Monte Carlo Simulation of Diffusion Magnetic Resonance Imaging

by

Ming Miao

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Applied Mathematics

Waterloo, Ontario, Canada, 2019

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The goal of this thesis is to describe, implement and analyse Monte Carlo (MC) algorithms for simulating the mechanism of diffusion magnetic resonance imaging (dMRI). As the inverse problem of mapping the sub-voxel micro-structure remains challenging, MC methods provide an important numerical approach for creating ground-truth data. The main idea of such simulations is first generating a large sample of independent random trajectories in a prescribed geometry and then synthesizing the imaging signals according to given imaging sequences.

The thesis starts by providing a concise introduction of the mathematical background for understanding dMRI. It then proceeds to describe the workflow and implementation of the most basic Monte Carlo method with experiments performed on simple geometries. A theoretical framework for error analysis is introduced, which to the best of the author's knowledge, has been absent in the literature. In an effort to mitigate the costly nature of MC algorithms, the geometrically adaptive fast random walk algorithm (GAFRW) is implemented, first invented by D.Grebenkov. Additional mathematical justification is provided in the appendix should the reader find details in the original paper by Grebenkov lacking. The result suggests that the GAFRW algorithm only provides moderate accuracy improvement over the crude MC method in the geometry modeled after white matter fibers. Overall, both approaches are shown to be flexible for a variety of geometries and pulse sequences.

Acknowledgements

I would like to thank all the people who made this thesis possible. My utmost gratitude goes towards my supervisor and mentor, Professor Edward R. Vrscay, who has supported me in every way during my graduate study. In addition to his invaluable guidance towards my research and writing, his inexhaustible love and patience is sometimes all that kept me going. I would also like to express my thanks to Professor Oleg Michailovich, from the department of Electrical and Computer Engineering, who introduced me to the exciting field of medical imaging and initiated this research project. Many thanks to Professor Zoran Miskovic for teaching me the course on stochastic processes in Physics and the discussion which cleared my confusion on first passage problems. I thank Dr. Denis Grebenkov for valuable discussions on the fast random walk algorithm.

Last but not least, I am thankful to all of my friends and family. Without their support I would not be able to fulfill my goals.

Dedication

To my parents.

Table of Contents

List of Figures

Chapter 1

Introduction

Since its advent in the mid-1980s, diffusion magnetic resonance imaging (dMRI) has become the state-of-the-art non-invasive technique for probing the micro-structure of the central nervous system. dMRI has shown to be capable of revealing cellular morphology at a finer scale compared to typical MRI resolution. The most widely-applied imaging technique is diffusion tensor imaging (DTI). Combined with tractography based on DTI derived quantities, scientists have been able to determine structure of the central nervous system.

Despite its great advancement and popularity, many questions in dMRI remain open, the most fundamental one being consistently mapping the diffusion-restricting geometry. Theoretically answering this question has been proven to be highly challenging. In addition, research that only uses real images taken from an MRI scanner can be expensive and sensitive to imaging artifacts. Therefore, numerical experiments is a popular research method in hope of gaining insights into the largely featureless MR signals.

Among various existing approaches, Monte Carlo methods are used for building a diffusion MR simulator in this study. Our methods are developed based on the fundamental principles of the formation of MR images. Careful implementation is carried out to perform simulations on relatively simple geometries. Mathematical justification are included wherever approximations are employed. I hope the our simulation schemes can make contribution to the vibrant area of diffusion MRI.

Chapter 2

Mathematical Foundation

2.1 Behaviour of protons in magnetic fields

We begin our exploration of the world of *magnetic resonance imaging* (MRI) by laying out the physical laws governing elementary particles. One of the most important discoveries in modern physics is that particles possess a quantum mechanical property known as *spin*. Although having no counterpart in classical physics, spin can be understood by the following two classical analogues:

- Spin induces a magnetic moment for a charged particle. As a consequence, charged particle can interact with the external magnetic field. This is similar to the everyday phenomena that a compass needle adjusts its orientation under the influence of the earth's magnetic field.
- Elementary particles possess angular momentum, referred to as the *intrinsic angular momentum*. This means that an external torque would cause the particle to *precess* in the same way that a classical spinning top would behave.

As a result, the external magnetic field can be used to manipulate the motion of protons. Our discussion will be focused on the spin-bearing water molecules as they are the main source of signal in diffusion MRI. We will explain how a signal and more importantly, an image can be generated by applying external magnetic fields of various temporal and spatial profile. The mathematical machinery needed is introduced along the way. Although a rigorous treatment of the MR physics requires quantum mechanics (for example, see [6]

or [7] for an excellent introduction), the theory laid out in this work is classical as it is sufficient for understanding diffusion MRI.

A modern commercial MRI scanner is able to produce three types of magnetic field: the main field, the radio frequency (rf) field and the gradient field. The cylindrical magnet and coils used for generating those fields are organized in a coaxial manner (see [17] for a scanner schematic). We will illustrate the role each field plays individually.

2.1.1 Main Field: Creating a net magnetization

Main field Generated by the main magnet, the *main field* or the *static field*, is the basis of any nuclear magnetic resonance (NMR) experiment. It is of constant and uniform profile for the whole duration of the MRI scanning process. It has an exceptionally strong field strength (1.5 to 3.0 T[1]), typically denoted as B_0. The direction of the main field always aligns with the axis of the cylindrical shaped scanner, denoted as the z-axis. In addition, the plane orthogonal to the main field is called the transverse plane, or xy plane.

Spin excess When spins experience no magnetic field, their orientation appears to be random. Once the main magnet is activated the spins in the scanned sample (e.g, part of a patient's body) immediately align themselves either towards or against the direction of the main field. If a water molecule has its magnetic moment pointing the same direction of the main field, it is referred as being in the *spin-up* state. Otherwise it is called to be in the *spin-down* state. At thermal equilibrium, more water molecules are in the spin-up state since it has lower energy, a phenomenon known as the *spin excess*. A computation based on statistical mechanics shows that ratio of the numbers of spin-up water molecules (N_+) versus spin down-ones (N_-) is given by the Boltzmann factor:

$$\frac{N_+}{N_-} = \exp\left(\frac{\gamma B_0 \hbar}{k_B T}\right), \tag{2.1}$$

where

- $\gamma = 2.38 \times 10^8 \text{rad} \cdot T^{-1} \cdot s^{-1}$ is the *proton gyromagnetic ratio*,
- $\hbar = 1.05 \times 10^{-34} J \cdot s$ is the reduced Planck constant,

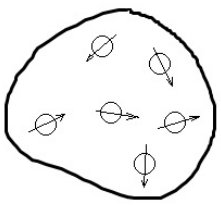
[1] Compared to the earth magnetic field strength which is only around 5×10^{-5} T.

- $k_B = 1.38 \times 10^{-23} J/K$ is the Boltzmann constant,
- T is the absolute temperature.

At room temperature this ratio is around 10^{-6} for $B_0 = 1.5$ T.

In a millimetic sub-sample (known as a *voxel* in MRI literature), we define the quantity called the spin *magnetization*, denoted by $\boldsymbol{M}$, by summing up the magnetic moment of all protons within (typically of the number around 10^{23}). Due to spin excess, a magnetization with the direction of the main field is produced (shown in Figure 2.1). The vector *magnetization density* $\boldsymbol{m}$ is defined as the magnetization per volume of sample. A derivation based on statistical mechanics shows that the equilibrium magnetization density is proportional to B_0 and ρ_0,

$$m_0 \propto \rho_0 \frac{\gamma^2 \hbar^2 B_0}{k_B T}. \tag{2.2}$$



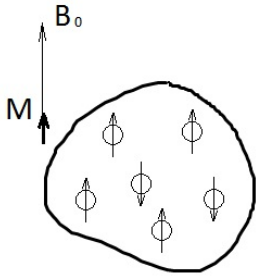


Figure 2.1: Spin orientations prior and after applying the main field.

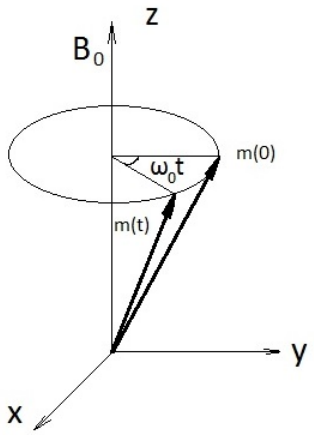


Figure 2.2: Time evolution of the magnetization according to the simple Bloch equation.

Unfortunately, it is impossible to measure this induced magnetization experimentally since its magnitude is extremely small in comparison to the main field strength. However, if this magnetization vector can be 'tilted' off the main field direction, in other words, if the induced magnetization has a non-zero xy component, there is hope to detect it.

Simple Bloch equation The time evolution of magnetization under the influence of an external magnetic field $\boldsymbol{B}$ is described by the *simple Bloch equation*:

$$\frac{d\boldsymbol{m}}{dt} = \gamma \boldsymbol{m} \times \boldsymbol{B}. \tag{2.3}$$

If the initial value of $\boldsymbol{m}(t)$ is $\boldsymbol{m}(0) = (m_{0x}, m_{0y}, m_{0z})^T$, and we set $\boldsymbol{B} = B_0\hat{\boldsymbol{z}}$, the solution of the simple Bloch equation can be written in the matrix form:

$$\boldsymbol{m}(t) = \boldsymbol{R}_z(-\omega_0 t)\boldsymbol{m}(0), \tag{2.4}$$

where $\omega = \gamma B_0$ is called the *Larmor frequency* and

$$\boldsymbol{R}_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

represents the transformation of the counterclockwise rotation about the $z-$axis by an angle of θ. Geometrically, the $z-$component of $\boldsymbol{m}(t)$ remains constant while the xy-component of $\boldsymbol{m}(t)$ rotates clockwise about the z-axis at Larmor frequency (shown in Figure 2.2). The z- and xy-component of $\boldsymbol{m}(t)$ are referred as the *longitudinal* and *transverse* magnetizations respectively. The transverse magnetization will be our main quantity of interest from now on. Notice that the x- and y- components of equation 2.4 can be combined into a complex form,

$$m_{xy}(t) = m_{xy}(0)e^{-i\omega_0 t}. \tag{2.5}$$

The question remains that how a non-zero transverse magnetization can be created in the first place. This is where the radio frequency (rf) pulse plays its role.

2.2 Signal detection

The simple Bloch equation predicts the Larmor precession of spins under influence of external magnetic field. Additional magnetic fields can be applied to manipulate spin motions.

