



Generation of realistic white matter substrates with controllable morphology for diffusion MRI simulations

Ross Callaghan

Supervisor: Dr. Gary Zhang

Co-Supervisor: Dr. Marco Palombo

Centre for Doctoral Programme in Medical Imaging

January 24, 2019

B

This work is supported by the EPSRC-funded UCL Centre for Doctoral Training in Medical Imaging (EP/L016478/1) and the Department of Health's NIHR-funded Biomedical Research Centre at UCLH



Abstract

Abstract goes here

Contents

Contents	i
List of Figures	iii
List of Algorithms	v
List of Acronyms	vii
1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	3
1.3 Project Aims and Scope	3
1.4 Report Overview	3
2 Background	5
2.1 MR Physics	5
2.1.1 Nuclear Magnetism	6
2.1.2 Magnetic Resonance	6
2.1.3 The Bloch Equations	8
2.1.4 Detecting the MR Signal	11
2.1.5 Spin Echoes	13
2.2 Diffusion MRI	14
2.2.1 The Bloch-Torrey Equations	17
2.3 Diffusion Simulation	18
2.3.1 Numerical Solutions	18
2.3.2 Monte-Carlo Simulations	18
2.4 Graphics Processing Units	20
3 Literature Review	23
3.1 Numerical Solutions	23
3.2 Monte-Carlo - Packages	24
3.3 Monte-Carlo - Numerical Phantoms	25

3.4	graphics processing unit (GPU) accelerated Monte-Carlo (MC) simulations	28
3.5	Conclusions	28
4	Contextual Fibre Growth (ConFiG)	31
4.1	ConFiG: Contextual Fibre Growth	31
4.1.1	Input to the algorithm	32
4.1.2	Creation of the Growth Network	33
4.1.3	Growth of a Fibre	34
4.1.4	Creation of Fibre Meshes	36
4.2	Experiments and Results	37
4.2.1	Effect of Choice of Growth Network	37
4.2.2	Demonstration of ConFiG	38
4.2.3	Comparison with Brute-Force Approach	38
4.3	Discussion and Conclusion	40
5	CUDAmino	43
5.1	CUDAmino Design	43
5.2	CUDAmino Experiments	45
5.3	Discussion and Conclusion	47
6	Future Plan	49
6.1	Improving ConFiG	49
6.2	Extending CUDAmino	50
6.3	Future Applications	51
6.4	Plan of Action	51
References		55

List of Figures

2.1	a) Illustration of the precessional motion of a single spin with magnetic moment μ in the presence of an external magnetic field B_0 . b) Many individual spins in an external magnetic field precess around the external field with random phase producing a net magnetisation in the direction of the B_0 field.	7
2.2	Nutation motion of an on-resonance spin in the presence of an radiofrequency (RF) field. Precession about the B_0 and B_1 fields create the spiralling motion in the laboratory frame.	8
2.3	Motion of a spin in the presence of a B_1 RF field in the rotating frame. This is identical to the nutation in Figure 2.2, however viewing from the rotating frame simplifies the motion.	11
2.4	The effective field, B_e , produced due to an off-resonance frequency Ω . The off-resonance effects produce an additional component of magnetic field along the z' axis.	12
2.5	The free induction decay described by Equation (2.32). Here, just the real channel is plotted.	13
2.6	a) Spin echo sequence and (b) an indication of the evolution of spins under a spin echo sequence. This shows how the 180° refocusing pulse acts to refocus the spins after a time TE.	13
2.7	The standard pulsed gradient spin echo sequence used in diffusion MRI (dMRI).	15

4.1	Schematic overview of the fibre growth algorithm. A fibre grows sequentially, moving from one node to the next, starting from the start point (top left, green node) toward the target (top left, blue node) along the edges defined by the Delaunay triangulation. Inset: The algorithm determining which node a fibre steps to at any given iteration. a) The possible nodes to step to are those which share an edge with the current node. b) From the edges available costs are calculated using eqs. (4.1) and (4.2). c) The fibre will grow along the edge with the lowest cost. d) From this new segment, the maximum diameter sustainable at a given node is calculated, giving each node a cost based on the maximum sustainable diameter. This cost will then be used in the calculation of edge weights (b) for future fibres. Note that although this figure illustrates the algorithm in 2D, in practice the algorithm grows fibres in 3D.	34
4.2	Simple example of metaball interactions. a) With two positive metaballs, the fields combine to attract the surfaces together. This is used to join individual segments into a continuous fibre. b) With one negative metaball (indicated by the flat grey circle) the surface of the metaball is repelled from the negative metaball. This is used to deform nearby fibres around one another.	37
4.3	Fibres generated using uniform grid (orange) and pseudo-random (blue) network nodes for increasing numbers of nodes.	37
4.4	a) Example substrates (cut into $30 \times 30 \times 30 \mu\text{m}^3$ cube) from the fibre growth algorithm, left to right: Zero macroscopic dispersion (60% density), 15° of macroscopic dispersion (30% density), 35° dispersed (25% density). b) Simulations for each substrate for varying permeabilities with $\text{SNR} = \infty$ and c) $\text{SNR} = 20$. Units of κ are $\mu\text{m}/\text{ms}$	39
4.5	Timing of brute force growth vs. the fibre growth algorithm along with a quadratic fit (brute-force) and linear fit (fibre growth algorithm). The fibre growth algorithm is clearly linear in the number of fibres, while brute force growth fits an order n^2 well.	40
5.1	Simple lo-res cuboid used for testing CUDAmino. Cuboid dimensions are $2 \mu\text{m} \times 5 \mu\text{m} \times 10 \mu\text{m}$ and it has 12 triangular faces.	44
5.2	CUDAmino simulations (dotted lines) versus analytic solutions (solid black lines) for diffusion in the three directions normal to faces of the cuboid in Figure 5.1. a is the separation between the faces in each direction. The simulated signals match the analytic signal well, with more deviation at higher b -value in general.	45
5.3	CUDAmino simulations (dotted lines) versus analytical solutions (solid black lines) simulations within the cuboid in Figure 5.1 varying the number of spins, n	47
6.1	Timetable of planned future work	52

List of Algorithms

2.1	Basic algorithm for taking a step in the random walk.	19
4.1	Fibre growing algorithm. Takes desired morphological priors (OD, ρ and d_0) as well as desired number of nodes in the growth network. From these initial fibre positions, targets and the growth network are generated before the main loop of the algorithm in which each fibre grows one-by-one. . .	32

List of Acronyms

MRI Magnetic resonance imaging

MRS Magnetic resonance spectroscopy

NMR Nuclear magnetic resonance

dMRI Diffusion MRI

WM White matter

RF Radiofrequency

FID Free induction decay

MC Monte-Carlo

BT Bloch-Torrey

PDE Partial differential equation

FDM Finite difference method

FEM Finite element method

CPU Central processing unit

GPU Graphics processing unit

ConFiG Contextual fibre growth

PGSE Pulsed gradient spin echo

SGP Short gradient pulse

CUDA Compute Unified Device Architecture

API Application program interface

BVH Bounding volume hierarchy

TBI Traumatic brain injury

ADC Apparent diffusion coefficient

OGSE Oscillating gradient spin echo

RWS Random Walk Simulator

DMS Diffusion Microscopist Simulator

FA Fractional anisotropy

TE Echo time

MS Multiple sclerosis

NN Neural network

1 Introduction

Chapter Contents

1.1	Motivation	1
1.2	Problem Statement	3
1.3	Project Aims and Scope	3
1.4	Report Overview	3

Introduction

The purpose of this chapter is to introduce the main motivations behind this project, from the historical relevance of diffusion MRI to the use of modern simulations for validating new diffusion MRI models. This chapter also sets out the problem that this project aims to address and the specific aims for addressing various aspects of it.

1.1 Motivation

In 1992, James Watson, co-discoverer of DNA, said “The brain is the last and grandest biological frontier, the most complex thing we have yet discovered in our universe.”[1]. One year later, Francis Crick, fellow discoverer of DNA, and Edward Jones published a commentary in *Nature* lamenting how little was understood about human neuroanatomy, saying that new techniques were needed beyond the contemporary tracer studies in non-human primates [2].¹

Just one year after that, in 1994, Basser et al. [4] showed that it was possible to use magnetic resonance imaging (MRI) to measure the movement of water along axons, providing the basis for exactly the kind of new technique Crick and Jones had felt was needed. This technique of using MRI for measuring the movement of water molecules is known as diffusion MRI (dMRI).

¹This introduction was inspired by Richard Passingham’s very nice foreword to *Diffusion MRI* by Berg and Behrens [3]

1 Introduction

In the 25 years since the work of Basser et al. the field of diffusion MRI has grown into a major topic of MRI research, generating thousands of publications per year. Diffusion MRI has found extensive use for imaging the brain, generating new techniques such as tractography which attempts to map out the connections in the brain *in vivo*. Another technique which takes advantage of dMRI is microstructure imaging, in which measurements of the diffusion of water in tissue is used to infer information about the structure of the tissue.

These techniques work because dMRI sensitises the MRI signal to the diffusive motion of water molecules. The environment in which the water molecules move will restrict the motion of the molecules and so will affect the MRI signal. This dependency of the dMRI signal on the environment in which water molecules diffuse can be exploited to infer information about the environment solely from the dMRI signal. Microstructure imaging attempts to do exactly this, infer information about the microstructural environment such as cell size and density from the dMRI signal. In order to infer meaningful information from the dMRI signal, models are typically used which relate microstructural features to the dMRI signal.

The validation of these microstructural models can be difficult since ground truth microstructural features are typically inaccessible *in vivo* and classical histological validation techniques have limitations such as disruption due to tissue extraction and preparation. One approach commonly taken for the validation of new models is simulation of the dMRI signal using well defined and controllable ground truth microstructural environments known as numerical phantoms.

Whilst these numerical phantoms often provide a valuable ground truth for simulation, they typically over simplify the complex microstructure of real tissue. An example of this is in white matter (WM), where fibres are commonly represented as straight cylinders[5, 6] whereas in real tissue, fibres have complex shapes with undulation and diameter variation[7, 8].

In order to validate models in scenarios that are as close to *in vivo* conditions, the realism of numerical phantoms should be improved, attempting to fully capture all of the microstructural variation found in *in vivo*.

Another application of numerical phantoms which is growing in popularity is in the direct computational modelling of microstructure. These techniques use machine learning or fingerprinting-style techniques to match simulated signals and the corresponding ground truth microstructure of the numerical phantom to the measured signal[9, 10, 11, 12]. Again, in this case, the more realistic the numerical phantoms used for simulation, the more accurately microstructural features can be estimated.

With increased complexity of numerical phantoms, however, the computational cost of dMRI simulations grows. Complex numerical phantoms are typically represented by polygon meshes and as the number of faces in a mesh grows, the computational cost of simulating the diffusion process grows.

In order to attempt to account for some of this increased computational complexity, adapting the dMRI simulations for the graphics processing unit (GPU) can provide a performance boost.

The GPU has a different architecture to the central processing unit (CPU), being designed for parallel processing of graphics data. Over the past decade, utilising the GPU for performing non-graphics calculations has become increasingly popular. In short, the

goal is to exploit the parallel design of the GPU to multiple calculations simultaneously. There are certain limitations on the kind of tasks that can be improved by the GPU, for instance, as a minimum the problem must be parallelisable. dMRI simulations are inherently parallel, meaning that adapting the simulations for GPU should be able to bring a performance increase.

1.2 Problem Statement

There is a need to be able to generate numerical phantoms that realistically represent WM microstructure, with controllable microstructural parameters such as axon packing density and orientation dispersion, and to improve the efficiency of dMRI simulations.

1.3 Project Aims and Scope

This report summarises work towards improving the realism of simulations of diffusion MRI. Realistic simulations allow models of the MRI signal to be validated using controllable and well known ground truth.

The mains aims of this work are as follows:

1. Develop and test a method for generating realistic WM numerical phantoms with realistic axon packing densities and orientation dispersion
 2. Develop and test a GPU accelerated dMRI simulator

In an effort to achieve aim 1, a method called ConFiG (**C**ontextual **F**ibre **G**rowth) is presented which ‘grows’ fibres densely attempting to respect some morphological priors to generate realistic WM numerical phantoms.

Additionally, to tackle aim 2, CUDAmino, a GPU implementation of Monte-Carlo (MC) dMRI simulations is presented. The GPU simulations are based on the MC simulator software in Camino [13, 5] and leverage NVIDIA’s CUDA parallel programming platform [14].

1.4 Report Overview

The rest of the report is arranged as follows: Chapter 2 outlines some of the physics behind diffusion MRI and how we simulate it as well as some background on GPU architecture and considerations for programming the GPU. Chapter 3 reviews current literature on dMRI simulations (including GPU dMRI simulations) and numerical phantom generation. Chapter 4 outlines ConFiG for generating new phantoms and some experiments assessing the performance of ConFiG in some simple situations and presenting examples of realistic substrates. Chapter 5 presents CUDAmino and some experiments assessing its performance against the CPU simulations. Chapter 6 discusses future perspectives for the work.

1 Introduction



2 Background

Chapter Contents

2.1	MR Physics	5
2.2	Diffusion MRI	14
2.3	Diffusion Simulation	18
2.4	Graphics Processing Units	20

Introduction

This chapter introduces the physics behind dMRI as well as some techniques used for the simulation of dMRI. Firstly, general MR physics is introduced building up from a single proton to the generation of the spin echo signal. The second section discusses how the diffusion of water molecules impacts the MR signal, leading to the diffusion signal attenuation. The third section introduces some simulation approaches which can be used to generate synthetic dMRI data and the final section discusses the GPU and how it may be used to accelerate dMRI simulations.

2.1 MR Physics

All forms of *in vivo* use of magnetic resonance have their origins in the 1940s when Purcell, Torrey and Pound independently and almost simultaneously with Bloch, Hansen and Packard detected radio frequency signals from nuclei in ordinary matter[15, 16, 17, 18]. This discovery gave birth to the field of nuclear magnetic resonance (NMR) which has become widespread, with applications in a number of areas including chemistry, biology, materials science and medical imaging[16, 19].

2 Background

Within the medical imaging context, NMR typically finds two uses, magnetic resonance imaging (MRI) and magnetic resonance spectroscopy (MRS)¹. Both of these uses are closely related: MRI is typically concerned with building images of internal structures in the body, whilst MRS is concerned with identifying the chemical composition of tissues in the body.

The theory behind NMR concerns the interaction between nuclei and magnetic fields. This section briefly introduces the NMR physics relevant to dMRI.

2.1.1 Nuclear Magnetism

The most important property of a nucleus for the application of NMR is nuclear spin. Spin is a property inherent to all subatomic particles and whilst it is a purely quantum effect, it can be thought of loosely as the particle spinning around its axis - much like a tiny planet [15]. A planet spinning about its axis will have an angular momentum associated with that rotation and similarly the spin of a particle behaves like an angular momentum. Unlike the angular momentum of a rotating planet, however, the spin is an intrinsic property of the particle itself and not a result of its motion[15].

MRI typically relies on NMR of ¹H nuclei in water molecules, which simply consist of a single proton. A proton carries a positive electric charge. Just as classically a rotating charge with angular momentum, \mathbf{L} , will produce a magnetic moment, $\mu = \gamma\mathbf{L}$, the intrinsic spin angular momentum, \mathbf{S} of a proton will produce a magnetic moment

$$\mu = \gamma\mathbf{S}, \quad (2.1)$$

where γ is the gyromagnetic ratio [15]. For a proton, $\gamma = 2.675 \times 10^8 \text{ rad s}^{-1} \text{ T}^{-1}$.

The fact that a proton has an intrinsic magnetic moment means that it will interact with magnetic fields and it is understanding this interaction that underpins NMR theory.

2.1.2 Magnetic Resonance

Classically, a magnetic moment, μ , placed in an external magnetic field, \mathbf{B}_0 , will feel a torque, τ , given by [20]

$$\tau = \mu \times \mathbf{B}_0. \quad (2.2)$$

At the same time, classical mechanics gives a relationship between the change in angular momentum and the torque as [20]

$$\frac{d\mathbf{L}}{dt} = \tau. \quad (2.3)$$

For a proton in its rest state, the only angular momentum is the intrinsic spin angular momentum, \mathbf{S} , so combining Equations (2.2) and (2.3) gives the equation of motion for a spin in an external magnetic field

$$\frac{d\mathbf{S}}{dt} = \mu \times \mathbf{B}_0. \quad (2.4)$$

¹The ‘N’ from NMR is dropped in the medical imaging context to avoid confusion with nuclear medicine and general squeamishness around the word nuclear

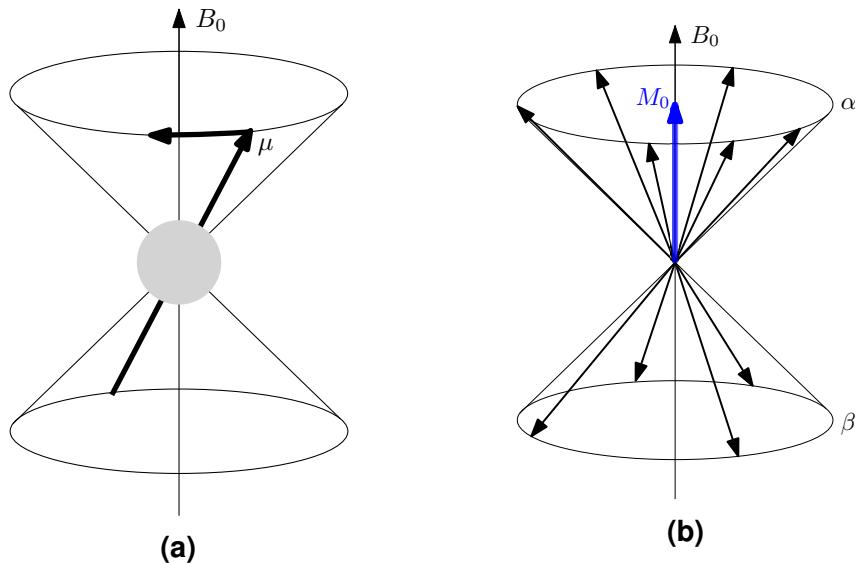


Figure 2.1: a) Illustration of the precessional motion of a single spin with magnetic moment μ in the presence of an external magnetic field B_0 . b) Many individual spins in an external magnetic field precess around the external field with random phase producing a net magnetisation in the direction of the B_0 field.

Since S is equivalent to μ/γ , this becomes

$$\frac{d\mu}{dt} = \gamma\mu \times B_0 . \quad (2.5)$$

This equation of motion can be solved in a few ways for the case of constant external magnetic field, with the result being that the magnetic moment precesses about the magnetic field with a frequency, ω_0 , given by[15]

$$\omega_0 = \gamma B_0 , \quad (2.6)$$

with B_0 being the external field strength. This precessional motion is illustrated in Figure 2.1a. The frequency ω_0 is known as the Larmor frequency and lies in the radiofrequency (RF) range for typical field strengths found in MRI machines (1.5 - 7 T).

In practice, it is not possible to observe the magnetic moment of a single spin. The quantity observed is rather the sum of the magnetic moments from many spins together, this is known as the net magnetisation.

In a sample, slightly more protons will align with the B_0 field than against it, meaning that the net magnetisation will be parallel to B_0 . Figure 2.1b shows a pictorial representation of the system of many spins producing a net magnetisation, M_0 aligned with B_0 .

Each of the spins will still be precessing about the magnetic field at the Larmor frequency but since they are out of phase with one another, all transverse components of the magnetisation cancel out when they combine and all that is left is a static longitudinal component.



