports parameter dependencies in composite models.

Without any dependencies, MDT defines the NODDI model (H. Zhang, Schneider, et al., 2012) as:

```
class NODDI (CompositeModelTemplate) :
    model_expression = '''
        S0 * ((Weight(w_csf) * Ball) +
              (Weight(w_ic) * NODDI_IC) +
              (Weight(w_ec) * NODDI_EC))
    '''
```

Where NODDI.IC and NODDI.EC are the compartment models of respectively the intra-cellular and the extra-cellular water compartments (H. Zhang, Schneider, et al., 2012).

To fully implement the NODDI model, we need to add the tortuosity assumption to the model definition (H. Zhang, Schneider, et al., 2012). The tortuosity assumption models the exchange of water between the intra-cellular and extra-cellular compartments and is defined as:

$$d_{ec\perp} = d_{ec\parallel} (w_{ec} / (w_{ec} + w_{ic})) \quad (5.4)$$

This states that the perpendicular diffusivity of the extra-cellular compartment  $d_{ec\perp}$  is fixed to the extra-cellular parallel diffusivity  $d_{ec\parallel}$  multiplied by the ratio of the extra-cellular water volume fraction. As such,  $d_{ec\perp}$  is no longer a free parameter but is fixed to the value of the dependency formulation, i.e. it is calculated from the values of the parameters  $d_{ec\parallel}$ ,  $w_{ec}$ , and  $w_{ic}$  and updated when the value of any of these changes.

The tortuosity dependency can be added to our NODDI definition using the `fixes` attribute of the composite model definitions. This attribute allows us to define arbitrary parameter dependencies to the parameters of a compartment model. The new model definition then becomes:

```

class NODDI (CompositeModelTemplate):
    model_expression = '''
        S0 * ((Weight(w_csf) * Ball) +
              (Weight(w_ic) * NODDI_IC) +
              (Weight(w_ec) * NODDI_EC))
    ...
    fixes = {
        'NODDI_EC.dperp0':
            'NODDI_EC.d * w_ec.w / (w_ec.w + w_ic.w)'
    }

```

Here, we use the naming format “<compartment>.<parameter>” to refer to a specific parameter of a specific compartment. For example, the string “NODDI-EC.dperp0” refers to the “dperp0” parameter of the NODDI-EC compartment, which refers to the parameter  $d_{ec\perp}$  in equation 5.4.

The “fixes” attribute can also be used to fix parameters to constant values or to lock parameters of two compartments together to ensure that they are optimized as a single value. For example, the NODDI model sets the diffusivities of the various compartments to a-priori values. Furthermore, the dispersion distribution and the orientation of the intra-cellular and extra-cellular compartments are optimized together. These dependencies can be added to our NODDI implementation by changing the “fixes” attribute of the NODDI model as follows:

```

class NODDI (CompositeModelTemplate):
    ...
    fixes = {
        'NODDI_EC.dperp0':
            'NODDI_EC.d * w_ec.w / (w_ec.w + w_ic.w)',
        'NODDI_IC.d': 1.7e-9,
        'NODDI_EC.d': 1.7e-9,
        'Ball.d': 3.0e-9,
        'NODDI_EC.kappa': 'NODDI_IC.kappa',
        'NODDI_EC.theta': 'NODDI_IC.theta',
        'NODDI_EC.phi': 'NODDI_IC.phi'
    }

```

### 5.2.5 Likelihood functions

Maximum likelihood estimation and MCMC sampling require the objective function to be stated in terms of a likelihood. Typical likelihood functions in MRI analysis are the Gaussian, Offset Gaussian and Rician (Alexander, 2009; Panagiotaki et al., 2012; Harms, Fritz, et al., 2017) likelihoods, each differing in the way they

account for the noise inherent to MR imaging. By default, MDT uses the Offset Gaussian likelihood model (Harms, Fritz, et al., 2017). This can be changed by defining the attribute `likelihood_function` of a composite model with the name of another likelihood model. For example, to use a Rician noise model with the Ball&Stick model defined earlier, we define the composite model as:

```
class BallStick(CompositeModelTemplate):
    model_expression = '''
        S0 * ( (Weight(w_ball) * Ball) +
              (Weight(w_stick) * Stick) )
    '''
    likelihood_function = 'Rician'
```

## 5.2.6 Initialization methods

Previous work has shown that providing a good starting point for a model can benefit the accuracy, precision and run time of non-linear optimization (Harms, Fritz, et al., 2017), and increase convergence times for MCMC sampling (Harms and Roebroek, 2018). By default, when optimizing a model, MDT will initialize any given model using an initialization dictionary, which functions similar to the Cascade Initialize of previous work (Harms, Fritz, et al., 2017). Cascading describes a method for obtaining a starting point for a complex model by first fitting a simpler model.

The automatic initialization can be disabled, or be overruled by providing a user-defined set of initialization values for the free parameters of a composite model. Similarly, for MCMC sampling, the user can provide an initial starting point, which is typically a maximum likelihood estimate from non-linear optimization. For example, to optimize the NODDI model given some input data we can use the following Python code:

```
mdt.fit_model('NODDI', input_data, 'output')
```

This would first compute a Ball&Stick model and use the angles of diffusion to initialize the NODDI model, as described in (Harms, Fritz, et al., 2017). To disable automatic initialization, we can use:

```
mdt.fit_model('NODDI', input_data, 'output',
              use_cascaded_inits=False)
```

Alternatively, if we would like to use the default initialization but overwrite the initial values for the NODDI orientation angles, we can use:

```

mdt.fit_model(
    'NODDI', input_data, 'output',
    initialization_data={
        'inits': {'NODDI_IC.theta': np.array(...),
                  'NODDI_IC.phi': np.array(...)}
    }
)

```

The last example would run a Ball&Stick estimate first, use that to initialize the angles and then overwrite that with the provided angles.

### 5.2.7 Defining automatic post-processing

Various microstructure models in diffusion MRI describe scalar maps that are not estimated directly but are computed from the estimated parameters after optimization or MCMC sampling. For example, the Orientation Dispersion Index (ODI) of the NODDI model is defined as (H. Zhang, Schneider, et al., 2012):

$$\text{ODI} = \frac{2}{\pi} \arctan(1/\kappa) \quad (5.5)$$

and can only be computed after optimization or MCMC sampling of the  $\kappa$  parameter of the NODDI model.

To automate the computation of additional maps, MDT allows defining post-processing routines for the composite models. For example, to automatically compute the ODI after optimizing the parameters of the NODDI model, we add a new function to the `extra_optimization_maps` attribute of the NODDI composite model:

```

class NODDI(CompositeModelTemplate):
    ...
    extra_optimization_maps = [compute_odi]

```

where `compute_odi` is a function accepting a dictionary of input maps and returns a dictionary of output maps:

```

def compute_odi(results):
    kappa = results['NODDI_IC.kappa']
    return {'ODI': (2 / pi) * arctan2(1.0, kappa)}

```

The `extra_optimization_maps` attribute allows multiple post-processing routines to be defined, and each routine can return multiple data volumes. Optionally, these post-processing routines can also be added to the

compartments models, allowing the post-processing routines to be reused by all composite models that use that compartment model. For post-processing the results of MCMC sampling, the same feature exists based on sampling results and can be defined with the `extra_sampling_maps` attribute.

Supplementary to these extra maps, MDT always computes a few generic statistical maps after optimization or MCMC sampling. Additional results after optimization include the variances and co-variances of each of the estimated model parameters (computed using the Fisher Information Matrix, see Chapter 4), the log-likelihood of the estimate and the Bayesian Information Criterion (Schwarz, 1978). These additional results are written as additional nifti files. Additional results after MCMC sampling include an estimation of the effective sample size (Harms and Roebroeck, 2018), a maximum a posteriori estimate, a maximum log-likelihood estimate, the average acceptance rate, and a univariate normal fit to the samples of each of the parameters of a model, providing a Gaussian mean and Gaussian standard deviation for each sampled free parameter.

## 5.3 Interfaces to MDT

MDT offers several interfaces for data processing and data visualization. Each of these interfaces differs in ease-of-use, complexity and programming freedom. For single subject data exploration there is a user friendly graphical user interface and a command line interface. For access to all features there is a well documented Python programming interface. For large group studies there is a batch fitting routine. For visualizing results, MDT offers a data visualization tool.



### 5.3.1 Graphical interface

The graphical interface allows single subject MRI analysis without in depth knowledge of programming. To obtain a maximum likelihood estimate of your model using the GUI you only need to provide the (pre-processed) MRI data and your acquisition settings. The GUI then guides you through the steps needed to fit your desired model to the dataset. For feature reuse, the GUI automatically creates a Python and a Bash/command line script file allowing reproducibility of the model fitting using a script. Figure 5.2 illustrates the GUI.

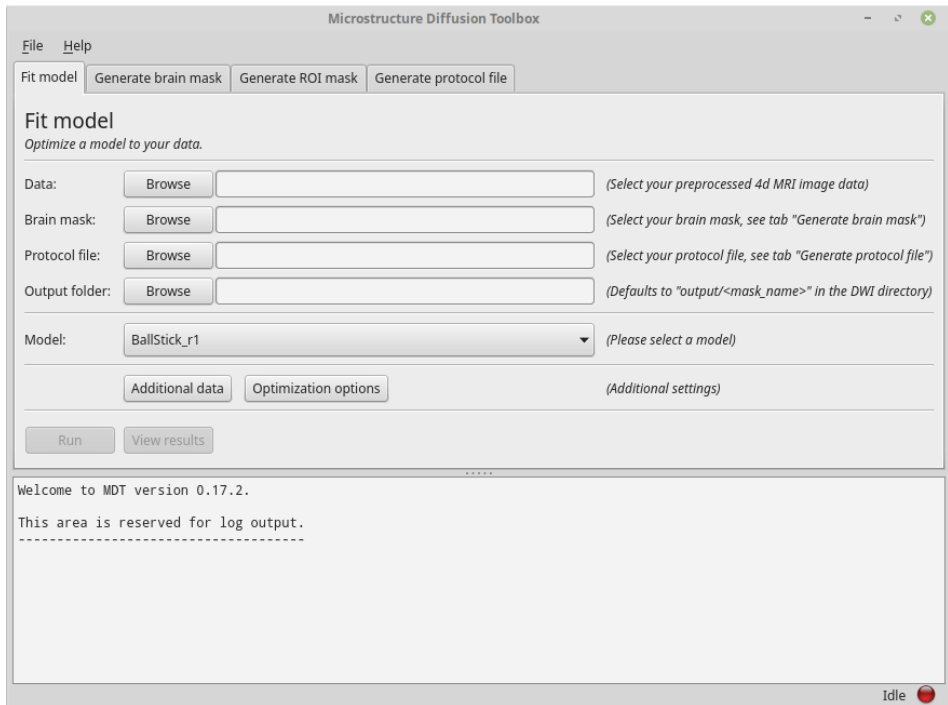


Figure 5.2: A screenshot of the MDT GUI

### 5.3.2 Command line interface

MDT has a rich selection of command line functions, allowing you to generate protocol files, create brain masks, apply mathematical operations on nifti files and fit a model. Code listing 5.1 illustrates the steps from data to model fitting using the MDT example data. The commands in code listing 5.1 can be executed in a command line console, like the `bash` shell on Linux or Mac, or the `cmd` prompt on Windows.

---

```
1 mdt-get-example-data /tmp
2
3 mdt-create-protocol blk_b2k.bvec blk_b2k.bval
4
5 mdt-create-mask data.nii.gz data.prtcl
6
7 mdt-model-fit NODDI \
8     blk_b2k_example_slices_24_38.nii.gz \
9     blk_b2k.prtcl \
10    data_mask.nii.gz
11
12 mdt-view-maps output/BallStick_r1
```

---

Listing 5.1: Command Line Interface (CLI) example of fitting NODDI to the MDT example data. Line 1 loads the example data, line 3 prepares the protocol, line 5 prepares the mask and line 7 to 10 fits the NODDI model to the data. Line 12 opens the MDT maps visualizer for inspecting the results.

### 5.3.3 Python interface

The Python interface provides access to all features of MDT, including model fitting and MCMC sampling. Code listing 5.2 illustrates the steps from data to model fitting using the MDT example data, following the same steps as the command line example above. The code in listing 5.2 can easily be extended to MCMC sampling, as shown in listing 5.3

```
1 import mdt
2
3 mdt.get_example_data('/tmp')
4
5 protocol = mdt.create_protocol(
6     bvecs='blk_b2k.bvec',
7     bvals='blk_b2k.bval',
8     out_file='blk_b2k.prtcl')
9
10 mdt.create_median_otsu_brain_mask(
11     'blk_b2k_example_slices_24_38',
12     'blk_b2k.prtcl')
13
14 input_data = mdt.load_input_data(
15     'blk_b2k_example_slices_24_38',
16     'blk_b2k.prtcl',
17     'blk_b2k_example_slices_24_38_mask')
18
19 mle = mdt.fit_model(
20     'NODDI', input_data, 'output')
21
22 mdt.view_maps(mle)
```

---

Listing 5.2: Python interface example of fitting NODDI to the MDT example data. Line 3 loads the example data, line 5 to 8 prepares the protocol, line 10 to 12 prepares the mask, line 14 to 17 load all the data into a structure and line 19 and 20 fit the NODDI model to the data using the prepared data. Line 22 opens the MDT maps visualizer for inspecting the Maximum Likelihood Estimate (mle).

---

```
23 ...
24
25 samples = mdt.sample_model(
26     'NODDI', input_data, 'output',
27     nmr_samples=1000,
28     burnin=0,
29     thinning=0,
30     initialization_data={'inits': mle}
31 )
```

---

Listing 5.3: MCMC sampling the NODDI model on the bundled MDT example data using Python. This code listing extends listing 5.2 with MCMC sampling. In this example we sample the NODDI model with 1000 samples without thinning and burn-in.

### 5.3.4 Batch fitting

For ease of use with group studies or population studies, MDT has batch fitting functionality that can automatically detect common directory layouts and automatically load each subject's data. A specific batch profile for the Human Connectome Project (HCP) studies is included, allowing MDT to automatically load all the subjects of the HCP MGH (Fan et al., 2016) and HCP WuMinn (Van Essen et al., 2012) studies. As an example, to fit the NODDI model to all HCP MGH or HCP WuMinn subjects the following command suffices:

```
mdt-batch-fit /hcp_data_dir/ NODDI
```

This command automatically detects the correct study and fits the selected model to all the subjects. This process can be interrupted and resumed at a later stage without losing intermediate results.