2.2.1 Rotating reference frame

We introduce the *rotating reference frame*, a reference frame that rotates clockwise about the z-axis at ω_0. The time evolution of the magnetization can be described in this new reference frame. Mathematically, this is equivalent of the change of variable:

$$\tilde{\boldsymbol{m}}(t) = \boldsymbol{R}_z(\omega_0 t)\boldsymbol{m}(t) \tag{2.6}$$

Taking the time derivative on both sides and plugging into equation 2.3, one gets the simple Bloch equation in the rotating reference frame:

$$\frac{d\tilde{\boldsymbol{m}}}{dt} = \gamma\tilde{\boldsymbol{m}}(t) \times \boldsymbol{B}_{\text{eff}}, \text{ where } \boldsymbol{B}_{\text{eff}} = \boldsymbol{R}_z(\omega_0 t)\boldsymbol{B} - \frac{\omega_0}{\gamma}\hat{\boldsymbol{z}}. \tag{2.7}$$

Let $\boldsymbol{B}$ take the form of a perturbed main field so that the effective magnetic field becomes

$$\boldsymbol{B} = B_0\hat{z} + \Delta\boldsymbol{B} \Rightarrow \boldsymbol{B}_{\text{eff}} = \boldsymbol{R}_z(\omega_0 t)\Delta\boldsymbol{B} + (1 - \frac{\omega_0}{\gamma B_0})B_0\hat{\boldsymbol{z}}. \tag{2.8}$$

The choice of $\omega_0 = \gamma B_0$ makes the second term vanish and the simple Bloch equation in the rotating frame reads

$$\frac{d\tilde{\boldsymbol{m}}}{dt} = \gamma\tilde{\boldsymbol{m}} \times \boldsymbol{R}_z(\omega_0 t)\Delta\boldsymbol{B}. \tag{2.9}$$

Unsurprisingly, when $\Delta\boldsymbol{B} = 0$, the solution in the rotating frame is a constant

$$\tilde{\boldsymbol{m}}(t) = \boldsymbol{m}(0).$$

2.2.2 Radio Frequency Pulses: Letting the spins dance

We now introduce our *radio frequency (rf) field* as a perturbation of the main field, which takes the form of

$$\Delta\boldsymbol{B} = B_1(t)\boldsymbol{R}_z(-\omega_0 t)\hat{\boldsymbol{x}}. \tag{2.10}$$

In other words, the rf-field has a potentially time-dependent magnitude[2] and a fixed direction measured in the rotating frame. Plugging into equation 2.9, one has

$$\frac{d\tilde{\boldsymbol{m}}}{dt} = \gamma\tilde{\boldsymbol{m}} \times (B_1(t)\hat{\boldsymbol{x}}). \tag{2.11}$$

By comparing equation 2.11 to equation 2.3, one can recognize the interpretation of equation 2.11: the vector $\tilde{\boldsymbol{m}}(t)$ rotates clockwise about the x-axis at an instantaneous angular velocity of $\gamma B_1(t)$. Therefore, the solution of equation 2.11 is

$$\tilde{\boldsymbol{m}}(t) = \boldsymbol{R}_x(-\theta(t))\tilde{\boldsymbol{m}}(0), \tag{2.12}$$

where

$$\theta(t) = \gamma \int_0^t B_1(t')dt' \tag{2.13}$$

is called the *flip angle*; and the matrix

$$\boldsymbol{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{2.14}$$

corresponds to the counterclockwise rotation about the x-axis by an angle of θ. Therefore, the magnetization vector can be tilted away from the z-axis by the applying the rf-field (the term 'radio frequency' is chosen due to the fact that the frequency of the field ω_0 falls in the typical frequency range of radio transmission). Moreover, if we want to flip the equilibrium magnetization to the xy plane, assuming $B_1 = 50\mu T$ being a constant, the duration of the so-called 90° *rf-pulse* is given by

$$\tau_{\frac{\pi}{2}} = \frac{\pi/2}{\gamma B_1} = 0.13\text{ms}. \tag{2.15}$$

The process of bring bringing the $\boldsymbol{m}$ vector onto the xy plane is known as *excitation*. As we can see the duration of this process is a very short[3] so it is treated as approximately instantaneous throughout the rest of this thesis (Figure 2.3).
The goal of creating a transverse magnetization is now achieved and we shall proceed to explain how a signal is generated from it.

[2] In practice, the time profile of $B_1(t)$ approximately takes the form of a sinc function.
[3] In contrast, a typical diffusion encoding gradients (introduced later) lasts tens of milliseconds.

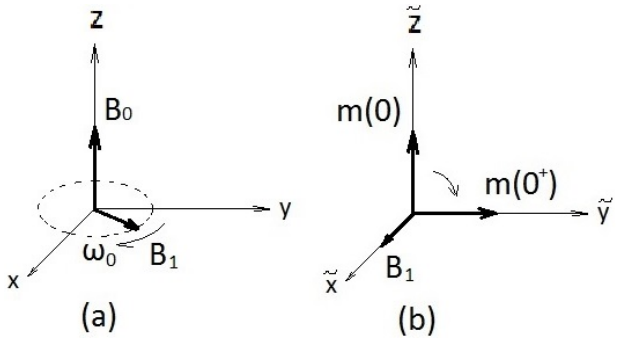


Figure 2.3: (a) Magnetic fields (measured in the lab frame): B_0, the static field; B_1, a rf-pulse. (b) Excitation due to a 90° rf-pulse.

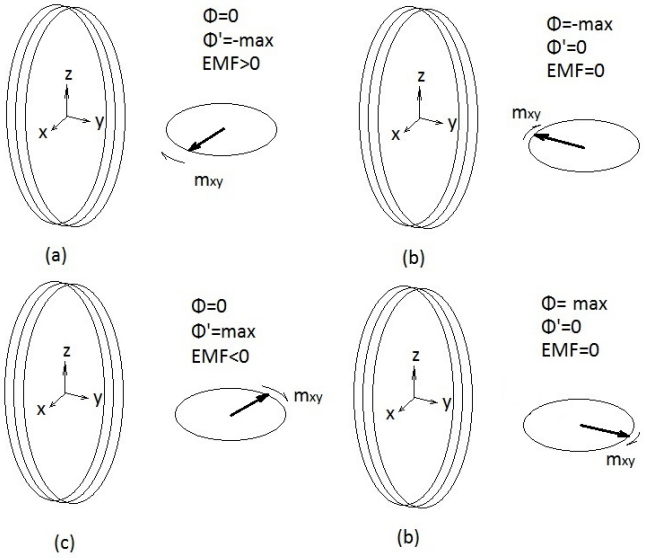


Figure 2.4: Oscillating EMF detected by the receiver coil.

2.2.3 Receiver Coil: measuring signals

A receiver coil surrounding the sample is placed in a MRI scanner in order to detect the MR signal. The orientation of the coil is orthogonal to the main field. Once the transverse magnetization is created by the 90° rf-pulse, it precesses about the z-axis at Larmor frequency ω_0 in the laboratory frame. As a result, a periodic magnetic field exists inside the receiver coil, which produces a measurable electromotive force (EMF) according to Faraday induction law (Figure 2.4). This EMF is called a *MR signal*.

The aforementioned EMF can be expressed mathematically. Let Ω be the whole sample and the proton density distribution is given by the function $\rho(\boldsymbol{r})$. Without loss of generality we let the axis of the receiver coil be the y-axis. The transverse magnetization density rotates at Larmor frequency (equation 2.5). The magnetic flux goes through the cross section of the coil and is proportional to the y-component pf the total transverse magnetization in the sample,

$$\Phi(t) \propto \int_{\Omega} \rho(\boldsymbol{r}) \mathrm{Im}(m_{xy}(t)) d\boldsymbol{r}. \tag{2.16}$$

The proportionality constant depends on the area of the cross section and the number turns of the receiver coil. The induced EMF, according to Faraday's induction law, is written as

$$\mathrm{EMF}(t) = -\frac{d\Phi}{dt} \propto \frac{d}{dt} \int_{\Omega} \rho(\boldsymbol{r}) m_0 \sin(\omega_0 t) d\boldsymbol{r} \propto \int_{\Omega} \rho(\boldsymbol{r}) \cos(\omega_0 t) d\boldsymbol{r}. \tag{2.17}$$

For simplicity the cosine function is replaced by a complex exponential. Dropping the proportional sign the signal measured at time t can be written as

$$E(t) = \int_{\Omega} d\boldsymbol{r} \rho(\boldsymbol{r}) e^{i\omega_0 t} = e^{i\omega_0 t} \int_{\Omega} d\boldsymbol{r} \rho(\boldsymbol{r}). \tag{2.18}$$

The crucial observation here is that the complex exponential factor is taken out of the integral. This is due to the fact that the local precession frequency is uniform across the whole sample.

Therefore, the signal generated by spins in a static field is just a sinusoidal function with a single frequency. This is not very useful in application since the only information we can extract from this signal is the total number of protons. To address this issue, inhomogeneous fields are necessary for encoding spatial information, in other words, producing an image.

2.3 The Gradient Field: Forming an Image

The most common inhomogeneous field is the *gradient magnetic field*. The spatial profile of a gradient field strength satisfies a linear function at any time. In Cartesian coordinates, the general expression for its field strength is given by,

$$B_g(\boldsymbol{r}, t) = \langle \boldsymbol{G}(t) \cdot \boldsymbol{r} \rangle = G_x(t) r_x + G_y(t) r_y + G_z(t) r_z, \tag{2.19}$$

where $\boldsymbol{G}(t) = (G_x(t), G_y(t), G_z(t))^T$ is called the gradient vector.

It is worth pointing out the difference between the magnetic field vector and the gradient vector: the former is always in the $z-$direction; while the later, orthogonal to the isosurfaces of the field strength, can be in arbitrary direction (Figure 2.5).

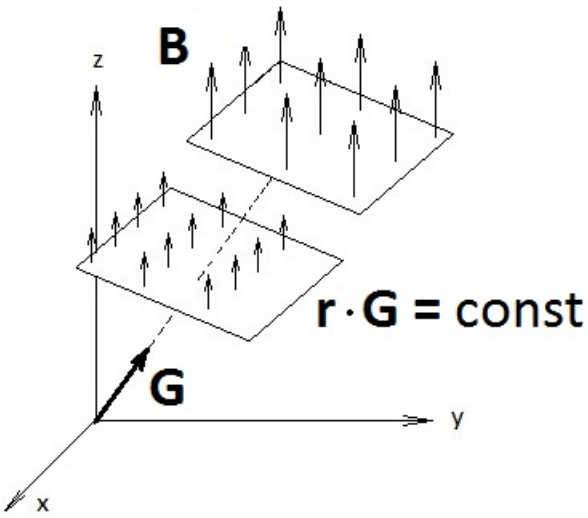


Figure 2.5: A general gradient field: $\boldsymbol{G}$, the gradient vector; $\boldsymbol{B}$, magnetic field vector; $\boldsymbol{r} \cdot \boldsymbol{G} = \text{const}$, field strength isosuraces.

Imaging equation In the rotating reference frame, the accumulated phase of the spins in the vicinity of $\boldsymbol{r}$ during time interval $[0, t]$ is given by

$$\begin{aligned} \varphi(\boldsymbol{r}, t) &= \int_0^t dt' B_g(\boldsymbol{r}, t') = \int_0^t dt' (G_x(t') r_x + G_y(t') r_y + G_z(t') r_z) \\ &= k_x r_x + k_y r_y + k_z r_z = \langle \boldsymbol{k}(t) \cdot \boldsymbol{r} \rangle, \end{aligned} \tag{2.20}$$

where

$$\boldsymbol{k}(t) = (k_x(t), k_y(t), k_z(t))^T = \int_0^t \boldsymbol{G}(t')dt'. \tag{2.21}$$

$\boldsymbol{k}(t)$ is called the *wave vector* and its components are referred to as *spatial frequencies*, for reasons which shall soon become clear. The set of all possible values of $\boldsymbol{k} = (k_x, k_y, k_z)$ is referred to as the *k-space*.

Therefore, the transverse magnetic moments of the spins in the vicinity of $\boldsymbol{r}$ are proportional to $\exp(i\varphi(\boldsymbol{r}, t))$. Let $\rho(\boldsymbol{r})$ be the spin density distribution, the signal measured at time t is given by the net transverse magnetization,

$$S(\boldsymbol{k}) = \int_\Omega d\boldsymbol{r}\rho(\boldsymbol{r})e^{i\langle \boldsymbol{k}\cdot\boldsymbol{r}\rangle}. \tag{2.22}$$

Equation 2.22 is called the *imaging equation*. We have shown that the signal is the Fourier transform of the proton density function. By Fourier inversion theorem, the function $\rho(\boldsymbol{r})$ can be reconstructed by taking the inverse Fourier transform of $S(\boldsymbol{k})$. Ignoring the proportionality constant, one has

$$\rho(\boldsymbol{r}) = \int_\Omega d\boldsymbol{k}S(\boldsymbol{k})e^{-i\langle \boldsymbol{k}\cdot\boldsymbol{r}\rangle}. \tag{2.23}$$

In practice, a sample of sufficient number of $S(\boldsymbol{k})$ values, known as a *k-space trajectory*, is need for computing the discrete Fourier inverse transform. It is clear that to acquire higher resolution of images one need a larger size k-space sample points. To execute such sampling process, an *imaging sequence* is applied, which consists of delicately designed combination of gradients and rf-pulses generated at precise moments. Readers interested in greater details can consult [6].