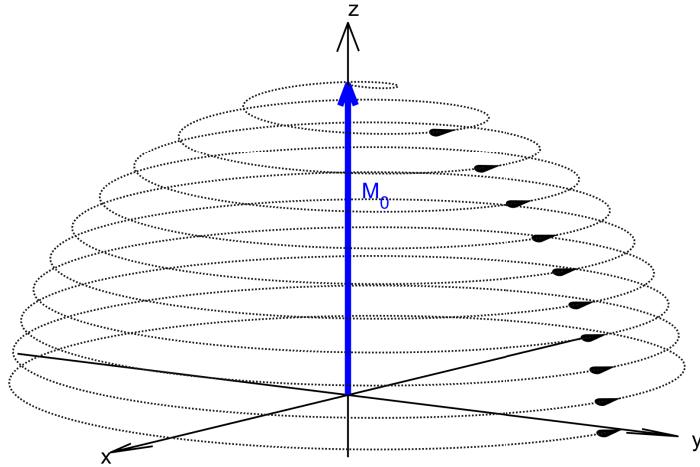


Figure 2.2: Nutation motion of an on-resonance spin in the presence of an RF field. Precession about the B_0 and B_1 fields create the spiralling motion in the laboratory frame.

In order to make measurements, the net longitudinal magnetisation needs to be ‘flipped’ into the transverse plane where it can be detected. This is achieved by applying a second magnetic field, B_1 , oscillating in the transverse plane. In much the same way as with B_0 , the magnetisation feels a torque from B_1 and begins to rotate about B_1 , away from the longitudinal axis. The two external fields act simultaneously on M_0 so the magnetisation will tip away from the z axis whilst still precessing about z with a frequency ω_0 . This kind of motion is known as nutation and is illustrated in Figure 2.2.

2.1.3 The Bloch Equations

The interaction between the magnetisation and magnetic fields is described by the Bloch equations - an empirical set of equations describing the evolution of magnetisation introduced by Felix Bloch in 1946[17].

The magnetisation arises from a sum of independent magnetic moments, meaning that we could represent the magnetisation as

$$\mathbf{M} = \sum_i \boldsymbol{\mu}_i, \quad (2.7)$$

with i indicating a sum over all the spins in the sample. This definition for \mathbf{M} can be combined with the equation of motion for a single spin, Equation (2.5), to give [20]

$$\frac{d\mathbf{M}}{dt} = \gamma \mathbf{M} \times \mathbf{B}. \quad (2.8)$$

In the presence of the main external field, \mathbf{B}_0 , the magnetisation will be static and aligned along the z axis. The x and y components of the magnetisation will have random

orientations and precess about \mathbf{B}_0 at the Larmor frequency with a mean amplitude of zero. This will give the components of Equation (2.8) as[21]

$$\frac{d\mathbf{M}_x(t)}{dt} = \gamma \mathbf{M}_y \mathbf{B}_0, \quad (2.9)$$

$$\frac{d\mathbf{M}_y(t)}{dt} = -\gamma \mathbf{M}_x \mathbf{B}_0, \quad (2.10)$$

$$\frac{d\mathbf{M}_z(t)}{dt} = 0. \quad (2.11)$$

To understand the interaction of the magnetisation with the \mathbf{B}_1 field, the oscillation of the field in the transverse plane needs to be described. Usually, the \mathbf{B}_1 field is circularly polarised to oscillate in the transverse plane so that the field can be described as

$$\mathbf{B}_1(t) = B_1 \cos(\omega t) \mathbf{e}_x - B_1 \sin(\omega t) \mathbf{e}_y, \quad (2.12)$$

where \mathbf{e}_x and \mathbf{e}_y are unit vectors in the x and y directions respectively.

The combined effect of the \mathbf{B}_0 and \mathbf{B}_1 fields can be seen from Equation (2.8) to get [21]

$$\frac{d\mathbf{M}_x(t)}{dt} = \gamma (\mathbf{M}_y(t)\mathbf{B}_0 - \mathbf{M}_z(t)\mathbf{B}_1 \sin(\omega t)), \quad (2.13)$$

$$\frac{d\mathbf{M}_y(t)}{dt} = \gamma (\mathbf{M}_z(t)\mathbf{B}_1 \cos(\omega t) - \mathbf{M}_x(t)\mathbf{B}_0), \quad (2.14)$$

$$\frac{d\mathbf{M}_z(t)}{dt} = \gamma (\mathbf{M}_x(t)\mathbf{B}_1 \sin(\omega t) - \mathbf{M}_y(t)\mathbf{B}_1 \cos(\omega t)). \quad (2.15)$$

These are the equations of motion of the magnetisation in the laboratory frame under the influence of the \mathbf{B}_0 and \mathbf{B}_1 and describe the kind of motion seen in Figure 2.2.

Relaxation

In order to get to the full Bloch Equations the concept of relaxation must be introduced. Relaxation is a term used to describe the way in which a spin system will return to equilibrium after being perturbed. The components of \mathbf{M} that are parallel to the \mathbf{B}_0 magnetic field relax differently to those perpendicular to the magnetic field leading to two relaxation terms being introduced into Equations (2.13) to (2.15).

The relaxation processes are exponential and described by two time constants, T_1 and T_2 . T_1 is the longitudinal relaxation time and describes the rate at which longitudinal magnetisation regrows after a perturbation. T_2 is the transverse relaxation time and describes the rate at which transverse magnetisation decays after a perturbation. T_2 is always shorter than T_1 since all the effects which contribute to T_1 contribute to T_2 relaxation, however T_2 relaxation is also affected by the spins going out of phase with one another. The relaxation process can be written as [21]

$$\frac{d\mathbf{M}_x(t)}{dt} = -\frac{\mathbf{M}_x(t)}{T_2}, \quad (2.16)$$

$$\frac{d\mathbf{M}_y(t)}{dt} = -\frac{\mathbf{M}_y(t)}{T_2}, \quad (2.17)$$

$$\frac{d\mathbf{M}_z(t)}{dt} = -\frac{\mathbf{M}_z(t) - \mathbf{M}_0}{T_1}. \quad (2.18)$$

2 Background

Combining Equations (2.13) to (2.15) and Equations (2.16) to (2.18) gives the full Bloch equations

$$\frac{d\mathbf{M}_x(t)}{dt} = \gamma (\mathbf{M}_y(t)\mathbf{B}_0 - \mathbf{M}_z(t)\mathbf{B}_1 \sin(\omega t)) - \frac{\mathbf{M}_x(t)}{T_2}, \quad (2.19)$$

$$\frac{d\mathbf{M}_y(t)}{dt} = \gamma (\mathbf{M}_z(t)\mathbf{B}_1 \cos(\omega t) - \mathbf{M}_x(t)\mathbf{B}_0) - \frac{\mathbf{M}_y(t)}{T_2}, \quad (2.20)$$

$$\frac{d\mathbf{M}_z(t)}{dt} = \gamma (\mathbf{M}_x(t)\mathbf{B}_1 \sin(\omega t) - \mathbf{M}_y(t)\mathbf{B}_1 \cos(\omega t)) - \frac{\mathbf{M}_z(t) - \mathbf{M}_0}{T_1}. \quad (2.21)$$

T_2 is used to refer to relaxation due to intrinsic spin-spin interactions which cause spins to accrue phase relative to one another and thus the magnitude of the net transverse magnetisation is reduced when taking the sum in Equation (2.7). Other effects can also contribute to the loss of transverse magnetisation, such as magnetic field inhomogeneities which can add to the T_2 relaxation. This is referred to as T_2^* , with

$$\frac{1}{T_2^*} = \frac{1}{T_2} + \frac{1}{T'_2}, \quad (2.22)$$

where T'_2 is the relaxation time associated with these external sources. The T'_2 effect can be negated using special MR pulse sequences which will be covered in Section 2.1.5, however the intrinsic T_2 relaxation cannot be avoided.

The Rotating Frame

To this point, everything has been described in a static Cartesian frame known as the laboratory frame. The lab frame is not the most convenient reference frame to analyse the NMR experiment in, however. Moving to a frame which is rotating about \mathbf{B}_0 (i.e. the z -axis) at a frequency ω matching the \mathbf{B}_1 field oscillation simplifies the maths of the system. The axes of this frame will be referred to as x' , y' and z' .

The components of the magnetisation in the rotating frame can be calculated from the lab frame components as [21]

$$\mathbf{M}'_x = \mathbf{M}_x \cos(\omega t) - \mathbf{M}_y \sin(\omega t), \quad (2.23)$$

$$\mathbf{M}'_y = \mathbf{M}_x \sin(\omega t) - \mathbf{M}_y \cos(\omega t), \quad (2.24)$$

$$\mathbf{M}'_z = \mathbf{M}_z. \quad (2.25)$$

The rotating frame Bloch equations can be calculated by combining these rotating frame magnetisation components with the lab frame Bloch equations[21]

$$\frac{d\mathbf{M}'_x(t)}{dt} = \Omega \mathbf{M}'_y(t) - \frac{\mathbf{M}'_x(t)}{T_2}, \quad (2.26)$$

$$\frac{d\mathbf{M}'_y(t)}{dt} = -\Omega \mathbf{M}'_x(t) + \gamma \mathbf{B}_1 \mathbf{M}'_z(t) - \frac{\mathbf{M}'_y(t)}{T_2}, \quad (2.27)$$

$$\frac{d\mathbf{M}'_z(t)}{dt} = -\gamma \mathbf{B}_1 \mathbf{M}'_y(t) - \frac{\mathbf{M}'_z(t) - \mathbf{M}_0}{T_1}, \quad (2.28)$$

where $\Omega = \omega_0 - \omega$ is the offset frequency between the \mathbf{B}_1 field frequency and the Larmor frequency.



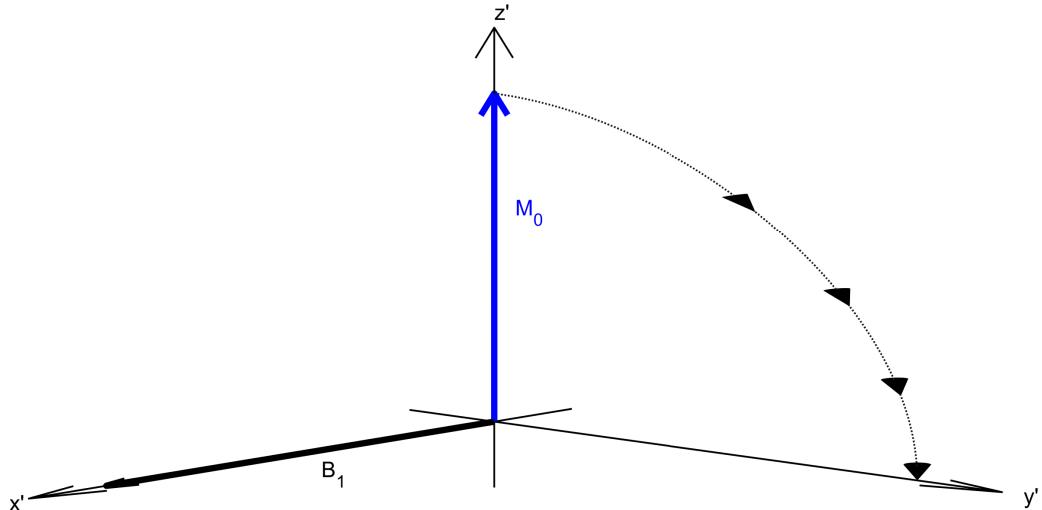


Figure 2.3: Motion of a spin in the presence of a B_1 RF field in the rotating frame. This is identical to the nutation in Figure 2.2, however viewing from the rotating frame simplifies the motion.

Since the frame is rotating with a frequency ω , the B_1 field appears static in the rotating frame. The precessional motion that is seen in the lab frame ($\omega_0 = \gamma B_0$) is reduced to a frequency Ω in the rotating frame. When $\Omega = 0$, meaning that B_1 oscillates at the Larmor frequency, the magnetisation simply precesses about the B_1 field towards the transverse plane as illustrated in Figure 2.3.

This situation is known as resonance - the frequency of the RF pulse matches the Larmor frequency, tipping the magnetisation away from the z' axis and into the transverse plane.

In the off-resonance case, an additional component of magnetic field with magnitude Ω/γ is produced in the z -direction. This results in an effective magnetic field, B_e , with a magnitude [21]

$$B_e = |B_e| = \sqrt{B_1^2 + \left(\frac{\Omega}{\gamma}\right)^2}. \quad (2.29)$$

The effective field is illustrated in Figure 2.4 with the additional component of Ω/γ resulting in an effective field that is no longer aligned with x' . Off-resonance effects can produce unwanted results meaning the spin does not get flipped as much as expected under an RF pulse which can result in signal losses.

2.1.4 Detecting the MR Signal

The detection and processing of NMR signals is a deep topic which could be the subject of its own book, however some very basic details of how a signal is formed are useful to go on from here.

2 Background

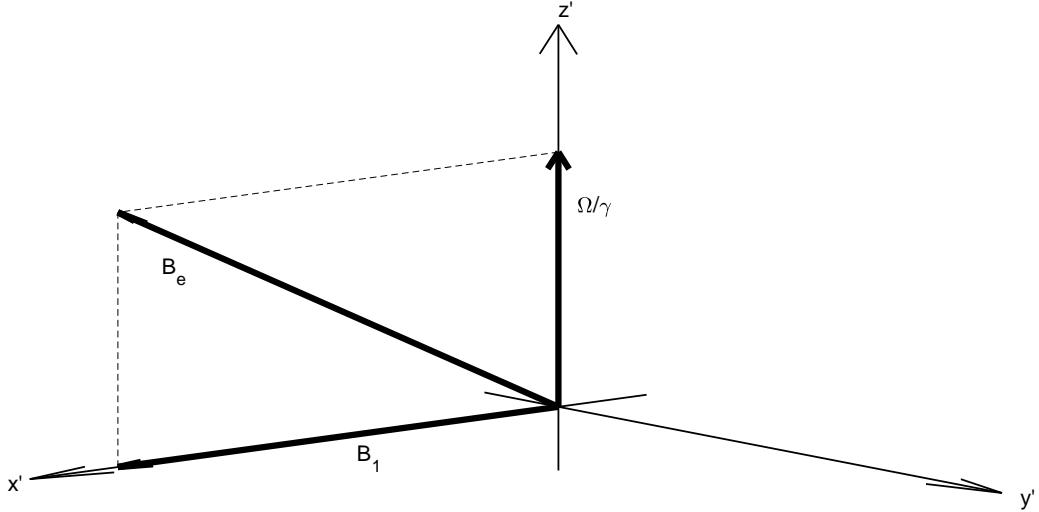


Figure 2.4: The effective field, B_e , produced due to an off-resonance frequency Ω . The off-resonance effects produce an additional component of magnetic field along the z' axis.

The reason for flipping the magnetisation into the transverse plane using B_1 fields is to make the magnetisation detectable. Transverse magnetisation precesses about B_0 at the Larmor frequency, sweeping its magnetic field around B_0 . A coil of wire placed near this precessing field will feel an electromotive force induced in it according to Faraday's Law of Induction[20].

Following a pulse that flips the magnetisation from M_0 aligned with z through an angle β towards x' , the x' -component of the magnetisation will be $M_0 \sin \beta$ and (ignoring relaxation) will then precess at the offset frequency, Ω , in the rotating frame. This will give the components of the magnetisation in the transverse plane over time as

$$M_x = M_0 \sin(\beta) \cos(\Omega t) \quad M_y = M_0 \sin(\beta) \sin(\Omega t). \quad (2.30)$$

The signal induced into the receiver coils is proportional to M_x and M_y and so the signal will also have an oscillating form similar to Equation (2.30). From the $\sin \beta$ term, it is clear that the maximum signal will arise when $\beta = 90^\circ$, meaning all the magnetisation is flipped into the transverse plane. Additionally, in a realistic experiment, there will be T_2^* relaxation so including this, the general form of the signal following a 90° pulse will be[21]

$$S_x = S_0 \cos(\Omega t) \exp(-t/T_2^*) \quad S_y = S_0 \sin(\Omega t) \exp(-t/T_2^*). \quad (2.31)$$

Generally NMR systems use something known as quadrature detection, meaning that both the x' and y' components of the magnetisation are measured simultaneously[15], giving the signal as a function of time as

$$\begin{aligned} S(t) &= S_x + iS_y, \\ &= S_0 \exp((i\Omega - 1/T_2^*)t). \end{aligned} \quad (2.32)$$



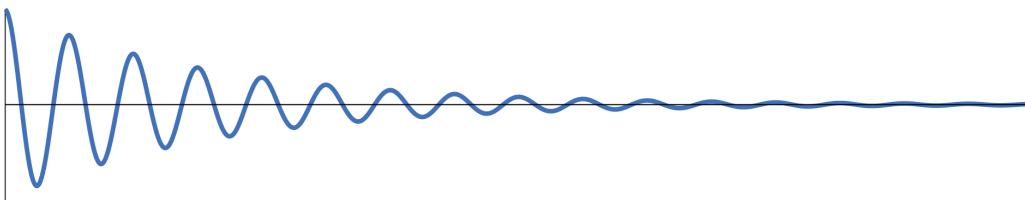


Figure 2.5: The free induction decay described by Equation (2.32). Here, just the real channel is plotted.

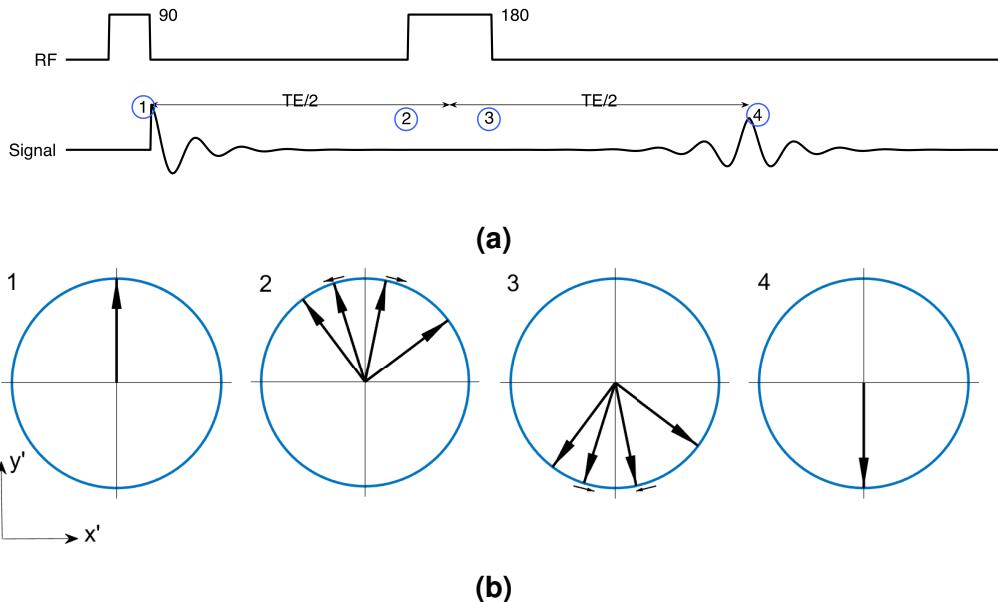


Figure 2.6: a) Spin echo sequence and (b) an indication of the evolution of spins under a spin echo sequence. This shows how the 180° refocusing pulse acts to refocus the spins after a time TE .

This time-domain signal is known as a free induction decay (FID) and has a typical form shown in Figure 2.5. If we neglect off-resonance effects, then the FID in the rotating frame will be a simple exponential decay.

$$S(t) = S_0 \exp(-t/T_2^*) \quad (2.33)$$

The FID is not commonly used for dMRI for a few reasons. Firstly, magnetic field gradients need to be introduced to make the signal sensitive to diffusion. Additionally, the T_2^* decay is often very rapid, so sequences known as spin echo sequences are used to remove the T_2' relaxation.