Since group studies often have a unique directory layout, MDT can recognize studies based on the directory layout described in the batch profiles. New batch profiles can be added dynamically, allowing the extension of this functionality to custom group studies.

### 5.3.5 Map visualizer

To visualize the scalar maps resulting from model fitting or sampling, MDT offers a visualization tool called the MDT maps visualizer. This visualization program is meant to visualize a single slice of multiple scalar maps for comparison across parameters, as illustrated in figure 5.3.

The provided data is plotted according to a text based plot configuration based on the YAML syntax (<https://yaml.org/>). This allows scripting figure plots and storing repeatable plot configurations for later reference. As an example, given the data, the results of figure 5.3 can be reproduced exactly by loading the configuration of listing 5.4 in the visualizer.

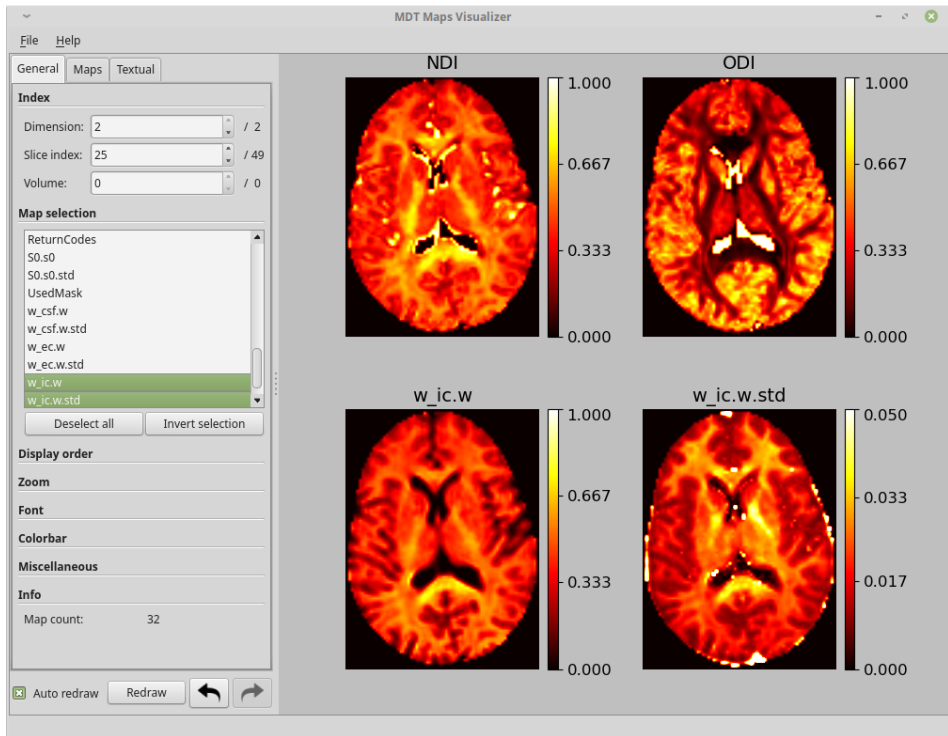


Figure 5.3: A screenshot of the MDT Maps visualizer showing model fitting results of the NODDI model on the NODDI example data (H. Zhang, Schneider, et al., 2012; Winston et al., 2014).

---

```
maps_to_show: [NDI, ODI, w_ic.w, w_ic.w.std]
slice_index: 25
zoom:
  p0: {x: 27, y: 21}
  p1: {x: 99, y: 118}
map_plot_options:
  NDI:
    scale: {use_max: true, use_min: true,
            vmax: 1.0, vmin: 0.0}
  ODI:
    scale: {use_max: true, use_min: true,
            vmax: 1.0, vmin: 0.0}
  w_ic.w:
    scale: {use_max: true, use_min: true,
            vmax: 1.0, vmin: 0.0}
  w_ic.w.std:
    scale: {use_max: true, use_min: true,
            vmax: 0.05, vmin: 0.0}
```

---

Listing 5.4: Example of a plot configuration for the MDT map visualizer. By storing the configurations, plots can be reproduced exactly.

## 5.4 Application examples

To demonstrate the flexibility of MDT with specific software, hardware setups, and models, we applied MDT to a range of datasets using multiple software and hardware combinations. The following subsections show a variation of examples, with each example using different data, software configurations and with varying hardware. Table 5.1 shows an overview of the the models, software and hardware settings used to compute each of the results.

<b>Model</b>	<b>Likelihood model</b>	<b>Estimation routine</b>	<b>Hardware</b>	<b>Operating system</b>
NODDI	Rician	Levenberg-Marquardt	AMD Fury X	Linux
CHARMED_in3	Offset-Gaussian	AMWG (MCMC)	NVIDIA Tesla K40c	Windows
Bingham-NODDI	Gaussian	Powell	Intel Xeon E5-2650v3	Windows

Table 5.1: Overview of the software settings and hardware we used to compute each of the application examples.

### 5.4.1 NODDI

Figure 5.4, shows MDT NODDI fitting results on the example data pre-supplied with the NODDI Matlab toolbox (H. Zhang, Schneider, et al., 2012; Winston et al., 2014), a two-shell dataset with 24 b-val 700s/mm<sup>2</sup>, 48 b-val 2000s/mm<sup>2</sup>, and 9 b<sub>0</sub> measurements (Winston et al., 2014). For these results we used the Levenberg-Marquardt optimization routine, the Rician likelihood model and computed it using an AMD Fury X graphics card on Linux. The total run time for whole brain fitting was 3 minute and 1 second. If we instead of Levenberg-Marquardt we would have used the Powell routine, the runtime would have dropped to 1 minute and 15 seconds, as computed on a single AMD Fury X graphics card.

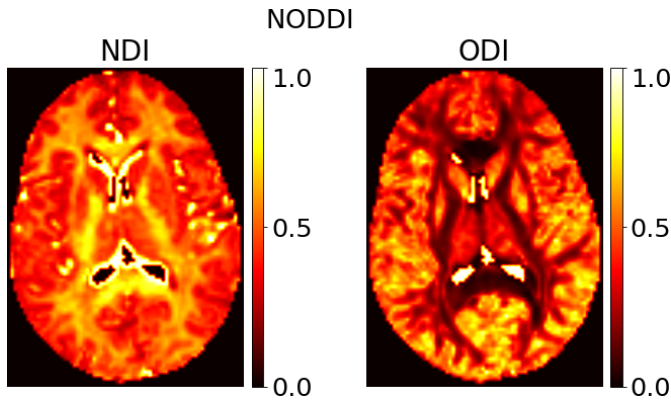


Figure 5.4: NODDI Neurite Dispersion Index (NDI) and Orientation Dispersion Index (ODI) results computed on the NODDI example dataset, using the Levenberg-Marquardt optimization method with the Rician likelihood model using an AMD Fury X graphics card running on Linux.



### 5.4.2 CHARMED\_in3

To illustrate the use of the MCMC sampling options in MDT, we sampled the CHARMED\_in3 model (Assaf and Basser, 2005) on subject mgh\_1003 of the HCP MGH datasets (Fan et al., 2016). This dataset is from the MGH study of the Human Connectome Project (HCP), scanned at 1.5mm isotropic with 4 shells of  $b=1000, 3000, 5000, 10,000, s/mm^2$ , with respectively 64, 64, 128, 393 directions and with 40  $b_0$  volumes. The high  $b$ -values make this dataset well suited for the CHARMED models which require such high  $b$ -values. The CHARMED model can be defined using one or more intra-neuronal compartments meant to resolve one or more fibre organizations (Assaf and Basser, 2005). We used the CHARMED model with three intra-neuronal compartments, signified as CHARMED\_in3 (Harms, Fritz, et al., 2017). We sampled this model using the default MCMC sampler in MDT using 10k samples with a burnin of 200 samples and using a maximum likelihood estimate as starting point, computed using an NVIDIA Tesla K40c graphics card. After sampling, MDT automatically computed a mean and standard deviation of the Fraction of Restricted (FR).

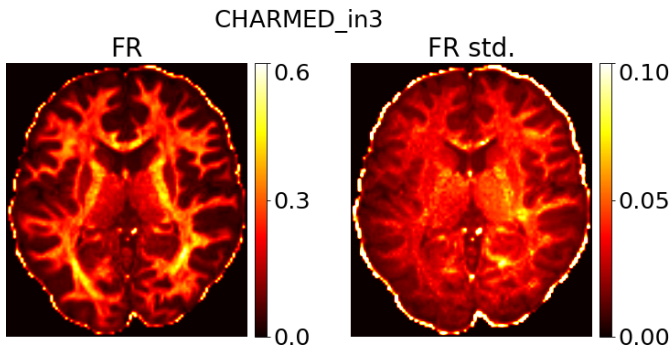


Figure 5.5: Three intra-neuronal compartment CHARMED Fraction Of Restricted (FR) mean and standard deviation maps computed using MCMC sampling on an HCP MGH dataset mgh\_1003 (Fan et al., 2016), This example used the Offset-Gaussian likelihood model and was computed on an NVIDIA Tesla K40c graphics card running on Microsoft Windows.

### 5.4.3 Bingham-NODDI

To illustrate the use of the batch fitting functionality in MDT, we computed Bingham-NODDI on one of HCP Wu-Minn datasets using the command:

```
mdt-batch-fit /wuminn_data_dir/ BinghamNODDI
```

This command suffices to compute the Bingham-NODDI (Tariq et al., 2016) on all 1200 subjects of the HCP Wu-Minn study (Van Essen et al., 2012). Each of these datasets is acquired at a resolution of 1.25mm isotropic with three shells of  $b=1000, 2000, 3000$  s/mm<sup>2</sup>, with 90 directions each and 18 b0 volumes. Figure 5.6 shows the hindered volume fraction ( $f_{hindered}$ ) of Bingham-NODDI model fitted on HCP Wu-Minn subject 100307. For these results we adapted the Bingham-NODDI to use an Gaussian likelihood model and then fitted it with the Powell routine using all ten cores of an Intel Xeon E5-2650v3 processor. After model fitting, MDT automatically computes a standard deviation estimate using the theory of the Fisher Information Matrix (see Chapter 4).

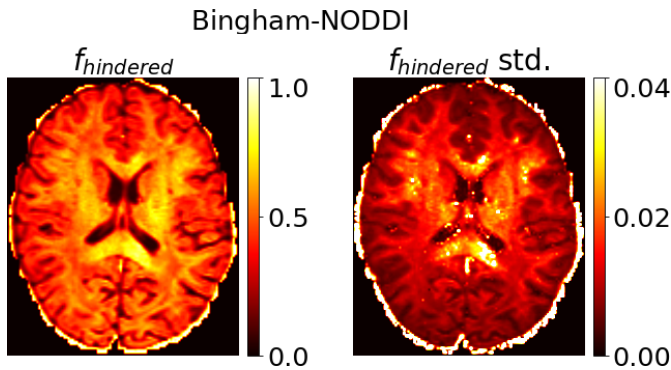


Figure 5.6: Bingham-NODDI hindered volume fraction ( $f_{hindered}$ ) and corresponding standard deviation computed using the Powell optimization routine and Fisher Information Matrix, using the Gaussian likelihood model. This example is computed using an Intel Xeon E5-2650v3 CPU running on Microsoft Windows with diffusion MRI data from subject 100307 of the Wu-Minn data project (Van Essen et al., 2012).

#### 5.4.4 Different data

Besides diffusion data, MDT can be used for other types of data and new models can be easily added. Pilot analysis have been performed for ex-vivo data, cortical diffusion analysis, mouse brains, monkey brains and spinal cord data, with varying MRI scanners ranging from 1.5T to 9.4T. As such we expect MDT to function with various kinds of data, scanned with various MRI scanners.

### 5.5 Discussion

We presented the design concepts and usage interfaces of the Microstructure Diffusion Toolbox (MDT), concluding with a few application examples. We showed how the modeling framework allows easy definition of microstructure models using multiple layers of reusable components. To use these models for data analysis purposes we illustrated the graphical user interface, the command line interface and the Python programming interface. As application examples we applied MDT to multiple datasets using multiple models using varying hardware configurations.

For GPU accelerated processing we presented a second toolbox called the Multi-threaded Optimization Toolbox (MOT). This package is used by MDT for all non-linear optimization and MCMC sampling of microstructure models. MOT is a pure numerical optimization package, agnostic of MRI microstructure modeling. By splitting the software stack into MDT and MOT we open the possibility of using the developed parallel routines for purposes other than that of MRI microstructure analysis. For example in (Kashyap et al., 2018) the authors use the MOT software package for combining the multiple channels of a multi-channel functional MRI acquisition. Potential other applications could be with genetical data or geological data.

A few other toolboxes with a similar purpose as MDT have been presented in the past (Cook et al., 2006; Fick, Wassermann, and Deriche, 2018; Hernandez-Fernandez et al., 2018). Although more verbose, the modeling framework in (Cook et al., 2006) is similar to that of MDT in that they also define composite models as a combination of compartments. In contrast to MDT, their software includes no GPU acceleration and is more rigid in the types of MRI data accepted as input. In (Fick, Wassermann, and Deriche, 2018), the authors propose an alternative modeling framework, but this toolbox does not include any parallelization strategies. The software presented in (Hernandez-Fernandez et al., 2018) includes GPU accelerated model fitting, but their modeling framework is not as advanced as that of (Cook et al., 2006; Fick, Wassermann, and Deriche, 2018) or that of MDT.

Their applicability is also limited to NVIDIA hardware and to the Linux operating system. In contrast, MDT works on Intel, AMD and NVIDIA hardware and on the Windows, Linux and Apple OSX operating systems.

Future extensions to the modeling framework can include multi-modality fitting. Diffusion MRI is relatively insensitive to water in the myelin compartment, due to its very low  $T_2$ , but relatively sensitive water in the intra- and extra-cellular compartments. Other modalities like susceptibility MRI, quantitative magnetic transfer imaging, and myelin water imaging are relatively sensitive to myelin and not as sensitive to intra and extra cellular compartments. These differences in contrast may prove beneficial in joint multi-modal analysis. Although MDT already offers susceptibility MRI and quantitative magnetic transfer models, it has no support yet for conjoint analysis. Future work can extend the current modeling framework to support fitting multiple models (with shared parameters) to multiple datasets at the same time.



# 6

## Summary and Discussion

## Summary and Discussion

The primary aim of this thesis was to improve the robustness and efficiency of the methods used in diffusion MRI microstructure modeling. A secondary aim was to provide the community with software to utilize the findings of this work. For the primary aim we studied maximum likelihood estimation in Chapter 2, MCMC sampling in Chapter 3 and uncertainty estimation in Chapter 4. For the secondary aim we created the Microstructure Diffusion Toolbox, discussed in Chapter 5. The first section of this chapter summarizes the scientific contributions of each of the individual chapters. The second section discusses this thesis and provides an outlook for future work in microstructure modeling.