2.4 Relaxation

In reality, the simple Bloch equation is an incomplete description for the time evolution of the magnetization vector. In addition to Larmor precession, experimental evidence indicates that immediately after excitation, the magnetization vector starts gradually restoring to its equilibrium value (and direction). Such phenomena is referred to as *relaxation*, which can be classified into the following two categories:

- T_1 relaxation, also known as the *longitudinal* or *spin-lattice* relaxation. The effect of T_1 relaxation is that the longitudinal magnetization regrows from zero to its full strength i.e., equilibrium strength $\boldsymbol{m}_0$. This is caused by the irreversible transfer of energy from the protons to the surrounding environment;
- T_2 relaxation, also known as the *transverse* or *spin-spin* relaxation. The effect of T_2 relaxation is that the transverse magnetization decays to zero. This occurs due to the loss of phase cohesion among nuclei.

In the rotating frame the differential equations for modeling those two types of relaxations are:

$$\frac{d\tilde{m}_z}{dt} = \frac{1}{T_1}(m_0 - \tilde{m}_z), \quad \frac{d\tilde{m}_{xy}}{dt} = -\frac{\tilde{m}_{xy}}{T_2}, \tag{2.24}$$

where T_1 and T_2 are time constants depending on the type of tissues. For the same tissue T_1 is always greater than T_2. Typically T_2 is around 100ms while T_1 can range from 200ms to 2s.

If the initial condition for $\tilde{\boldsymbol{m}}$ is $(0, m_0, 0)$, the solutions are simply

$$\tilde{m}_x(t) = 0, \quad \tilde{m}_y(t) = m_0 e^{-\frac{t}{T_2}}, \quad \tilde{m}_z(t) = m_0(1 - e^{-\frac{t}{T_1}}). \tag{2.25}$$

The *full Bloch equation* in the laboratory frame is given by

$$\frac{d\boldsymbol{m}}{dt} = \gamma \boldsymbol{m} \times \boldsymbol{B} - \frac{1}{T_2}(m_x \hat{\boldsymbol{x}} + m_y \hat{\boldsymbol{y}}) + \frac{1}{T_1}(m_0 - m_z) m_x \hat{\boldsymbol{z}}. \tag{2.26}$$

Since time constants T_1 and T_2 are both tissue-dependent[4], both relaxation mechanisms are utilized to create contrasts in MR images.

[4]Typical magnitudes of T_1 and T_2 are hundreds and tens of milliseconds respectively, see [5] for a study on relaxation times of various tissues.

2.5 Spin Echoes and Diffusion

Twelve years after the discovery of NMR in 1938, it was Hahn who first recognized its application for studying molecule diffusion[15]. Here we only introduce the phenomenon of spin echo and the diffusion process qualitatively. The relevant mathematical machinery will be introduced in later sections.

2.5.1 Spin echo experiment

We present a description of the spin echo experiment to demonstrate the effect of an 180° rf-pulse.

Consider a sample of water molecules experiencing a time-independent inhomogeneous magnetic field $(B_0 + \Delta B(\boldsymbol{r}))\hat{\boldsymbol{z}}$ (assuming $\Delta B(\boldsymbol{r}) \ll B_0$ everywhere). At time $t = 0$ the usual 90° rf-pulse is applied and a measurable signal appears. It can be observed experimentally, that the signal gradually vanishes as time elapses. However, the signal decay rate can not be fully explained by the T_1 and T_2 relaxation effects.

Attempting to recover the signal loss, Hahn applied a second rf-pulse with a 180° flip angle[5] at $t = TE/2$. A measurable signal reemerged at $t = TE$, which is known as the *spin echo phenomenon*. The 180° rf-pulse is sometimes referred to as the *refocusing* pulse. The reemerged signal is appropriately named an *echo* and TE is the *echo time*.

Let us examine the mechanism of signal decay and reemergence in the rotating frame (the tildes over the axis labels in subsection 2.2.1 are dropped for notation simplicity). A schematic explanation is shown in Figure 2.6. Without loss of generality, at $t = 0$ all spins are assumed to be flipped towards the positive $y-$axis by the 90° pulse. The spins within the vicinity of location $\boldsymbol{r}$ immediately start to precess at the local Larmor frequency, $\gamma\Delta B(\boldsymbol{r})$. Thus, their accumulated phase at time t, $t < TE^-/2$ is given by

$$\phi(\boldsymbol{r}, t) = \pi/2 - \gamma\Delta B(\boldsymbol{r})t. \tag{2.27}$$

The of the 180° pulse rotates the spin vectors by 180° about the x-axis, phase-wise this means

$$\phi(\boldsymbol{r}, TE^+/2) = -\phi(\boldsymbol{r}, TE^-/2) = -\pi/2 + \gamma\Delta B(\boldsymbol{r})TE/2. \tag{2.28}$$

[5]This can be achieved, for example, by doubling either the duration or strength of the excitation pulse

The spins proceed to precess at their local Larmor frequencies afterwards. Thus for $t > TE/2$, one has

$$\phi(\boldsymbol{r}, t) = \phi(\boldsymbol{r}, TE^+/2) - \gamma \Delta B(\boldsymbol{r})(t - TE/2) = -\pi/2 + \gamma \Delta B(\boldsymbol{r})(TE - t). \tag{2.29}$$

The second term vanishes when $t = TE$ and

$$\phi(\boldsymbol{r}, TE) = -\pi/2 \quad \text{for all } \boldsymbol{r}. \tag{2.30}$$

Therefore, an echo is observed at echo time.

The above analysis suggests all spin vectors realign at echo time, predicting an identical signal. However, experimental evidence shows that echo is still weaker than the initial signal. This discrepancy is due to the unaccounted diffusion of the spin-bearing particles. Qualitatively speaking, the precise cancellation of the second term in equation (27) relies on the assumption that the spin stays at location $\boldsymbol{r}$ for the whole duration of the experiment. On the other hand, if the spins are mobile, the spins will experience different local precession frequencies along their trajectories. This incomplete phase cancellation is referred to as the *dephase*.

Overall, the moral of the story is that the more restricted spin diffusion is, the stronger signal measured in the spin echo experiment. It is therefore possible to 'label' the trajectories of water molecules via a spatially dependent magnetic field. The ultimate goal of Diffusion MRI is to establish the relationship between the geometry of neural pathways and the diffusion weighted MR signal.

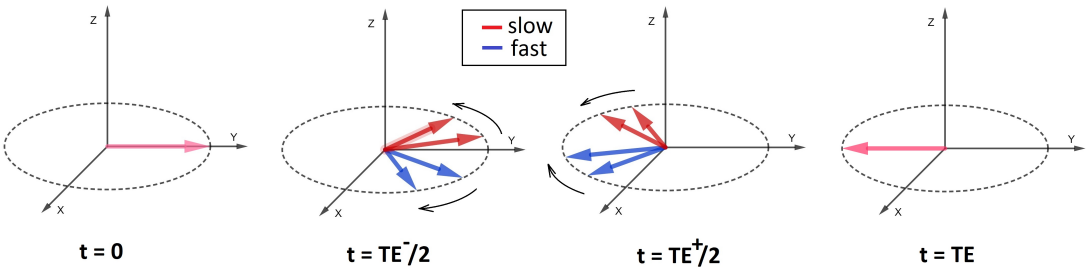


Figure 2.6: (a) 90° pulse applied at $t = 0$, all spin orientations are set as y^+; (b) spins loses phase cohesion due to field inhomogeneity; (c) 180° pulse applied at $t = TE/2$, spins rotate 180° about x-axis; (d) spins refocus at $t = TE$.

2.5.2 Diffusion: self vs mutual

The term 'diffusion' is used with ambiguity in scientific literature. In the context of molecular motions, diffusion may refer to mutual diffusion or self-diffusion, which are two distinct physical phenomena. It is therefore worth clarifying the difference between them. Readers interested in the exact mechanism of both types of diffusion can find more details in [30].

Mutual-diffusion occurs in systems containing multiple substances. A concentration inhomogeneity (also known as gradient) of any substance causes mass fluxes in an effort to nullify the gradient. For example, one can consider the process of ink being poured into clear water. The mutual diffusion coefficient, $D_M(m^2 \cdot s^{-1})$, is used to characterize how sensitive the mass flux is to the concentration gradient.

Self-diffusion is the random (stochastic) translational motion of molecules due to their thermal energy and collisions with surrounding molecules or boundaries. The individual self-diffusing trajectories are highly irregular and unpredictable while their statistical properties can be well characterized. For example, no net displacement is observed so the diffusing substance makes no net motion. In addition, molecules starting from the same location 'spread' out over time and a parameter $D(\mathrm{m}^2 \cdot \mathrm{s}^{-1})$, known as the self-diffusion coefficient, is used to quantify how fast such 'spreading' takes place. The magnitude of self-diffusion coefficient ranges from $10^{-20}\mathrm{m}^2 \cdot \mathrm{s}^{-1}$ for solids to $1\mathrm{m}^2 \cdot \mathrm{s}^{-1}$ for gases. We are mostly interested in the diffusion of water molecules, whose diffusion coefficient $D = 2.3 \times 10^{-9}\mathrm{m}^2 \cdot \mathrm{s}^{-1}$ at room temperature.

2.6 Mathematical machinery

Ever since the discovery of Brownian motion in 1827, finding an appropriate mathematical model for such phenomena has attracted great attention from biologists, physicists and mathematicians alike. In this section, we introduce two of the most prominent tools, namely the theories of diffusion equation and Markov processes.

2.6.1 Diffusion equation

Derivation We first derive the diffusion equation in the context of mutual diffusion.

Let A, B be two types of substance exist in a solution. As a classical example, one may think of A as pollen grains and B as water. Let $c(\boldsymbol{r}, t)$ be the time- and spatially-dependent concentration of A, where c has unit $mol \cdot m^{-3}$. The *mass flux density* $\boldsymbol{J}(\boldsymbol{r}, t)$, a time-dependent vector field describes the rate of transport of substance A. More precisely, consider an infinitesimal surface, with area ΔS, passing through point $\boldsymbol{r}$. The amount of A particles passing through the tiny surface during time interval $[t, t + \Delta t)$ is equal to the magnitude of $\boldsymbol{J}\Delta S\Delta t$ when ΔS and Δt are small. The direction of $\boldsymbol{J}$ is orthogonal to the surface ΔS. Mass conservation gives rise to the *continuity equation*,

$$\frac{\partial c(\boldsymbol{r}, t)}{\partial t} + \nabla \cdot \boldsymbol{J} = 0, \tag{2.31}$$

where $\nabla\cdot$ is the divergence operator.