2.1.5 Spin Echoes

It is possible to undo the effects of T_2' by designing a pulse sequence to ‘refocus’ the spins, forming what is known as a spin echo. Spin echo sequences also enable localisation by combining RF pulses with magnetic field gradients.



2 Background

The first spin echo sequence was introduced by Edwin Hahn in 1950 [22]. The simplest sequence to form a spin echo consists of a 90° pulse to excite the spins followed by a 180° pulse after a delay. This sequence is shown in the diagram in Figure 2.6 along with how the signal varies during the pulse sequence.

Figure 2.6b represents how the magnetisation evolves through the pulse sequence with the four diagrams corresponding to the points marked in Figure 2.6a. At point 1, immediately following the 90° pulse, all the magnetisation has been flipped into the transverse plane and is in phase - meaning all the magnetic moments of the spins point in the same direction in the x-y plane.

The field inhomogeneities cause the different spins to feel slightly different magnetic fields and so precess at slightly different frequencies. This causes the spins to lose phase-coherence as indicated at point 2, and so the signal decays with T_2^* .

At point 3, following the 180° pulse, the spins remain out of phase with one another, but the 180° pulse has flipped their orientations across the x' -axis. The magnetic field the spins feel is still the same, so despite their flip in orientation, they still precess in the same direction. This means that the evolution that caused the spins to dephase begins to rewind and bring the spins back into phase coherence. After a time equal to the time between the 90° and 180° pulses, at point 4, the spins will be brought completely back in phase - or, refocused - and the spin echo is formed.

The signal at point 4 will still be lower in magnitude than that at point 1 since the T_2 relaxation will still occur as it is an inherent property of the matter. The spin echo does, however, refocus the B_0 inhomogeneities. The time between the 90° pulse and the formation of the echo is known as the echo time (TE).

The spin echo sequence shown in Figure 2.6 forms the basis of the standard pulsed gradient spin echo (PGSE) diffusion MRI sequence which is introduced in the following section, along with a description of the physics behind diffusion MRI.

2.2 Diffusion MRI

Diffusion MRI (dMRI) sensitises the MRI signal to the motion of water molecules due to diffusion. The following section describes the physics behind diffusion and how the diffusion impacts the MRI signal.

The diffusion process is driven by the Brownian motion of particles in fluids. The thermal kinetic energy of particles causes them to move around rapidly, however particles frequently collide with each other (for instance, molecules in water at room temperature experience around 60 billion collisions per second [23]) creating a very tortuous, random path.

Diffusion MRI sensitises the MR signal to this motion by exploiting the dephasing of spins as a result of magnetic field gradients.

The magnetic field will generally have a uniform component from the main B_0 field, and spatially and/or time varying components due to deliberate magnetic field gradients or typically unwanted effects such as magnetic susceptibility inhomogeneities and concomitant fields [20]. In general, $B(\mathbf{r}, t)$, the magnitude of the magnitude of the magnetic field at a position \mathbf{r} at time t is given by

$$B(\mathbf{r}, t) = |\mathbf{B}| = |B_0 \hat{\mathbf{z}} + \Delta \mathbf{B}(\mathbf{r}, t)|, \quad (2.34)$$

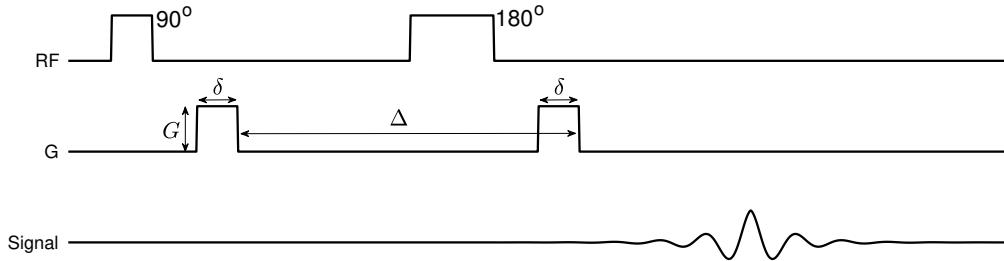


Figure 2.7: The standard pulsed gradient spin echo sequence used in dMRI.

where $\Delta\mathbf{B}(\mathbf{r}, t)$ accounts for all of the variation in the magnetic field away from B_0 . Note that $\Delta\mathbf{B}(\mathbf{r}, t)$ is a vector quantity which may have components in the \hat{x} and \hat{y} directions.

An idealised expression for $\Delta\mathbf{B}(\mathbf{r}, t)$ often applied to MRI assumes that all of the change in the magnetic field is due to an applied magnetic field gradient, $\mathbf{g}(\mathbf{r}, t)$, which only has a significant \hat{z} component. This means that Equation (2.34) can be written as

$$\begin{aligned} B(\mathbf{r}, t) &= |B_0\hat{\mathbf{z}} + (\mathbf{g}(\mathbf{r}, t) \cdot \mathbf{r})\hat{\mathbf{z}}|, \\ &= B_0 + \mathbf{g}(\mathbf{r}, t) \cdot \mathbf{r}(t). \end{aligned} \quad (2.35)$$

Magnetic field gradients introduce a deliberate variation in the magnetic field which, according to Equation (2.6), causes the Larmor frequency to vary spatially as well temporally.

Since the Larmor frequency varies spatially, spins in different locations will precess at different frequencies and accrue a phase shift relative to spins in different locations. The incremental phase, $d\phi$, accrued for a single spin, i , in an infinitesimal time, dt , is given by

$$d\phi_i = \gamma B(\mathbf{R}_i(t), t) dt, \quad (2.36)$$

where γ is the gyromagnetic ratio and $\mathbf{R}_i(t)$ is the position of the particle at time t .

Putting Equation (2.35) into Equation (2.36) and integrating over the time of the diffusion experiment will give the total phase accrued for a single spin:

$$\phi(\mathbf{R}_i(t)) = \gamma B_0 t + \gamma \int_0^t \mathbf{g}(\mathbf{R}_i(t'), t') \cdot \mathbf{R}_i(t') dt', \quad (2.37)$$

The first term in this equation is the phase accrued due to the main magnetic field which will be the same for all spins in the system. The second term is the phased accrued due to the gradient, which will be dependent on the motion of each individual spin. The dot product here indicates that only displacement projected onto the gradient direction affects the phase, allowing the gradient direction to be used to probe the diffusion in different directions.

The first diffusion MR sequence, introduced by Stejskal and Tanner in 1965[24], is the PGSE sequence, shown in Figure 2.7. The PGSE sequence consists of a standard spin echo sequence with a pair of gradient pulses added either side of the refocussing pulse. In the ideal case, each pulse is rectangular with a gradient strength, G , and duration, δ and they are separated by a time, Δ .

The effect of this pulse sequence can be simplified by considering the case when $\delta \ll \Delta$. This is known as the short gradient pulse (SGP) approximation and means that the motion of spins during the pulses can be ignored.

2 Background

Under the SGP approximation, the phase accrued by a spin at a position \mathbf{R}_0 during a pulse at a time t_0 will be

$$\phi(\mathbf{R}_0) = \gamma B_0 \delta + \gamma \delta \mathbf{g}(\mathbf{R}_0) \cdot \mathbf{R}_0 . \quad (2.38)$$

Here, \mathbf{R}_0 refers to the spin position at time $t = t_0$ and is not time-dependent, because under the SGP approximation the motion of spins during the pulse is assumed to be negligible.

The 180° pulse in the PGSE sequence is crucial, since it flips the orientation of the spins' magnetic moments as in the spin echo sequence in Figure 2.6. Not only does this refocus T_2^* effects as outlined in Section 2.1.5, but it means that the phase accrued during the second gradient pulse has opposite sign to the phase accrued during the first spin.

A spin which is at a position \mathbf{R}_0 during the first pulse and then diffuses to a position \mathbf{R}_1 during the second pulse will therefore have a phase shift of

$$\Delta\phi(\mathbf{R}_1 - \mathbf{R}_0) = \gamma \delta \mathbf{g} \cdot (\mathbf{R}_1 - \mathbf{R}_0) . \quad (2.39)$$

The B_0 term from Equation (2.38) is the same for both gradient pulses, meaning that during the subtraction in Equation (2.39) it cancels and the only phase shift comes from the diffusive motion.

The total MR signal comes from an ensemble of spins, each with their own random Brownian motion and thus, from Equation (2.39), their own relative phase shift. To get to the total MR signal, we need to consider the probability that a particle starts at position \mathbf{R}_0 (i.e. the initial spin density, $\rho(\mathbf{R}_0)$) and the probability that a particle which starts at \mathbf{R}_0 moves to \mathbf{R}_1 during the time Δ , $P(\mathbf{R}_0, \mathbf{R}_1, \Delta)$. Putting these together, gives an expression for the total MR signal[25, 24]:

$$S(\mathbf{g}, \Delta) = S(\mathbf{0}, \Delta) \int \int \rho(\mathbf{R}_0) P(\mathbf{R}_0, \mathbf{R}_1, \Delta) e^{i \gamma \delta \mathbf{g} \cdot (\mathbf{R}_1 - \mathbf{R}_0)} d\mathbf{R}_0 d\mathbf{R}_1 . \quad (2.40)$$

This quantity, $P(\mathbf{R}_0, \mathbf{R}_1, \Delta)$, is known as the diffusion propagator and is of great interest for diffusion MRI. $P(\mathbf{R}_0, \mathbf{R}_1, \Delta)$ encodes the information about the environment in which the spins are diffusing. For diffusion in an isotropic, homogeneous medium, the diffusion propagator is a Gaussian distribution [25]:

$$P(\mathbf{R}_0, \mathbf{R}_1, t) = (4\pi Dt)^{-3/2} \exp\left(-\frac{(\mathbf{R}_1 - \mathbf{R}_0)^2}{4Dt}\right) . \quad (2.41)$$

In the case of Gaussian diffusion, Equation (2.40) can be solved analytically and will give an MR signal attenuation (that is, $S(t)/S(0)$) which is also Gaussian[24, 25]

$$E(g, \Delta) = \exp(-\gamma^2 g^2 \delta^2 D \Delta) . \quad (2.42)$$

The general form of this expression, accounting for finite duration gradient pulses, can also be analytically derived to give the Stejskal-Tanner equation [24, 26]

$$\ln(E) = -\gamma^2 g^2 \delta^2 D(\Delta - \delta/3) , \quad (2.43)$$

$$= -bD , \quad (2.44)$$



where $b = \gamma^2 g^2 \delta^2 (\Delta - \delta/3)$ is the so-called b -value which describes the strength of the diffusion encoding.

In general, we can formulate an equation similar to Equation (2.40) without requiring any assumptions on the gradients as [25, 5]

$$S(\mathbf{g}, t) = S(\mathbf{0}, t) \int_{-\infty}^{\infty} P(\phi, t) e^{i\phi} d\phi, \quad (2.45)$$

where the phase, ϕ will be given by Equation (2.37) and $P(\phi, t)$ is the probability density function of the phase distribution after a time t .

In the case of restricted diffusion (i.e. diffusion in an inhomogeneous or anisotropic environment) the form of the diffusion propagator becomes more complex and closed form solutions of Equations (2.40) and (2.45) are only possible for certain simple geometries and assumptions.

Analytical solutions can be found for some simple restricting geometries such as spheres, cylinders and parallel plates [27, 28, 29]. For more complex environments, however, an analytical solution is intractable and we must rely on simulations to approximate the diffusion MRI signal.

2.2.1 The Bloch-Torrey Equations

As well as describing the dMRI signal considering the microscopic diffusion of spins, a macroscopic formulation can be derived considering Fick's laws of diffusion. The combination of the Bloch equations (Equations (2.26) to (2.28)) with Fick's second law of diffusion leads to the Bloch-Torrey (BT) equations, proposed by H. C. Torrey in 1956[30, 31, 32]:

$$\frac{\partial \mathbf{M}(\mathbf{r}, t)}{\partial t} = \gamma \mathbf{M} \times \mathbf{B}(\mathbf{r}, t) - \frac{M_x \hat{\mathbf{x}} + M_y \hat{\mathbf{y}}}{T_2} - \frac{(M_z - M_0) \hat{\mathbf{z}}}{T_1} + \nabla \cdot (\mathbf{D} \nabla \mathbf{M}). \quad (2.46)$$

This version of the BT equation is sometimes referred to as the standard BT equation. An additional term can be added to account for the evolution of the magnetisation due to a flow, described by a velocity field \mathbf{v} to get the generalised BT equation [33]. This flow term is often dropped, assuming no net flow, in the application to dMRI, leaving Equation (2.46). \mathbf{D} is the diffusion tensor, a generalisation of the diffusion coefficient, D , to allow for anisotropic diffusion. In short, this means that diffusion happens at a different rate in different directions. It is the \mathbf{D} term which encodes the evolution of the magnetisation when there is diffusion.

In general, the BT equations cannot be solved analytically, apart from in some simple cases such as isotropic free diffusion. For instance, the solution to the BT equations in isotropic free diffusion can be shown to give the expected Stejskal-Tanner equation, Equation (2.43).

For complex geometries, as with the case above, we must rely on computational methods to come to a solution for the BT equations.



2.3 Diffusion Simulation

Diffusion simulations attempt to evaluate Equation (2.45) or Equation (2.45) computationally. Simulation approaches broadly fall into two categories: numerical solutions of the BT equation and MC simulations of the diffusion dynamics. This section introduces these techniques, highlighting some of the similarities and differences between them.

At a high level, all diffusion simulations have three common components: the substrate, the diffusion dynamics and the measurement. The substrate describes the environment in which the diffusion is taking place. A common example of this is parallel cylinders representing axons in white matter. The diffusion dynamics describe our understanding of the processes underlying the diffusive motion of molecules and the measurement describes how this diffusive motion results in a synthetic dMRI signal.

2.3.1 Numerical Solutions

Numerical solution approaches generally attempt to solve the Bloch-Torrey equation [30]. In literature, the BT equation excluding the relaxation and flow terms is sometimes referred to as the standard BT equation and the full BT equation in Equation (2.46) is referred to as the generalised BT equation [34].

Numerical solution approaches combine both the dynamics and the measurement components of the diffusion simulation by solving for the magnetisation in Equation (2.46). The third component, the substrate, defines boundary conditions required for the solution of the equation.

There are two typical methods for solving the partial differential equation (PDE) in Equation (2.46), finite difference methods (FDMs) and finite element methods (FEMs). Finite difference methods evaluate the PDE using a local Taylor expansion at discrete points which are generally uniformly separated in each spatial as well as the temporal dimension [35]. FDMs are an efficient method for solving PDEs when the problem can fit into a rectangular grid, however they are less effective when applied to complex geometries [36, 35].

Finite element methods subdivide the domain into small elements which are simple geometric shapes, though unlike the FDMs, they do not have to form a regular grid, but rather an arbitrary mesh. In each element, the PDE solution is approximated by simple functions such as a linear combination of polynomials. The combination of all of these local approximations can be solved to give a numerical solution of the PDE across the whole domain [37]. FEMs are generally more complex to formulate and implement than FDMs, however the added complexity can be worth the effort for more difficult problems in which FDM may be ineffective [38].

2.3.2 Monte-Carlo Simulations

Monte-Carlo methods take a different approach to the simulation of the dMRI signal. Monte-Carlo methods simulate the Brownian motion of a large number of particles, simulating the motion of each particle individually, along with the MR acquisition to generate the dMRI signal.



There exist many different implementations of the Monte-Carlo simulation of dMRI [39, 40, 41, 42, 5, 43, 44], however the underlying principles are similar for all of them. The following is a general description of the MC simulation process, however the specifics for each different implementation may vary.

Most early MC studies used simple, easily parametrised substrates like regularly packed cuboids [44] or cylinders [43]. As computational power has increased, so too has the capacity for more and more complex substrates. This includes cylinders with randomly distributed radii [5], undulating cylinders [40], beading cylinders [45] and meshes generated from high resolution microscopy of tissue [46].

The Brownian motion of particles is typically simulated as a random walk of many independent particles. The time domain is discretised into many time points and at each time point each particle takes a random step through the substrate. One step of the random walk can be briefly summarised as shown in Algorithm 2.1.

Algorithm 2.1 Basic algorithm for taking a step in the random walk.

```

generate randomly oriented step vector
check if step crosses a barrier
while step crosses barrier do
    amend step according to barrier interaction (e.g. elastic reflection)
    repeat barrier checking on amended step
update the particle position

```

Following the Brownian motion, each particle in the simulation will have taken many steps giving each particle a unique trajectory that it has traversed. The incremental phase, $\Delta\phi$, accrued at each step can be calculated from a discrete version of Equation (2.37). Under the assumption of uniform B_0 , only the gradient term matters, giving

$$\Delta\phi = \gamma \mathbf{g}(\mathbf{r}(t), t) \cdot \mathbf{r}(t) \Delta t, \quad (2.47)$$

where Δt is the duration of the step and $\mathbf{g}(\mathbf{r}(t), t)$ and $\mathbf{r}(t)$ are the gradient and particle position during that step respectively.

The phase accumulation in Equation (2.47) for each spin in the simulation can be combined with Equation (2.45) to approximate the total signal for the dMRI acquisition as

$$S = \sum_j e^{i\Phi_j}, \quad (2.48)$$

where Φ_j is the total phase accrued for each spin.

Monte-Carlo simulations are a powerful tool for dMRI simulation due to their ability to handle any arbitrary substrate and MR pulse sequence. Additionally, MC simulations can be modified to account for effects that are more difficult to formulate for analytical and numerical solutions of the diffusion equation such as semi-permeable membranes, membrane-particle interactions and spatially and/or temporally varying T_1 , T_2 and diffusivities.

2 Background

A drawback of MC simulation, particularly for complex substrates is the need to simulate enough spins to mimic the ensemble behaviour of spins *in vivo* as well as enough discrete time points to adequately capture the dynamics through the pulse sequence. The huge number of calculations required to handle large simulations can be alleviated by exploiting the inherent parallel nature of the problem to run simulations in parallel on a CPU cluster or, even more effectively, a GPU cluster.

2.4 Graphics Processing Units

The graphics processing unit (GPU) is one of the main processing units in a modern computer, along with the central processing unit (CPU). The GPU was developed, as the name suggests, to handle graphics operations. Rendering an image to a screen requires a value to be updated for every pixel, and since this calculation is essentially the same for all pixels, the GPU was developed to be effective at performing many copies of the same operation simultaneously. A modern GPU may be able to run upwards of 50,000 operations at a time.

Conversely, the CPU was historically developed to be able to run one set of sequential instructions very effectively. Modern CPUs typically have a few (4 - 32) processing cores, each of which is designed to run sequential programs efficiently. This allows them to run multiple sets of instructions at a time, however the design philosophy is different to the GPU. The GPU is designed to run *the same* operation multiple times in parallel, the CPU is designed to be flexible to run a few different operations very effectively.

In 2007, NVIDIA released Compute Unified Device Architecture (CUDA) [14], a parallel computing platform and application program interface (API) allowing programmers to write software to use a CUDA-enabled GPU for general purpose computing. Since then, programming for the GPU has become increasingly popular with developers taking advantage of performance increases that can come from parallel programming across a wide range of fields, from physics simulations to deep learning.

Other vendors have also produced similar APIs to CUDA, most prominently OpenCL[47], however in this project CUDA was used due to its compatibility with the NVIDIA GPUs available.

The CUDA API enables software running on the CPU (known as the host) to launch what are called ‘kernels’ on the GPU (known as the device) as well as handling the memory transfers between the host and device. Kernels are the parallel functions which will run on the device.