### 6.1 Summary

#### Chapter 2: Robust and fast nonlinear optimization of diffusion MRI microstructure model

Most of the authors in microstructure modeling prefer the use of non-linear optimization routines for their microstructure modeling. Often mentioned optimization routines are the Levenberg-Marquardt (Jespersen et al., 2010; Assaf, Freidlin, et al., 2004; Assaf, Blumenfeld-Katzir, et al., 2008) and Nelder-Mead Simplex routine (Landman, Bazin, and Prince, 2007). The initialization strategy is often not specified, although some authors mention a grid search prior to optimization (H. Zhang, Schneider, et al., 2012). In Chapter 2 we studied these two aspects of non-linear optimization, the choice of optimization routine and the choice of initialization strategy.

We tested three optimization routines against each other on a multitude of models and datasets, the Levenberg-Marquardt, Nelder-Mead simplex and a newly proposed routine, the Powell method. Based on robustness, accuracy, precision and run time we recommend the Powell method as preferred optimization routine for dMRI modeling. We recommend using a single optimization strategy for all microstructure models in a study to minimize possible bias from the optimization strategy.

We recommend the use of cascades as an initialization strategy. In cascaded optimization, the parameters of a complex model are initialized with corresponding and already estimated parameters of a simpler model. Applied recursively, this can lead to a cascade of models to fit, with each new model slightly more complex than the previous model. As an extension to cascaded initialization, we proposed fixing the diffusion orientation estimates of a model using orientations from Ball&Stick estimates. Fixing the orientation parameters reduces the degrees

of freedom in a model, improving run time and fitting robustness of the remaining parameters.

### Chapter 3: Robust and fast Markov Chain Monte Carlo sampling of diffusion MRI microstructure models

Some authors prefer the use of MCMC sampling for their model fitting purposes (Behrens et al., 2003; Sotiropoulos et al., 2013; Wegmann, Eklund, and Villani, 2017). Whereas maximum likelihood estimation provides only a point estimate of the fitted model parameters, MCMC recovers the entire posterior distribution of the model parameters given the data. In addition to a point estimate, the posterior distribution provides parameter uncertainties and parameter correlations. The aim of chapter 3 was to provide a generally applicable, scientifically validated MCMC methodology for microstructure modeling. To this end, we studied several methodological aspects of MCMC including the choice of proposal distribution, burn-in, thinning and number of samples to draw.

There are many different MCMC strategies proposed in the literature, each defined uniquely by their proposal distribution. In Chapter 3 we focused on self adjusting univariate proposal distributions and recommend the Adaptive Metropolis Within Gibbs routine based on its theoretical soundness and its general robust performance. For burn-in and thinning we provide empirical and theoretical arguments against their use and recommend to use them sparingly, if at all. To determine the number of MCMC samples to draw, we propose the effective sample size theory to predict the number of samples required for the estimate of the posterior to reach a predetermined precision.

### Chapter 4: Fast quantification of uncertainty in non-linear diffusion MRI models

Parameter uncertainties are typically ignored in diffusion MRI studies, although recognized for their benefits in functional MRI (Chen et al., 2012; Woolrich et al., 2004). In Chapter 4 we discuss and compare two methods for computing parameter uncertainties, the Fisher Information Matrix (FIM) and MCMC sampling. We then show how these uncertainty estimates could be used for artifact detection and increasing the power in group studies.

We first compared the FIM against gold standard MCMC using multiple datasets and models. We show that the parameter uncertainties obtained using the FIM show close correspondence to that of MCMC, yet computed in only a fraction of the time. We then proposed two use cases for the computed parameter uncertainties, artifact detection and variance reduction in group statistics. Results



show that parameter uncertainties are often high in regions with white matter acquisition artifacts, allowing the uncertainties to be used for single subject artifact detection. For group studies we recommend the use of variance weighted averaging. Doing so can reduce the overall variance in group statistics and reduce the effect of data artifacts without discarding data from the analysis. Since the power of a study depends partially on the variance in the estimates, lowering the variance in group statistics increases the power of a study.

## Chapter 5: MDT, the Microstructure Diffusion Toolbox

To make the findings in this thesis reproducible and easily applicable to other studies, we additionally present our research findings in the form of a software package. This software package, the Microstructure Diffusion Toolbox (MDT) is dedicated to diffusion MRI microstructure modeling. Since the models and algorithms in MDT have been discussed in the previous chapters, Chapter 5 focuses on the design concepts, user interfaces and application examples of MDT.

This software package contains all the optimization and MCMC sampling routines mentioned in this thesis. By default, it uses the Powell optimization routine and the Adaptive Metropolis Within Gibbs sampling routine for all microstructure fitting and MCMC sampling purposes. In addition, MDT automatically computes parameter uncertainties using the Fisher Information Matrix after every microstructure model optimization.

Unique to MDT is the combination of high level model definitions with high performance parallel processing. For maximum performance, we re-implemented all models and methods for execution on graphical processing units (GPU's). Using GPU's, MDT can process thousands of voxels simultaneously, making it one of the fastest microstructure modeling tools currently available. To hide away most of the GPU programming complexity, we created a microstructure modeling framework. Users can interact with this modeling framework using the high level Python programming language, while MDT creates the necessary GPU code in the background.

## 6.2 Discussion

### 6.2.1 Limitations

A large part of this thesis was dedicated to evaluating different optimization and MCMC sampling routines for their use in microstructure modeling. Although time did not permit us to evaluate all available routines, it can still be considered a limitation of this thesis.

To make this research feasible, we re-implemented several optimization and sampling routines in a language suitable for execution on GPU's. Without the speed-up of GPU's, some of the experiments in this thesis would have computed for over sixty weeks instead of one week. While this speed-up is a virtue, the need to re-implement all the optimization and sampling routines put a constraint on the choice of algorithms. For example, the optimization routine we currently recommend for microstructure modeling is the Powell conjugate gradient method (Powell, 1964). This is a relatively straightforward optimization routine with a small codebase and low memory requirements, making it ideal for GPU programming. Nevertheless, this routine is from 1964 and is superseded by later algorithms such as NEWUOA (Powell, 2006) and LINCOA (Powell, 2015) from the same author. These routines are also vastly more complex than the conjugate gradient method so we decided to focus on the simpler algorithms first.

The choice in optimization routines was also limited by a methodological choice. That is, we only compared local optimization routines and deliberately ignored global optimization routines like simulated annealing (Bohachevsky, M. E. Johnson, and Stein, 2008), self-organizing migrating algorithm (Zelinka, 2004) and particle swarm optimization (Eberhart and Kennedy, 2002). There are two major justifications for this choice. First, global optimization routines are rarely used and most microstructure modeling authors prefer the use of local optimization routines (Assaf, Freidlin, et al., 2004; Assaf, Blumenfeld-Katzir, et al., 2008; Jespersen et al., 2010; Landman, Bazin, and Prince, 2007). Second, if the function is well behaved or if a local optimum is good enough, these global optimization routines may not be as fast as local optimization routines in finding the local optimum.

The MCMC comparisons were limited by similar constraints and choices as with the optimization routines. That is, we re-implemented several MCMC samplers to benefit from the speed-up provided by GPU's, and this constraint our choice in sampling routines. As a methodological choice we decided to only compare univariate updating MCMC samplers, thereby ignoring multivariate updating MCMC samplers like the t-walk sampler (Christen and Foxy, 2010) and the No-U-Turn sampler (Hoffman and Gelman, 2011). This choice for univariate updating MCMC samplers is partly justified by previous work, as most microstructure modelers use univariate MCMC samplers (Behrens et al., 2003; Sotiropoulos et al., 2013; Wegmann, Eklund, and Villani, 2017).

### 6.2.2 Future research

The tools developed in this thesis can potentially be extended to different branches of MRI research. Here we hope to give an overview of ideas for future research which build upon the work of this thesis.

Future extensions to MDT could include multi-modality fitting, i.e. fitting multiple models (with shared parameters) to multiple datasets at the same time. Due to its very low T2, diffusion MRI is relatively insensitive to the myelin compartment but relatively sensitive to the size of intra- and extra-cellular compartments. Other modalities like susceptibility MRI, quantitative magnetic transfer imaging, and myelin water imaging are relatively sensitive to myelin and not as sensitive to the compartment sizes. These differences in contrast may prove beneficial in joined analysis. MDT already offers multiple susceptibility MRI and quantitative magnetic transfer models that can be combined with diffusion MRI models. For example, MDT already includes the time-dependent ActiveAx method (De Santis, D. K. Jones, and Roebroek, 2016), a multi-modality relaxometry and diffusion model. At the moment defining these joined models is relatively involved because MDT was designed around single dataset, single model fitting. Future work could extend the current modeling framework to support simultaneous fitting of multiple models to multiple datasets, providing a cleaner programming semantic.

Another possible line of research is including Automatic Differentiation to MDT. Automatic Differentiation (AD) can compute the derivative of a computer program by making repetitive use of the chain rule (Wengert, 1964; Griewank, 1989). This technique has the performance and exactness of analytical derivatives without needing to program the derivative explicitly. We currently use numerical derivatives to compute the first derivatives in the Levenberg-Marquardt routine and the second derivatives for the Fisher Information Matrix. The use of AD would make these derivatives more accurate and faster to compute.

The modeling framework in MDT could also be used for experimental design optimization, similar to (Alexander, 2008). In a typical dMRI study, the experimental design is based on expert knowledge. That is, given the desired microstructure model, a dMRI expert will propose acquisition settings such that the acquired data has sufficient information to fit the desired microstructure model. The key idea in experimental design optimization is to take a more systematic approach to the choice of acquisition settings. Instead of an expert, experimental design optimization systematically optimizes the acquisition settings such that the chosen model can be fitted more precisely to the acquired data. With a large collection of models and the possibility of computing the Cramér Rao Lower Bound, MDT is a highly suitable toolbox for implementing experimental design optimization.

At the moment MDT is very focused on processing microstructure models which use signal equations. An alternative class of models that could be added to MDT are simulation based models like Extended Phase Graphs (Weigel, 2015). Instead of a model equation, this class of models simulate the state and relaxation of water molecules with the tissue specifications as free parameters. Since this kind of models simulate a physical process they are often slower to compute than signal

equation models. The use of GPU's could prove beneficial to this class of models as well.

Another model fitting paradigm suitable for inclusion in MDT would be MR fingerprinting (Ma et al., 2013). This technique first prepares a dictionary of possible signal evolutions. A matching or pattern recognition algorithm is then used to select a signal vector or a weighted set of signal vectors from the dictionary that best correspond to the observed signal evolution (Ma et al., 2013). Both the dictionary creation step as well as the pattern recognition step can potentially be accelerated using GPU programming.

There are other optimization approaches for microstructure modeling which avoid the use of non-linear optimization altogether. These approaches typically linearize the optimization problem (Daducci et al., 2015; Canales-Rodríguez et al., 2015; Novikov, Jelescu, and Fieremans, 2015). For instance, AMICO (Daducci et al., 2015; Canales-Rodríguez et al., 2015) constructs a convex formulation with a single global minimum by choosing a finite dictionary of atoms. This technique has already been successfully applied to the NODDI, ActiveAx (Daducci et al., 2015) and VERDICT (Bonet-Carne, Johnston, et al., 2019) models. It might be possible to extend the modeling framework in MDT to include convex formulations of each of the compartments. This could effectively lead to a generalized AMICO implementation with GPU accelerated model fitting, built on top of MDT.

In the current MCMC sampling approach we initialize the MCMC sampler using a point estimate derived from a prior maximum log likelihood estimate. This initialization could be extended to also initialize the parameter priors and parameter proposal distributions based on parameter uncertainty estimates. Or, from a more Bayesian viewpoint, the priors of a model could be set to posterior estimates of a simpler model, similar to our cascaded initialization technique from Chapter 2. This could increase efficiency of the MCMC sampler by restricting the search to known high probability areas.

The results of this thesis could also be interesting to tractometry (Bells et al., 2011) frameworks. Tractometry is a method which maps voxel-wise microstructure measurements on reconstructed white matter pathways. In effect, this provides a tract based assessment of the underlying microstructure. In Chapter 2 we discussed "cascade fixed" as a method to fix the axonal orientations of a complex model based on orientation estimates from a simpler model. We hinted that this technique could prove essential in correctly mapping microstructural measurements of a complex model on white matter pathways reconstructed using another, simpler model. The uncertainty measurements from Chapter 4 could additionally be mapped to the white matter pathways, to provide tract-wise microstructure measurement uncertainty.

### 6.2.3 Conclusion

Diffusion MRI promises a structural understanding of the human brain with numerous applications in both scientific and clinical disciplines. However, the interpretation of dMRI images is not a straightforward process and requires modeling of the dMRI signal. While the primary focus is often on the choice of models, small implementation differences and choice of algorithms can make modeling results irreproducible and incomparable between software packages. A single shared modeling framework, like the one proposed in this thesis, can merge the knowledge from both the model developers and the computer scientists, providing a solid foundation for future dMRI applications.





# A

## Appendix to Chapter 2

### A.1 Parameter transformations

To enforce boundary conditions on the parameters and to scale the parameters to a range better suited for optimization, we use encoding and decoding transformations on the model parameters. Table A.1 shows the model encoding and decoding transformations used in this work. For models that share the same parameter and that have a different transformation we have listed the specific transformation per parameter. The cosine and sin transformations are used to restrain the parameter within a given set of lower and upper bounds. The sin encode transformation  $\bar{f}_{\sin}(x)$  is generated by:

$$\bar{f}_{\sin}[\mathbf{lb}, \mathbf{ub}](x) = \arcsin \sqrt{\left| \frac{x - \mathbf{lb}}{\mathbf{ub} - \mathbf{lb}} \right|} \quad (\text{A.1})$$

and the sin decoding transformation is given by  $f_{\sin}$ :

$$f_{\sin}[\mathbf{lb}, \mathbf{ub}](\bar{x}) = \sin^2(\bar{x}) \cdot (\mathbf{ub} - \mathbf{lb}) + \mathbf{lb} \quad (\text{A.2})$$

with a similar transformation for  $\bar{f}_{\cos}$  and  $f_{\cos}$ .