As mentioned in the previous section, mutual diffusion means particles relocate due to concentration imbalance. Assuming no net flow (bulk motion) exists in the system, the most commonly used *constitutive relation* linking the mass concentration and its flux is *Fick's law*. The relation states that the flux density is a linear function of the negative concentration gradient:

$$\boldsymbol{J} = -D_M \nabla c(\boldsymbol{r}, t), \tag{2.32}$$

where ∇ is the gradient operator and D_M is the mutual diffusion coefficient matrix, known as the *diffusion tensor*. In general, D_M is a symmetric positive-definite and can be spatially dependent. The diffusion is said to be *isotropic* if D_M is proportional to the identity matrix. As an example, for diffusion of spherical particles through a liquid with low Reynolds

number, the diffusion coefficient is given by the *Stokes-Einstein relation*[6]:

$$D_M = \frac{k_B T}{6\pi\eta r}, \tag{2.33}$$

where k_B is the Boltzmann's constant; T is the absolute temperature; η is the viscosity of the liquid; and r is the spherical radius of the particle.

Combining equations 2.31 and 2.32 yields the well-known *diffusion equation*:

$$\frac{\partial c(\boldsymbol{r}, t)}{\partial t} = \nabla \cdot (D_M \nabla c(\boldsymbol{r}, t)). \tag{2.34}$$

In the case of isotropic diffusion, equation 2.34 is reduced to

$$\frac{\partial c(\boldsymbol{r}, t)}{\partial t} = D_M \nabla^2 c(\boldsymbol{r}, t), \tag{2.35}$$

where ∇^2 is the Laplacian operator.

Boundary conditions In applications, additional information is often needed to find meaningful solutions of the diffusion equation, which may include *initial conditions* (ICs) and *boundary conditions* (BCs). The initial condition $c(\boldsymbol{r}, 0) = c_0(\boldsymbol{r})$ specifies the concentration profile prior to diffusion. The boundary conditions depend on the physical and chemical nature of the diffusing domain. Common BCs include:

- *Dirichlet* or the *1st kind*, where the value of $c(\boldsymbol{r}, t)$ is prescribed on the boundary. The *homogeneous* case reads

  $$c(\boldsymbol{r}, t) = 0 \quad \text{for} \quad \boldsymbol{r} \in \partial\Omega \tag{2.36}$$

  and is associated with an *absorbing* boundary. Physically, one can the substance disappears once it touches the boundary, perhaps due to chemical reactions.

- *Neumann* or the *2nd kind*, where the normal component of the boundary mass flux is prescribed. This is equivalent to specifying the normal derivative of $c(\boldsymbol{r}, t)$. The homogeneous Neumann BC reads

  $$D_M \frac{\partial c}{\partial n}(\boldsymbol{r}, t) = 0 \quad \text{for} \quad \boldsymbol{r} \in \partial\Omega, \tag{2.37}$$

[6]Notice that in the formula diffusion coefficient reflects a compromise between the internal energy of particles and 'stickiness' of their surrounding medium.

where $\boldsymbol{n}$ is the unit exterior normal vector at the boundary. The homogeneous Neumann BC applies when the boundary is completely insulating and as a result, the total mass is conserved.

- *Robin* or the *3rd kind*, where the linear combination of $c(\boldsymbol{r}, t)$ and $\frac{\partial c}{\partial n}(\boldsymbol{r}, t)$ is prescribed. In diffusion NMR the most encountered Robin BC reads

$$D_M \frac{\partial c}{\partial n}(\boldsymbol{r}, t) + hc(\boldsymbol{r}, t) = 0 \quad \text{for} \quad \boldsymbol{r} \in \partial\Omega, \tag{2.38}$$

  where the parameter h is a measure of boundary permeability, called the *surface relaxivity* in MRI literature. Physically Robin BC's correspond to scenarios where a fixed portion of diffusing substance in the vicinity of the boundary is absorbed while the rest is kept inside the region. Robin BCs reduces to the Neumann BCs When $h = 0$.

Analytical solutions of the diffusion equation in simple geometries are recorded in the classical text by J.Crank [8], mostly solved by separation of variables and integral transforms[7]. Numerically, finite difference and finite element solvers are widely available as well.

[7] Moreover, [33], a recently published book contains over a thousand analytical solutions.

2.6.2 Markov processes

The diffusion equations is a continuum model. On the microscopic level, the self-diffusion of each water molecule is influenced by the constant collisions (of average frequency around 10^{21}Hz) from the surrounding water molecules. The complexity of this many-body problem makes it hopeless to look for deterministic solutions. On the other hand, at experimental time scales, the collision forces have short memory[8] and thus can be treated as Gaussian white noise. Thus, at 'large' times, molecule positions can be treated as random variables. The observed trajectories are realizations of a *stochastic (or random) process* , a continuum of random variables.

We now present the stochastic formalism of diffusion. Let $\Omega \subset \mathbb{R}^d (d = 1, 2, 3)$ be an open and connected domain. The position process $\left\{\boldsymbol{r}(t)\right\}_{t\geqslant 0}$ is modeled as a *continuous random walk* with potential interactions with the boundary of Ω. The aforementioned self-diffusion coefficient D is defined as

$$D = \lim_{t\to\infty} \frac{\mathbb{E}\left[\left\|\boldsymbol{r}(t) - \boldsymbol{r}_0\right\|^2\right]}{2dt}, \tag{2.39}$$

where $\mathbb{E}[\cdot]$ is the expectation operator. Therefore, the self-diffusion coefficient represents the long-time growth rate of the second moment of the displacement size.

A random walk is an example of *Markov processes*. Physically, the Markov assumption asserts that the all future states of the process are statistically independent to its history with the possible exception of the current state. This is a well-researched area in probability theory and the literature on the topic is abundant[9].

Characterization Only two functions are needed in order to characterise such processes completely, which are the *initial distribution* $\rho(\boldsymbol{r}_0)$ and the *transition probability* $G_t(\boldsymbol{r}, \boldsymbol{r}')$. Typically, the initial distribution is assumed to be uniform. The transition probability is also sometimes called *the diffusion propagator* or *the Green's function*. $G_t(\boldsymbol{r}, \boldsymbol{r}')$ has the following interpretation: Given that the initial position is $\boldsymbol{r}$, the conditional probability of $\boldsymbol{r}(t)$ to be found in the ε-ball centered at $\boldsymbol{r}'$ is proportional to $G_t(\boldsymbol{r}, \boldsymbol{r}')d\boldsymbol{r}'$ as $\varepsilon \to 0$.

[8] A water molecule almost never gets the chance of moving freely for any experimentally distinguishable time-scale. That means almost right after one collision, another one occurs. The collision history is therefore quickly 'forgotten'.

[9] See [20] for a full a rigorous treatment and [22],[35] for applications in natural sciences.

To see why the knowledge of $\rho(\boldsymbol{r}_0)$ and $G_t(\boldsymbol{r}, \boldsymbol{r}')$ is sufficient for characterizing the process, we consider a finite set of times of ascending order,

$$0 < t_1 < \cdots < t_n,$$

where $n = 1, 2, \ldots$. The *$n-$th order probability density function (pdf)* of the random process $\left\{\boldsymbol{r}(t)\right\}_{t\geqslant 0}$ is defined as the joint density function of variates $(\boldsymbol{r}_0, \boldsymbol{r}(t_1), \ldots, \boldsymbol{r}(t_n))$, known as a *discrete skeleton* of the process $\left\{\boldsymbol{r}(t)\right\}_{t\geqslant 0}$. By applying the law of conditional probabilities and invoking the Markov assumption, one can write the pdf of any order in terms of $\rho(\boldsymbol{r}_0)$ and $G_t(\boldsymbol{r}, \boldsymbol{r}')$:

$$f_n(\boldsymbol{r}_0, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_n; t_1, \ldots, t_n) = \rho(\boldsymbol{r}_0) G_{t_1}(\boldsymbol{r}_0, \boldsymbol{r}_1) \cdots G_{t_n - t_{n-1}}(\boldsymbol{r}_{n-1}, \boldsymbol{r}_n). \tag{2.40}$$

The transition probabilities can be thought as the 'building blocks' of Markov processes. Furthermore, the transition probability satisfies the *Chapman-Kolmogorov equation*, a semigroup property,

$$G_t(\boldsymbol{r}_0, \boldsymbol{r}_2) = \int_\Omega d\boldsymbol{r}_1 G_{t_1}(\boldsymbol{r}_0, \boldsymbol{r}_1) G_{t-t_1}(\boldsymbol{r}_1, \boldsymbol{r}_2) \quad \text{for any } t_1 \text{ and } t \text{ s.t.} \quad 0 < t_1 < t. \tag{2.41}$$

In applications, the transition probability is the most crucial quantity for studying a Markov process since it governs the time evolution. In next discussion, it will be shown that $G_t(\boldsymbol{r}, \boldsymbol{r'})$ satisfies the diffusion equation as well.

2.7 Green's Function

The relation between the micro- and macroscopic formalism of diffusion explains why the transition probability $G_t(\boldsymbol{r}, \boldsymbol{r}')$ is also referred to as the Green's function. A heuristic explanation is given below[10].

Imagine that one walker is randomly chosen from a large ensemble of walkers. The walker is labeled so it can be located. We subsequently 'close our eyes' for a time duration of t. At time t we open our eyes and hope to find the walker in the vicinity of $\boldsymbol{r}$. The probability of success is proportional to the first order pdf,

$$f(\boldsymbol{r}, t) = \int_\Omega d\boldsymbol{r}_0 \rho(\boldsymbol{r}_0) G_t(\boldsymbol{r}_0, \boldsymbol{r}). \tag{2.42}$$

Notice that the initial position $\boldsymbol{r}_0$ is integrated over since the walker is randomly sampled.

As we have no knowledge of the walker's movement preferences, we can only look for more populated regions in the hope of increasing our probability of success. Hence, it is reasonable to assume that such probability is proportional to the concentration $c(\boldsymbol{r}, t)$,

$$c(\boldsymbol{r}, t) = k f(\boldsymbol{r}, t), \quad k > 0. \tag{2.43}$$

Substituting equations 2.42,2.43 into the equation 2.35 and replacing D_M by D, one gets

$$0 = \frac{\partial c(\boldsymbol{r}, t)}{\partial t} - D\nabla^2 c(\boldsymbol{r}, t) = \int_\Omega d\boldsymbol{r}_0 k \rho(\boldsymbol{r}_0) \left(\frac{\partial G_t(\boldsymbol{r}_0, \boldsymbol{r})}{\partial t} - D\nabla^2_{\boldsymbol{r}'} G_t(\boldsymbol{r}_0, \boldsymbol{r}) \right),$$

where $\nabla^2_{\boldsymbol{r}'}$ denotes the Laplacian is taken with respect to the second argument of G_t.

Let us assume that $\rho(\boldsymbol{r}_0)$ is smooth and compactly supported; $G_t(\boldsymbol{r}, \boldsymbol{r}')$ is C^1 in t and C^2 in $\boldsymbol{r}'$[11]; and Ω is arbitrary. Invoking the fundamental lemma of calculus of variation, one concludes that the integrand can only be zero. Therefore, $G_t(\boldsymbol{r}, \boldsymbol{r}')$ satisfies the diffusion equation,

$$\frac{\partial G_t(\boldsymbol{r}_0, \boldsymbol{r})}{\partial t} = D\nabla^2_{\boldsymbol{r}'} G_t(\boldsymbol{r}_0, \boldsymbol{r}). \tag{2.44}$$

[10] A more rigorous derivation using the Kramers–Moyal expansion can be found in [22].
[11] C^k means $k^{-\text{th}}$ order continuously differentiable.

The initial condition of $G_t(\boldsymbol{r}, \boldsymbol{r}')$ should be the Dirac delta function centred at $\boldsymbol{r}$,

$$G_{o^+}(\boldsymbol{r}, \boldsymbol{r}') = \delta(\boldsymbol{r}' - \boldsymbol{r}). \tag{2.45}$$

To see this is true, we first notice that according to the probabilistic interpretation, $G_{o^+}(\boldsymbol{r}, \boldsymbol{r}')$ is normalized and equals zero everywhere else except for $\boldsymbol{r} = \boldsymbol{r}'$, where it blows up to infinity. Setting $t = 0$ in equation 2.42 yields

$$f(\boldsymbol{r}, 0) = \rho(\boldsymbol{r}_0) = \int_\Omega d\boldsymbol{r}_0 \rho(\boldsymbol{r}_0) G_0(\boldsymbol{r}_0, \boldsymbol{r}), \tag{2.46}$$

which is exactly the sifting property statement of the delta function.