When a kernel is launched it will generate a large number of threads to execute the function on the device, with each thread essentially performing one copy of the kernel function with different inputs. For instance, each thread could render one pixel to the screen for the graphics example.

If different inputs cause the threads to take a different path through the code (for instance, different branches at an if-else statement) this causes a problem known as *thread divergence*. Thread divergence reduces the performance of software since the GPU is designed for all threads to run in step with one another, so when one thread takes a different branch all of the other threads must wait for that thread to finish before they can continue.

Thread divergence and handling memory transfers, both between the host and device and between memory spaces on the device, are two of the difficulties that come with developing for the GPU and must be considered when designing algorithms. There are many more technical points to take into account when developing GPU software such as ensuring threads don't write to the same memory address at the same time and ensuring the data structures are optimal for parallel access. Chapter 5 will discuss how some of these problems were addressed in order to get MC dMRI simulations running on the GPU.



2 Background



3 Literature Review

Chapter Contents

3.1	Numerical Solutions	23
3.2	Monte-Carlo - Packages	24
3.3	Monte-Carlo - Numerical Phantoms	25
3.4	GPU accelerated MC simulations	28
3.5	Conclusions	28

Introduction

This chapter presents some examples of the use of numerical phantoms from literature. It represents a review of the contemporary literature in dMRI simulation, highlighting the state of the art and presenting some weakness which this project aims to address.

3.1 Numerical Solutions

Many of the studies presenting a numerical solution are focused on the validation of the technique and improvements to various algorithms rather than the direct use of the technique involving numerical phantoms.

One application of FDM solutions of the Bloch-Torrey equations is in the simulation of dMRI signals from histological images. For example, Chin et al. [48] simulate the signal from segmented histological images of mouse spinal cord white matter, showing that the fast and slow components of a biexponential decay of diffusion attenuation do not arise from a contribution from each of the intra and extracellular components. Hwang et al. [49] extend this technique to 3D, showing that their FDM solutions agree well with analytical solutions for hexagonally packed cylinders.

3 Literature Review

Xu et al. [50] develop a matrix based FDM, also testing their solution on hexagonally packed cylinders, showing a reduction in error compared to a conventional FDM. This FDM is used by the same group to investigate the sensitivity of dMRI to intracellular structure [51]. A numerical phantom of cells represented as densely packed spheres with spherical nuclei at the centre of each sphere. The FDM is used to show that an oscillating gradient spin echo (OGSE) sequence is more sensitive to changes in the nucleus size than a PGSE sequence. Similarly, Xu et al. [52] use histology based FDM simulations to investigate the efficacy of an axon diameter technique based on an OGSE sequence. They show that the OGSE technique is able to distinguish axons of a lower diameter than traditional PGSE techniques.

The first example of an FEM solution known to the author is presented by Hagslätt et al. [36]. In this study, rather than solving the BT equation, the FEM is used to solve for the diffusion propagator [31, 25]. From the diffusion propagator, the diffusion attenuated signal calculated from this based on an assumption of infinitely narrow gradient pulses. A good agreement is shown between simulation and theoretical solutions for a range of simple geometries (parallel plates, a lamellar system and hexagonally packed cylinders).

More studies have recently begun investigating diffusion simulation using FEMs. Moroney et al. [53] present an FEM solution of the standard BT equation (Equation (2.46) without the relaxtion and flow terms) for numerical analysis of dMRI experiments in the short gradient pulse limit. FEM results are compared to analytical solutions and MC simulations in simple geometries, showing that the FEM is more accurate than MC simulations, whilst taking less time to run.

Nguyen et al. [54] also present an FEM solution of the standard BT equation, showing its application to diffusion simulation with more general gradient waveforms. The FEM solution is shown to be more accurate in some simple geometries than a finite volume method, with second order accuracy in both the spatial and temporal domains. Three example applications to questions in dMRI are demonstrated using this FEM [54]. One shows that an infinitely thin membrane can be used to approximate a thick membrane. The second shows that the apparent diffusion coefficient (ADC) approaches the value predicted by mathematical homegenisation for long diffusion times. Finally, a model of a neuron is presented as a spherical body, with cylindrical axons and dendrites protruding. The ADC is shown to approach a steady state faster with a smaller neuronal body.

Beltrachini et al. [34] present a solution of the generalised BT equation, extending the FEM of Nguyen et al. [54] to include the relaxation and flow terms. This FEM improves on some of the restrictions in the FEM, making the simulations more stable through the use of an implicit scheme that is stable for coarser discretisations without compromising the validity of the result.

3.2 Monte-Carlo - Packages

Historically, most studies utilising MC simulation used in-house developed MC simulation software [55, 44, 56, 57], however in more recent years and as the complexity of situations possible to simulate has grown, a range of MC simulation packages have been released for public use.



Hall and Alexander [5] introduced MC simulation as part of the Camino diffusion MRI toolkit [13] in the context of simulating swelling cylinders as a model of the effect of ischaemic stroke, however the MC framework is very general and can be used to simulate any arbitrary from a 3D mesh.

Balls and Frank [42] present DiffSim, a dMRI simulation framework which embeds the MCell [58, 59, 60] cellular microphysiology simulator within an MRI simulator for synthesising the dMRI signal. DiffSim is used to simulate myelinated white matter [61], showing that an analytical solution model by Sen and Basser [62] holds for an **SPG!** (**SPG!**) approximation or long diffusion time, however with more realistic pulse sequence parameters, the numerical simulations show lower anisotropy than the analytical model.

Landman et al. [41] developed the DW-MRI Random Walk Simulator (RWS) showing, as an example of its flexibility and reproducibility, a range of geometrical models for white matter damage, including healthy straight cylinders, bulging cylinders, crimped cylinders and broken cylinders.

Yeh et al. [39] present Diffusion Microscopist Simulator (DMS), showing a range of diffusion substrates ranging from simple parallel uniform cylinders to more complex undulating, beading or crossing arrangements of fibres. A recent extension of DMS shows more complex white matter numerical phantom including orientation dispersion, tortuosity, beading and nodes of Ranvier [63], however the range of orientation dispersion and axon densities achieved does not reach typical *in vivo* values. As of the writing of this review, the DMS software package has not been publicly released.

3.3 Monte-Carlo - Numerical Phantoms

The above packages, as well as dMRI simulation software developed in-house in various research groups, have been used to investigate the diffusion signal in many different numerical phantoms.

A common target of microstructure imaging is the estimation of axonal diameter density. As mentioned above, FDM approaches have been used to investigate this [48, 52], whilst this has been the subject of MC simulation studies as well. Alexander et al. [64] use Camino to simulate a series of numerical phantoms of parallel cylinders with radii drawn from a Gamma distribution for the validation of a technique for orientationally invariant indices of axon diameter and density.

Recently Nilsson et al. [65], investigated the theoretical resolution limit for cylinder diameter estimation using diffusion MRI. Analytic expressions based on the Gaussian phase distribution approximation [25] were used for the intracellular signal and validated with MC simulations to determine a d_{min} , the diameter below which a cylinder cannot be differentiated from a cylinder with diameter approaching zero. The resolution limit for clinical scanners was found to be between 4 - 8 μm . This suggests a limitation on the level of microstructural detail that can be estimated using current clinical MRI machines.

Another problem commonly investigated using numerical phantoms is that of exchange between the intra and extracellular compartments of tissues. Permeability is difficult to control and vary in physical or biological phantoms, so numerical phantoms offer a unique tool with which to explore permeability and exchange models.



3 Literature Review

Nilsson et al. [66, 67] and Fieremans et al. [68] investigate the Kärger model [69], a model for exchange between two signal bearing compartments. These three studies all use similar numerical phantoms made of straight cylinders in which there is some probability that on encountering a barrier, the spin will pass through the barrier, exchanging spins between the compartments.

In their first study, Nilsson et al. [66] use simulations and experimental data to draw the conclusion that it is necessary to include exchange in a model containing two compartments, one of which is restricted. Fieremans et al. [68] show that the Kärger model is able to describe the signal for long diffusion times and sufficiently impermeable membranes, however at larger permeabilities, the Kärger model underestimates the value of the permeability. Nilsson et al. also investigate the effectiveness of the Kärger model at estimating the intracellular water fraction, showing that the Kärger model has a negative bias, underestimating the intracellular water fraction by up to 25% when there is high permeability when compared to a computational made by building a database of simulated signals [67].

Nilsson et al. [40] also investigate the importance of axonal undulation on diffusion MRI measurements. In this experiment, numerical phantoms consisting of axons with either sinusoidal or helical undulations were used in MC simulations to investigate the impact on a range of dMRI measured parameters. Nilsson et al. show that undulation affects essentially all of the parameters they tested derived from dMRI, for instance, undulation results in an overestimation of axonal diameter when using models that assume axons are straight [40].

Budde et al. [45] use MC simulations to investigate the effect of neurite beading, showing that beading is sufficient to explain the decrease in ADC after ischaemic stroke. Numerical phantoms consisting of straight cylinders with increasing amounts of beading introduced are simulated, showing a decrease in ADC in both the intra and extracellular spaces with increased beading.

Lin et al. [70] investigate the effect of traumatic brain injury (TBI) on acdMRI derived parameters. Using a numerical phantom consisting of straight cylinders representing axons, ellipsoids representing glial cells and the extracellular space, the effects of TBI are investigated by varying various parameters such as the size of the glial cells, the permeability of the cylinders and the spacing of the cylinders [70]. Using this technique, Lin et al. conclude that the inconsistencies amongst previous dMRI based TBI studies [71, 72, 73] are due to differences in the timing between the onset of TBI and the diffusion measurement, arguing that different processes drive the TBI at different timings, leading to different dMRI characteristics.

Lin et al. [74] similarly investigate the effect of myelin water exchange on various dMRI derived parameters. In this work, their representation of WM is slightly different, choosing nested cylinders to represent the intra-axonal and myelin-water compartments. Using MC simulations on these cylinders, they show correlations between dMRI derived parameters such as ADC and fractional anisotropy (FA).

Lam et al. [75] produce an empirical model of the extra axonal space using a series of MC simulations based on both regularly and randomly packed cylinders. The model is based on the diffusion spectrum [76], modelling diffusion in densely packed cylinders as diffusion in a series of pores with a small chance of exchange between the pores. The empirical model agrees closely with MC simulated data.



Some studies combine analytical solutions of the Bloch-Torrey equation and MC simulations. Rensonnet et al. [77] used this combined simulation to synthesise signals for parallel and crossing cylinders. The intracellular component is modelled using an analytical solution for diffusion within a cylinder based on Grebenkov's multiple correlation function approach [78]. The extracellular compartment, which is much more complex geometrically, is simulated using Monte-Carlo simulations. This hybrid approach yields simulation results which are indistinguishable from pure MC simulation whilst being quicker and more precise than a purely MC approach [77].

In a further study, Rensonnet et al. [79] use this approach to assess the validity of the superposition approximation of crossing fascicles (i.e. that the total signal from crossing fascicles is the sum of the signal from each fascicle independently). They are able to show that the signal differences between the superposition approximation and a full simulation of interwoven fascicles is small enough compared to typical noise levels in clinical dMRI data, that the superposition approximation is sufficient to describe the signal. A drawback to this hybrid approach is that the intra and extracellular compartments are treated as distinct, non-interacting compartments, meaning that membrane permeability is not accounted for.

An emerging application for dMRI simulations in the direct computational modelling of microstructure. The first example found for this type of modelling is actually the 2010 work by Nilsson et al. [67] mentioned above for evaluating the Kärger model. In recent years however, this idea has reemerged, partially thanks to the popularity of machine learning and emergence of MR fingerprinting.

One area of application these approaches have found is in the estimation of axonal permeability. Nedjati-Gilani et al. [12] use a machine learning technique known as random forest regression to learn the relationship between dMRI signal and microstructural parameters. They simulate diffusion using Camino in a range parallel cylinder substrates with different microstructural parameters including membrane permeability to build a dataset to train the random forest regression. The random forest regression is shown to estimate membrane permeability well, performing better than the Kärger model and an application to multiple sclerosis (MS) is presented, showing *in vivo* results consistent with pathology.

Palombo et al. [11] verify this method using a cuprizone treated *in vivo* mouse model. Cuprizone is a well known mouse model of WM demyelination, which is important as demyelination is hypothesised to affect axonal permeability. The random forest approach achieves accurate and robust estimation of microstructural parameters which match expected microstructure changes from electron microscopy and gains more specific information that typical dMRI measures such as ADC and FA.

Hill et al. [10] extend this work to use a deep neural network neural network (NN) in place of the random forest regression. They are able to show that the NN outperforms the random forest approach and an application in *in vivo* estimates microstructural parameters in the biologically plausible range.

Rensonnet et al. [9] also attempt to use computational models to directly estimate microstructural parameters using dMRI simulations. Using the hybrid approach mentioned above [77], they generate a dictionary of dMRI simulation signals for various combinations of microstructural parameters and crossing fascicles. Measured signals are then compared against this dictionary to find the entry that most closely matches the



3 Literature Review

measured signal to get an estimate of the microstructural parameters. They show that their approach achieves accurate and robust estimates of microstructural parameters and shows good correspondence with histology compared to traditional closed-form models when applied to an *in vivo* mouse model of spinal cord injury.

3.4 GPU accelerated MC simulations

There have been a couple of studies attempting to modify MC dMRI simulations for the GPU. The first, by Waudby and Christodoulou [80] implements a simple random walk on the GPU, with a rejection sampling scheme to handle substrate boundaries. In this case, a step crossing a boundary is ignored rather than reflected.

Unlike other geometries studied in this report, the Waudby and Christodoulou work uses a binary representation for simple shapes, meaning that the collision check is as straightforward as checking whether a spin moves into a region of space that is disallowed. One downside of this approach is that complex geometries cannot be easily represented in this binary manner.

Waudby and Christodoulou show that the GPU implementation is able to replicate unoptimised CPU and analytical solutions for simple geometries whilst achieving a $1000\times$ speedup over the CPU implementation. It is mentioned, that with CPU optimisation, this difference should reduce to around $20\times$.

A second, more recent, GPU accelerated dMRI simulation has been reported by Nguyen et al. [81]. This implementation handles arbitrary meshes with two approaches, one with an octree acceleration scheme and one following a similar binary approach to Waudby and Christodoulou. The octree scheme gives them a $45\text{-}65\times$ acceleration over Camino with the binary GPU version achieving a $2000\times$ acceleration. The binary representation in this work uses a uniform grid to divide the space. Each cell in the grid is assigned a 1 or a 0 based on whether or not it is inside the mesh and a step is considered to have collided with the mesh if it steps from a cell with a 1 to a cell with a 0.

This approach enables them to achieve a massive speedup in their simulation, however, it is limited by the resolution of the grid and the size of the smallest features in the mesh. Nguyen et al. show that if the resolution is not sufficient, the simulated signal will not accurately represent the signal from the true mesh. For relatively small substrates, this is not a problem as the grid can be made fine enough, however for large substrates, the memory requirements may become excessive.

Currently, the GPU dMRI simulator presented by Nguyen et al. is not publicly available.

3.5 Conclusions

There has been significant work into the use of numerical phantoms in the simulation of diffusion MRI signals. One thing which is common throughout most of the works is a simple model of WM. Typically, WM is represented as a set of densely packed, straight, parallel cylinders, which oversimplifies the underlying complexity of the microstructure. Lee et al. [8] use electron microscopy to reconstruct a 3D section of mouse corpus callosum, showing that real WM contains a large amount of microstructural complexity,



including orientation dispersion, undulation and diameter variations.

Generating WM substrates which accurately represent real microstructure is important for kinds of model validation studies mentioned in Section 3.3 because if a model is validated using overly simplified simulations, it may not be very robust when applied to real data.

Realistic WM representations are important for the kind of computational models mentioned in Section 3.3 for similar reasons. Another element which is particularly important for building the training datasets or dictionaries for these kinds of computational models is controllability. The WM numerical phantoms need to be realistic, but also must be able to be generated in a controlled manner so that we can generate substrates covering then relevant ranges of microstructural parameters needed to build robust computational models.

These are the main driving reasons behind the work presented in Chapter 4 which aims to build these realistic WM substrates with controllable morphology.

3 Literature Review



30 |

4 Contextual Fibre Growth (ConFiG)

Chapter Contents

4.1	ConFiG: Contextual Fibre Growth	31
4.2	Experiments and Results	37
4.3	Discussion and Conclusion	40

Introduction

This chapter introduces ConFiG, a method for generating white matter (WM) numerical phantoms with more realistic orientation dispersion and packing density. The algorithm behind ConFiG is introduced, describing how each white matter fibre is grown individually to generate a densely packed substrate.

Some experiments are presented, one of which explores how different inputs to ConFiG impacts of the shape of resulting fibres. An application of the method to dMRI is demonstrated with simulations of diffusion-weighted MR signal in three example substrates with differing orientation-dispersions, packing-densities and permeabilities.

4.1 ConFiG: Contextual Fibre Growth

In this section we describe ConFiG which grows fibres one-by-one avoiding intersection between fibres whilst attempting to ensure that the resulting substrate has desired morphological properties such as orientation dispersion, diameter distribution and packing density. The algorithm is broken into a few stages:

- the definition of inputs to the algorithm
- the generation of the network on which fibres grow
- the method by which each fibre grows
- the meshing procedure to create 3D meshes

Pseudocode for the first three of these points is shown in Algorithm 4.1. The rest of this section details each of the above stages.

4 Contextual Fibre Growth (ConFiG)

Algorithm 4.1 Fibre growing algorithm. Takes desired morphological priors (OD , ρ and d_0) as well as desired number of nodes in the growth network. From these initial fibre positions, targets and the growth network are generated before the main loop of the algorithm in which each fibre grows one-by-one.

```

procedure FIBREGROWTH( $OD$ ,  $\rho$ ,  $d_0$ , numNodes)
    [startPoints, targets]  $\leftarrow$  GETINITIALPOINTS( $OD$ ,  $\rho$ ,  $d_0$ )            $\triangleright$  Section 4.1.1
    [DT, D]  $\leftarrow$  INITIALISENETWORK(numNodes, startPoints, targets)  $\triangleright$  Section 4.1.2
    numFibre  $\leftarrow$  number of entries in startPoints
    for i in 1:numFibre do                                      $\triangleright$  Initialise the fibre structures
        fibres(i).node(0) = startPoints(i)
    for i in 1:numFibre do                                      $\triangleright$  Main growth loop (Section 4.1.3)
        terminated  $\leftarrow$  false
        j  $\leftarrow$  0
        while not terminated do
            GETCANDIDATES(fibres(i).node(j), DT)            $\triangleright$  Figure 4.1a
            • candidates  $\leftarrow$  DT.nodes sharing edge with fibres(i).node(j)
            GETBESTSTEP(candidates, D, targets)              $\triangleright$  Figure 4.1b
            • costs  $\leftarrow$  costs for candidates from Eq. (1) given targets, D
            • bestStep  $\leftarrow$  candidate with minimum cost
            TAKESTEP(fibres(i), bestStep)                   $\triangleright$  Figure 4.1c
            • fibres(i).node(j+1)  $\leftarrow$  bestStep
            UPDATETRIANGULATION(fibres(i), DT, D)           $\triangleright$  Figure 4.1d
            • segment  $\leftarrow$  vector from fibre(i).node(j) to fibre(i).node(j+1)
            •  $D_{new}$   $\leftarrow$  distance from DT.nodes to segment
            •  $D \leftarrow \min(D, D_{new})$ 
            ISTERMINATED(fibres(i), targets)
            • if fibre(i) has reached target:
                terminated  $\leftarrow$  true
            • elseif fibre(i) has no possible node to move to:
                terminated  $\leftarrow$  true
            • else:
                terminated  $\leftarrow$  false
                j  $\leftarrow$  j + 1
    end procedure

```

4.1.1 Input to the algorithm

The morphology of the final substrate will depend on the inputs to the algorithm which can be split into two general categories: parameters defining the fibre population(s), and parameters defining the space in which fibres grow.