Parameter	Model (& compartment)	Encoding transformation	Decoding transformation
$d_{\parallel}$	Tensor	$\bar{f}_{\sin}[0, 10^{-10}]$	$f_{\sin}[0, 10^{-10}]$
$d_{\perp_1}$	Tensor	$\bar{f}_{\sin}[0, 10^{-10}]$	$f_{\sin}[0, 10^{-10}]$
$d_{\perp_2}$	Tensor	$\bar{f}_{\sin}[0, 10^{-10}]$	$f_{\sin}[0, 10^{-10}]$
$\psi$	-	$\bar{\psi} =  \psi  \bmod \pi$	$\psi =  \bar{\psi}  \bmod \pi$
$\theta$	-	$\bar{\theta} =  \theta  \bmod \pi$	$\theta =  \bar{\theta}  \bmod \pi$
$\phi$	-	$\bar{\phi} =  \phi  \bmod \pi$	$\phi =  \bar{\phi}  \bmod \pi$
$\kappa$	-	$\bar{f}_{\sin}[10^{-5}, 2\pi]$	$f_{\sin}[10^{-5}, 2\pi]$
$d_{\parallel}$	CHARMED	$\bar{f}_{\sin}[10^{-10}, 5 \cdot 10^{-9}]$	$f_{\sin}[10^{-10}, 5 \cdot 10^{-9}]$
$d_{\perp_1}$	CHARMED	$\bar{f}_{\sin}[10^{-10}, 3 \cdot 10^{-9}]$	$f_{\sin}[10^{-10}, 3 \cdot 10^{-9}]$
$d_{\perp_2}$	CHARMED	$\bar{f}_{\sin}[10^{-10}, 3 \cdot 10^{-9}]$	$f_{\sin}[10^{-10}, 3 \cdot 10^{-9}]$
$d$	CHARMED	$\bar{f}_{\sin}[10^{-10}, 3 \cdot 10^{-9}]$	$f_{\sin}[10^{-10}, 3 \cdot 10^{-9}]$

Table A.1: Parameter transformations, the units of the diffusivities is in  $m^2/s$ , other units are dimensionless.

## A.2 Cascaded model initializations

While in an Cascade S0 (CS) we only initialize S0 to the unweighted signal level, in a CI and CF cascade we initialize and/or fix more parameters. Table A.2 and Table A.3 show the model initializations for the more intricate CI and CF cascades. The target model (first column) is the final model we wish to optimize, the source model (second column) is the model we use as a source of initialization. In the CI table, the last column shows which parameters are initialized from source model to target model. In the CF table, the last column shows which parameters are fixed instead of initialized. Please note that the cascades are defined recursively. For example, if we optimize CHARMED\_in[n] using CI, we use Ball&Sticks\_in[n] as a source model. Ball&Sticks\_in[n] in turn uses Ball&Sticks\_in[n-1] as a source of initialization, and so forth until the source model is S0. The initialization and fixes are indicated with an assignment operation. For example,  $\text{Tensor}.\theta = \text{Stick0}.\theta$  indicates that the  $\theta$  parameter of the Tensor compartment in the target model is initialized with the optimized  $\theta$  parameter maps of the Stick0 model. In all the tables we use the convention that initializations and fixes with similar named models are abbreviated to a single statement. For example, initializing/fixing  $\text{Stick0}.\theta$  of the previous model to the  $\text{Stick0}.\theta$  of the next model is abbreviated as just  $\text{Stick0}.\theta$ .

Target model	Source model	Cascade parameters inits
Ball&Sticks_in1	S0	S0.s0
Ball&Sticks_in[n]	Ball&Sticks_in[n-1]	S0.s0, $\text{Stick}[n].\{\theta, \phi\}$ , $w_{\text{stick}\{0,1,2,\dots,n-1\}}$
CHARMED_in[n]	Ball&Sticks_in[n]	S0.s0, $\text{CHARMED}[n].\{\theta, \phi\} = \text{Stick}[n].\{\theta, \phi\}$ , $\text{Tensor}.\{\theta, \phi\} = \text{Stick0}.\{\theta, \phi\}$ , $w_{\text{res}\{0,1,2,\dots,n\}} = w_{\text{stick}\{0,1,2,\dots,n\}}$
Tensor	Ball&Sticks_in1	S0.s0, $\text{Tensor}.\{\theta, \phi\} = \text{Stick0}.\{\theta, \phi\}$
NODDI	Ball&Sticks_in1	S0.s0, $\text{NODDI.in}.\{\theta, \phi\} = \text{Stick0}.\{\theta, \phi\}$ , $w_{\text{csf}} = w_{\text{ball}}$ , $w_{\text{in}} = w_{\text{stick0}}/2$ , $w_{\text{ex}} = w_{\text{stick0}}/2$

Table A.2: The initializations in a Cascade Initialize (CI) cascade.

Target model	Source model	Overriding cascade fixes
CHARMED_in[n]	Ball&Sticks_in[n]	$\text{CHARMED}[n].\{\theta, \phi\} = \text{Stick}[n].\{\theta, \phi\}$
NODDI	Ball&Sticks_in1	$\text{NODDI.in}.\{\theta, \phi\} = \text{Stick0}.\{\theta, \phi\}$

Table A.3: The Cascade Fixed (CF) fixed parameters.



# B

## Appendix to Chapter 4

### B.1 Numerical Hessian

To compute the Hessian we use a numerical differentiation routine with multiple step sizes and extrapolations to provide an estimate with a  $\mathcal{O}(h^6)$  order of accuracy. For a single step size vector  $\mathbf{d}$ , we compute each element of the Hessian using a second order Taylor expansion central difference,

$$\mathbf{H}_{ij}(\mathbf{x}) := \frac{1}{4\mathbf{d}_i\mathbf{d}_j} [ \begin{aligned} &l(\mathbf{x} + \mathbf{e}_i\mathbf{d}_i + \mathbf{e}_j\mathbf{d}_j) \\ &-l(\mathbf{x} + \mathbf{e}_i\mathbf{d}_i - \mathbf{e}_j\mathbf{d}_j) \\ &-l(\mathbf{x} - \mathbf{e}_i\mathbf{d}_i + \mathbf{e}_j\mathbf{d}_j) \\ &+l(\mathbf{x} - \mathbf{e}_i\mathbf{d}_i - \mathbf{e}_j\mathbf{d}_j) \end{aligned} ] \quad (\text{B.1})$$

where  $\mathbf{x} \in \mathcal{R}^n$  is the parameter vector,  $l(\mathbf{x})$  is the log-likelihood function and  $\mathbf{e}_k$  is a zeros vector with only element  $k$  set to one. We evaluate the Hessian multiple times with exponentially diminishing steps and with the largest step size chosen such that  $\mathbf{x} \pm \mathbf{d}$  is within bounds and  $\mathbf{d}$  is within predefined upper and lower limits. In this work we evaluate the Hessian for five different step sizes  $\mathbf{d}$  with each step half the previous step. We then apply Richardson extrapolation (Burg and Erwin, 2009) twice to produce three estimates with a sixth order of accuracy. These three approximations we extrapolate again using Wynn's epsilon algorithm (Weniger, 1991) to arrive at a single final estimate.

## B.2 Uncertainty propagation

This appendix provides two illustrations of uncertainty propagation, one example using Ball&Stick Fraction of Stick and one example using Tensor Fractional Anisotropy.

Uncertainty propagation of the Ball&Stick Fraction of Stick can be defined as follows. For a two Stick Ball&Stick model, the Fraction of Stick is defined as:

$$\text{FS} = w_0 + w_1 \quad (\text{B.2})$$

The analytical gradient of this function is given by:

$$\nabla_{\text{FS}} = (w_0, w_1) \quad (\text{B.3})$$

The covariance matrix of the weights can be defined as:

$$\Sigma_{\mathbf{w}} = \begin{pmatrix} \sigma_{w_0}^2 & \sigma_{w_0 w_1} \\ \sigma_{w_1 w_0} & \sigma_{w_1}^2 \end{pmatrix} \quad (\text{B.4})$$

with  $\sigma_{w_i}^2$  denoting the variance of weight  $w_i$ , and  $\sigma_{w_i w_j}$  denoting the covariances of weights  $w_i$  and  $w_j$ . When evaluated, these quantities are taken from the covariance matrix provided by the FIM.

Using equation 4.6, we can write the uncertainty propagation as:

$$\sigma_{\text{FS}}^2 = \nabla_{\text{FS}} \Sigma_{\mathbf{w}} \nabla_{\text{FS}}^{\top} \quad (\text{B.5})$$

which simplifies to:

$$\sigma_{\text{FS}}^2 = w_0^2 \sigma_{w_0}^2 + w_1^2 \sigma_{w_1}^2 + 2w_0 w_1 \sigma_{w_0 w_1} \quad (\text{B.6})$$

By evaluating expression B.6 using the point estimates, variance estimates and covariance estimates of the weights, we can compute the variance in the FS metric.

Uncertainty propagation of Tensor FA is slightly more complex considering FA is not a linear function of its inputs. The Tensor FA can be defined in terms of the three Tensor diffusivities (the eigenvalues of the diffusion Tensor) as:

$$\text{FA} = \sqrt{\frac{1}{2} \frac{\sqrt{(d_0 - d_1)^2 + (d_1 - d_2)^2 + (d_0 - d_2)^2}}{\sqrt{d_0^2 + d_1^2 + d_2^2}}} \quad (\text{B.7})$$

The derivative of FA with respect to the first diffusivity can be written as:

$$\frac{\partial \text{FA}}{\partial d_0} = \frac{2d_0d_1d_2 + d_0^2(d_1 + d_2) - d_1^2d_2 - d_1d_2^2 - d_1^3 - d_2^3}{2^{3/2}\sqrt{d_0^2 + d_1^2 + d_2^2}\sqrt{d_0^2 - d_0(d_1 + d_2) + d_1^2 - d_1d_2 + d_2^2}} \quad (\text{B.8})$$

and similar derivatives can be derived for the second and third diffusivity by suitable permutations of the diffusivity indices. The analytical gradient of FA,  $\nabla_{\text{FA}}$  can now be defined as:

$$\nabla_{\text{FA}} = \left( \frac{\partial \text{FA}}{\partial d_0}, \frac{\partial \text{FA}}{\partial d_1}, \frac{\partial \text{FA}}{\partial d_2} \right) \quad (\text{B.9})$$

The covariance matrix of the diffusivities can be defined as:

$$\Sigma_{\text{d}} = \begin{pmatrix} \sigma_{d_0}^2 & \sigma_{d_0d_1} & \sigma_{d_0d_2} \\ \sigma_{d_1d_0} & \sigma_{d_1}^2 & \sigma_{d_1d_2} \\ \sigma_{d_2d_0} & \sigma_{d_2d_1} & \sigma_{d_2}^2 \end{pmatrix} \quad (\text{B.10})$$

with  $\sigma_{d_i}^2$  denoting the variance of diffusivity  $d_i$ , and  $\sigma_{d_id_j}$  denoting the covariances of diffusivities  $d_i$  and  $d_j$ .

Using equation 4.6, we can define the uncertainty propagation of FA as:

$$\sigma_{\text{FA}}^2 = \nabla_{\text{FA}} \Sigma_{\text{d}} \nabla_{\text{FA}}^{\top} \quad (\text{B.11})$$

By evaluating expression B.11 using the point estimates of the diffusivities together with the corresponding variance and covariance estimates from the FIM, we can compute the propagated variance in the FA metric.



# Bibliography

- Aboitiz, F. et al. (Dec. 1992). "Fiber composition of the human corpus callosum." In: *Brain research* 598.1-2, pp. 143–53.
- Alexander, D. C. (Aug. 2008). "A general framework for experiment design in diffusion MRI and its application in measuring direct tissue-microstructure features." In: *Magnetic resonance in medicine* 60.2, pp. 439–48. DOI: 10.1002/mrm.21646.
- (2009). "Modelling, Fitting and Sampling in Diffusion MRI". In: *Visualization and Processing of Tensor Fields. Mathematics and Visualization*. Ed. by L. D. and W. J. Springer, Berlin, Heidelberg, pp. 3–20. DOI: 10.1007/978-3-540-88378-4\_1.
- Alexander, D. C., T. B. Dyrby, et al. (2017). "Imaging brain microstructure with diffusion MRI: Practicality and applications". In: *NMR in Biomedicine* September, pp. 1–26. DOI: 10.1002/nbm.3841.
- Alexander, D. C., P. L. Hubbard, et al. (2010). "Orientationally invariant indices of axon diameter and density from diffusion MRI". In: *NeuroImage* 52.4, pp. 1374–1389. DOI: 10.1016/j.neuroimage.2010.05.043.
- Andersson, J., M. Jenkinson, and S. Smith (2010). *Non-linear registration, aka spatial normalisation. FMRIB technical report TR07JA2*. Tech. rep. Oxford: FMRIB.
- Andersson, J. L. and S. N. Sotiropoulos (2016). "An integrated approach to correction for off-resonance effects and subject movement in diffusion MR imaging". In: *NeuroImage* 125, pp. 1063–1078. DOI: 10.1016/j.neuroimage.2015.10.019.
- Arras, K. O. (1998). *An Introduction To Error Propagation: Derivation, Meaning and Examples of Equation  $CY = FXCXFT$* . Tech. rep. Swiss Federal Institute of Technology Lausanne. DOI: 10.3929/ethz-a-010113668.
- Assaf, Y., D. C. Alexander, et al. (2013). "The CONNECT project: Combining macro- and micro-structure". In: *NeuroImage* 80, pp. 273–282. DOI: 10.1016/j.neuroimage.2013.05.055.
- Assaf, Y. and P. J. Basser (2005). "Composite hindered and restricted model of diffusion (CHARMED) MR imaging of the human brain". In: *NeuroImage* 27.1, pp. 48–58. DOI: 10.1016/j.neuroimage.2005.03.042.
- Assaf, Y., T. Blumenfeld-Katzir, et al. (2008). "AxCaliber: A method for measuring axon diameter distribution from diffusion MRI". In: *Magnetic Resonance in Medicine* 59.6, pp. 1347–1354. DOI: 10.1002/mrm.21577.
- Assaf, Y., R. Z. Freidlin, et al. (2004). "New modeling and experimental framework to characterize hindered and restricted water diffusion in brain white matter". In: *Magnetic Resonance in Medicine* 52.5, pp. 965–978. DOI: 10.1002/mrm.20274.