Combining equations 2.34 and 2.45, we have shown that the transition probability is in fact the Green's function of the diffusion equation. Unless stated otherwise, the term 'diffusion' in the rest of the thesis means self-diffusion as diffusion NMR/MRI is mostly applied for studying self-diffusion.

2.7.1 Boundary Conditions for Propagators

Boundary conditions introduced in subsection 2.6.1 can be imposed for Green's functions. Their probabilistic interpretation are examined below. We restrict our discussion to only homogeneous BC's due to their relevance to diffusion NMR.

Given that a particle's initial position is $\boldsymbol{r}_0 \in \Omega$, we are interested in the event where the particle reaches the boundary for the first time at time t . Let $P_t(\boldsymbol{r}_0, \boldsymbol{r})$ be a function such that, the probability of the event of interest can be expressed as a surface integral over $\partial\Omega$,

$$\int_{\partial\Omega} P_t(\boldsymbol{r}_0, \boldsymbol{r}) d\boldsymbol{r}. \tag{2.47}$$

The integration over the second argument of P_t means that we are not interested in the position of the arrival point.

On the other hand, consider the opposite of the aforementioned event, namely, the particle stays in the *interior* of Ω for the whole duration of $(0, t)$. Its probability is a volume integral of propagator over Ω,

$$\int_\Omega G_t(\boldsymbol{r}_0, \boldsymbol{r}) d\boldsymbol{r}. \tag{2.48}$$

Similar to before, the final position is integrated over.

By design, integrals 2.47 and 2.48 should add up to 1,

$$\int_{\partial\Omega} P_t(\boldsymbol{r}_0, \boldsymbol{r}) d\boldsymbol{r} + \int_{\Omega} G_t(\boldsymbol{r}_0, \boldsymbol{r}) d\boldsymbol{r} = 1. \tag{2.49}$$

Differentiating with respect to time yields

$$\int_{\partial\Omega} \frac{\partial P_t(\boldsymbol{r}_0, \boldsymbol{r})}{\partial t} d\boldsymbol{r} + \int_{\Omega} \frac{\partial G_t(\boldsymbol{r}_0, \boldsymbol{r})}{\partial t} d\boldsymbol{r} = 0. \tag{2.50}$$

Since $G_t(\boldsymbol{r}_0, \boldsymbol{r})$ satisfies the diffusion equation, the time derivative of G_t can replaced by its Laplacian (w.r.t.$\boldsymbol{r}$) times D. Applying the divergence theorem, one can rewrite the second integral of equation 2.50 as

$$\int_{\Omega} \frac{\partial G_t(\boldsymbol{r}_0, \boldsymbol{r})}{\partial t} d\boldsymbol{r} = \int_{\Omega} D\nabla \cdot \nabla G_t(\boldsymbol{r}_0, \boldsymbol{r}) d\boldsymbol{r} = \int_{\partial\Omega} D\nabla G_t(\boldsymbol{r}_0, \boldsymbol{r}) \cdot \boldsymbol{n} d\boldsymbol{r} = \int_{\partial\Omega} D \frac{\partial G_t(\boldsymbol{r}_0, \boldsymbol{r})}{\partial n} d\boldsymbol{r},$$

where $\boldsymbol{n}$ is the outward normal unit vector of the surface $\partial\Omega$. Therefore,

$$\int_{\partial\Omega} \left(\frac{\partial P_t(\boldsymbol{r}_0, \boldsymbol{r})}{\partial t} + D \frac{\partial G_t(\boldsymbol{r}_0, \boldsymbol{r})}{\partial n} \right) d\boldsymbol{r} = 0. \tag{2.51}$$

Assuming the integrand in equation 2.51 is continuous and $\partial\Omega$ arbitrary, we conclude that

$$\frac{\partial P_t(\boldsymbol{r}_0, \boldsymbol{r})}{\partial t} = -D \frac{\partial G_t(\boldsymbol{r}_0, \boldsymbol{r})}{\partial n} \quad \text{for} \quad \boldsymbol{r} \in \partial\Omega. \tag{2.52}$$

We are can now examine the three main types of boundary conditions individually ($\boldsymbol{r} \in \partial\Omega$ for all three cases).

- BC of the 1st kind,

  $$G_t(\boldsymbol{r}_0, \boldsymbol{r}) = 0. \tag{2.53}$$

  Rigorously speaking, the arguments of G_t can only be interior points of $\partial\Omega$ for discussing its probabilistic interpretation. Therefore, equation 2.53 should be understood in the limiting sense where the second argument of G_t approaches the boundary,

  $$G_t(\boldsymbol{r}_0, \boldsymbol{r}) = \lim_{\substack{\boldsymbol{r}' \in \Omega \\ \text{dist}(\boldsymbol{r}', \partial\Omega) \to 0}} G_t(\boldsymbol{r}_0, \boldsymbol{r}') = 0.$$

  The Dirichlet BC suggests that particles are not allowed to stay close to the boundary. If they do, their Brownian motion shall be terminated. Imposing 'artificial' Dirichlet BC is a common technique for solving first exit problems, since one is only interested in the part of the trajectory before the first visit to a certain region.

- BC of the 2nd kind:
$$D\frac{\partial G_t(\boldsymbol{r}_0,\boldsymbol{r})}{\partial n}=0. \tag{2.54}$$
The time derivative of P_t equals zero according to equation 2.52, which means that P_t must be time independent. Since $P_{0^+}(\boldsymbol{r}_0,\boldsymbol{r})=0$ (the particle can not jump to the boundary instantaneously), $P_t(\boldsymbol{r}_0,\boldsymbol{r})\equiv 0$. This implies that at no time can the particle ever exit the confining domain. Therefore, a reflection must occur every time the walker collide with the boundary.

- BC of the 3rd kind:
$$D\frac{\partial G_t(\boldsymbol{r}_0,\boldsymbol{r})}{\partial n}+hG_t(\boldsymbol{r}_0,\boldsymbol{r})=0. \tag{2.55}$$
Using equation 2.52 once again, one obtains
$$\frac{\partial P_t(\boldsymbol{r}_0,\boldsymbol{r})}{\partial t}=hG_t(\boldsymbol{r}_0,\boldsymbol{r}). \tag{2.56}$$
Since here $G_t(\boldsymbol{r}_0,\boldsymbol{r})$ can be positive, so is the case for the time derivative of P_t. This means the P_t (the probability of escaping) grows over time and becomes positive. Therefore, termination and reflection are both possible at the boundary.

Some problems have *mixed boundary conditions*, the case where BCs of different kinds hold for different sections of the boundary.

2.7.2 Gaussian Propagator and Wiener Process

The most famous example of the transition probability is the *Gaussian heat kernel*, also known as the *fundamental solution* of the diffusion equation. In the case of unrestricted diffusion, G_t is defined in terms of the solution of the following initial value problem:
$$\begin{aligned}\frac{\partial G_t(\boldsymbol{r},\boldsymbol{r}')}{\partial t}&=D\nabla^2 G_t(\boldsymbol{r},\boldsymbol{r}'),\quad \boldsymbol{r}\in\mathbb{R}^d\\ G_t(\boldsymbol{r},\boldsymbol{r}')&=\delta(\boldsymbol{r}-\boldsymbol{r}').\end{aligned} \tag{2.57}$$
This initial value problem can solved by taking the d-dimensional Fourier transform of both sides of equation 2.57. The fundamental solution turns out to be the pdf of a three-dimensional Gaussian (or normal) distribution:
$$G_t(\boldsymbol{r},\boldsymbol{r}')=\frac{1}{(4\pi Dt)^{\frac{d}{2}}}\exp\left(-\frac{\|\boldsymbol{r}-\boldsymbol{r}'\|^2}{4Dt}\right). \tag{2.58}$$

As a result, it is established that the each component of $\boldsymbol{r} - \boldsymbol{r}'$, i.e the displacement during a time interval of length t, is a Gaussian random variable with mean 0 and variance $2Dt$.

The unrestricted diffusion gives rise to the *Wiener process*, arguably the most famous continuous-time stochastic process. The significance of Wiener processes in stochastic processes is arguably comparable to that of Gaussian distributions in elementary probability and statistics. One possible way of defining it is the following:

Definition 1. *The process $\{W_t\}_{t\geqslant 0}$ is a Wiener process if it satisfies the following conditions:*

1. *$W_0 = 0$;*
2. *(independent increment) For $\forall n \in \mathbb{N}$ and any choice of $t_1 < \cdots < t_n$ the random variables*
$$(W_{t_1} - W_{t_0}), (W_{t_2} - W_{t_1}), \ldots, (W_{t_n} - W_{t_{n-1}})$$
are mutually independent;
3. *(Gaussian increment) For $s < t$, random variable $(W_t - W_s)$ is Gaussian with mean 0 and variance $(t - s)$;*
4. *(continuous path) The probability of a realization of $\{W_t\}_{t\geqslant 0}$(aka, a sample path) to be a continuous w.r.t. t is one.*

Some useful properties of Wiener process are:

1. (auto-covariance) $\mathbb{E}[W_{t_1} W_{t_2}] = \min(t_1, t_2)$;
2. (regularity) Sample paths of Wiener process are nowhere-differentiable with probability 1;
3. (Reflection principle) If the path of a Wiener process W_t reaches a value $W_s = a$ at time $t = s$, then the subsequent path after time s has the same distribution as the reflection of the subsequent path about the value a.

A simulated sample path is shown in Figure 2.7. Alternative definitions and additional properties of the Wiener processes can be found in[35].

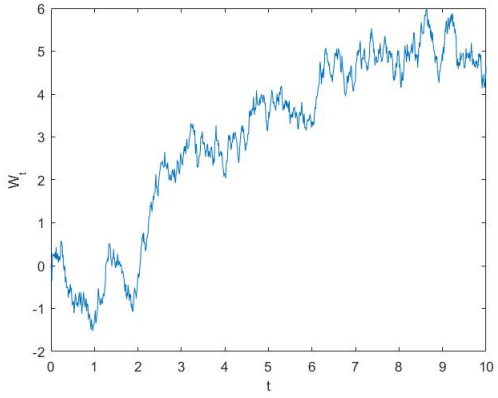


Figure 2.7: Standard Wiener process: a sample path

2.7.3 Summary

To summarize the present and previous sections, here we collect key ingredients required for formulating a general diffusion problem.

Let $\Omega \in \mathbb{R}^d$ be a open and connected domain, where $d = 1, 2, 3$. A *restricted Brownian motion* (RBM) in Ω, $\{\boldsymbol{r}(t)\}_{t \geqslant 0}$, is a Markov process with

- state space $\Omega \cup \partial\Omega$;
- initial distribution $\rho(\boldsymbol{r})_0$, a probability distribution over Ω;
- transition probability $G_t(\boldsymbol{r}, \boldsymbol{r}')$, the Green's function for the diffusion equation defined by equations 2.34,2.45 and some BCs (equation 2.53, 2.54, 2.55).

Our next goal is to combine the theory of NMR with the formulation of restricted Brownian motions.

2.8 Diffusion signal

Armed with the mathematics of diffusion, we are now able to define and calculate the signal attenuation due to spin diffusion (at least theoretically). In particular, the Multiple Correlation Function (MCF) Approach, established by Grebenkov (2007) in[11], is presented in this section.