Fibre parameters Fibre parameters include the desired orientation dispersion (OD), packing density (ρ) and diameter distribution ($P(d_0)$). These three parameters determine the initial settings for each individual fibre. Each fibre is defined by a starting point and a target point towards which it will grow as well as an initial fibre diameter, d_0 . These parameters for each fibre are determined from OD , ρ and $P(d_0)$ by packing circles



with the diameters drawn from $P(d_0)$ up to a density of ρ in 2 dimensions. Orientation dispersion is introduced by moving the target points of fibres relative to the starting points. For instance, planar dispersion can be introduced by splitting the substrate into planes and rotating these relative to one another.

Each fibre is allowed to shrink its diameter if it is necessary to fit into spaces close to other fibres. The maximum amount of shrinkage permitted is a controllable parameter, specified as a percentage of the initial fibre diameter.

Due to the stochastic nature of the algorithm, the final substrate is not guaranteed to have the exact morphological properties as input in the priors, however these inputs give the target morphology that ConFiG will attempt to produce.

Additionally, if the user wishes, the starting point, target point and diameter for each fibre can be directly input, rather than allowing ConFiG to generate them, in order to specify particular fibre configurations such as crossing fibre bundles.

Space defining parameters Parameters defining the space in which the fibres grow are used to define a discretisation of the space which is necessary to make the algorithm run in a practically feasible time. Ideally, the space in which the fibres can grow is a continuous space, so there are an infinite number of positions a fibre can occupy. Practically, this is intractable, so in ConFiG the space is discretised into a set of points which define nodes that the fibres can occupy.

Naturally, the choice of the density and arrangement of node points will have an impact on the substrate that is produced. Too few nodes will result in fibres that have very long, straight segments and may introduce intersections between fibres. Using more nodes will reduce overlap between fibres at the cost of more memory usage and slower growth of the fibres.

The arrangement of the nodes will also affect the morphology of the final substrate. For instance, placing nodes on a uniform grid may produce fibres with unnaturally angular paths. If the density of points on a uniform grid becomes sufficiently high, these angular bends are insignificant compared to the diameter and the fibres will have more natural shapes. For large substrates however, the number of nodes required to satisfy this condition becomes intractably large. For this reason, the nodes used are typically pseudo-randomly distributed to ensure broadly uniform coverage of the space, whilst keeping the number of nodes required lower. The impact of the choice of node points is investigated in Section 4.2.1.

4.1.2 Creation of the Growth Network

In order to embed information about the local environment at each node, the first step of the algorithm is generating the paths that fibres can take between the nodes as well as defining a maximum diameter that can be sustained at each node to avoid intersection which will be denoted by d_i , for a node, i . These paths define a network along which the fibres may grow.

4 Contextual Fibre Growth (ConFiG)

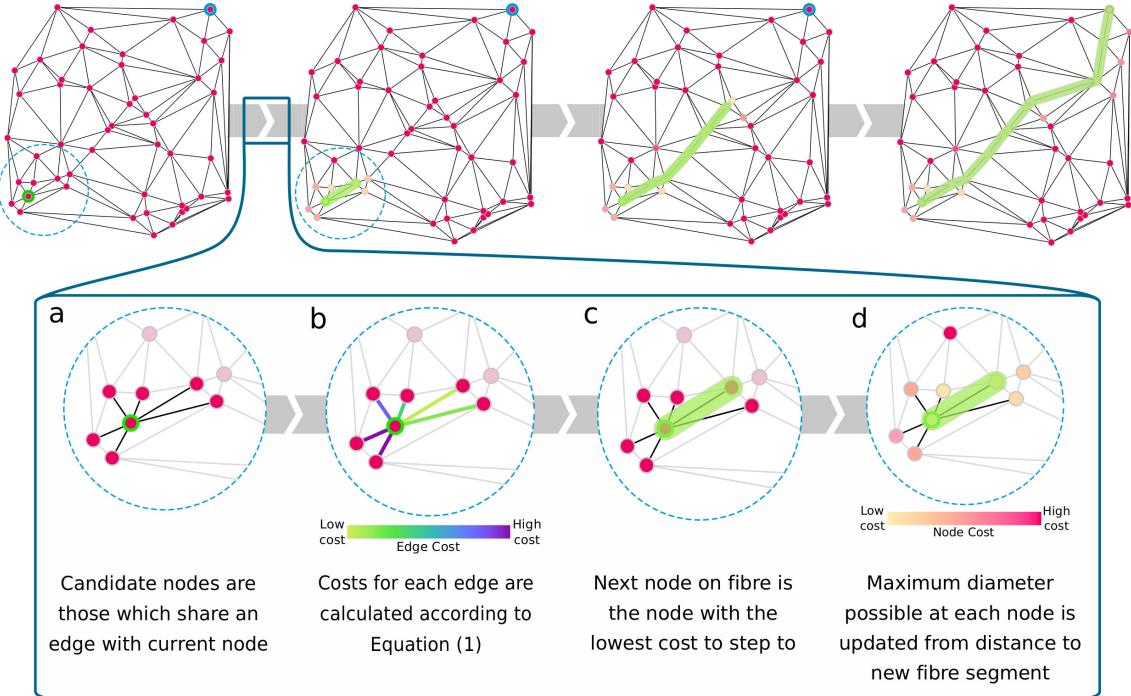


Figure 4.1: Schematic overview of the fibre growth algorithm. A fibre grows sequentially, moving from one node to the next, starting from the start point (top left, green node) toward the target (top left, blue node) along the edges defined by the Delaunay triangulation. Inset: The algorithm determining which node a fibre steps to at any given iteration. a) The possible nodes to step to are those which share an edge with the current node. b) From the edges available costs are calculated using eqs. (4.1) and (4.2). c) The fibre will grow along the edge with the lowest cost. d) From this new segment, the maximum diameter sustainable at a given node is calculated, giving each node a cost based on the maximum sustainable diameter. This cost will then be used in the calculation of edge weights (b) for future fibres. Note that although this figure illustrates the algorithm in 2D, in practice the algorithm grows fibres in 3D.

The paths between nodes are defined by the Delaunay triangulation[82] of the nodes which creates a fully connected network. This triangulation creates edges between nearby nodes, encoding information about the local connectivity at each node. Nodes that become occupied by a fibre will be inaccessible to any future fibres, which is one way in which intersection is minimised between fibres.

The maximum diameter, d_i , at each node encodes information on the amount of space available at each node. Where d_i is small, that node is close to an existing fibre, so any subsequent fibre passing through that node will have to shrink its diameter to d_i in order to prevent intersections. Allowing fibres to contextually shrink their diameter allows fibres to occupy spaces which would otherwise be unavailable.

4.1.3 Growth of a Fibre

Each individual fibre grows by moving the head of the fibre from node to node according to a cost function which attempts to ensure that the fibre moves towards its target whilst

avoiding intersection. The main steps in the growth of a single fibre are shown in Figure 4.1.

The first step in the growth of a fibre is determining which nodes are the possible next nodes can step to, referred to as candidate nodes. From a given starting node, s , the candidate nodes are any of the nodes which share an edge with s .

The choice of which candidate node a fibre steps to from the current node is determined by a cost function. The cost function consists of two terms, one which penalises moving away from the target point, t , and one which penalises moving to a position where d_i is low, meaning the fibre diameter would have to shrink. The cost function for a fibre at a position, s , to move to an candidate node, c , given a target point, t , is $l = l_t + f l_d$, where

$$l_t = \frac{1}{2} \cdot \frac{\|s - c\|}{1 + \|s - c\|} \cdot \left(1 - \frac{(c - s) \cdot (t - s)}{\|c - s\| \|t - s\|} \right), \quad (4.1)$$

$$l_d = \max \left(0, \frac{1}{d_0} (d_0 - d_i) \right), \quad (4.2)$$

d_0 is the desired radius of the fibre and f is a weighting factor between the two terms. In this work, f is fixed to 0.2 to more strongly weight growth toward the target.

Equation (4.1) is the term penalising moving away from the target. The dot product between the vector to the candidate and the vector to the target ensures that the minimum cost occurs when the candidate is directly aligned with the target. This term of the cost function is inspired by the work of Bicho et al. [83] who use the space colonisation algorithm [84, 85] to simulate crowds.

Equation (4.2) is the term penalising moving to a position where the radius of the fibre must shrink. For radii lower than the desired radius of the fibre, d_0 , Equation (4.2) grows linearly with distance from d_0 . For radii greater than or equal to d_0 , Equation (4.2) is zero, meaning that regions of empty space are equally weighted.

The next node for a fibre will be the candidate node which has the lowest cost according to Equations (4.1) and (4.2). This method of finding a path through the triangulation by choosing the lowest cost node at each position amounts to a greedy best-first pathfinding approach with a heuristic given by Equations (4.1) and (4.2).

With the next node chosen, the value of d_i needs to be updated for other nearby nodes. All nodes have d_i set to the Euclidean distance between the node and the surface of the new section of fibre if that distance is less than the current value of d_i . This is illustrated in Figure 4.1d.

Any nodes which now lie within the fibre have d_i set to zero. Nodes with $d_i = 0$ are disallowed from future steps, meaning that once a fibre has grown, no future fibres can connect to any nodes within the fibre. This, in addition to shrinking the radius of future fibres according to d_i at each node means that the fibres grow in an almost completely non-intersecting manner. Since the value of d_i is set based on fibre-to-point distances, there can be cases in which the fibres would intersect when the closest point between two fibre sections is not at one of the fibre nodes. In order to account for this, a meshing process developed which can deform fibres around one another. This is described in Section 4.1.4.

4 Contextual Fibre Growth (ConFiG)

The fibre growth algorithm will output a set of fibres which are defined by a series of nodes and the diameter of the fibre at each node. These are written into the Stockley-Wheal-Cole (SWC) format[86], a format commonly used to store cellular morphology information.

4.1.4 Creation of Fibre Meshes

In order to create 3D meshes to be used in dMRI simulations, a meshing process was developed using 3D modelling software Blender (<https://blender.org>). Fibres are meshed one-by-one using the Blender “SWC Mesher” add-on (https://github.com/mcellteam/swc_mesher) which uses Blender metaballs to make a mesh.

In Blender, a metaball is an implicit surface defined as the isosurface of a so-called directing structure. This directing structure can be seen the source of a static field. For instance a spherical isosurface can be formed with a directing structure which mimics the electric field a point charge. When multiple metaballs come close to one another, the fields will combine to form a surface that merges the two spheres together. An example of metaball interactions is shown in Figure 4.2.

By placing metaballs along the skeleton of each fibre, with the path and diameters given from the fibre growth algorithm, a smooth surface is formed for each fibre. It is this implicit surface, created using metaballs that the SWC mesher add-on creates. This implicit surface can be turned into an explicit surface (i.e. a mesh of vertices and faces) in Blender, which can then be refined by progressively smoothing and reducing the number of faces in the mesh to create a mesh which can be used in dMRI simulations.

This process can be used to mesh each fibre individually, however issues can arise with intersection of fibres, as mentioned in Section 4.1.3. In order to account for this, a contextual meshing algorithm was developed. The metaball surface for one fibre is created using the SWC Mesher. This surface is then turned into a mesh as described above, however the metaballs are retained. The metaball potential is then turned negative, meaning that rather than attracting any future nearby metaball surfaces, it will repel them, as shown in Figure 4.2b. This means that subsequent fibres which are meshed very close to, or overlapping with existing fibres will deform organically to resolve the intersection, thus creating a series of completely non-intersecting fibre meshes which can be used by the dMRI simulator.

The deformation introduced by the contextual meshing process has two effects. As well as helping to prevent intersection between fibres, the deformation produces fibres with more organic non-circular cross sections, better mimicking realistic mythologies. This is vastly different to the majority of previous WM numerical phantoms which model fibres as circular or elliptic cylinders.



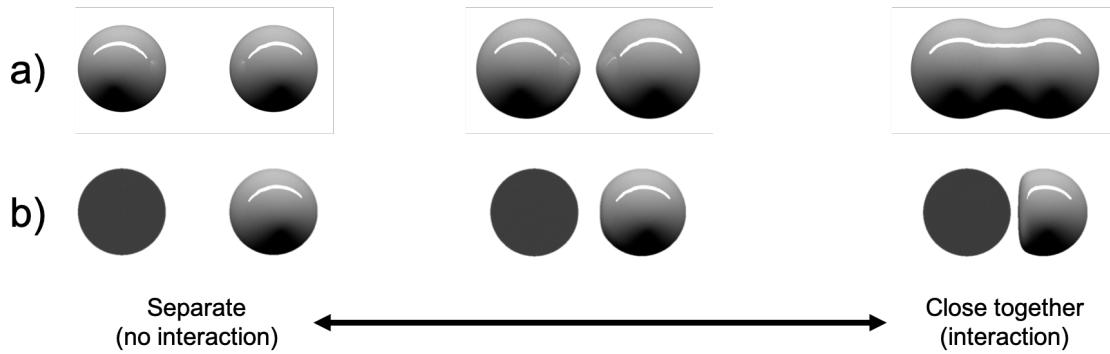


Figure 4.2: Simple example of metaball interactions. a) With two positive metaballs, the fields combine to attract the surfaces together. This is used to join individual segments into a continuous fibre. b) With one negative metaball (indicated by the flat grey circle) the surface of the metaball is repelled from the negative metaball. This is used to deform nearby fibres around one another.

4.2 Experiments and Results

4.2.1 Effect of Choice of Growth Network

As mentioned in Section 4.1.1, the choice of the node points in the network will affect the morphology of the resulting substrate. In order to investigate this, a qualitative experiment was performed in which a single fibre was grown on a network either a) nodes on a uniform grid or b) pseudorandom nodes. In each case, the number of nodes was increased and the resulting fibre investigated.

The fibre was defined by a start point $(20, 0, 0)$ μm , target point $(0, 0, 50)$ μm and diameter $1 \mu\text{m}$. This configuration was chosen so that the fibre would not have a path that directly followed one of the 90° or 45° lines in the uniform grid. Node points were

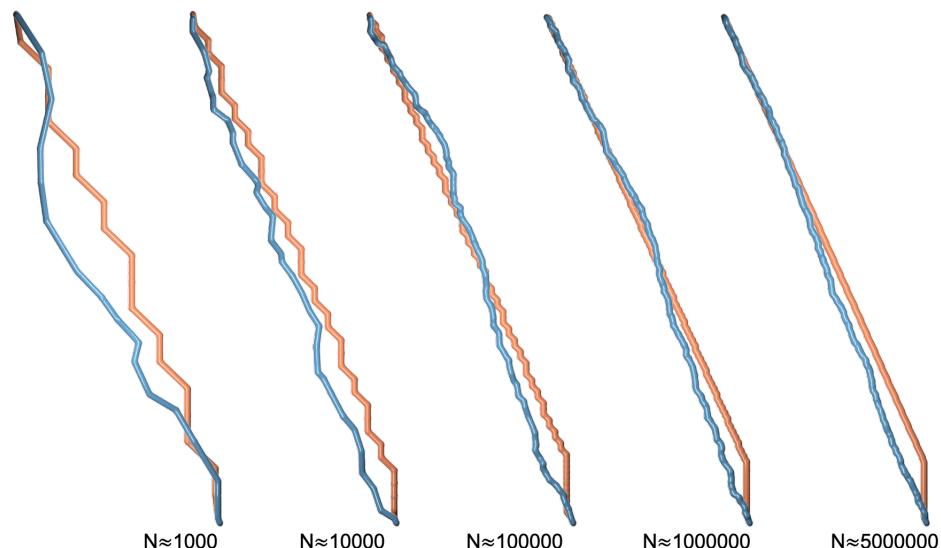


Figure 4.3: Fibres generated using uniform grid (orange) and pseudo-random (blue) network nodes for increasing numbers of nodes.

4 Contextual Fibre Growth (ConFiG)

initialised in either a uniform grid or pseudorandomly within the space [-5, -5, -5] to [25, 5, 55] to ensure coverage of the space in which the fibre would grow. The number of source points used was $N \approx 1000, 10000, 100000, 1000000, 5000000$.

The resulting fibres can be seen in Figure 4.3, where orange fibres are grown using the uniform grid and blue fibres using pseudorandom points. In both cases, as the number of nodes increases, the resulting fibre has more of a smooth, straight path between start and target. The uniform grid fibres, have a much more angular, structured path due to being forced to grow on the grid, while the pseudorandom fibres more irregular paths, which could be considered more ‘organic’ looking.

4.2.2 Demonstration of ConFiG

To demonstrate the potential of ConFiG, three substrates at different (dispersion, packing density) conditions were generated: (0°, 60%), (15°, 30%) and (35°, 25%), shown in Figure 4.4a. Each substrate is grown using 5×10^6 pseudo-randomly placed source nodes for the growth network, giving a network with 3.88×10^7 edges and a mean distance between any given node and its neighbours of $0.29 \mu\text{m}$. The packing densities chosen represent the highest densities achievable using ConFiG for each dispersion condition.

For the 0° dispersed substrate, initial diameters were drawn from a gamma distribution with mean $d_0 = 2 \mu\text{m}$ and standard deviation $\sigma_d = 0.2 \mu\text{m}$. The 15° and 35° substrates were generated with $d_0 = 1.2 \mu\text{m}$ and $\sigma_d = 0.2 \mu\text{m}$ in order to show the flexibility of ConFiG to generate substrates with different diameter distributions as well as orientation dispersion and packing density. Diameters were limited to be permitted to shrink to 25% of the original fibre diameter in order to fit into space.

For each substrate, the Pulsed-Gradient-Spin-Echo (PGSE) signal was simulated in Camino[13] using 5×10^5 diffusing spins and 5×10^3 discrete time steps, uniformly distributed with bulk-diffusivity $D_0=2 \mu\text{m}^2/\text{ms}$. To show the range of simulation possibilities available, three different membrane permeabilities ($\kappa=0, 0.0025, 0.0050 \mu\text{m}/\text{ms}$) were also imposed. The simulated PGSE measurement parameters were: $\delta/\Delta = 1/40 \text{ ms}$ and 50 b-values from 0 to $9 \text{ ms}/\mu\text{m}^2$ along x-, y- and z-directions.

The corresponding direction-averaged simulated PGSE signals at different permeabilities are shown with $\text{SNR} = \infty$ in Figure 4.4b and $\text{SNR} = 20$ in Figure 4.4c. The signal decays to a lower value as the dispersion increases and density decreases, as expected.

4.2.3 Comparison with Brute-Force Approach

ConFiG was compared against the naïve brute-force approach to fibre growth. The brute-force approach grows fibres one segment at a time and checks for collisions between the new segment and all existing fibres. Each new segment is chosen from one of 128 candidate directions on a cone aligned with the previous segment, with each direction being weighted according to Equation (4.1).