- Assaf, Y. and O. Pasternak (2008). "Diffusion tensor imaging (DTI)-based white matter mapping in brain research: A review". In: *Journal of Molecular Neuroscience* 34.1, pp. 51–61. DOI: 10.1007/s12031-007-0029-0.
- Basser, P. J. and C. Pierpaoli (June 1996). "Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI." In: *Journal of magnetic resonance. Series B* 111.3, pp. 209–19.
- Basser, P. J., J. Mattiello, and D. LeBihan (1994). "MR diffusion tensor spectroscopy and imaging." In: *Biophysical journal* 66.1, pp. 259–67. DOI: 10.1016/S0006-3495(94)80775-1.
- Bastiani, M. et al. (2019). "Automated quality control for within and between studies diffusion MRI data using a non-parametric framework for movement and distortion correction". In: *NeuroImage* 184. September 2018, pp. 801–812. DOI: 10.1016/j.neuroimage.2018.09.073.
- Beaulieu, C. (2002). *The basis of anisotropic water diffusion in the nervous system - A technical review*. DOI: 10.1002/nbm.782.
- Behrens, T. E. J. et al. (2003). "Characterization and Propagation of Uncertainty in Diffusion-Weighted MR Imaging". In: *Magnetic Resonance in Medicine* 50.5, pp. 1077–1088. DOI: 10.1002/mrm.10609.
- Bells, S. et al. (2011). "Tractometry: comprehensive multi-modal quantitative assessment of white matter along specific tracts." In: *Proc. Int. Soc. Magn. Res. Med., Montreal*. Montreal.
- Benitez, A. et al. (2014). "White matter tract integrity metrics reflect the vulnerability of late-myelinating tracts in Alzheimer's disease". In: *NeuroImage: Clinical* 4, pp. 64–71. DOI: 10.1016/j.nicl.2013.11.001.
- Bennett, K. M. et al. (Oct. 2003). "Characterization of continuously distributed cortical water diffusion rates with a stretched-exponential model". In: *Magnetic Resonance in Medicine* 50.4, pp. 727–734. DOI: 10.1002/mrm.10581.
- Bergers, E. et al. (Dec. 2002). "Axonal damage in the spinal cord of MS patients occurs largely independent of T2 MRI lesions." In: *Neurology* 59.11, pp. 1766–71.
- Bohachevsky, I. O., M. E. Johnson, and M. L. Stein (2008). "Generalized Generalized Simulated Simulated Annealing for Function Function Optimization Optimization". In: 28.3, pp. 209–217.
- Bonet-Carne, E., A. Daducci, et al. (2016). "Non-invasive quantification of prostate cancer using AMICO framework for VERDICT MRI". In: *Annual Meeting of the International Society of Magnetic Resonance in Medicine (ISMRM), Singapore, 7-13 May*, pp. 3465–3465.
- Bonet-Carne, E., E. Johnston, et al. (2019). "VERDICT-AMICO: Ultrafast fitting algorithm for non-invasive prostate microstructure characterization". In: *NMR in Biomedicine* 32.1, pp. 1–15. DOI: 10.1002/nbm.4019.
- Brent, R. P. (1973). *Algorithms for function minimization without derivatives*. Dover Publications.

- 
- Burg, C. and T. Erwin (2009). "Application of Richardson extrapolation to the numerical solution of partial differential equations". In: *Numerical Methods for Partial Differential ...* 72035. June. DOI: 10.1002/num.
- Callaghan, P. T. (1993). *Principles of Nuclear Magnetic Resonance Microscopy*. Oxford science publications. Clarendon Press.
- Canales-Rodríguez, E. J. et al. (2015). "Spherical deconvolution of multichannel diffusion MRI data with non-Gaussian noise models and spatial regularization". In: *PLoS ONE* 10.10. DOI: 10.1371/journal.pone.0138910.
- Chen, G. et al. (2012). "NeuroImage FMRI group analysis combining effect estimates and their variances". In: *NeuroImage* 60.1, pp. 747–765. DOI: 10.1016/j.neuroimage.2011.12.060.
- Chib, S. and E. Greenberg (1995). "Understanding the Metropolis-Hastings Algorithm". In: *The American Statistician* 49.4, pp. 327–335.
- Christen, J. A. and C. Foxy (2010). "A general purpose sampling algorithm for continuous distributions (the t-walk)". In: *Bayesian Analysis* 5.2, pp. 263–282. DOI: 10.1214/10-BA603.
- Christensen, R. et al. (2010). *Bayesian Ideas and Data Analysis: An Introduction for Scientists and Statisticians*. 3th. Boca Raton, Florida: CRC Press, p. 516.
- Chung, S. W., Y. Lu, and R. G. Henry (2006). "Comparison of bootstrap approaches for estimation of uncertainties of DTI parameters". In: *NeuroImage* 33.2, pp. 531–541. DOI: 10.1016/j.neuroimage.2006.07.001.
- Cochran, W. G. (1937). "Problems Arising in the Analysis of a Series of Similar Experiments". In: *Supplement to the Journal of the Royal Statistical Society* 4.1, p. 102. DOI: 10.2307/2984123.
- Cook, P. a. et al. (2006). "Camino: Open-Source Diffusion-MRI Reconstruction and Processing". In: *14th Scientific Meeting of the International Society for Magnetic Resonance in Medicine*. Vol. 14, p. 2759.
- Cramer, H. (1946). *Mathematical methods of statistics*. English. Princeton: Princeton University Press.
- Daducci, A. et al. (2015). "Accelerated Microstructure Imaging via Convex Optimization (AMICO) from diffusion MRI data". In: *NeuroImage* 105, pp. 32–44. DOI: 10.1016/j.neuroimage.2014.10.026.
- De Santis, S., Y. Assaf, et al. (2014). "Improved precision in CHARMED assessment of white matter through sampling scheme optimization and model parsimony testing". In: *Magnetic Resonance in Medicine* 71.2, pp. 661–671. DOI: 10.1002/mrm.24717.
- De Santis, S., M. Drakesmith, et al. (2014). "Why diffusion tensor MRI does well only some of the time: Variance and covariance of white matter tissue microstructure attributes in the living human brain". In: *NeuroImage* 89, pp. 35–44. DOI: 10.1016/j.neuroimage.2013.12.003.
- De Santis, S., D. K. Jones, and A. Roebroeck (2016). "Including diffusion time dependence in the extra-axonal space improves in vivo estimates of axonal

- diameter and density in human white matter". In: *NeuroImage* 130, pp. 91–103. DOI: 10.1016/j.neuroimage.2016.01.047.
- Dietrich, O., J. G. Raya, S. B. Reeder, M. Ingrisch, et al. (2008). "Influence of multichannel combination, parallel imaging and other reconstruction techniques on MRI noise characteristics". In: *Magnetic Resonance Imaging* 26.6, pp. 754–762. DOI: 10.1016/j.mri.2008.02.001.
- Dietrich, O., J. G. Raya, S. B. Reeder, M. F. Reiser, et al. (2007). "Measurement of signal-to-noise ratios in MR images: Influence of multichannel coils, parallel imaging, and reconstruction filters". In: *Journal of Magnetic Resonance Imaging* 26.2, pp. 375–385. DOI: 10.1002/jmri.20969.
- Eberhart, R. and J. Kennedy (2002). "A new optimizer using particle swarm theory". In: pp. 39–43. DOI: 10.1109/mhs.1995.494215.
- Fan, Q. et al. (2016). "MGH-USC Human Connectome Project datasets with ultrahigh b-value diffusion MRI". In: *NeuroImage* 124, pp. 1108–1114. DOI: 10.1016/j.neuroimage.2015.08.075.
- Farooq, H. et al. (2016). "Microstructure Imaging of Crossing (MIX) White Matter Fibers from diffusion MRI". In: *Scientific Reports* 6.September, pp. 1–9. DOI: 10.1038/srep38927.
- Ferizi, U., T. Schneider, E. Panagiotaki, et al. (2014). "A ranking of diffusion MRI compartment models with in vivo human brain data". In: *Magnetic Resonance in Medicine* 72.6, pp. 1785–1792. DOI: 10.1002/mrm.25080.
- Ferizi, U., T. Schneider, M. Tariq, et al. (2013). "The importance of being dispersed: A ranking of diffusion MRI models for fibre dispersion using in vivo human brain data." In: *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention* 16.Pt 1, pp. 74–81.
- Fick, R., D. Wassermann, and R. Deriche (2018). "Mipy: An Open-Source Framework to improve reproducibility in Brain Microstructure Imaging". In: *Annual Meeting of the Organization for Human Brain Mapping*.
- Fields, R. D. (July 2008). "White matter in learning, cognition and psychiatric disorders". In: *Trends in Neurosciences* 31.7, pp. 361–370. DOI: 10.1016/j.tins.2008.04.001.
- Fieremans, E. et al. (2013). "Novel white matter tract integrity metrics sensitive to Alzheimer disease progression". In: *American Journal of Neuroradiology* 34.11, pp. 2105–2112. DOI: 10.3174/ajnr.A3553.
- Filley, C. M. (2005). "White matter and behavioral neurology". In: *Annals of the New York Academy of Sciences* 1064, pp. 162–183. DOI: 10.1196/annals.1340.028.
- Gao, F. and L. Han (2012). "Implementing the Nelder-Mead simplex algorithm with adaptive parameters". In: *Computational Optimization and Applications* 51.1, pp. 259–277. DOI: 10.1007/s10589-010-9329-3.

- 
- Garyfallidis, E. et al. (2014). "Dipy, a library for the analysis of diffusion MRI data". In: *Frontiers in Neuroinformatics* 8, February, pp. 1–17. DOI: 10.3389/fninf.2014.00008.
- Gelderen, P. van et al. (1994). "Water diffusion and acute stroke". In: *Magnetic Resonance in Medicine* 31.2, pp. 154–163. DOI: 10.1002/mrm.1910310209.
- Gelman, A., G. Roberts, and W. Gilks (1996). "Efficient Metropolis Jumping Rules". In: *Bayesian statistics 5*, pp. 599–608.
- Gelman, A., J. B. Carlin, et al. (2013). *Bayesian Data Analysis*. CRC Press.
- Geman, S. and D. Geman (1984). "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6.6*, pp. 721–741. DOI: 10.1109/TPAMI.1984.4767596.
- Geyer, C. J. (1991). "Markov Chain Monte Carlo Maximum Likelihood". In: *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*. 1. Minneapolis: Interface Foundation of North America, pp. 156–163.
- Gong, L. and J. M. Flegal (2016). "A Practical Sequential Stopping Rule for High-Dimensional Markov Chain Monte Carlo". In: *Journal of Computational and Graphical Statistics* 25.3, pp. 684–700. DOI: 10.1080/10618600.2015.1044092.
- Gorgolewski, K. et al. (2011). "Nipype: A Flexible, Lightweight and Extensible Neuroimaging Data Processing Framework in Python". In: *Frontiers in Neuroinformatics* 5, August. DOI: 10.3389/fninf.2011.00013.
- Griewank, A. (1989). *On Automatic Differentiation*. Tech. rep., pp. 83–108.
- Gu, X. et al. (2017). "Bayesian Diffusion Tensor Estimation with Spatial Priors". In: *International Conference on Computer Analysis of Images and Patterns*. Vol. 47. 1-2, pp. 372–383. DOI: 10.1007/978-3-319-64689-3\_30.
- Gudbjartsson, H. and S. Patz (1995). "The Rician distribution of noisy MRI data". In: *Magnetic Resonance in Medicine* 34.6, pp. 910–914. DOI: 10.1002/mrm.1910340618.
- Haario, H., E. Saksman, and J. Tamminen (2005). "Componentwise adaptation for high dimensional MCMC". In: *Computational Statistics* 20.2, pp. 265–273. DOI: 10.1007/BF02789703.
- Hagmann, P. et al. (Oct. 2006). "Understanding Diffusion MR Imaging Techniques: From Scalar Diffusion-weighted Imaging to Diffusion Tensor Imaging and Beyond". In: *RadioGraphics* 26.suppl\_1, S205–S223. DOI: 10.1148/rg.26si065510.
- Harms, R. L., F. J. Fritz, et al. (July 2017). "Robust and fast nonlinear optimization of diffusion MRI microstructure models". In: *NeuroImage* 155, October 2016, pp. 82–96. DOI: 10.1016/j.neuroimage.2017.04.064.
- Harms, R. L. and A. Roebroeck (Dec. 2018). "Robust and Fast Markov Chain Monte Carlo Sampling of Diffusion MRI Microstructure Models". In: *Frontiers in Neuroinformatics* 12, December, pp. 1–18. DOI: 10.3389/fninf.2018.00097.

- Hastings, W. K. (Apr. 1970). "Monte Carlo Sampling Methods Using Markov Chains and Their Applications". In: *Biometrika* 57.1, p. 97. DOI: 10.2307/2334940.
- Herculano-Houzel, S. (2009). "The human brain in numbers: a linearly scaled-up primate brain". In: *Frontiers in Human Neuroscience* 3.November, pp. 1–11. DOI: 10.3389/neuro.09.031.2009.
- Hernandez-Fernandez, M. et al. (2018). "Using GPUs to accelerate computational diffusion MRI: From microstructure estimation to tractography and connectomes". In: *bioRxiv* 188.November 2018, p. 371278. DOI: 10.1101/371278.
- Hernández, M. et al. (2013). "Accelerating Fibre Orientation Estimation from Diffusion Weighted Magnetic Resonance Imaging Using GPUs". In: *PLoS ONE* 8.4, e0130915. DOI: 10.1371/journal.pone.0061892.
- Hoffman, M. D. and A. Gelman (2011). "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo". In: *Journal of Machine Learning Research* 15, pp. 1593–1623.
- Horsfield, M. A. and D. K. Jones (2002). "Applications of diffusion-weighted and diffusion tensor MRI to white matter diseases - A review". In: *NMR in Biomedicine* 15.7-8, pp. 570–577. DOI: 10.1002/nbm.787.
- Jackman, S. (Oct. 2009). *Bayesian Analysis for the Social Sciences*. Chichester, UK: John Wiley & Sons, Ltd. DOI: 10.1002/9780470686621.
- Jelescu, I. O. and M. D. Budde (2017). "Design and Validation of Diffusion MRI Models of White Matter". In: *Frontiers in Physics* 5.November. DOI: 10.3389/fphy.2017.00061.
- Jelescu, I. O., J. Veraart, V. Adisetiyo, et al. (2015). "One diffusion acquisition and different white matter models: How does microstructure change in human early development based on WMTI and NODDI?" In: *NeuroImage* 107, pp. 242–256. DOI: 10.1016/j.neuroimage.2014.12.009.
- Jelescu, I. O., J. Veraart, E. Fieremans, et al. (2015). "Caveats of non-linear fitting to brain tissue models of diffusion". In: *ISMRM 2015*. Vol. 23, p. 88040.
- Jensen, J. H. et al. (2005). "Diffusional kurtosis imaging: The quantification of non-Gaussian water diffusion by means of magnetic resonance imaging". In: *Magnetic Resonance in Medicine* 53.6, pp. 1432–1440. DOI: 10.1002/mrm.20508.
- Jespersen, S. N. et al. (2010). "Neurite density from magnetic resonance diffusion measurements at ultrahigh field: Comparison with light microscopy and electron microscopy". In: *NeuroImage* 49.1, pp. 205–216. DOI: 10.1016/j.neuroimage.2009.08.053.
- Jeurissen, B. et al. (2014). "Multi-tissue constrained spherical deconvolution for improved analysis of multi-shell diffusion MRI data". In: *NeuroImage* 103, pp. 411–426. DOI: 10.1016/j.neuroimage.2014.07.061.
- Johnson, A. A., G. L. Jones, and R. C. Neath (2013). "Component-Wise Markov Chain Monte Carlo: Uniform and Geometric Ergodicity under Mixing and Composition". In: *Statistical Science* 28.3, pp. 360–375. DOI: 10.1214/13-STS423.