Statistical definition First, let us assume an infinite ensemble of spin-bearing particles undergoes restricted Brownian motions defined in subsection 2.7.3. In addition, during the time interval $[0, T]$ an general magnetic field Ω, denoted as $B_t(\boldsymbol{r})\hat{\boldsymbol{z}}$, where T is the time when the MR signal is measured.

At any time t, a spin-bearing Brownian particle located at $\boldsymbol{r}(t)$ experiences a local, instantaneous magnetic field strength of $B_t(\boldsymbol{r}(t))$. Consequently, its instantaneous Larmor frequency equals to $\gamma B_t(\boldsymbol{r}(t))$. For simpler notation, the constant γ can be absorbed into $B_t(\boldsymbol{r}(t))$. Therefore, the total phase accumulation at time T is equal to the following time integral,

$$\varphi = \int_0^T dt B_t(\boldsymbol{r}(t)), \tag{2.59}$$

which is a random variable as the trajectory $\boldsymbol{r}(t)$ is stochastic. Hence, at time T, the spin has a transverse magnetic moment of $e^{i\varphi}$. The *diffusion weighted signal* or simply *diffusion signal*[12] is defined as the ensemble average of transverse magnetic moments:

$$S = \mathbb{E}\left[e^{i\varphi}\right], \tag{2.60}$$

where the expectation is taken over all possible trajectories.

Moment expansion In general, this expectation value is very difficult to calculate despite the simple appearance of its definition. This is due to the fact that the expectation is an integral over uncountably-infinite degrees of freedom as well as the extreme complexity of the random Brownian trajectories themselves. By Taylor expansion, the signal can be

[12]In practice, this signal is obtained by taking the ratio of two signals measured with and without the diffusion encoding gradient (introduced later), denoted as E and E_0 respectively. As a result, S sometimes has the notation of E/E_0, which is referred to as the *normalized signal* or *sign attenuation* due to diffusion (as $|S| \leqslant 1$).

formally written in terms of contributions from all moments of φ:

$$\mathbb{E}\left[e^{i\varphi}\right] = \mathbb{E}\left[\sum_{n=0}^{\infty} \frac{(i\varphi)^n}{n!}\right] = \sum_{n=0}^{\infty} \frac{i^n}{n!}\mathbb{E}\left[\varphi^n\right]. \tag{2.61}$$

We turn our attention to the n-th moment of φ, which can be written in terms of a n-fold time integral:

$$\begin{aligned}
\mathbb{E}\left[\varphi^n\right] &= \mathbb{E}\left[\int_0^T dt B_t(\boldsymbol{r}(t))\right]^n \\
&= \int_0^T dt_1 \cdots \int_0^T dt_n \mathbb{E}\left[B_{t_1}(\boldsymbol{r}(t_1)) \ldots B_{t_n}(\boldsymbol{r}(t_n))\right] \\
&= n! \int_0^T dt_1 \int_{t_1}^T dt_2 \cdots \int_{t_{n-1}}^T dt_n \mathbb{E}\left[B_{t_1}(\boldsymbol{r}(t_1)) B_{t_2}(\boldsymbol{r}(t_2)) \ldots B_{t_n}(\boldsymbol{r}(t_n))\right].
\end{aligned} \tag{2.62}$$

Notice that from the second to the third line of the above derivation, the originally unordered dummy variables $t_1, \ldots, t_n$ are arranged into an ascending order. The key observation justifying this step is that the n-variate integrand,

$$\mathbb{E}\left[B_{t_1}(\boldsymbol{r}(t_1)) \ldots B_{t_n}(\boldsymbol{r}(t_n))\right],$$

is symmetric, in the sense that switching any pair of dummy variables has no effect on the integrand's value. The n-dimensional cube $[0,T]^n$ can be partitioned into $n!$ subsets in each of which the dummy variables are always in ascending order[13]. Each one of those subsets contributes equally to the integral over $[0,T]^n$, resulting an extra factor of $n!$.

Multiple Correlation Function Often in practice, the time and spatial dependency of $B_t(\boldsymbol{r})$ is multiplicatively separable,

$$B_t(\boldsymbol{r}(t)) = f(t)B(\boldsymbol{r}), \tag{2.63}$$

where $B(\boldsymbol{r})$ and $f(t)$ are called the *spatial* and the *effective temporal profile* of the magnetic field. Taking $f(t)$ outside of the expectation as it is deterministic, one can rewrite 2.62 as,

$$\mathbb{E}\left[\varphi^n\right] = n! \int_0^T dt_1 f(t_1) \cdots \int_{t_{n-1}}^T dt_n f(t_n) \mathbb{E}\left[B(\boldsymbol{r}(t_1)) \ldots B(\boldsymbol{r}(t_n))\right]. \tag{2.64}$$

[13]For example, for an arbitrary point $(t_1, t_2) \in (0,T)^2$, $2! = 2$ orderings are possible ($t_1 < t_2$, $t_2 < t_1$, ignore equal signs); for $(t_1, t_2, t_3) \in (0,T)^3$, $3! = 6$ orderings are possible ($t_1 < t_2 < t_3$, $t_1 < t_3 < t_2$, $t_2 < t_1 < t_3$, $t_2 < t_3 < t_1$, $t_3 < t_1 < t_2$, $t_3 < t_2 < t_1$); etc.

The integrand above,

$$\mathbb{E}\left[B(\boldsymbol{r}(t_1))\dots B(\boldsymbol{r}(t_n))\right], \tag{2.65}$$

is called the *multiple correlation function* (MCF)[14]. To get a more compact expression, we rewrite equation 2.64 as

$$\mathbb{E}\left[\varphi^n\right] = n!\langle\mathbb{E}\left[B(\boldsymbol{r}(t_1))\cdots B(\boldsymbol{r}(t_n))\right]\rangle_n, \tag{2.66}$$

where $\langle\cdots\rangle_n$ denotes the *n-fold ordered time average* operator, or simply the time average. For a function, $h(t_1,\cdots,t_n)$, where $t_1<\cdots<t_n$, we define

$$\langle h\rangle_n = \int_0^T dt_1 f(t_1)\cdots\int_{t_{n-1}}^T dt_n f(t_n)h(t_1,\cdots,t_n). \tag{2.67}$$

2.8.1 Spatial and temporal profiles

In most case, the spatial profile $B(\boldsymbol{r})$ in equation 2.63 takes the form of a linear gradient field:

$$B(\boldsymbol{r}) = \boldsymbol{r}\cdot\boldsymbol{g}, \tag{2.68}$$

where $\boldsymbol{g}$ is a constant vector known as the *diffusion-encoding gradient* or simply *diffusion gradient*. In dMRI, the imaging sequence is augmented by the diffusion encoding sequence. Non-linear gradient fields includes parabolic and cosine fields are rarely encountered applications. Theoretical studies, however, do exist on some special profiles such as parabolic and cosine profiles[13],[12].

Some common properties of the temporal profile $f(t)$ in equation 2.63 are listed below. We also consider their physical implications on corresponding sequences. Furthermore, notable sequences are described.

1. $f(t)$ is a dimensionless scalar function and its maximum absolute value equals one. In the case where $f(t)\equiv 1$ for $0\leqslant t\leqslant T$, the sequence is called the *free induction decay*(FID). Physically, the only rf-pluse applied is the 90° excitation pulse while a steady gradient field is present in the whole duration of $[0,T]$.

[14]Recall that for a stochastic process with zero mean, $\{\boldsymbol{X}(t)\}_{t\geqslant 0}$ ($\mathbb{E}[\boldsymbol{X}(t)]=0$ for all t), its correlation function is defined as $\mathbb{E}[\boldsymbol{r}(t_1)\boldsymbol{r}(t_2)]$. The term multiple correlation function is appropriately coined.

2. $f(t)$ is piecewise continuous and changes its sign at the moments when the 180° rf-pulse is applied since physically the sign of the phase particles is inverted instantaneously (approximately speaking). For example in spin echo experiments we have

$$f(T^-/2) = -f(T^+/2).$$

Mathematically, the spin echo (SE) sequence is equivalent to the following one known as *gradient echo*(GE) sequence: at time $t = T/2$ the original gradient is turned off and a new gradient with the exact same spatial profile but with opposite direction is activated simultaneously[15].

3. To form an echo, $f(t)$ needs to satisfy the *rephasing condition*,

$$\int_0^T f(t)dt = 0. \tag{2.69}$$

The rephasing condition ensures that an immobile spin always has zero net phase gain (recall Figure 2.6). The most common sequence whose temporal profile satisfying equation 2.69 is the pulse-gradient spin-echo (PGSE) sequence, first introduced by Stejska and Tanner in 1965[32]. Its $f(t)$ (idealized) consists of two antisymmetic rectangular pulses. In piecewise form,

$$f(t) = \begin{cases} 0 & t < t_1 \\ 1 & t_1 < t < t_1 + \delta \\ 0 & t_1 + \delta < t < t_1 + \Delta \\ -1 & t_1 + \delta < t < t_1 + \Delta + \delta \\ 0 & t_1 + \Delta + \delta < t < TE \end{cases},$$

where sequence paramters δ and Δ are called the *pulse duration* and *diffusion time*. The first pulse must end in the first half of the sequence, implying that $t_1 + \delta \leqslant TE/2$. The case where $t_1 = 0$, $\delta = \Delta = TE/2$ corresponds to the Hahn's spin echo experiment. A less idealized version of $f(t)$ for PGSE uses trapezoidal pulses instead of rectangular ones, taking the *ramp time*, the time for the gradient to reach it maximum strength, into account.

Other commonly used sequences in diffusion MRI can be found in [11].

[15] GE is actually preferred over SE in imaging since the technology allows responsive gradient control.

2.8.2 Gaussian diffusion and Stejskal-Tanner formula

In this section, we apply the stochastic formulation to compute the signal due to free diffusion, as a 'Hello, World!' example of diffusion MRI. The result is the celebrated Stejskal-Tanner (S-T) formula[16], which forms the basis of diffusion tensor imaging.

It was shown earlier that for free diffusion ($\Omega = \mathbb{R}^3$), the position process along any direction is a Wiener process. If the free diffusion is isotropic, the displacement vector at time t is given by

$$\boldsymbol{r}(t) - \boldsymbol{r}_0 = \Delta\boldsymbol{r}(t) = (W_t^1, W_t^2, W_t^3), \tag{2.70}$$

where W_t^1, W_t^2, W_t^3 are independent Gaussian random variables with mean 0 and variance $2Dt$. As a result, under the influence of a linear gradient field with gradient vector $\boldsymbol{g}$ and temporal profile $f(t)$, the dephase is equal to

$$\varphi = \int_0^T dt f(t)(\boldsymbol{g}\cdot\boldsymbol{r}(t)) = \int_0^T dt f(t)(\boldsymbol{g}\cdot(\boldsymbol{r}_0 + \Delta\boldsymbol{r}(t)) = \int_0^T dt f(t)\boldsymbol{g}\cdot\Delta\boldsymbol{r}(t). \tag{2.71}$$

The term with $\boldsymbol{r}_0$ vanishes due to the rephasing condition (2.69). In the component form, equation 2.71 reads

$$\varphi = g_1\phi_1 + g_2\phi_2 + g_3\phi_3, \tag{2.72}$$

where

$$\phi_i = \int_0^T f(t)W_t^i dt \quad \text{for} \quad i = 1, 2, 3. \tag{2.73}$$

Random variables ϕ_i's are independent since processes W_t^i's are independent. In addition, ϕ_i's are Gaussian since it is a linear functional of a Wiener process[17]. Therefore, φ follows a Gaussian distribution. This significantly simplifies the problem as in general, the exact distribution of φ is untractable even for the simplest geometries.