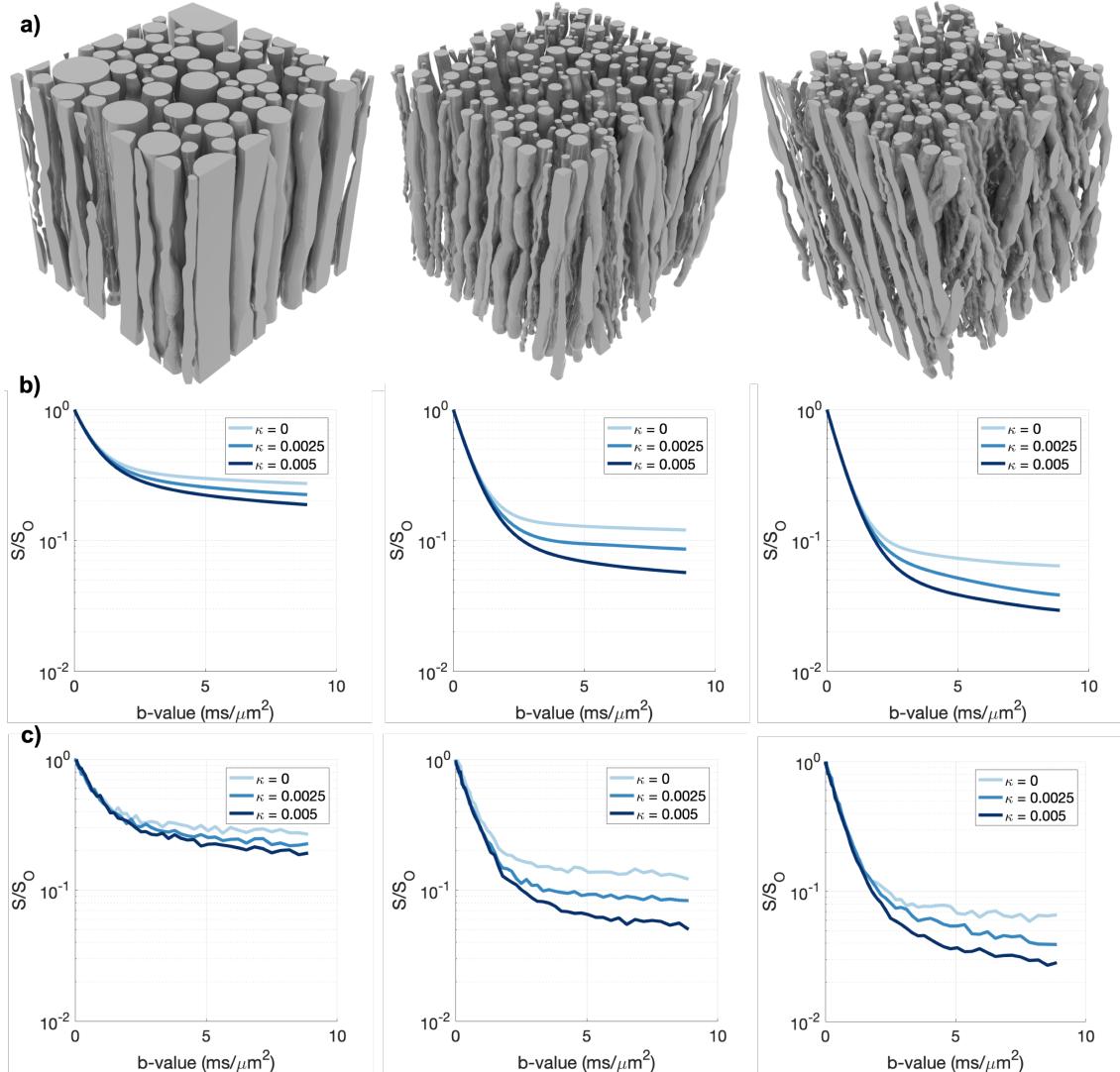


Figure 4.4: a) Example substrates (cut into $30 \times 30 \times 30 \mu\text{m}^3$ cube) from the fibre growth algorithm, left to right: Zero macroscopic dispersion (60% density), 15° of macroscopic dispersion (30% density), 35° dispersed (25% density). b) Simulations for each substrate for varying permeabilities with $\text{SNR} = \infty$ and c) $\text{SNR} = 20$. Units of κ are $\mu\text{m}/\text{ms}$.

Substrates were grown with both the brute-force approach and ConFiG using the same starting and target points and initial diameters. These initial parameters were determined by packing circles with gamma distributed radii (mean $d_0 = 2 \mu\text{m}$, standard deviation $\sigma = 0.6 \mu\text{m}$) into a $40 \mu\text{m} \times 40 \mu\text{m}$ square up to a packing density of 60%. Target points were set as $40 \mu\text{m}$ directly above the starting points to define a substrate with 0° macroscopic orientation dispersion. This resulted in a substrate with a total of 54 initial fibres.

The fibre growth algorithm used 1×10^6 randomly distributed points for the Delaunay triangulation giving a mean distance between points of $0.5 \mu\text{m}$, matching the brute force approach which used a segment length of $0.5 \mu\text{m}$ for each new fibre segment.

From these initial parameters, fibres were grown using a subset of $n = 1, 5, 10, 15, 20, 25, 30, 40$ fibres and the growth was timed. Each value of n was timed

4 Contextual Fibre Growth (ConFiG)

5 times with and the mean taken to reduce single-run timing fluctuations.

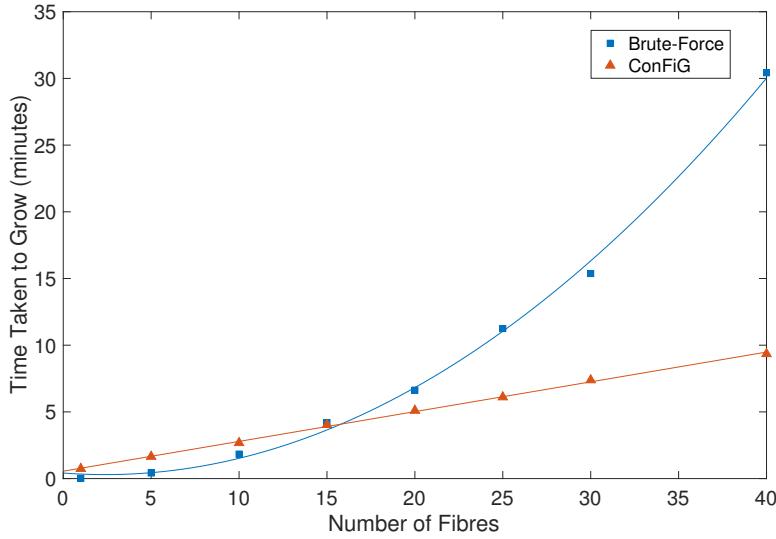


Figure 4.5: Timing of brute force growth vs. the fibre growth algorithm along with a quadratic fit (brute-force) and linear fit (fibre growth algorithm). The fibre growth algorithm is clearly linear in the number of fibres, while brute force growth fits an order n^2 well.

Figure 4.5 shows the timing results of the brute-force approach versus the fibre growth algorithm. The fibre growth algorithm has approximately $\mathcal{O}(n)$ complexity with n being the number of fibres. Conversely, the brute-force algorithm shows $\mathcal{O}(n^2)$ complexity owing to the fact that every new segment has to check for collisions with all existing fibres.

The fibre growth algorithm has a higher $n = 0$ offset which is caused by the overhead in calculating the Delaunay triangulation for the growth network. This causes the brute-force approach to have better performance at low n , while at higher n (approaching the > 100 fibres needed for a realistic dMRI voxel) the linearity of the fibre growth algorithm gives it much faster performance.

4.3 Discussion and Conclusion

ConFiG shifts perspective from previous works attempting to pack together fibres, by trying to mimic natural fibre genesis. This approach represents a major step towards very high fibre packing, enabling us to reach the highest dispersion at the highest packing density reached so far, to our knowledge. Our $(15^\circ, 30\%)$ and $(35^\circ, 25\%)$ represent an average $\sim 50\%$ and $\sim 200\%$ improvement, respectively, over the best previously reported results of $(10^\circ, 20\%)$ [63].



40 ||::|

One limitation of the ConFiG approach is that the resulting substrate will depend on the choice of the growth network. This is briefly investigated in Figure 4.3, showing that choosing a uniform grid can introduce regularities in the substrate which could be considered unrealistic. The dependence of the final substrate on the number and arrangement of network nodes should be investigated further, to gain a better understanding of which arrangement of nodes will be necessary to produce a desired substrate.

The substrates presented in Figure 4.4 are just a few examples of the kinds of substrates that can be produced using our ConFiG method. By varying the setup of the morphological controls and start and target points, many different fibre configurations can be produced. Currently, fibres will attempt to grow in a straight line between the start and target points, meaning that certain configurations such as kissing bundles cannot be represented. However, the algorithm can in principle be extended to allow for series of target points, allowing the definition of a desired ‘path’ of a fibre.

This behaviour of attempting to grow in a straight line from start to target is due to the dot product term in the cost function which was inspired by Bicho et al. [83]. This method may not be ideal however, since the dot product is very local measure, so at any point you only know the direction towards the target, but have no information about any ‘desired’ path. An alternative approach would be to adjust the cost function so that rather than maximising a dot product, it would minimise the distance to desired path which could allow any arbitrary path to be specified.

Additionally, some input parameter settings cannot be achieved. For instance, trying to grow a substrate with both very high density and very high dispersion will result in a final substrate that does not reach the density required. The reason for this could be a combination of limitations of the algorithm in restricting growth to a discrete network and also the fact that some morphological settings are practically infeasible. This limitation, however, also applies to the fibre packing and brute force growth approaches.

One weakness of the fibre-growth algorithm is that since the fibre diameters are calculated from a fibre-to-point distance, there can still be some small amount of overlap between fibres. This is solved using the meshing process in Blender to deform the regions of slight overlap between neighbouring fibres.

To conclude, the proposed ConFiG approach, using the fully connected growth network, is shown to be more efficient than a ‘brute-force’ growth approach. The fact that ConFiG is linear with the number of fibres makes it far more efficient for high numbers of fibres. For instance, a realistic voxel will need hundreds or thousands of fibres which will become impractically slow for the ‘brute-force’ approach, whilst remaining manageable for our algorithm. This efficiency, along with the high density and orientation dispersion achieved means that ConFiG represents a promising step forward in the construction of ultra-realistic numerical phantoms of WM.

4 Contextual Fibre Growth (ConFiG)



5 CUDAmico

Chapter Contents

5.1	CUDAmico Design	43
5.2	CUDAmico Experiments	45
5.3	Discussion and Conclusion	47

Introduction

This chapter introduces CUDAmico, a GPU accelerated dMRI simulator based on the MC simulator that is part of the Camino software package [13, 5]. The main differences between CUDAmico and Camino are outlined and well as presenting some experiments assessing the performance of CUDAmico relative to Camino.

5.1 CUDAmico Design

The basic Monte-Carlo dMRI simulation algorithm outlined in Algorithm 2.1 does not change when adapting the simulations to run on the GPU, however some changes are necessary to make the simulations efficient for the GPU.

CUDAmico is, of course, designed to exploit the parallel nature of the GPU. Since each spin is simulated independently of all of the other spins, the problem is inherently parallel and we can use each execution thread to simulate a single spin.

The problem could also be parallelised along the time dimension, however that can lead to a problem known as a race condition. A race condition occurs when the program tries to do two things at once, but one should rely on the other so they must be done in order. In this instance, you could get one thread trying to take the $(t + 1)^{\text{th}}$ step before the t^{th} step is finished. For this reason, CUDAmico is only parallelised along the number of spins.

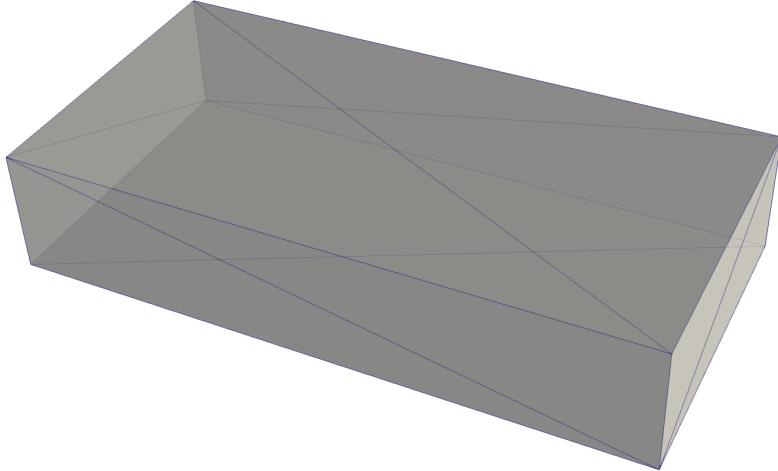


Figure 5.1: Simple lo-res cuboid used for testing CUDAmico. Cuboid dimensions are $2 \mu\text{m} \times 5 \mu\text{m} \times 10 \mu\text{m}$ and it has 12 triangular faces.

Another potential pitfall of GPU simulations, mentioned in Section 2.4, is thread divergence. In particular, the while loop in Algorithm 2.1 could be a source of thread divergence. Different threads will have spins in different locations, so may end up with different numbers of iterations of the loop to amend the step. The impact of this will be accentuated by meshes with small structures where faces are close to each other, so there may be many multiple reflections.

In order to try and minimise thread divergence, CUDAmico follows the technique of Waudby and Christodoulou [80] who use rejection sampling. The rejection sampling technique simply checks whether any given step has crossed a face and if it has, then the step is rejected and the spin does not move. Provided the steps are small relative to the size of the structures in the substrate, the difference between rejection sampling and reflection is minimal [87].

Another consideration when designing CUDAmico, was to limit the number of unnecessary collision checks. A complex mesh may have hundreds of thousands or millions of faces, if each step in the simulation has to check all faces, it will be unnecessarily slow. The step should, ideally, only check for collisions with those faces which are nearby to it, in doing so it may be possible to reduce the number of collision checks from hundreds of thousand to a few tens of faces.

This problem of reducing the number of faces required for collision checks is a well studied problem in the field of ray-traced graphics rendering. A number of spatial partitioning schemes have been developed to accelerate collision checks such as the simple uniform grid [88, 89], octree [88] and bounding volume hierarchy (BVH)[90].

So far, aligning with the CPU Camino implementation, CUDAmico uses a simple uniform grid acceleration scheme. In short, this uniformly subdivides the space in which the mesh sits, meaning that each step need only check collisions for the triangles within the subspace that the step goes through.

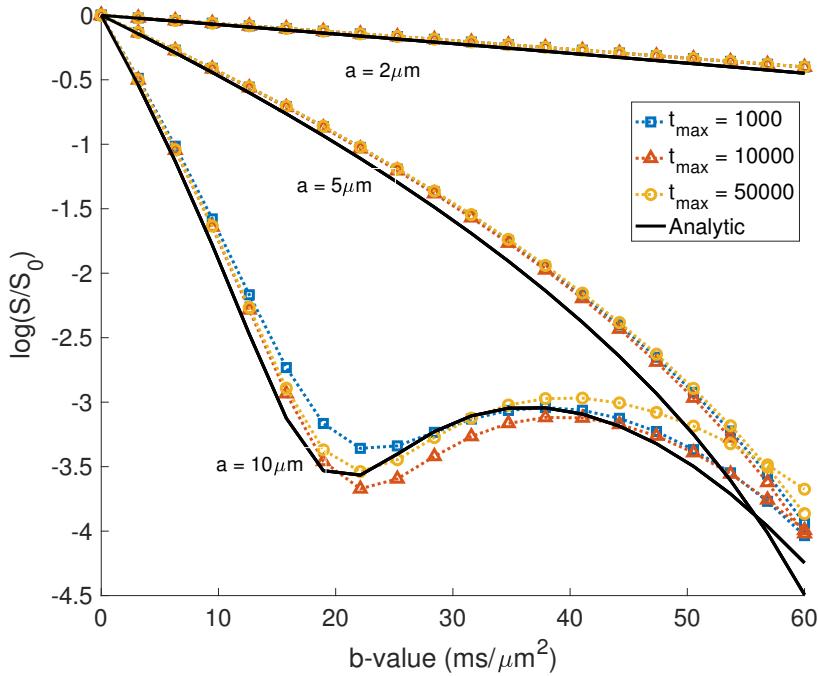


Figure 5.2: CUDAmino simulations (dotted lines) versus analytic solutions (solid black lines) for diffusion in the three directions normal to faces of the cuboid in Figure 5.1. a is the separation between the faces in each direction. The simulated signals match the analytic signal well, with more deviation at higher b -value in general.

5.2 CUDAmino Experiments

In order to test CUDAmino’s performance against Camino, two experiments were carried out. Both experiments were carried out using a simple cuboid mesh as shown in Figure 5.1, which has dimensions $2\text{ }\mu\text{m} \times 5\text{ }\mu\text{m} \times 10\text{ }\mu\text{m}$ and 12 faces. This simple substrate was chosen since the analytic solution for diffusion between two planes with a separation, a , is well known [29, 91].

The first experiment investigated the effect of the number of timesteps on the resulting simulated signal. As the number of timesteps increases, the step length decreases and the rejection sampling approach of CUDAmino should approach the reflection approach of Camino.

Simulations were run in both Camino and CUDAmino with $n = 100000$ spins and $t_{max} = 1000, 10000, 50000$ and impermeable membranes. Simulated dMRI signals were generated using a standard PGSE sequence with $\Delta = 50$ ms, $\delta = 0.1$ ms (very small δ to approximate SGP regime) and b -value ranging from 0 to $60\text{ ms}/\mu\text{m}^2$. CUDAmino experiments are run on an NVIDIA TITAN Xp GPU and Camino experiments are run on a 3.1GHz Intel Core i7 CPU.

Figure 5.2 shows the results of the tests varying the number of timesteps for CUDAmino against the analytic solution for each direction normal to the planes of the cuboid. Here a is the separation between the planes. In general, CUDAmino matches the the analytic signal, with slightly more deviation from the analytic solution at higher b -value. Whilst Figure 5.2 gives the impression that the error is highest for the largest



	2 μm		5 μm		10 μm	
	CUDA	Camino	CUDA	Camino	CUDA	Camino
$t_{max} = 10^3$	0.0178	0.0174	0.0198	0.0205	0.0111	0.0076
$t_{max} = 10^4$	0.0177	0.0187	0.0173	0.0204	0.0081	0.0088
$t_{max} = 5 \times 10^4$	0.0180	0.0180	0.0210	0.0228	0.0102	0.0074
$n = 10^3$	0.0133	0.0180	0.0626	0.0493	0.0283	0.0087
$n = 10^4$	0.0173	0.0207	0.0242	0.0267	0.0154	0.0109
$n = 10^5$	0.0170	0.0179	0.0198	0.0180	0.0083	0.0076

Table 5.1: Mean error between simulations and analytic results for the cuboid in Figure 5.1. Values are absolute difference between simulated signal and analytic signal, $|S_{simulated} - S_{analytic}|/S_0$, for the two experiments described in Section 5.2. Here CUDAmico is shortened to CUDA to save space. In each case, the value across 5 repeats had insignificant variance.

separation, $a = 10 \mu\text{m}$, this is in part a consequence of the logarithm on the y-axis.

Table 5.1 shows the mean error between both the CUDAmico and Camino signals and the analytic solution for each value of t_{max} . This shows that the mean error is of a similar order for all three values of a . In general, the error reduces with higher t_{max} , though this is not the case for $a = 2 \mu\text{m}$. Additionally, the errors are similar for both CUDAmico and Camino, suggesting that the difference between the rejection sample and reflection approaches is minimal, at least for this simple substrate.

A similar experiment was carried out, varying the number of spins rather than the number of timesteps. In this case, t_{max} was fixed to 5000 and $n = 1000, 10000, 100000, 1000000$. The same PGSE sequence parameters were used.

Figure 5.3 shows the results of the tests varying n for CUDAmico against the analytic solutions. Again, the simulations generally match the analytic solutions well, with higher n matching more closely as would be expected. As in the t_{max} varying case, the deviation from the analytic solution tends to increase with increased b -value.

Again, Table 5.1 shows the mean error between both the CUDAmico and Camino signals and the analytic solution. Similarly to the varying t_{max} case, the error is comparable between CUDAmico and Camino and for both tends to decrease slightly with increasing n .