- 
- Jones, D. K. (2003). "Determining and visualizing uncertainty in estimates of fiber orientation from diffusion tensor MRI". In: *Magnetic Resonance in Medicine* 49.1, pp. 7–12. DOI: 10.1002/mrm.10331.
- Jones, D. K., T. R. Knösche, and R. Turner (2013). "White matter integrity, fiber count, and other fallacies: The do's and don'ts of diffusion MRI". In: *NeuroImage* 73, pp. 239–254. DOI: 10.1016/j.neuroimage.2012.06.081.
- Kamagata, K. et al. (2016). "Neurite orientation dispersion and density imaging in the substantia nigra in idiopathic Parkinson disease". In: *European Radiology* 26.8, pp. 2567–2577. DOI: 10.1007/s00330-015-4066-8.
- Kashyap, S. et al. (2018). "Effect of optimised coil-combinations on high-resolution laminar fMRI at 9.4T". In: *Proc. 26th Annual Meeting ISMRM, Paris (2018)*. Paris.
- Kass, R. E. et al. (May 1998). "Markov Chain Monte Carlo in Practice: A Roundtable Discussion". In: *The American Statistician* 52.2, p. 93. DOI: 10.2307/2685466.
- Kay, S. M. (1993). *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.
- Kunz, N. et al. (2014). "Assessing white matter microstructure of the newborn with multi-shell diffusion MRI and biophysical compartment models". In: *NeuroImage* 96, pp. 288–299. DOI: 10.1016/j.neuroimage.2014.03.057.
- Landman, B., P. L. Bazin, and J. Prince (2007). "Diffusion tensor estimation by maximizing rician likelihood". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1–8. DOI: 10.1109/ICCV.2007.4409140.
- Le Bihan, D. et al. (Nov. 1986). "MR imaging of intravoxel incoherent motions: application to diffusion and perfusion in neurologic disorders." In: *Radiology* 161.2, pp. 401–407. DOI: 10.1148/radiology.161.2.3763909.
- Lebel, C., S. Caverhill-Godkewitsch, and C. Beaulieu (2010). "Age-related regional variations of the corpus callosum identified by diffusion tensor tractography". In: *NeuroImage* 52.1, pp. 20–31. DOI: 10.1016/j.neuroimage.2010.03.072.
- Levenberg, K. (1944). "a Method for the Solution of Certain Non – Linear Problems in Least Squares". In: *Quarterly of Applied Mathematics* 2.2, pp. 164–168. DOI: citeulike-article-id:10796881.
- Li, H. et al. (June 2017). "Impact of transcytolemmal water exchange on estimates of tissue microstructural properties derived from diffusion MRI". In: *Magnetic Resonance in Medicine* 77.6, pp. 2239–2249. DOI: 10.1002/mrm.26309.
- Link, W. A. and M. J. Eaton (2012). "On thinning of chains in MCMC". In: *Methods in Ecology and Evolution* 3.1, pp. 112–115. DOI: 10.1111/j.2041-210X.2011.00131.x.
- Liu, J. S. (2004). *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. New York, NY: Springer New York. DOI: 10.1007/978-0-387-76371-2.

- Liu, J. S., F. Liang, and W. H. Wong (Mar. 2000). "The Multiple-Try Method and Local Optimization in Metropolis Sampling". In: *Journal of the American Statistical Association* 95.449, p. 121. DOI: 10.2307/2669532.
- Liu, Z. et al. (2010). "Quality control of diffusion weighted images". In: *Medical Imaging 2010: Advanced PACS-based Imaging Informatics and Therapeutic Applications*. Vol. 7628. DOI: 10.1117/12.844748.
- Ma, D. et al. (Mar. 2013). "Magnetic resonance fingerprinting". In: *Nature* 495.7440, pp. 187–192. DOI: 10.1038/nature11971.
- MacEachern, S. N. and L. M. Berliner (Aug. 1994). "Subsampling the Gibbs Sampler". In: *The American Statistician* 48.3, p. 188. DOI: 10.2307/2684714.
- Martino, L., V. Elvira, and F. Louzada (2017). "Effective sample size for importance sampling based on discrepancy measures". In: *Signal Processing* 131, pp. 386–401. DOI: 10.1016/j.sigpro.2016.08.025.
- Metropolis, N. et al. (June 1953). "Equation of State Calculations by Fast Computing Machines". In: *The Journal of Chemical Physics* 21.6, pp. 1087–1092. DOI: 10.1063/1.1699114.
- Mitchel, H. H. et al. (1945). "The chemical composition of the adult human body and its bearing on the biochemistry of growth." In: *Journal of Biological Chemistry* 168, pp. 625–637.
- Moeller, S. et al. (2010). "Multiband multislice GE-EPI at 7 tesla, with 16-fold acceleration using partial parallel imaging with application to high spatial and temporal whole-brain fMRI". In: *Magnetic Resonance in Medicine* 63.5, pp. 1144–1153. DOI: 10.1002/mrm.22361.
- Muller, P. (1994). *Metropolis based posterior integration schemes*. Tech. rep. Duke University, Durham: Institute of Statistics and Decision Sciences.
- Neil, J. et al. (2002). "Diffusion tensor imaging of normal and injured developing human brain - A technical review". In: *NMR in Biomedicine* 15.7-8, pp. 543–552. DOI: 10.1002/nbm.784.
- Nelder, J. A. and R. Mead (1965). "A simplex method for function minimization". In: *Computer Journal* 7.4, pp. 308–313. DOI: 10.1093/comjnl/7.4.308.
- Nilsson, M. et al. (2013). *The role of tissue microstructure and water exchange in biophysical modelling of diffusion in white matter*. DOI: 10.1007/s10334-013-0371-x.
- Novikov, D. S., V. G. Kiselev, and S. N. Jespersen (2018). "On modeling". In: *Magnetic Resonance in Medicine* 79.6, pp. 3172–3193. DOI: 10.1002/mrm.27101.
- Novikov, D. S., J. Veraart, et al. (July 2018). "Rotationally-invariant mapping of scalar and orientational metrics of neuronal microstructure with diffusion MRI". In: *NeuroImage* 174. December 2016, pp. 518–538. DOI: 10.1016/j.neuroimage.2018.03.006.
- Novikov, D. S., I. O. Jelescu, and E. Fieremans (2015). "From Diffusion Signal Moments to Neurite Diffusivities, Volume Fraction and Orientation Distribution: An Exact Solution". In: *Proc. 23rd Annual Meeting ISMRM, Toronto, Canada* 23, p. 469.

- 
- Oguz, I. et al. (2014). "DTIPrep: quality control of diffusion-weighted images". In: *Frontiers in Neuroinformatics* 8, January, pp. 1–11. DOI: 10.3389/fninf.2014.00004.
- Panagiotaki, E. et al. (2012). "Compartment models of the diffusion MR signal in brain white matter: A taxonomy and comparison". In: *NeuroImage* 59.3, pp. 2241–2254. DOI: 10.1016/j.neuroimage.2011.09.081.
- Pawitan, Y. (2013). *In all likelihood*. 1 edition. Oxford: Oxford University Press, p. 544.
- Perge, J. A. et al. (2009). "How the Optic Nerve Allocates Space, Energy Capacity, and Information". In: *Journal of Neuroscience* 29.24, pp. 7917–7928. DOI: 10.1523/jneurosci.5200-08.2009.
- Pfefferbaum, A. et al. (2000). "Age-related decline in brain white matter anisotropy measured with spatially corrected echo-planar diffusion tensor imaging". In: *Magnetic Resonance in Medicine* 44.2, pp. 259–268. DOI: 10.1002/1522-2594(200008)44:2<259::AID-MRM13>3.0.CO;2-6.
- Powell, M. (1964). "An efficient method for finding the minimum of a function of several variables without calculating derivatives". In: *The Computer Journal* 7.2, pp. 155–162. DOI: 10.1093/comjnl/7.2.155.
- (2006). *The NEWUOA software for unconstrained optimization without derivatives*. Tech. rep., pp. 255–297. DOI: 10.1007/0-387-30065-1\_16.
- (2015). "On fast trust region methods for quadratic models with linear constraints". In: *Mathematical Programming Computation* 7.3, pp. 237–267. DOI: 10.1007/s12532-015-0084-4.
- Press, W. et al. (2007). *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge University Press, p. 1235. DOI: citeulike-article-id:1386464.
- Rao, C. R. (1945). "Information and the accuracy attainable in the estimation of statistical parameters". In: *Bulletin of the Calcutta Mathematical Society* 37, pp. 81–89.
- Ravenzwaaij, D. van, P. Cassey, and S. D. Brown (2016). "A simple introduction to Markov Chain Monte-Carlo sampling". In: *Psychonomic Bulletin and Review*, pp. 1–12. DOI: 10.3758/s13423-016-1015-8.
- Rios, L. M. and N. V. Sahinidis (2013). "Derivative-free optimization: A review of algorithms and comparison of software implementations". In: *Journal of Global Optimization* 56.3, pp. 1247–1293. DOI: 10.1007/s10898-012-9951-y.
- Robert, C. P. (2015). *The Metropolis-Hastings algorithm*. Tech. rep. Wiley StatsRef: Statistics Reference Online, pp. 1–15. DOI: 10.1002/9781118445112.stat07834.
- Robert, C. P. and G. Casella (2004). *Monte Carlo Statistical Methods*. Springer Texts in Statistics. New York, NY: Springer New York. DOI: 10.1007/978-1-4757-4145-2.
- Roberts, G. O. and J. S. Rosenthal (2007). "Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms". In: *Journal of Applied Probability* 44, March 2005, pp. 458–475. DOI: 10.1239/jap/1183667414.



- Roberts, G. O. and J. S. Rosenthal (2009). "Examples of adaptive MCMC". In: *Journal of Computational and Graphical Statistics* 18.2, pp. 349–367. DOI: 10.1198/jcgs.2009.06134.
- Rodriguez, C. L. et al. (Oct. 2013). "Inadequacies of the Fisher information matrix in gravitational-wave parameter estimation". In: *Physical Review D* 88.8, p. 084013. DOI: 10.1103/PhysRevD.88.084013.
- Rowan, T. H. (1990). "Functional Stability analysis of numerical algorithms". Phd. Thesis. University of Texas at Austin.
- Sairanen, V., A. Leemans, and C. M. Tax (2018). "Fast and accurate Slice-wise Outlier Detection (SOLID) with informed model estimation for diffusion MRI data". In: *NeuroImage* 181.May, pp. 331–346. DOI: 10.1016/j.neuroimage.2018.07.003.
- Santis, S. D., Y. Assaf, B. Jeurissen, et al. (2016). "T1 relaxometry of crossing fibres in the human brain". In: *NeuroImage* 141, pp. 133–142. DOI: 10.1016/j.neuroimage.2016.07.037.
- Santis, S. D., Y. Assaf, and D. K. Jones (2012). "Using the biophysical CHARMED model to elucidate the underpinnings of contrast in diffusional kurtosis analysis of diffusion-weighted MRI". In: *Magnetic Resonance Materials in Physics, Biology and Medicine* 25.4, pp. 267–276. DOI: 10.1007/s10334-011-0292-5.
- Schwarz, G. (1978). "Estimating the dimension of a model". en. In: *The Annals of Statistics* 6.2, pp. 461–464. DOI: 10.1214/aos/1176344136.
- Shahar, D. J. (2017). "Minimizing the Variance of a Weighted Average". In: *Open Journal of Statistics* 7, pp. 216–224. DOI: 10.4236/ojs.2017.72017.
- Sherlock, C., P. Fearnhead, and G. O. Roberts (2010). "The Random Walk Metropolis: Linking Theory and Practice Through a Case Study". In: *Statistical Science* 25.2, pp. 172–190. DOI: 10.1214/10-STS327.
- Sjölund, J. et al. (2018). "Bayesian uncertainty quantification in linear models for diffusion MRI". In: *NeuroImage* 175.November 2017, pp. 272–285. DOI: 10.1016/j.neuroimage.2018.03.059.
- Smith, R. L. (Dec. 1984). "Efficient Monte Carlo Procedures for Generating Points Uniformly Distributed over Bounded Regions". In: *Operations Research* 32.6, pp. 1296–1308. DOI: 10.1287/opre.32.6.1296.
- Smith, S. M. (2002). "Fast robust automated brain extraction". In: *Human Brain Mapping* 17.3, pp. 143–155. DOI: 10.1002/hbm.10062.
- Smith, S. M. et al. (July 2006). "Tract-based spatial statistics: Voxelwise analysis of multi-subject diffusion data". In: *NeuroImage* 31.4, pp. 1487–1505. DOI: 10.1016/j.neuroimage.2006.02.024.
- Sotak, C. H. (2002). "The role of diffusion tensor imaging in the evaluation of ischemic brain injury - a review." In: *NMR Biomed* 15.7-8, pp. 561–569. DOI: 10.1002/nbm.786.
- Sotiropoulos, S. N. et al. (2013). "RubiX: Combining spatial resolutions for bayesian inference of crossing fibers in diffusion MRI". In: *IEEE Transactions on Medical Imaging* 32.6, pp. 969–982. DOI: 10.1109/TMI.2012.2231873.