Consequently, to characterise the distribution of φ completely we only need its first two moments. Using the mean and autocovariance functions of Wiener processes, $\mathbb{E}[W_t] = 0$ and $\mathbb{E}[W_{t_1}W_{t_2}] = 2D\min(t_1, t_2)$, one derives the mean and variance of ϕ_i's,

$$\mathbb{E}[\phi_i] = \int_0^T f(t)\mathbb{E}[W_t]\,dt = 0, \tag{2.74}$$

[16] In their seminal paper [32], Stejskal and Tanner derived the result using the PDE approach (see section 2.9). The stochastic approach, in the author's opinion, is more intuitive and physically motivated.

[17] We can approximate equation 2.73 by its Riemann sum, which is a linear combination of jointly distributed Gaussian random variables. The Riemann sum is therefore Gaussian. The limit of the Riemann sum remains Gaussian as well

$$\begin{aligned}
\mathbb{E}\left[\phi_i^2\right] &= \int_0^T dt_1 f(t_1) \int_0^T dt_2 f(t_2) \mathbb{E}\left[W_{t_1} W_{t_2}\right] \\
&= 2\int_0^T dt_1 f(t_1) \int_{t_1}^T dt_2 f(t_2) \mathbb{E}\left[W_{t_1} W_{t_2}\right] \\
&= 2\int_0^T dt_1 f(t_1) \int_{t_1}^T dt_2 f(t_2) 2D t_1 \\
&= 4D\langle t_1 \rangle_2.
\end{aligned} \tag{2.75}$$

Therefore, $\mathbb{E}\left[\varphi\right] = 0$. Since ϕ_i's are mutually independent, the variance of φ is equal to

$$\mathbb{E}\left[\varphi^2\right] = 2D \sum_{i=1}^{3} g_i^2 \mathbb{E}\left[\phi_i^2\right] = 4D\langle t_1 \rangle_2 (g_1^2 + g_2^2 + g_3^2) = 4D\gamma^2 g^2 \langle t_1 \rangle_2. \tag{2.76}$$

Recall that the characteristic function of the Gaussian random variable X following $N(\mu, \sigma^2)$ is given by

$$\mathbb{E}\left[e^{itX}\right] = \exp\left(i\mu t - \frac{1}{2}\sigma^2 t^2\right). \tag{2.77}$$

Setting $t = 1$ in equation 2.77, the characteristic function has the value of

$$\mathbb{E}\left[e^{iX}\right] = \exp\left(i\mu - \frac{1}{2}\sigma^2\right). \tag{2.78}$$

Therefore, the diffusion signal $\mathbb{E}\left[e^{i}\varphi\right]$ equals to

$$S = \mathbb{E}\left[e^{i\varphi}\right] = \exp\left(-2D\gamma^2 g^2 \langle t_1 \rangle_2\right). \tag{2.79}$$

The gyromagnetic ratio γ is brought back as we dropped it earlier for the ease of derivation. The result given by equation 2.79 is the well known *Stejskal-Tanner formula*.

b-value Isolating the diffusion coefficient D in equation 2.79, all the other factors are combined into the so-called *b-value*,

$$b = 2\gamma^2 g^2 \langle t_1 \rangle_2, \tag{2.80}$$

and the result 2.79 is written as a simple exponential function of b,

$$S = \exp\left(-bD\right). \tag{2.81}$$

The b-value 'summarizes' the sequence information into a single scalar. It is also an indicator of how sensitive the signal is to the diffusivity.

One may notice that our expression for the b-value seem different compared to the one in literature[32][7],

$$b = \gamma^2 g^2 \int_0^T dt \left(\int_0^t f(t') \right)^2 . \tag{2.82}$$

They are in fact equivalent. it is shown in [11] that

$$\langle t_1 - t_2 \rangle_2 = \int_0^T dt \left(\int_0^t f(t') \right)^2 . \tag{2.83}$$

The left hand side of equation 2.83 can be shown to be equal to $2\langle t_1 \rangle_2$. Due to the linearity of $\langle \cdot \rangle_n$, we have

$$\begin{aligned}
&2\langle t_1 \rangle_2 - \langle t_1 - t_2 \rangle_2 = \langle t_1 + t_2 \rangle_2 \\
=&\int_0^T dt_1 f(t_1) \int_{t_1}^T dt_2 f(t_2)(t_1 + t_2) \\
=&\frac{1}{2} \int_0^T dt_1 f(t_1) \int_0^T dt_2 f(t_2)(t_1 + t_2) \text{ (the integrand is symmetric)} \\
=&\frac{1}{2} \int_0^T dt_1 f(t_1) \int_0^T dt_2 f(t_2) t_1 + \frac{1}{2} \int_0^T dt_1 f(t_1) \int_0^T dt_2 f(t_2) t_2 .
\end{aligned}$$

Both of the double integrals in the last step vanish under the rephasing condition 2.69. Therefore, the consistency between different forms of b-value is shown.

For a given type of sequence, to derive the closed form expression for b-value, one needs to calculate the time integral in any of the aforementioned expressions. The most commonly used b-value in dMRI is the one for PGSE sequences, which states that

$$b = \gamma^2 g^2 \delta^2 (\Delta - \frac{\delta}{3}). \tag{2.84}$$

The b-value for FID is given by

$$b = \gamma^2 g^2 T^3 / 3. \tag{2.85}$$

If the diffusion is anisotropic, the scalar diffusion coefficient is replaced by the diffusion tensor and the exponent of the S-T formula becomes a quadratic form[26],

$$S = \exp\left(-b \hat{\boldsymbol{g}}^T \boldsymbol{D} \hat{\boldsymbol{g}}\right), \tag{2.86}$$

in which we assume that $\boldsymbol{D}$ is a constant matrix.

Even when studying complex confining geometry such as human brain, where the assumption of free diffusion clearly fails, the S-T formula is still taken as the most common *analytic model* for the signal. For example, what *Diffusion Tensor Imaging* (DTI)[2] essentially does is measuring enough signals under various gradient directions in order to fit the model given by equation 2.86. The estimated diffusion tensor $\boldsymbol{D}$ is computed by the least square method. From the fitted tensor, one may further compute its eigenvalues and eigenvectors to find its principal axis orientation, which is then taken as the nerve fiber orientation at each voxel.

2.8.3 Spectral Representation

Our utmost interest lies in investigating the case of restricted diffusion. The presence of interacting boundary complicates the solution of the signal attenuation significantly. Since the result of Gaussian diffusion is no longer valid, in general, moments of all order $\mathbb{E}[\varphi^n]$ are now required to compute the signal.

It was established earlier that the n-th order moment is the ordered time average of the MCF (equation 2.64). We now consider expressing the MCF according to its probabilistic interpretation.

The MCF (equation 2.65) can be expressed in terms of the joint pdf of random positions $\boldsymbol{r}(t1), \ldots, \boldsymbol{r}(t_n)$,

$$\mathbb{E}\left[B(\boldsymbol{r}(t_1)) \cdots B(\boldsymbol{r}(t_n))\right] = \int_\Omega d\boldsymbol{r}_1 \cdots \int_\Omega d\boldsymbol{r}_n f_n(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n; t_1, \ldots, t_n) B(\boldsymbol{r}_1) \cdots B(\boldsymbol{r}_n), \tag{2.87}$$

The joint pdf can be constructed using the initial distribution and the Green's function. Substituting equation 2.40 into 2.87 and integrating over the initial and final positions, one arrives at the integral form of the MCF,

$$\begin{aligned}\mathbb{E}\left[B(\boldsymbol{r}(t_1)) \cdots B(\boldsymbol{r}(t_n))\right] = \int_\Omega d\boldsymbol{r}_0 \rho(\boldsymbol{r}_0) \int_\Omega d\boldsymbol{r}_1 G_{t_1}(\boldsymbol{r}_0, \boldsymbol{r}_1) B(\boldsymbol{r}_1) \cdots \\ \int_\Omega d\boldsymbol{r}_n G_{t_n - t_{n-1}}(\boldsymbol{r}_{n-1}, \boldsymbol{r}_n) B(\boldsymbol{r}_n) \int_\Omega d\boldsymbol{r}_{n+1} G_{T-t_n}(\boldsymbol{r}_n, \boldsymbol{r}_{n+1}).\end{aligned} \tag{2.88}$$

The final position, needs to be integrated over in order to address the information that the particle still remains inside Ω at time t_n[18]. We can now fully appreciate the seemingly innocuous step of arranging the dummy variables into ascending order in equation 2.62, without which the Green's functions can not be used as the time parameter in $G_t(\boldsymbol{r}, \boldsymbol{r}')$ should be positive.

Using the method of separation of variables, the Green's function can be expressed in terms of the Laplacian eigenvalues and eigenfunctions,

$$G_t(\boldsymbol{r}, \boldsymbol{r}') = \sum_m u_m^*(\boldsymbol{r}) e^{-D\lambda_m t} u_m(\boldsymbol{r}'), \tag{2.89}$$

[18]It can be harmlessly omitted boundary is reflective as it is guaranteed that all particles remains inside at all time.

where λ_m and $u_m(\boldsymbol{r})$ satisfy the Helmholtz equation with appropriate BC (we use the Robin BC as an example),

$$\nabla^2 u_m(\boldsymbol{r}) + \lambda_m u_m(\boldsymbol{r}) = 0, \tag{2.90}$$

$$D\frac{\partial u_m(\boldsymbol{r})}{\partial n} + h u_m(\boldsymbol{r}) = 0, \quad \text{for} \quad \boldsymbol{r} \in \partial\Omega. \tag{2.91}$$

Depending on the dimension of the domain m can be a multiple index. Plugging equation 2.89 into the integral form of MCF yields the infinite series form, which reads

$$\begin{aligned}\mathbb{E}\left[B(\boldsymbol{r}(t_1))\cdots B(\boldsymbol{r}(t_n))\right] = \sum_{m_1}\sum_{m_2}\cdots\sum_{m_{n+1}} U_{m_1} e^{-D\lambda_{m_1}t_1}\mathcal{B}_{m_1 m_2}\times \\ e^{-D\lambda_{m_2}(t_2-t_1)}\mathcal{B}_{m_2 m_3}\cdots\mathcal{B}_{m_n m_{n+1}} e^{-D\lambda_{m_{n+1}}(T-t_n)}\tilde{U}^*_{m_{n+1}},\end{aligned} \tag{2.92}$$

where

$$\begin{aligned}U_m &= \int_\Omega d\boldsymbol{r}_0 \rho(\boldsymbol{r}_0) u_m(\boldsymbol{r}_0); \\ \tilde{U}^*_m &= \int_\Omega d\boldsymbol{r} u^*_m(\boldsymbol{r});\end{aligned} \tag{2.93}$$

and

$$\mathcal{B}_{mm'} = \int_\Omega d\boldsymbol{r} u^*_m(\boldsymbol{r}) B(\boldsymbol{r}) u_{m'}(\boldsymbol{r}). \tag{2.94}$$

This is also known as the spectral representation of the MCF.

It is well-known that $\{u_m(\boldsymbol{r})\}_m$ forms an complete orthonormal basis for the function space $L^2(\Omega)$ (after normalization if necessary). We can view coefficients[19] U_m's, $\tilde{U}^*_m$'s as the coordinates of the initial distribution $\rho(\boldsymbol{r}_0)$ and the constant function $\mathbb{1}(\boldsymbol{r}) = 1$ under the eigenbasis. Similarly, coefficients $\mathcal{B}_{mm'}$ has the interpretation of the eigenbasis representation of the 'Larmor precession operator',

$$\mathcal{B}: \ L^2(\Omega) \mapsto L^2(\Omega) \ [\mathcal{B}\chi](\boldsymbol{r}) = B(\boldsymbol{r})\chi(r). \tag{2.95}$$

In light of the linear algebra analogy, it is natural to introduce the some infinite-dimensional matrices so we can express the MCF in a compact matrix form.