The main reason to develop CUDAmico for the GPU was to speed up the simulations by exploiting the parallelism on the GPU. To test the speedup achieved, each simulation was timed. Table 5.2 shows the results of timing the CUDAmico and Camino simulations. CUDAmico achieves a shorter running time in each case, with most simulations having a $100\times$ speedup compared to Camino. The one case in which CUDAmico does not achieve a $100\times$ speedup is the $n = 1000$ case in which it has a $15\times$ speedup. This is likely due to $n = 1000$ not being enough spins to fully exploit the parallelism of the GPU and being limited by overheads such as copying data from the CPU to GPU.



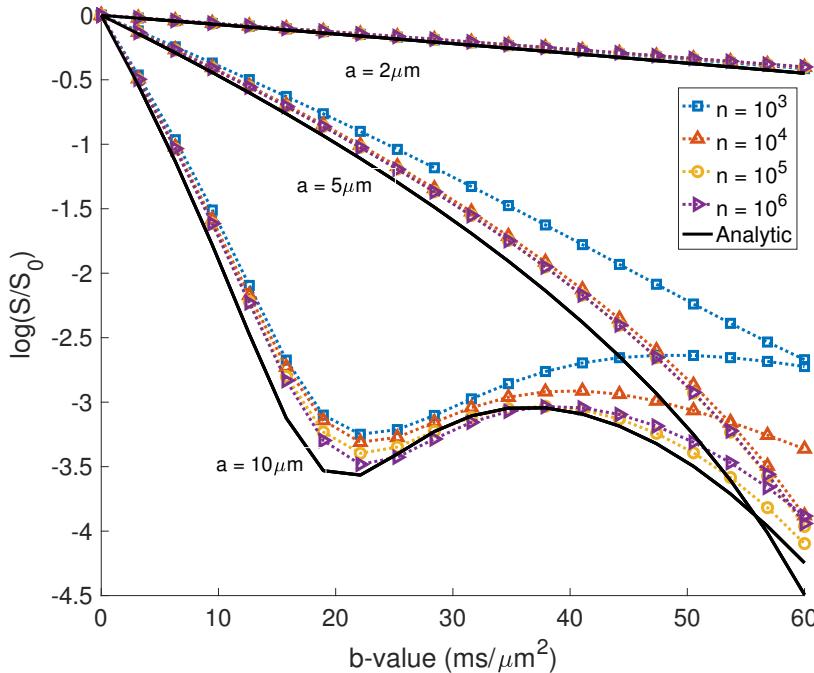


Figure 5.3: CUDAMino simulations (dotted lines) versus analytical solutions (solid black lines) simulations within the cuboid in Figure 5.1 varying the number of spins, n .

	Running time (s)		Speedup
	CUDAMino	Camino	
$t_{max} = 10^3$	2.0876 ± 0.003	241.71 ± 7.49	$115.7\times$
$t_{max} = 10^4$	19.964 ± 0.037	2707.3 ± 27.3	$110.5\times$
$t_{max} = 5 \times 10^4$	97.739 ± 0.039	10762 ± 68.4	$111.2\times$
$n = 10^3$	0.7797 ± 0.008	12.059 ± 0.478	$15.46\times$
$n = 10^4$	1.1385 ± 0.006	114.82 ± 0.775	$110.9\times$
$n = 10^5$	10.004 ± 0.034	1105.1 ± 11.2	$110.5\times$

Table 5.2: Timing results for CUDAMino versus Camino. For all experiments, CUDAMino shows faster running time with lower variance across runs. The typical speedup is around $100\times$.

5.3 Discussion and Conclusion

CUDAMino, as presented here, still has some way to go to fully replicate Camino's diffusion simulations. There are a number of features lacking, such as permeability and full reflections, however this represents a good first step towards a fully-featured GPU dMRI simulator.

As shown in Figures 5.2 and 5.3, CUDAMino is able to match the analytic result in a simple substrate. Additionally Table 5.1 shows that the difference between the simulated signals in CUDAMino and Camino is small in the simple scenario presented. This is a promising result as it shows that CUDAMino is able to match analytic results as well as the

5 CUDAmico

results of Camino well. There should be further investigation, however, to investigate how CUDAmico performs in other simple substrates compared to both Camino and analytical results.

The performance boost that CUDAmico has over Camino, running simulations 15-100 \times faster shows the promise that CUDAmico has in making dMRI simulations much more efficient. Most of the simulations had a speedup of over 100 \times , with only the simulations with a small number of spins having the 15 \times speedup. For most complex cases, the number of spins required will be significantly higher than 1000, suggesting that in practice, CUDAmico may see typical speedups around 100 \times .



6 Future Plan

Chapter Contents

6.1	Improving ConFiG	49
6.2	Extending CUDAmino	50
6.3	Future Applications	51
6.4	Plan of Action	51

Introduction

This chapter lays out the future plan for the remainder of the PhD. Sections 6.1 to 6.3 outline various avenues for improvement for both ConFiG and CUDAmino as well as some potential applications for them. Section 6.4 brings these potential ideas for improvement together into a concrete plan of future work for this project.

6.1 Improving ConFiG

As mentioned in 4.3, the dependence of ConFiG substrates on the growth network requires more thorough investigation. One planned experiment to investigate this will be to grow substrates initialised in a simple manner in which we know the optimal packing, such as a simple square packing of parallel cylinders. We can then grow substrates with varying setups of the growth network and compare the result to the parallel cylinder case. This will give an understanding of how the density and arrangement of network nodes impacts the packing density achievable in the resulting substrate.

A potential improvement to ConFiG is to use more clever generation of the network nodes. Currently, nodes are generated throughout the entire space in which the fibres will grow. This is not necessary, however, since only the area immediately surrounding the fibres will matter for growth. For this reason, it should be possible to generate the network only in the immediate area in which the fibres are growing, allowing for a more dense sampling of the space.

6 Future Plan

A further potential improvement to network generation could be to generate the nodes based on the desired path of the fibre. For instance, it may be possible to generate a dense cloud of nodes around the desired path of each fibre and sparsely fill the remaining space to prevent oversampling of space where fibres are unlikely to grow.

There are also potential avenues for improvement in the process of finding the best step among the neighbours at any given node. Essentially, this boils down to adding/removing/adjusting terms in the cost function. One potential improvement, mentioned in Section 4.3, is to adjust the target following term to minimise the distance to a desired path. This will allow for any arbitrary target path, rather than simply the straight line between start and target that is currently used, allowing undulation control to be more easily added.

Additionally, it may be possible to incorporate connectivity information from the network to optimise growth. Since a node becomes inaccessible when it is within a fibre, the number of neighbours at any given node contains information about how many of its neighbours are within existing fibres. This may be used to penalise moving to areas with low connectivity, since the likelihood of becoming stuck is higher. Furthermore, connectivity from second neighbours may be incorporated to give some information more than just one step ahead.

One additional improvement would be to allow the nodes in the network to move slightly. For instance, when a fibre grows very close to a node, it may be possible to move that node away from the fibre, so that future fibres will not have to shrink as much when they access that node. One drawback of this approach is that by adjusting the physical position of the node, but not its connectivity, you may introduce intersections between fibres. It may be possible that any intersection can be solved using the Blender meshing procedure, however this should be investigated if dynamic adjustment of the node positions is to be implemented.

6.2 Extending CUDAMino

As it stands, CUDAMino lacks some of the features of the diffusion MRI simulator in Camino. One feature which is yet to be added is support for simulation of permeable membranes. This can be added so that when there is a collision between a step and a face, there is a probability that the step will go through the face.

The performance of CUDAMino relative to Camino should be investigated further, characterising the exact differences between the two in more simple cases and assessing the performance on more complex substrates, such as those produced by ConFiG too.

Another feature would be to support reflections as the interaction between a step and a barrier as an alternative to rejection sampling. Although this may cause branch divergence as mentioned in Section 5.1, the impact of this is not yet quantified and may be outweighed by requiring a larger number of steps for rejection sampling.

As discussed in Section 5.1, there are many acceleration possible for ray tracing-like problems. For instance, the octree structure may be used to further accelerate CUDAMino since the space partitioning is adaptive to the mesh, meaning that the space is more finely discretised in regions where the mesh is complex. Another alternative, the bounding volume hierarchy and its variants, similarly subdivide the space, but have been shown to be more efficient than octrees for GPU ray traced rendering[92].



A comparison of the various space partitioning schemes for dMRI simulation, both in terms of computational time and memory usage, could be very informative.

6.3 Future Applications

As mentioned in Chapter 1, realistic WM substrates could have applications in validation of novel microstructural models of WM. To this end, ConFiG is planned to be used as part of the ISBI 2019/2020 MRI White Matter Reconstruction Challenge (MEMENTO: For more information see <https://my.vanderbilt.edu/memento/>). The MEMENTO challenge aims to test the state of the art microstructure imaging techniques, evaluating participants' ability to estimate microstructural parameters, predict unseen signal and evaluate sensitivity and specificity of potential biomarkers. The MEMENTO challenge presents an opportunity for ConFiG to make an important contribution to the advancement of the microstructure imaging field.

Additionally, ConFiG may be used to train a machine learning tool directly for the estimation of parameters, in a similar manner to Hill et al. and Palombo et al. [10, 11] for estimating permeability and the fingerprinting approach of Rensonnet et al. [9]. The basic idea would be to generate a range of WM substrates covering a realistic range of microstructure parameters. Simulated dMRI signals could then be generated using Camino or CUDAmino to generate microstructure-signal pairs. These could then be used to train a machine learning tool to infer the microstructure directly from the signal. This approach could yield better estimations of microstructural parameters than the previous studies due to ConFiG's ability to produce more realistic WM substrates. Rather than simple straight cylinders, ConFiG will be able to incorporate complex features such as orientation dispersion and diameter variations.

6.4 Plan of Action

An expected timeline for the remainder of the PhD is shown in Figure 6.1. The main work moving forward in this project will be to improve and apply ConFiG for the generation of realistic WM substrates. Three papers are planned to be completed by the end of the PhD:

1. A presentation of ConFiG in its current state, including the experiments shown in this report and the planned experiment mentioned in Section 6.1.
2. The extension of ConFiG to incorporate the concurrent growth of fibres.
3. The application of ConFiG to train a machine learning tool for microstructural parameter estimation including the complex morphological features that can be generated with ConFiG

Paper 1 should be completed relatively soon after the MPhil Upgrade, since a lot of the work has already been presented in this report and conference submissions (see Publications appendix). The other two papers are expected to require 6-8 months each for the development of the methods, experiments and writing up. Around four months at the end of the PhD is set aside for writing the thesis.

6 Future Plan

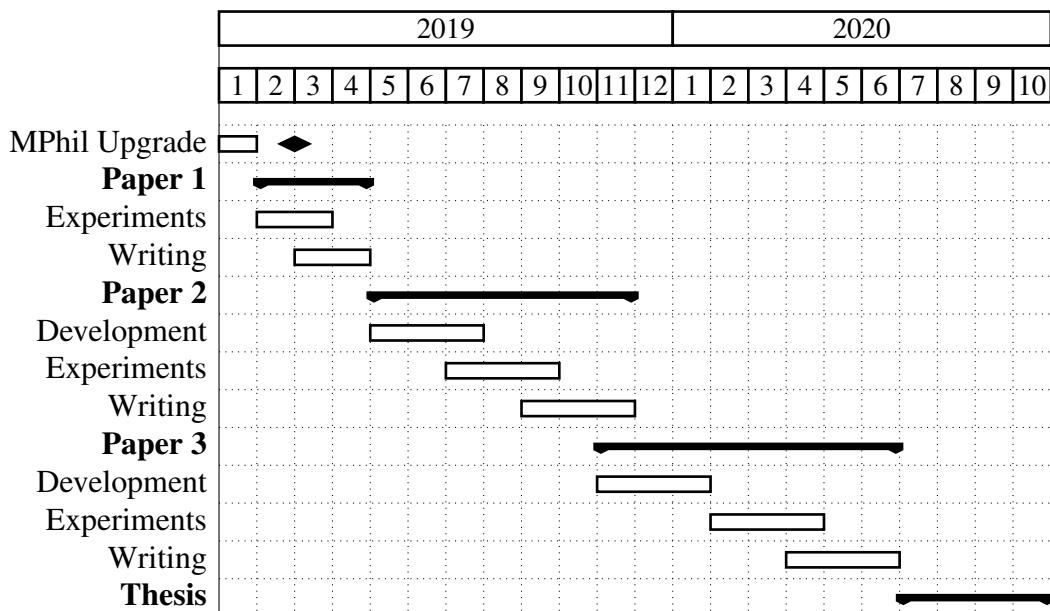


Figure 6.1: Timetable of planned future work



Publications

Relevant Publications

Under Review

1. Ross Callaghan, Noam Shemesh, Daniel C. Alexander, Hui Zhang, and Marco Palombo. “Towards a more realistic and flexible white matter numerical phantom generator for diffusion MRI simulation”. Submitted to *27th Annual Meeting of the International Society for Magnetic Resonance in Medicine (ISMRM)*, 2019
2. Ross Callaghan, Daniel C. Alexander, Hui Zhang and Marco Palombo. “Contextual fibre growth to generate realistic axonal packing for diffusion MRI simulation”. Submitted to *26th International Conference on Information Processing in Medical Imaging (IPMI)*, 2019
3. Ross Callaghan, Daniel C. Alexander, Hui Zhang and Marco Palombo. “Contextual fibre growth to generate realistic axonal packing for diffusion MRI simulation”. Submitted to *25th Annual Meeting of the Organization for Human Brain Mapping*, 2019

Other Publications

1. Ross Callaghan, Bhavana Solanky, Sotirios Bisdas, Hui Zhang and Enrico De Vita. “Comparing MEGA editing techniques for in-vivo measurement of 2-hydroxyglutarate”. *Proceedings of the 26th Annual Meeting of the International Society for Magnetic Resonance in Medicine (ISMRM)*, 2018

6 Future Plan



References

- [1] S. Ackerman, *Discovering the Brain*. Washington, DC: Institute of Medicine and National Academy of Sciences, The National Academies Press, 1992. (Cited on page 1)
- [2] F. Crick and E. Jones, “Backwardness of human neuroanatomy,” *Nature*, vol. 361, pp. 109–110, 1993. (Cited on page 1)
- [3] H. Johansen-Berg and T. E. J. Behrens, *Diffusion MRI: From Quantitative Measurement to In vivo Neuroanatomy: Second Edition*. 2013. (Cited on page 1)
- [4] P. J. Basser, J. Mattiello, and D. Lebihan, “Estimation of the Effective Self-Diffusion Tensor from the NMR Spin Echo,” *Journal of Magnetic Resonance, Series B*, vol. 103, no. 3, pp. 247–254, 1994. (Cited on page 1)
- [5] M. G. Hall and D. C. Alexander, “Convergence and Parameter Choice for Monte-Carlo Simulations of Diffusion MRI,” *IEEE Transactions on Medical Imaging*, vol. 28, no. 9, pp. 1354–1364, 2009. (Cited on pages 2, 3, 17, 19, 25 and 43)
- [6] A. Leemans, J. Sijbers, M. Verhoye, A. Van Der Linden, and D. Van Dyck, “Mathematical framework for simulating diffusion tensor MR neural fiber bundles,” *Magnetic Resonance in Medicine*, vol. 53, no. 4, pp. 944–953, 2005. (Cited on page 2)
- [7] M. Nilsson, D. V. Westen, F. Sta, and J. La, “The role of tissue microstructure and water exchange in biophysical modelling of diffusion in white matter,” *Magnetic Resonance Materials in Physics, Biology and Medicine*, vol. 26, no. 3, pp. 345–370, 2013. (Cited on page 2)
- [8] H.-H. Lee, K. Yaros, J. Veraart, J. Pathan, F.-X. Liang, S. G. Kim, D. S. Novikov, and E. Fieremans, “Electron microscopy 3-dimensional segmentation and quantification of axonal dispersion and diameter distribution in mouse brain corpus callosum,” *bioRxiv*, vol. 357491, 2018. (Cited on pages 2 and 28)

References

- [9] G. Rensonnet, B. Scherrer, G. Girard, A. Jankovski, S. K. Warfield, B. Macq, J.-P. Thiran, and M. Taquet, “Towards microstructure fingerprinting: Estimation of tissue properties from a dictionary of Monte Carlo diffusion MRI simulations,” *NeuroImage*, vol. 184, no. May 2018, pp. 964–980, 2018. (Cited on pages 2, 27 and 51)
- [10] I. Hill, M. Palombo, M. Santin, F. Branzoli, A.-C. Philippe, D. Wassermann, M.-S. Aigrot, B. Stankoff, H. Zhang, S. Lehericy, A. Petiet, D. C. Alexander, O. Ciccarelli, and I. Drobnjak, “Deep neural network based framework for in-vivo axonal permeability estimation,” in *26th Annual Meeting of the International Society for Magnetic Resonance in Medicine (ISMRM)*, 2018. (Cited on pages 2, 27 and 51)
- [11] M. Palombo, I. Hill, M. D. Santin, F. Branzoli, A.-C. Philippe, D. Wassermann, M.-S. Aigrot, B. Stankoff, H. Zhang, S. Lehericy, A. Petiet, D. C. Alexander, and I. Drobnjak, “Machine learning based estimation of axonal permeability: validation on cuprizone treated in-vivo mouse model of axonal demyelination,” in *Proc. Joint Annual Meeting ISMRM-ESMRMB, Paris, France*, 2018. (Cited on pages 2, 27 and 51)
- [12] G. L. Nedjati-Gilani, T. Schneider, M. G. Hall, N. Cawley, I. Hill, O. Ciccarelli, I. Drobnjak, C. A. Wheeler-Kingshott, and D. C. Alexander, “Machine learning based compartment models with permeability for white matter microstructure imaging,” *NeuroImage*, 2017. (Cited on pages 2 and 27)
- [13] P. a. Cook, Y. Bai, K. K. Seunarine, M. G. Hall, G. J. Parker, and D. C. Alexander, “Camino: Open-Source Diffusion-MRI Reconstruction and Processing,” *14th Scientific Meeting of the International Society for Magnetic Resonance in Medicine*, vol. 14, p. 2759, 2006. (Cited on pages 3, 25, 38 and 43)
- [14] J. Nickolls, I. A. N. Buck, and M. Garland, “Scalable Parallel with CUDA,” *Queue*, vol. 6, no. April, 2008. (Cited on pages 3 and 20)
- [15] M. H. Levitt, *Spin Dynamics: Basics of Nuclear Magnetic Resonance*. 2008. (Cited on pages 5, 6, 7 and 12)
- [16] P. B. Barker, A. Bizzi, N. De Stefano, R. Gullapalli, and D. D. M. Lin, *Clinical MR Spectroscopy*. 2009. (Cited on page 5)
- [17] F. Bloch, “Nuclear Induction,” *Physical Review*, vol. 652, no. 1938, pp. 460–474, 1946. (Cited on pages 5 and 8)
- [18] E. M. Purcell, H. C. Torrey, and R. V. Pound, “Resonance Absorption by Nuclear Magnetic Moments in a Solid,” *Physical Review*, vol. 69, no. 1-2, 1946. (Cited on page 5)
- [19] N. M. Salibi and M. A. Brown, *Clinical MR spectroscopy : first principles*. Wiley-Liss, 1998. (Cited on page 5)