- 
- Stepišnik, J. (1993). "Time-dependent self-diffusion by NMR spin-echo". In: *Physica B: Physics of Condensed Matter* 183.4, pp. 343–350. DOI: 10.1016/0921-4526(93)90124-O.
- Stone, J. E., D. Gohara, and G. Shi (May 2010). "OpenCL: A Parallel Programming Standard for Heterogeneous Computing Systems". In: *Computing in Science & Engineering* 12.3, pp. 66–73. DOI: 10.1109/MCSE.2010.69.
- Sundgren, P. C. et al. (May 2004). "Diffusion tensor imaging of the brain: review of clinical applications". In: *Neuroradiology* 46.5, pp. 339–350. DOI: 10.1007/s00234-003-1114-x.
- Syková, E. and C. Nicholson (Oct. 2008). "Diffusion in Brain Extracellular Space". In: *Physiological Reviews* 88.4, pp. 1277–1340. DOI: 10.1152/physrev.00027.2007.
- Tariq, M. et al. (2016). "Bingham-NODDI: Mapping anisotropic orientation dispersion of neurites using diffusion MRI". In: *NeuroImage* 133, pp. 207–223. DOI: 10.1016/j.neuroimage.2016.01.046.
- Timmers, I. et al. (Mar. 2015). "White matter microstructure pathology in classic galactosemia revealed by neurite orientation dispersion and density imaging." In: *Journal of inherited metabolic disease* 38.2, pp. 295–304. DOI: 10.1007/s10545-014-9780-x.
- Tournier, J. D., F. Calamante, and A. Connelly (2007). "Robust determination of the fibre orientation distribution in diffusion MRI: Non-negativity constrained super-resolved spherical deconvolution". In: *NeuroImage* 35.4, pp. 1459–1472. DOI: 10.1016/j.neuroimage.2007.02.016.
- Tournier, J.-D. et al. (Nov. 2004). "Direct estimation of the fiber orientation density function from diffusion-weighted MRI data using spherical deconvolution". In: *NeuroImage* 23.3, pp. 1176–1185. DOI: 10.1016/j.neuroimage.2004.07.037.
- Tuch, D. S. (2004). "Q-ball imaging". In: *Magnetic Resonance in Medicine* 52.6, pp. 1358–1372. DOI: 10.1002/mrm.20279.
- Turchin, V. F. (Jan. 1971). "On the Computation of Multidimensional Integrals by the Monte-Carlo Method". In: *Theory of Probability & Its Applications* 16.4, pp. 720–724. DOI: 10.1137/1116083.
- Vallisneri, M. (2008). "Use and abuse of the Fisher information matrix in the assessment of gravitational-wave parameter-estimation prospects". In: *Physical Review D - Particles, Fields, Gravitation and Cosmology* 77.4, pp. 1–20. DOI: 10.1103/PhysRevD.77.042001.
- Van Essen, D. et al. (Oct. 2012). "The Human Connectome Project: A data acquisition perspective". In: *NeuroImage* 62.4, pp. 2222–2231. DOI: 10.1016/j.neuroimage.2012.02.018.
- Vats, D., J. M. Flegal, and G. L. Jones (Dec. 2015). "Multivariate Output Analysis for Markov chain Monte Carlo". In: *ArXiv e-prints* 1, pp. 1–52.
- Versteeg, E. et al. (2018). "Probabilistic Tractography for Complex Fiber Orientations with Automatic Model Selection". In: *Computational Diffusion MRI*. Ed. by E. Kaden et al. Cham: Springer International Publishing, pp. 117–128.

- Walhovd, K. B., H. Johansen-Berg, and R. T. K arad ttir (2014). "Unraveling the secrets of white matter - Bridging the gap between cellular, animal and human imaging studies". In: *Neuroscience* 276, pp. 2–13. DOI: 10.1016/j.neuroscience.2014.06.058.
- Wang, P. C. and T. E. Shoup (2011). "Parameter sensitivity study of the Nelder-Mead Simplex Method". In: *Advances in Engineering Software* 42.7, pp. 529–533. DOI: 10.1016/j.advengsoft.2011.04.004.
- Wedeen, V. J. et al. (2005). "Mapping Complex Tissue Architecture With Diffusion Spectrum Magnetic Resonance Imaging". In: 1386, pp. 1377–1386. DOI: 10.1002/mrm.20642.
- Wegmann, B., A. Eklund, and M. Villani (2017). "Bayesian Rician regression for neuroimaging". In: *Frontiers in Neuroscience* 11.OCT. DOI: 10.3389/fnins.2017.00586.
- Weigel, M. (2015). "Extended phase graphs: Dephasing, RF pulses, and echoes - Pure and simple". In: *Journal of Magnetic Resonance Imaging* 41.2, pp. 266–295. DOI: 10.1002/jmri.24619.
- Wen, Q. et al. (2015). "Clinically feasible NODDI characterization of glioma using multiband EPI at 7 T". In: *NeuroImage: Clinical* 9, pp. 291–299. DOI: 10.1016/j.nicl.2015.08.017.
- Wengert, R. E. (1964). "A simple automatic derivative evaluation program". In: *Communications of the ACM* 7.8, pp. 463–464. DOI: 10.1145/355586.364791.
- Weniger, E. J. (1991). "On the derivation of iterated sequence transformations for the acceleration of convergence and the summation of divergent series". In: *Computer Physics Communications* 64.1, pp. 19–45. DOI: 10.1016/0010-4655(91)90047-O.
- Werring, D. J. et al. (1999). "Diffusion tensor imaging of lesions and normal-appearing white matter in multiple sclerosis". In: *Neurology* 52.8, pp. 1626–1626. DOI: 10.1212/WNL.52.8.1626.
- Whitcher, B. et al. (2008). "Using the wild bootstrap to quantify uncertainty in diffusion tensor imaging". In: *Human Brain Mapping* 29.3, pp. 346–362. DOI: 10.1002/hbm.20395.
- Winston, G. P. et al. (Feb. 2014). "Advanced diffusion imaging sequences could aid assessing patients with focal cortical dysplasia and epilepsy". In: *Epilepsy Research* 108.2, pp. 336–339. DOI: 10.1016/j.eplepsyres.2013.11.004.
- Woolrich, M. W. et al. (2004). "Multilevel linear modelling for fMRI group analysis using Bayesian inference". In: *NeuroImage* 21.4, pp. 1732–1747. DOI: 10.1016/j.neuroimage.2003.12.023.
- Xu, J. et al. (2013). "Evaluation of slice accelerations using multiband echo planar imaging at 3T". In: *NeuroImage* 83, pp. 991–1001. DOI: 10.1016/j.neuroimage.2013.07.055.
- Yablonskiy, D. A., G. L. Bretthorst, and J. J. Ackerman (Oct. 2003). "Statistical model for diffusion attenuated MR signal". In: *Magnetic Resonance in Medicine* 50.4, pp. 664–669. DOI: 10.1002/mrm.10578.

- 
- Yap, P.-T., Y. Zhang, and D. Shen (2015). "Iterative Subspace Screening for Rapid Sparse Estimation of Brain Tissue Microstructural Properties". In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI*. Ed. by N. Navab et al. Cham: Springer International Publishing, pp. 223–230.
- Zelinka, I. (2004). "SOMA - Self-Organizing Migrating Algorithm". In: *New Optimization Techniques in Engineering*. Ed. by G. C. Onwubolu and B. V. Babu. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 167–217. DOI: 10.1007/978-3-540-39930-8\_7.
- Zhang, H., P. L. Hubbard, et al. (2011). "Axon diameter mapping in the presence of orientation dispersion with diffusion MRI". In: *NeuroImage* 56.3, pp. 1301–1315. DOI: 10.1016/j.neuroimage.2011.01.084.
- Zhang, H., T. Schneider, et al. (2012). "NODDI: Practical in vivo neurite orientation dispersion and density imaging of the human brain". In: *NeuroImage* 61.4, pp. 1000–1016. DOI: 10.1016/j.neuroimage.2012.03.072.



# Valorisation

Knowledge valorisation refers to the “process of creating value from knowledge, by making knowledge suitable and/or available for social (and/or economic) use and by making knowledge suitable for translation into competitive products, services, processes and new commercial activities” - as detailed in *“Regulation governing the attainment of doctoral degrees”*, Maastricht University (2018). As the work presented in this thesis can potentially be applied in commercial products, in this chapter I take the opportunity to elaborate on the various valorisable aspects of the work in this thesis.

Diffusion MRI (dMRI) is one of the preferred approaches for investigating the brain’s white matter microstructure in vivo. In a scientific setting, dMRI can be used to further a structural understanding of the human brain, while in a clinical setting it can be used as a diagnostic instrument for detecting, for instance, acute ischemia (lack of oxygen) in the human brain.

Due to the nature of a dMRI scan, the measured signal provides only an indirect view of the underlying cellular structures. This makes modeling essential in order to extract quantitative measures from diffusion MRI data. Many models have been proposed over the years, each differing in underlying assumptions and applicable tissue types. To infer information from the dMRI signal, most of these models need to be fitted to the data. This model fitting typically depends on non-linear analysis methods which have their respective problems in robustness, accuracy, precision and run-time. Improving on these can have a large impact in both the scientific and clinical domain.

## The choice of algorithms

Since the quality and run-time of the dMRI analysis depend on the methods used, the investigation in the choice of algorithms can save both time and money.

In the first two chapters of this thesis we compared various optimization and Markov Chain Monte Carlo (MCMC) sampling algorithms on their usefulness for dMRI modeling. We noted that a smart choice of algorithm can heavily influence the run-time and quality of the results. For example, by showing that burn-in and thinning in MCMC sampling should be avoided, we can save hours of computation time for every dataset. In addition, we showed that using the same optimization routine for different models can improve the quality of the model comparisons. Since the used optimization routines influences the results, using the same routine for different models removes a possible confound from a

study, increasing the effect size and leading to a lower requirement in required participants.

The results from the third chapter also allow for a potential reduction of required participants in a dMRI study. The use of weighted averaging can reduce the effect of white matter artifacts in the averaged results. As such, subjects that previously would have been removed from consideration can now still be included in the group statistics as the weighted averaging will take care of removing or dampening the effect of the white matter artifacts. In addition, weighted averaging promises the lowest possible standard deviation of the mean. This in turn provides for a higher effect size when comparing two different populations. Both these effects can reduce the number of participants required for a study, reducing the number of expensive dMRI scans required.

## Advantages of GPU computing

Next to the choice of analysis methods, the specific implementation can affect the analysis computation times. Since every increase in (d)MRI resolution leads to a power of three increment in the number of voxels, we need faster hardware and software to keep up with future datasets. Recently, Graphical Processing Units (GPU's) have seen increased usage in data sciences due to their large parallel computation possibilities. The use of a GPU is however not straightforward as software needs to be specifically adapted to take advantages of their compute power.

To take advantage of graphical processing units in dMRI modeling, we developed multiple open-source software packages with native support for GPU computations. The first of these packages, the MOT package, contains GPU accelerated implementations of all the non-linear optimization routines, all Markov Chain Monte Carlo sampling methods and the Fisher Information Matrix computation as used in this thesis. The second software package, the MDT package, contains highly optimized implementations of a large set of dMRI models. By making these routines open-source we bring all the advantages of this thesis to the community and wider public.

The reduction in computation time associated with GPU computing has several advantages. First of all, it allows researchers to analyze datasets faster, allowing for a higher throughput in modeling analysis. Second, it allows for more and faster iterations when developing new dMRI models, making research more time efficient. As a third advantage, it could reduce the need for a compute cluster. Since GPU's are between 30 and 60 times more efficient than a central processing unit (CPU) for dMRI modeling (see Chapter 4), providing researchers with a

---

single good GPU can prevent the need for a CPU cluster, potentially saving large amounts of money.

## The modeling framework

Although GPU computing is a large selling point of the MDT and MOT software packages presented in this work, the MDT package has another useful feature, a large scale modeling framework. This modeling framework provides researchers with several advantages. Foremost, by bundling various dMRI models in one software package we allow for model comparisons using a single optimization routine for all models. This will increase effect size, leading to a reduction of required participants. Another useful aspect is that by providing a model building framework we allow researchers to reuse existing model implementations. This can save time when developing new models as the existing models are highly optimized and considered correct.

As a final selling point, the presented software packages can perhaps form the bridge between the scientific and the clinical domains. If the optimization and MCMC sampling routines can be verified for use in a medical setting, the MDT and MOT software packages could be used a model implementation framework to bring new dMRI models to the clinic.

## Beyond diffusion MRI

One of the software packages written for this thesis is a stand-alone non-linear optimization toolbox with parallel processing capabilities, the Multithreaded Optimization Toolbox (MOT). While this software was essential to accelerating the computations in diffusion MRI analysis, it is in essence a general purpose optimization toolbox, capable of being extended to various other domains of science. For example, already within the field of magnetic resonance imaging, one may think of accelerating functional MRI computations, quantitative magnetic transfer models and structural MRI models. Outside of MRI, MOT may prove of use in genetic data analysis, protein folding, cancer cell research, microscopy analysis or any other field where a large amount of parallelizable optimization computations are required.





# Acknowledgments

First and foremost, I would like to express my gratitude towards my supervisors Rainer and Alard for giving me the opportunity to pursue a doctorate degree at Maastricht University.

Rainer, I would like to especially thank you for the position at your company such that I had funding for the first three years of my research.

Alard, I am immensely indebted to you for your help, support and guidance over the course of our many years together. I have learned a great deal from you during our time together. Practical things like new tools and new ideas, but also mindful things, like believing in oneself and to not be afraid of change. Your encouragement and enthusiasm is often what kept me going when I felt my work was not good enough. Thank you for mentoring me.

Derek, Chantal, Mark, Erika, Elena, Maxime, thank you all for the collaborations, the cheerful retreat and the overall good time in Cardiff. Your energy, spirit and dedication are a joy to work with. Special thanks for Derek for inviting me to present my work at the CONNECT meeting.

Tony, Rudy, Alexandra, from the university in Bonn, thank you for the pleasant collaboration the past few years and thank you for believing in me and my work already from the early day onwards, this was of great support to me.

Gary Hui Zhang, I would like to express my great appreciation for the numerous times you gave feedback to about early versions of MDT. Without your input, it would not have been the software it is today.

Silvia de Santis, please accept my gratitude for the countless e-mails you replied to concerning various diffusion MRI modeling details. The CHARMED implementation in MDT is indebted to you.

Frank Thuijsman, my mentor and confidant for many years. You taught me the value of feedback and taught me I should trust myself and my work. I value the contact we have and I am very grateful for your time and wisdom on many occasions. Initially, you were the one who brought me to the field of neuroscience, after a lengthy discussion about my master thesis topic. This thesis is a result of that road taken.

Christl en Riny, thank you both for your help and support concerning the paperwork and other formalities. This PhD would not have been completed without your help.

## Acknowledgments

---

To Arko, Franc, Sven, Sanne, Michael, Anna, Annemarie, Johannes, Bram and Tim from my CBC-lab group, to Ben, Nikos, Sri from other groups at the university, and to Ralph, Quintin and others from Brain Innovation, thank you all for the good time together. Even though I worked from home a lot, I still appreciate coming to the office to have a nice chat now and then.