- [20] E. Haacke, R. Brown, M. Thompson, and R. Venkatesan, “Magnetic resonance imaging: physical principles and sequence design. 1999,” *New York: A John Wiley and Sons*, 1999. (Cited on pages 6, 8, 12 and 14)
- [21] R. A. de Graaf, *In Vivo NMR Spectroscopy: Principles and Techniques: 2nd Edition.* 2007. (Cited on pages 9, 10, 11 and 12)
- [22] E. L. Hahn, “Spin Echoes,” *Physical Review*, vol. 80, no. 4, pp. 580–594, 1950. (Cited on page 14)
- [23] M. W. Denny, *Air and Water: The Biology and Physics of Life’s Media.* Air and Water: The Biology and Physics of Life’s Media, Princeton University Press, 1993. (Cited on page 14)
- [24] E. O. Stejskal and J. E. Tanner, “Spin diffusion measurements: Spin echoes in the presence of a time-dependent field gradient,” *The Journal of Chemical Physics*, vol. 42, no. 1, pp. 288–292, 1965. (Cited on pages 15 and 16)
- [25] W. Price, “Pulsed-field gradient nuclear magnetic resonance as a tool for studying translational diffusion: Part 1. Basic theory,” *Concepts Magn. Reson.*, vol. 9, pp. 299–336, 1997. (Cited on pages 16, 17, 24 and 25)
- [26] P. W. Kuchel, G. Pages, K. Nagashima, V. Sendhil, V. Vijayaragavan, V. Nagarajan, and K. H. Chuang, “Stejskal–Tanner Equation Derived in Full,” *Concepts in Magnetic Resonance*, vol. 40, no. 5, pp. 205–214, 2012. (Cited on page 16)
- [27] C. H. Neuman, “Spin echo of spins diffusing in a bounded medium,” *The Journal of Chemical Physics*, vol. 60, no. 11, pp. 4508–4511, 1974. (Cited on page 17)
- [28] B. Balinov, B. Jönsson, P. Linse, and O. Söderman, “The NMR self-diffusion method applied to restricted diffusion. simulation of echo attenuation from molecules in spheres and between planes,” 1993. (Cited on page 17)
- [29] P. T. Callaghan, “Pulsed-Gradient Spin-Echo NMR for Planar, Cylindrical, and Spherical Pores under Conditions of Wall Relaxation,” 1995. (Cited on pages 17 and 45)
- [30] H. C. Torrey, “Bloch equations with diffusion terms,” *Physical Review*, vol. 104, no. 3, pp. 563–565, 1956. (Cited on pages 17 and 18)
- [31] P. T. Callaghan, *Principles of Nuclear Magnetic Resonance Microscopy*, vol. 19. 1991. (Cited on pages 17 and 24)
- [32] W. S. Price, *NMR studies of translational motion.* 2009. (Cited on page 17)
- [33] J. Jeener, “Macroscopic molecular diffusion in liquid NMR, revisited,” *Concepts in Magnetic Resonance Part B: Magnetic Resonance Engineering*, vol. 14, no. 2, pp. 79–88, 2002. (Cited on page 17)
- [34] L. Beltrachini, Z. A. Taylor, and A. F. Frangi, “An efficient finite element solution of the generalised Bloch-Torrey equation for arbitrary domains,” *Mathematics and Visualization*, no. FEM, pp. 3–14, 2016. (Cited on pages 18 and 24)



References

- [35] C. Grossmann and H.-G. Roos, *Numerical Treatment of Partial Differential Equations*. Springer, 2007. (Cited on page 18)
- [36] H. Hagslätt, B. Jönsson, M. Nydén, and O. Söderman, “Predictions of pulsed field gradient NMR echo-decays for molecules diffusing in various restrictive geometries. Simulations of diffusion propagators based on a finite element method,” *Journal of Magnetic Resonance*, vol. 161, no. 2, pp. 138–147, 2003. (Cited on pages 18 and 24)
- [37] D. L. Logan, E. Veitch, C. Carson, K. R. Burrell, V. Gould, E. Wagner, D. L. Logan, E. Veitch, C. Carson, K. R. Burrell, V. Gould, and E. Wagner, *A First Course in the Finite Element Method Fourth Edition*, vol. 147. 2007. (Cited on page 18)
- [38] A. Iserles, *A first course in the numerical analysis of differential equations*. 2009. (Cited on page 18)
- [39] C.-H. Yeh, B. Schmitt, D. Le Bihan, J.-R. Li-Schlittgen, C.-P. Lin, and C. Poupon, “Diffusion Microscopist Simulator: A General Monte Carlo Simulation System for Diffusion Magnetic Resonance Imaging,” *PLoS ONE*, vol. 8, no. 10, p. e76626, 2013. (Cited on pages 19 and 25)
- [40] M. Nilsson, J. Lätt, F. Ståhlberg, D. van Westen, and H. Hagslätt, “The importance of axonal undulation in diffusion MR measurements: A Monte Carlo simulation study,” *NMR in Biomedicine*, vol. 25, no. 5, pp. 795–805, 2012. (Cited on pages 19 and 26)
- [41] B. A. Landman, J. A. D. Farrell, S. A. Smith, D. S. Reich, P. A. Calabresi, and P. C. M. Van Zijl, “Complex geometric models of diffusion and relaxation in healthy and damaged white matter,” *NMR in Biomedicine*, vol. 23, no. 2, pp. 152–162, 2010. (Cited on pages 19 and 25)
- [42] G. T. Balls and L. R. Frank, “A simulation environment for diffusion weighted MR experiments in complex media,” *Magnetic Resonance in Medicine*, vol. 62, no. 3, pp. 771–778, 2009. (Cited on pages 19 and 25)
- [43] J. C. Ford and D. B. Hackney, “Numerical model for calculation of apparent diffusion coefficients (ADC) in permeable cylinders - Comparison with measured ADC in spinal cord white matter,” *Magnetic Resonance in Medicine*, vol. 37, no. 3, pp. 387–394, 1997. (Cited on page 19)
- [44] a. Szafer, J. Zhong, and J. C. Gore, “Theoretical model for water diffusion in tissues.,” *Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine*, vol. 33, no. 5, pp. 697–712, 1995. (Cited on pages 19 and 24)
- [45] M. D. Budde and J. A. Frank, “Neurite beading is sufficient to decrease the apparent diffusion coefficient after ischemic stroke,” *PNAS*, vol. 107, no. 32, pp. 14472–14477, 2010. (Cited on pages 19 and 26)



- [46] E. Panagiotaki, M. G. Hall, H. Zhang, B. Siow, M. F. Lythgoe, and D. C. Alexander, “High-Fidelity Meshes from Tissue Samples for Diffusion MRI Simulations,” in *MICCAI*, pp. 404–411, 2010. (Cited on page 19)
- [47] J. E. Stone, D. Gohara, and G. Shi, “OpenCL: A Parallel Programming Standard for Heterogeneous Computing Systems,” *Computing in Science & Engineering*, vol. 12, pp. 66–73, 2010. (Cited on page 20)
- [48] C. L. Chin, F. W. Wehrli, S. N. Hwang, M. Takahashi, and D. B. Hackney, “Biexponential diffusion attenuation in the rat spinal cord: Computer simulations based on anatomic images of axonal architecture,” *Magnetic Resonance in Medicine*, vol. 47, no. 3, pp. 455–460, 2002. (Cited on pages 23 and 25)
- [49] S. N. Hwang, C. L. Chin, F. W. Wehrli, and D. B. Hackney, “An image-based finite difference model for simulating restricted diffusion,” *Magnetic Resonance in Medicine*, vol. 50, no. 2, pp. 373–382, 2003. (Cited on page 23)
- [50] J. Xu, M. D. Does, and J. C. Gore, “Numerical study of water diffusion in biological tissues using an improved finite difference method Numerical study of water diffusion in biological tissues,” *Physics in medicine and biology*, vol. 52, pp. N111–N126, 2007. (Cited on page 24)
- [51] J. Xu, M. D. Does, and J. C. Gore, “Sensitivity of MR diffusion measurements to variations in intracellular structure: Effects of nuclear size,” *Magnetic Resonance in Medicine*, vol. 61, no. 4, pp. 828–833, 2009. (Cited on page 24)
- [52] J. Xu, H. Li, K. D. Harkins, X. Jiang, J. Xie, H. Kang, M. D. Does, and J. C. Gore, “Mapping mean axon diameter and axonal volume fraction by MRI using temporal diffusion spectroscopy,” *NeuroImage*, vol. 103, pp. 10–19, 2014. (Cited on pages 24 and 25)
- [53] B. F. Moroney, T. Stait-Gardner, B. Ghadirian, N. N. Yadav, and W. S. Price, “Numerical analysis of NMR diffusion measurements in the short gradient pulse limit,” *Journal of Magnetic Resonance*, vol. 234, pp. 165–175, 2013. (Cited on page 24)
- [54] D. V. Nguyen, J. R. Li, D. Grebenkov, and D. Le Bihan, “A finite elements method to solve the Bloch-Torrey equation applied to diffusion magnetic resonance imaging,” *Journal of Computational Physics*, vol. 263, no. FEM, pp. 283–302, 2014. (Cited on page 24)
- [55] H.-g. Lipinski, “Monte Carlo simulation of extracellular diffusion in brain tissues Monte Carlo simulation of extracellular diffusion in brain tissues,” *Physics in medicine and biology*, vol. 35, p. 441, 1990. (Cited on page 24)
- [56] G. J. Stanisz, A. Szafer, G. A. Wright, and R. M. Henkelman, “An Analytical Model of Restricted Diffusion in Bovine Optic-Nerve,” *Magnetic Resonance in Medicine*, vol. 37, pp. 103–111, 1997. (Cited on page 24)

References

- [57] A. Duh, A. Mohorič, and J. Stepišnik, “Computer simulation of the spin-echo spatial distribution in the case of restricted self-diffusion,” *Journal of Magnetic Resonance*, vol. 148, no. 2, pp. 257–266, 2001. (Cited on page 24)
- [58] J. R. Stiles, D. Van Helden, T. M. Bartol, E. E. Salpeter, and M. M. Salpeter, “Miniature endplate current rise times less than 100 microseconds from improved dual recordings can be modeled with passive acetylcholine diffusion from a synaptic vesicle.,” *Proceedings of the National Academy of Sciences*, vol. 93, no. 12, pp. 5747–5752, 1996. (Cited on page 25)
- [59] J. R. Stiles and T. M. Bartol, *Monte Carlo Methods for Simulating Realistic Synaptic Microphysiology using MCELL*. 2001. (Cited on page 25)
- [60] R. A. Kerr, T. M. Bartol, B. Kaminsky, M. Dittrich, J.-C. J. Chang, S. B. Baden, T. J. Sejnowski, and J. R. Stiles, “Fast Monte Carlo Simulation Methods for Biological Reaction-Diffusion Systems in Solution and on Surfaces.,” *SIAM J Sci Comput*, vol. 30, no. 6, p. 3126, 2008. (Cited on page 25)
- [61] G. T. Baxter and L. R. Frank, “A computational model for diffusion weighted imaging of myelinated white matter,” *NeuroImage*, vol. 75, pp. 212–220, 2013. (Cited on page 25)
- [62] P. N. Sen and P. J. Basser, “A model for diffusion in white matter in the brain,” *Biophysical Journal*, vol. 89, no. 5, pp. 2927–2938, 2005. (Cited on page 25)
- [63] K. Ginsburger, F. Poupon, J. Beaujoin, D. Estournet, F. Matuschke, J.-F. Mangin, M. Axer, and C. Poupon, “Improving the realism of white matter numerical phantoms: A step toward a better understanding of the influence of structural disorders in diffusion MRI,” *Frontiers in Physics*, vol. 5, no. FEB, pp. 1–18, 2018. (Cited on pages 25 and 40)
- [64] D. C. Alexander, P. L. Hubbard, M. G. Hall, E. A. Moore, M. Ptito, G. J. Parker, and T. B. Dyrby, “Orientationally invariant indices of axon diameter and density from diffusion MRI,” *NeuroImage*, vol. 52, no. 4, pp. 1374–1389, 2010. (Cited on page 25)
- [65] M. Nilsson, S. Lasić, I. Drobnjak, D. Topgaard, and C. F. Westin, “Resolution limit of cylinder diameter estimation by diffusion MRI: The impact of gradient waveform and orientation dispersion,” *NMR in Biomedicine*, vol. 30, no. 7, pp. 1–13, 2017. (Cited on page 25)
- [66] M. Nilsson, J. Lätt, E. Nordh, R. Wirestam, F. Ståhlberg, and S. Brockstedt, “On the effects of a varied diffusion time in vivo: is the diffusion in white matter restricted?,” *Magnetic Resonance Imaging*, vol. 27, no. 2, pp. 176–187, 2009. (Cited on page 26)
- [67] M. Nilsson, E. Alerstam, R. Wirestam, F. Ståhlberg, S. Brockstedt, and J. Lätt, “Evaluating the accuracy and precision of a two-compartment Karger model using Monte Carlo simulations,” *Journal of Magnetic Resonance*, vol. 206, no. 1, pp. 59–67, 2010. (Cited on pages 26 and 27)



- [68] E. Fieremans, D. S. Novikov, J. H. Jensen, and J. A. Helpern, “Monte Carlo study of a two-compartment exchange model of diffusion,” *NMR in Biomedicine*, vol. 23, no. 7, pp. 711–724, 2010. (Cited on page 26)
- [69] J. Kärger, H. Pfeifer, and W. Heink, “Principles and Application of Self-Diffusion Measurements by Nuclear Magnetic Resonance,” *Advances in Magnetic and Optical Resonance*, vol. 12, no. C, pp. 1–89, 1988. (Cited on page 26)
- [70] M. Lin, H. He, G. Schifitto, and J. Zhong, “Simulation of changes in diffusion related to different pathologies at cellular level after traumatic brain injury,” *Magnetic Resonance in Medicine*, vol. 76, no. 1, pp. 290–300, 2016. (Cited on page 26)
- [71] T. A. G. M. Huisman, L. H. Schwamm, P. W. Schaefer, W. J. Koroshetz, N. Shetty-Alva, Y. Ozsunar, O. Wu, and A. G. Sorensen, “Diffusion Tensor Imaging as Potential Biomarker of White Matter Injury in Diffuse Axonal Injury,” *American Journal of Neuroradiology*, vol. 25, no. 3, pp. 370–376, 2004. (Cited on page 26)
- [72] J. J. Bazarian, J. Zhong, B. Blyth, T. Zhu, V. Kavcic, and D. Peterson, “Diffusion Tensor Imaging Detects Clinically Important Axonal Damage after Mild Traumatic Brain Injury: A Pilot Study,” *Journal of Neurotrauma*, vol. 24, no. 9, pp. 1447–1459, 2007. (Cited on page 26)
- [73] D. R. Rutgers, P. Fillard, G. Paradot, M. Tadié, P. Lasjaunias, and D. Ducreux, “Diffusion tensor imaging characteristics of the corpus callosum in mild, moderate, and severe traumatic brain injury,” *American Journal of Neuroradiology*, vol. 29, no. 9, pp. 1730–1735, 2008. (Cited on page 26)
- [74] M. Lin, H. He, Q. Tong, Q. Ding, X. Yan, T. Feiweier, and J. Zhong, “Effect of myelin water exchange on DTI-derived parameters in diffusion MRI: Elucidation of TE dependence,” *Magnetic Resonance in Medicine*, vol. 1660, pp. 1650–1660, 2017. (Cited on page 26)
- [75] W. W. Lam, S. Jbabdi, and K. L. Miller, “A model for extra-axonal diffusion spectra with frequency-dependent restriction,” *Magnetic Resonance in Medicine*, vol. 73, no. 6, pp. 2306–2320, 2015. (Cited on page 26)
- [76] J. Stepišník, “Time-dependent self-diffusion by NMR spin-echo,” *Physica B: Physics of Condensed Matter*, vol. 183, no. 4, pp. 343–350, 1993. (Cited on page 26)
- [77] G. Rensonnet, D. Jacobs, B. Macq, and M. Taquet, “A hybrid method for efficient and accurate simulations of diffusion compartment imaging signals,” in *Proc. of SPIE*, vol. 9681, p. 968107, 2015. (Cited on page 27)
- [78] D. Grebenkov, “Laplacian Eigenfunctions in NMR. I. A Numerical Tool,” *Concepts in Magnetic Resonance Part A*, vol. 32, no. 4, pp. 277–301, 2008. (Cited on page 27)

References

- [79] G. Rensonnet, B. Scherrer, S. K. Warfield, B. Macq, and M. Taquet, “Assessing the validity of the approximation of diffusion-weighted-MRI signals from crossing fascicles by sums of signals from single fascicles,” *Magnetic Resonance in Medicine*, vol. 2345, pp. 2332–2345, 2017. (Cited on page 27)
- [80] C. A. Waudby and J. Christodoulou, “GPU accelerated Monte Carlo simulation of pulsed-field gradient NMR experiments,” *Journal of Magnetic Resonance*, vol. 211, no. 1, pp. 67–73, 2011. (Cited on pages 28 and 44)
- [81] K.-v. Nguyen, E. Hernández-garzón, and J. Valette, “Efficient GPU-based Monte-Carlo simulation of diffusion in real astrocytes reconstructed from confocal microscopy,” *Journal of Magnetic Resonance*, vol. 296, pp. 188–199, 2018. (Cited on page 28)
- [82] P. B. Delaunay, “Sur la sphere vide,” *Bulletin of Academy of Sciences of the USSR*, 1934. (Cited on page 34)
- [83] A. D. L. Bicho, R. A. Rodrigues, S. R. Musse, C. R. Jung, M. Paravisi, and L. P. Magalhães, “Simulating crowds based on a space colonization algorithm,” *Computers and Graphics (Pergamon)*, vol. 36, no. 2, pp. 70–79, 2012. (Cited on pages 35 and 41)
- [84] A. Runions, M. Fuhrer, B. Lane, P. Federl, A.-G. Rolland-Lagan, and P. Prusinkiewicz, “Modeling and visualization of leaf venation patterns,” *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 702–711, 2005. (Cited on page 35)
- [85] A. Runions, B. Lane, and P. Prusinkiewicz, “Modeling trees with a space colonization algorithm,” *Natural Phenomena*, pp. 63–70, 2007. (Cited on page 35)
- [86] E. Stockley, H. Cole, A. Brown, and H. Wheal, “A system for quantitative morphological measurement and electronic modelling of neurons: three-dimensional reconstruction,” *J Neurosci Methods*, vol. 47, no. 1-2, pp. 39–51, 1993. (Cited on page 36)
- [87] H. Jóhannesson and B. Halle, “Solvent diffusion in ordered macrofluids: A stochastic simulation study of the obstruction effect,” *Journal of Chemical Physics*, vol. 104, no. 17, pp. 6807–6817, 1996. (Cited on page 44)
- [88] A. Fujimoto, T. Tanaka, and K. Iwata, “Arts: Accelerated Ray-Tracing System,” *IEEE Computer Graphics and Applications*, 1986. (Cited on page 44)
- [89] J. Amanatides and A. Woo, “A fast voxel traversal algorithm for ray tracing,” in *Eurographics '87*, 1987. (Cited on page 44)
- [90] T. L. Kay and J. T. Kajiya, “Ray tracing complex scenes,” *ACM SIGGRAPH Computer Graphics*, 1986. (Cited on page 44)



- [91] P. Linse and O. Söderman, “The Validity of the Short-Gradient-Pulse Approximation in NMR Studies of Restricted Diffusion. Simulations of Molecules Diffusing between Planes, in Cylinders and Spheres,” *Journal of Magnetic Resonance*, vol. 116, pp. 77–86, 1995. (Cited on page 45)
- [92] M. G. Chajdas and R. Westermann, “Quantitative Analysis of Voxel Raytracing Acceleration Structures,” in *Pacific Graphics Short Papers*, pp. 85–90, 2014. (Cited on page 50)



References



Todo list

References



Acknowledgements

Cheers