Michael, you have been a voice of wisdom on multiple occasions, in retrospect it is perhaps unfortunate I only heeded your advice years later. Nonetheless, I did write papers in the end. An additional thank you goes out for introducing me to a new hobby, bouldering, which is indeed good fun.

Sven, thank you for the interesting discussions about biology and science during our time together, and a special thanks for all the cookies and tea you brought to the office.

Sri, your jokes, creativity and good mood have often cheered my day. Thank you for a lot of things, among which cheering me up, introducing me to Indian food and teaching me how to eat with my hands. Additionally, thank you for your occasional pieces of advice, such as that I should advocate my toolbox more.

Franc, your passion for MRI was contagious, you always got excited by new experiments and you could only relax when all the spins aligned. I will not forget our nice walks together in the mountains around Maastricht, our thought-provoking conversations about space and time, and your numerous lessons in physics. Thank you bro.

Arko, I would like to thank you for so many things. You were always there for me when I needed someone to talk to. Your thoughtful and considerate pieces of advice have often helped me sort things out. I look back fondly on so many good times we had together, be it playing MtG, buying furniture or plants, road-tripping, witnessing your marriage, or just sipping tea, coffee or whiskey together. I hope I will be seeing you around.

Christophe, Lily, thank you for being my first new friends in Maastricht. Christophe, I really enjoyed playing table tennis with you, even though you let me win only once per year.

Ralf, mijn dankwoord zou incompleet zijn zonder jou. Jij hebt ontzettend veel voor me betekend, zowel voor als tijdens mijn onderzoek. Ik kijk met plezier terug op vele diepgaande discussies, leuke stedentrips en ontspannende wandelingen. Daarnaast rij ik nog met veel plezier in die oude mercedes rond die je me verkocht hebt. Je hebt me veel geleerd en meegegeven, daar blijf ik je altijd dankbaar voor.

Marjo, bedankt voor je vele hulp en steun in de afgelopen jaren. Ik heb veel van je mogen leren, nieuwe inzichten met betrekking tot software ontwikkeling, de

---

waarde van het 'upward managen' en misschien nog wel het belangrijkste, over hoe je werk en privé in balans kan houden.

Peter, ik weet nog dat wij op de bassisschool samen onze eigen stad ontwierpen. Architect ben ik uiteindelijk niet geworden, maar mijn gevoel voor creëren is onverminderd aanwezig. Ik waardeer onze vriendschap en proost graag op nog veel mooie momenten samen.

Suus en Hein, ik vind het nog steeds bijzonder dat wat eenvoudig begon met pianolessen heeft kunnen uitgroeien tot zo'n mooie vriendschap. Jullie enthousiasme voor muziek en andere mooie dingen heeft mij iets moois meegegeven voor het leven. Suus, ik herinner me nog vele mooie uren samen achter de piano, muziek makend en pratend over het leven. Meer dan een piano docent ben je ook een docent in de levenskunst. Hein, jouw tomeloze enthousiasme voor alles rondom muziek is inspirerend. Ik herinner me met veel plezier nog ons gezamenlijk project, waarbij we muziek en software hebben weten te combineren tot een ware opus infinity.

Joke, Joop, Myrte, Daniël, Iris, het is moeilijk uit te drukken hoezeer ik het contact met jullie waardeer. Joke, jou wil ik met veel liefde bedanken voor je steun en troost in moeilijke tijden, je aandachtig luisteren, je vele zorgen en je pep-talks over reeds vele jaren. Ik hoop dat je Liefde Loont project ooit gemeengoed wordt, de wereld heeft het nodig. Joop, graag zou ik jou eveneens willen bedanken voor je vele hulp en aandacht, en zou ik je in het bijzonder willen bedanken voor al je fantastische maaltijden. Myrte, Daniel, bedankt voor het delen van tips en trucs rondom promovieren, en in het algemeen, bedankt voor alle gezellige momenten. Iris, met veel liefde zou ik jou willen bedanken voor alle gedeelde momenten van reflectie, voor je aandachtig luisteren en in het bijzonder voor het mij leren genieten van de fijne spijzen en dranken op deze aarde.

Opa, Oma, gelukkig heb ik jullie beiden nog kunnen vertellen hoe veel jullie voor mij hebben betekend. Ik herinner me nog met veel vreugde talloze mooie discussies over wetenschap, filosofie en het leven in het algemeen. Jullie worden gemist.

Oma Mizi, u bent de coolste oma die ik ken. Ik waardeer het zeer dat u altijd bereid bent om mee te denken en altijd paraat staat om tot actie over te gaan. Daarnaast weet u altijd en overal de mooiste dingen vandaan te halen om een nieuw huis direct wat op te fleuren.

Friedemann, Wendie, ontzettend bedankt voor de prachtige cover die jullie voor me ontworpen hebben. Een mooie cover tilt het boekje toch echt naar een ander niveau.

Bas, Job, jullie beiden zijn mij veel waard. Sebastiaan, ik heb zeer genoten van onze lange discussies over taal, geneeskunde en de menselijke psyche. Daarnaast

heb jij me laten zien dat er meer in het leven is dan een computer. Zonder jouw voorbeeld zou ik niet zoveel gereisd en gezien hebben. Job, haarscherp maar met een komische noot weet je me altijd wel een steuntje in de rug te geven. Jij hebt me laten zien hoe je van je passie je werk kunt maken, dat zal ik altijd als voorbeeld bij me houden.

Pap, Marja, met veel liefde wil ik jullie bedanken voor jullie aandachtig luisteren en de vele momenten van steun in de afgelopen jaren. Jullie stonden altijd voor me klaar, als het goed ging, maar ook op de momenten dat het leven even tegengaat. Ik weet dat ik bij jullie altijd terecht kan. Marja, graag extra bedankt voor je inzichten in de menselijke psyche, ik denk dat het geholpen heeft. Pa, ik geniet ontzettend van onze momenten samen. Of dat nou fietsend, zeilend, wandelend is of als we samen muziek maakten, het gaf me vaak energie voor een hele week.

Muttie, Gerrit, met veel liefde wil ik jullie bedanken voor jullie eeuwige steun en toeverlaat. Ik waardeer het ontzettend dat jullie me het gevoel geven dat er altijd een thuis is om even terug te trekken als de wereld weer wat te veel wordt. Gerrit, extra bedankt voor al je zakelijke tips en trucs, in problematische tijden halen die veel stress weg. Lieve Ma, jij bent en blijft voor altijd mijn numero uno muttie. Ik vind het heel bijzonder dat jij dag en nacht voor mij klaar staat. Jouw doorzettingsvermogen en energie zijn een voorbeeld voor mij.

Lieve Sanne, er zijn zoveel dingen waar ik je voor zou willen bedanken dat het boekje dubbel zo dik zou worden als ik ze allemaal zou noemen. Als collega zou ik je graag willen bedanken voor je begeleiding bij de laatste paar hoofdstukken van mijn thesis. Je hulp en inzicht bij het schrijven worden zeer gewaardeerd. Als geliefde zou ik je graag willen bedanken voor je eeuwige steun en vertrouwen in mij. Hoeveel nachten je wel niet naar mij geluisterd hebt omdat ik het even allemaal niet meer zag zitten en hoeveel weken en maanden gezeur en gekreun je allemaal hebt weten te doorstaan. Ik vind dat heel bijzonder en dank jou met veel liefde. Jij bent mijn schattekous. Mijn lief.

Robbert Harms  
Maastricht  
2019-10-25





# Research output

## Publications

### Part of this thesis

1. **R. L. Harms**, F. J. Fritz, S. Schoenmakers and A. Roebroeck (2019). "Fast quantification of uncertainty in non-linear diffusion MRI models for artifact detection and more power in group studies". *In review at NeuroImage*
2. **R. L. Harms** and A. Roebroeck (2019). "GPU accelerated MRI analysis with the Microstructure Diffusion Toolbox (MDT)", *In preparation*.
3. **R. L. Harms** and A. Roebroeck (Dec. 2018c). "Robust and Fast Markov Chain Monte Carlo Sampling of Diffusion MRI Microstructure Models". In: *Frontiers in Neuroinformatics* 12.December, pp. 1–18
4. **R. L. Harms**, F. J. Fritz, A. Tobisch, R. Goebel, and A. Roebroeck (July 2017). "Robust and fast nonlinear optimization of diffusion MRI microstructure models". In: *NeuroImage* 155.October 2016, pp. 82–96

### Co-authored

1. M. Drakesmith, **R. Harms**, S. U. Rudrapatna, G. D. Parker, C. J. Evans, and D. K. Jones (Dec. 2019). "Estimating axon conduction velocity in vivo from microstructural MRI". in: *NeuroImage* 203.September, p. 116186
2. F. J. Fritz, S. Sengupta, **R. L. Harms**, D. Tse, B. A. Poser, and A. Roebroeck (Aug. 2019). "Ultra-high resolution and multi-shell diffusion MRI of intact ex vivo human brains using kT-dSTEAM at 9.4T". in: *NeuroImage*, p. 116087
3. E. Schwarz et al. (2019). "Reproducible grey matter patterns index a multivariate, global alteration of brain structure in schizophrenia and bipolar disorder". In: *Translational Psychiatry* 9.1
4. A. Tobisch, R. Stirnberg, **R. L. Harms**, T. Schultz, A. Roebroeck, M. M. Breteler, and T. Stöcker (2018). "Compressed sensing diffusion spectrum imaging for accelerated diffusion microstructure MRI in long-term population imaging". In: *Frontiers in Neuroscience* 12.SEP, pp. 1–19



5. S. Sengupta, F. J. Fritz, **R. L. Harms**, S. Hildebrand, D. H. Tse, B. A. Poser, R. Goebel, and A. Roebroeck (2018). "High resolution anatomical and quantitative MRI of the entire human occipital lobe ex vivo at 9.4 T". in: *NeuroImage* 168.March 2017, pp. 162–171

## Conference proceedings

1. **R. L. Harms** and A. Roebroeck (2018b). "More certainty about your uncertainty in diffusion MRI microstructure estimates". In: *Proc. 26th Annual Meeting ISMRM, Paris (2018)*
2. **R. L. Harms** and A. Roebroeck (2018d). "Robust and fast MCMC sampling of diffusion MRI microstructure models". In: *Proc. 26th Annual Meeting ISMRM, Paris (2018)*
3. S. Kashyap, F. J. Fritz, **R. L. Harms**, L. Huber, D. Ivanov, A. Roebroeck, B. A. Poser, and K. Uludağ (2018). "Effect of optimised coil-combinations on high-resolution laminar fMRI at 9.4T". in: *Proc. 26th Annual Meeting ISMRM, Paris (2018)*. Paris
4. F. Fritz, S. Sengupta, **R. L. Harms**, B. A. Poser, and A. Roebroeck (2018). "Ultra-high resolution multi-shell dMRI and tractography of the ex vivo human brain using kT-dSTEAM at 9.4T". in: *Proc. 26th Annual Meeting ISMRM, Paris (2018)*
5. **R. L. Harms** and A. Roebroeck (2017b). "The Maastricht Diffusion Toolbox (MDT): Modular, GPU accelerated, dMRI microstructure modeling". In: *Proc. 25th Annual Meeting ISMRM, Honolulu (2017)*
6. **R. L. Harms**, R. Goebel, and A. Roebroeck (2017). "Statistical power and confounds for diffusion MRI microstructure modeling". In: *Proc. 25th Annual Meeting ISMRM, Honolulu (2017)*
7. **R. L. Harms** and A. Roebroeck (2017a). "The Maastricht Diffusion Toolbox (MDT): Modular, GPU accelerated, dMRI microstructure modeling". In: *Proc. 23rd Annual Meeting OHBM, Vancouver, Canada (2015)*
8. **R. L. Harms**, R. Goebel, and A. Roebroeck (2016). "Diffusion microstructure in the population: variability and effect size of biophysical compartment model parameters over 100 subjects". In: *Proc. 24th Annual Meeting ISMRM, Singapore (2016)*
9. F. Fritz, S. Foxley, S. Sengupta, **R. L. Harms**, S. Caspers, K. Zilles, D. Tse, B. A. Poser, K. L. Miller, and A. Roebroeck (2016). "Whole human brain diffusion MRI at 450um post mortem with dwSSFP and a specialized 9.4T RF-coil". In: *Proc. 24th Annual Meeting ISMRM, Singapore (2016)*

- 
10. **R. L. Harms**, S. de Santis, M. Bastiani, R. Goebel, and A. Roebroek (2015). "Diffusion Microstructure Modelling Using a Modular and Extensible GPU Accelerated Toolkit". In: *Proc. 23rd Annual Meeting ISMRM, Toronto, Canada (2015)*. Toronto
  11. **R. L. Harms**, M. Bastiani, J. Xu, E. Yacoub, R. Goebel, and A. Roebroek (2014). "White matter microstructure modeling from multi-shell multi-TE diffusion MRI". in: *Proc. 20th Annual Meeting OHBM, Hamburg (2014)*



# About the author

Robbert Harms was born in Helmond, the Netherlands, on 29th October 1988. Following his interest in logic, computation and mathematics he enrolled at the Department of Data Science and Knowledge Engineering at Maastricht University. In 2010 he was granted his bachelor on the basis of his research and thesis, supervised by Dr. Evgueni N. Smirnov and Dr. Georgi Nalbantov, on the automatic classification of paintings according to art periods. During his bachelor years he also started his own business, delivering software solutions to small and medium-sized enterprises.

Robbert continued his education at the same department, pursuing a Master of Science in Operations Research. In 2012 he graduated with distinction on a master thesis on Diffusion MRI Analysis, under the supervision of Prof. Dr. Frank Thuijsman and Dr. Alard Roebroek. He remained part-time self-employed alongside the Master program.

After graduating, he first continued his work in his own company, but fulltime. However, he missed the challenges of academic life and applied for a PhD position. A dual position was offered at Brain Innovation, granted by Prof. Dr. Rainer Goebel, where he worked as a data analyst for the IMAGEMEND (Imaging Genetics for Mental Disorders) consortium while simultaneously starting as a PhD student, studying Diffusion MRI under the supervision of both Rainer Goebel and Alard Roebroek. When his three year contract ended he continued his PhD research at Maastricht University, now funded by an ERC Starting Grant MULTICONNECT awarded by Alard Roebroek. The open-source software presented as a milestone achievement of this applied research led to several educational stipends at a number of significant conferences, and led to one abstract merit award. Recently he has become employed as a Machine Learning Expert at Altoida, a company focusing on pre-symptomatic Alzheimer's detection.