[19] They are called the generalized Fourier coefficients.

Let $\Lambda = [\delta_{mm'}\lambda_m]$, $U = [U_m]$, $\mathcal{B} = [B_{mm'}]$ and $\tilde{U}^* = [\tilde{U}^*_m]$, where $\delta_{mm'}$ is the Kronecker delta so that

$$\mathbb{E}\left[B(\boldsymbol{r}(t_1))\cdots B(\boldsymbol{r}(t_n))\right] = Ue^{-D\Lambda t_1}\mathcal{B}e^{-D\Lambda(t_2-t_1)}\mathcal{B}\cdots\mathcal{B}e^{-D\Lambda(T-t_n)}\tilde{U}^*. \tag{2.96}$$

Plugging the expression above into equation 2.61, we arrive at the final expression of the diffusion signal in all its glory,

$$\mathbb{E}\left[e^{i\varphi}\right] = \sum_{n=0}^{\infty} i^n \langle Ue^{-D\Lambda t_1}\mathcal{B}e^{-D\Lambda(t_2-t_1)}\mathcal{B}\cdots\mathcal{B}e^{-D\Lambda(T-t_n)}\tilde{U}^*\rangle_n. \tag{2.97}$$

The MCF approach is therefore also known as the *matrix formalism*.

Theoretically speaking, we reached the conclusion that the Laplacian eigenvalues and the eigenfuncions for the domain Ω provides the complete, natural language for studying the diffusion signal. Albeit more cumbersome than the stochastic definition (equation 2.59,2.60), the spectral/matrix formalism involves only deterministic quantities.

2.9 Bloch-Torrey Equation

A mathematically equivalent description for the diffusion NMR phenomenon was proposed by Torrey[34]. We briefly present it in this section for the sake of completeness though it does not serve as the theoretical foundation of our study.

To introduce the Bloch-Torrey (BT) equation, we need to return to our view of treating the diffusing substance as a continuum when we first introduced diffusion equation[20]. Let $m(\boldsymbol{r}, t)$ be the time-dependent, complex valued, local transverse magnetization density (measured in the rotating reference frame); and $\omega(\boldsymbol{r}, t)$ be the local precession frequency, determined by the field inhomegeneity.
The *Bloch-Torrey partial differential equation*, or simply *Bloch-Torrey equation*, is given by

$$\frac{\partial m}{\partial t} = -i\omega(\boldsymbol{r}, t)m + D\nabla^2 m, \tag{2.98}$$

where D is the diffusion coefficient. The diffusion signal is defined as the total magnetization,

$$S(t) = \int_{\Omega} m(\boldsymbol{r}, t)d\boldsymbol{r}. \tag{2.99}$$

A heuristic derivation is provide below.

The time-evolution of $m(\boldsymbol{r}, t)$ solely due to the magnetic field encoding is described by the simple Bloch equation:

$$\frac{\partial m}{\partial t} = -i\omega(\boldsymbol{r}, t)m. \tag{2.100}$$

Recall two facts we established earlier:

1. At thermal equilibrium, the magnetization is proportional to the concentration of the spin-bearing substance (see relation 2.2).
2. The concentration satisfies the diffusion equation (given the absence of bulk motion).

As a result, even without the influence of the external inhomegeneous field, local magnetization changes alongside with the spin concentration (in other words, diffusion). Hence, it

[20]The respective points of view represented by the stochastic and the PDE formalism is similar to the dichotomy between the Lagrangian and the Eulerian approaches in fluid dynamics. Namely, the focus of an observer is either a fluid particle or a specific location

is reasonbale to assume that the time evolution of $m(\boldsymbol{r}, t)$ due to diffusion is governed by its Laplacian.
The BT-equation combines the effects both driving forces via a simple addition, since the spin precession does not affect diffusion in anyway and vice versa.

Relaxation effects can also be incorporated. For example, if $T_2(\boldsymbol{r})$ is the local transverse relaxation rate the Bloch-Torrey PDE becomes

$$\frac{\partial m}{\partial t} = -i\omega(\boldsymbol{r}, t)m + D\nabla^2 m - \frac{m}{T_2(\boldsymbol{r})}. \tag{2.101}$$

The same three kinds of BCs discussed before can be imposed on the boundary, depending on the physical property of the membrane. If $t = 0$ is the exact moment of excitation by the 90° rf-pulse, $m(\boldsymbol{r}, 0)$ is taken as uniform since the initial concentration is uniform and all magentizations share the same initial phase (recall Figure 2.3).

Remark Having covered the fundamentals of dMRI, we would like to reminder readers that despite the same underlying physical processes, different equations describing various measurement outcomes are only meaningful at the length scales of those measurements take place.

Within the duration of a typical MRI sequence, spin displacements defined in subsection 2.7.3 are of several mircometers, a length scale we call *microscopic*; diffusion signals defined by equation 2.60 or 2.99 have contribution from spins within a voxel of several millimeters, a length scale we call *macroscopic*; the magnetization density in the BT equation are measured over regions whose length scales fall somewhere in between, which are referred to as *mesoscopic*[21]; and finally, the signal in the imaging equation 2.22 contains the spatial information of the whole sample (such as a part of the brain), which has the length scale of several centimeters.

[21] If the regions are so small that the diffusing substance can longer be viewed as a continuum, the notion of 'density' breaks down. On the other hand, if the regions are so large that most of the spins are not able to escape (within the sequence duration), spatial variation of $m(\boldsymbol{r}, t)$ becomes negligible and the PDE degenerates into an ODE.

2.9.1 New results

Some interesting theoretical developments happened in recent years. The sketched description of two of those are included below.

An alternative matrix formalism of the diffusion weighted signal is proposed[18], in which the basis functions are chosen as the eigenfucntions of the BT operator,

$$\mathcal{L} = i\omega(\boldsymbol{r}, t) + D\nabla^2, \tag{2.102}$$

instead of the Laplacian eigenfunctions used for expressing the MCF. Compared to the MCF approach, the computation of the time evolution using this new matrix formalism is trivial, as one only needs to multiply the each BT eigenfunctions by an exponential decay factor and take superposition. The trade-off however, is that the coordinates of the initial profile in the eigenbasis are much harder to compute. The reason for that is the BT eigenfunctions are non-orthogonal since the BT operator is not Hermitian.

A generalization of the Bloch-Torrey equation was proposed in [31]. Its basic idea treats the position $\boldsymbol{r}$ and the spin magnetization $\boldsymbol{m}$ as coordinates in the phase space [22]. Since the process $\boldsymbol{r}(t)$ is the reflected Brownian motion and $\boldsymbol{m}(t))$ evolves according to local Larmor frequencies, the time evolution of the joint process $(\boldsymbol{r}(t), \boldsymbol{m}(t))$ is governed by a system of stochastic differential equations. The generalised Bloch-Torrey PDE is defined as the Fokker-Planck equation describing the time evolution of the joint pdf of the paired processes[23].

[22] Analogous to the position and momentum pair in Hamiltonian dynamics.

[23] An analogy is the Kramers' equation, the PDE describing the joint process of velocity and position of Brownian motions. see [22] for a treatment of Fokker-Planck equations.

2.10 Summary and comparison

So far, it has been shown that the diffusion signal can be defined or calculated by three distinct approaches:

1. Statistical approach: Signal is the ensemble average of time integrals along Brownian trajectories (equation 2.59,2.60);
2. PDE approach: Signal is the volume integral of the solution of the Bloch-Torrey equation (equation 2.98.2.99).
3. MCF (or Matrix, Spectral) apporach: Signal is given by the contributions from all moments, which are in turn expressed as products of infinite-dimensional matrices (equation 2.97).

we would like compare those three approaches in terms of their mathematical consistencies and applications in numerical computations.

2.10.1 Relations between the three approaches

All formalisms can be derived from one another.

The matrix formalism can be alternatively derived from the PDE approach[11]. After expanding the magnetization function ($m(\boldsymbol{r}, t)$ in 2.98)in the Laplacian eigenbasis, one can plug the expansion into the Bloch-Torrey PDE. By comparing coefficients on both sides, one realises that the time dependent coefficients satisfy a linear system of infinitely many coupled ordinary differential equations (ODE). Rewriting the system of ODEs into its matrix form, one can formally express its solution as a matrix exponential.

On the other hand, it is well-known that linear parabolic PDEs and stochastic differential equations are linked through the *Feynman-Kac formula*[29], which states that the solution of the PDEs are in fact the conditional expectation of some stochastic processes. The statistical and the PDE formalism are almost certainly linked by the complexed version of Feynmann-Kac formula, which is still not yet proven. Further investigation is beyond the scope of this study.

2.10.2 Numerical computation

All three approaches can serve as the basis of numerical computation. We here discuss their advantages, difficulties and weaknesses individually.

MC simulation The stochastic formulation naturally gives rise to Monte Carlo (MC) simulation schemes, where the integral 2.59 and the expectation 2.60 are approximated by their finite-dimensional counterparts. In principle, MC simulations are intuitive to implement and enjoy the greatest flexibility since its large number of degree of freedom allows researchers to control physical processes at the most fundamental level. For example, it is straightforward to modify the MC simulation scheme should the diffusion is modeled as fractional Brownian motion instead. As for MC methods in general, they do not suffer from the curse of dimensionality since their convergence rates are determined by the central limit theorem. An additional perk of MC methods is that once the dephase distributions are well sampled in all spatial dimensions, the computation of signals in arbitrary gradient directions is trivial. Therefore, the focus in our simulations is to obtain the empirical dephase distribution.

For MC simulations in complex geometries, collision detection between spins and boundaries can be tricky to implement. The main disadvantages of MC simulations are low accuracy and high computation cost. In later chapters, we shall discuss those advantages and shortcomings in greater detail.

Camino[16], an open-sourced[24] toolkit originally designed for dMRI processing, now includes the functionality for MC simulations. To improve the computational efficiency for collision detection, works such as [36], [28] used the octree data structure to partition the 3D space.

Solving BT equation The PDE-based approach is currently the most popular choice for diffusion NMR/MRI simulations. Thanks to the maturity of algorithms for solving parabolic PDEs such as finite difference and finite element methods, a well-designed Bloch-Torrey PDE solver can simulate the diffusion signal accurately and at a low cost. Furthermore, in cases where one only interested in the total signal, but also the local magnetization density[25], solving BT equations is the only viable option. Unlike MC schemes however,

[24] http://camino.cs.ucl.ac.uk/index.php?n=Tutorials.MCSimulator

[25] in the mesoscopical sense

PDE-based simulations do suffer from the curse of dimensionality since the spatial grid size increases exponentially w.r.t the number of dimensions. As the grid sizes grows, so does the size of the linear system governing the time evolution. Previous studies on this approach includes [27] and [4], etc.

Matrix products Within simple geometries exhibiting high degrees of symmetry, simulation base on the matrix approach is highly efficient and accurate. Thanks of the rapid growth of Laplacian eigenvalues w.r.t their indices, the matrix exponentials in 2.97 can be well approximated by their low-dimensional truncations. Therefore, the computation boils down to multiplications and additions of matrices provided the solutions to the eigenvalue problem are known. For slabs, disks and spheres, the Laplacian eigenfunctions are well-known special functions. The values of eigenvalues are zeros of the eigenfunctions, which depend on the boundary conditions[26]. Closed-form expressions for the matrix elements 2.93, 2.94 are available for the aforementioned geometries [11] and their derivations can be quite involved (see Appendix B). MCFAL, a MATLAB package that implements matrix-based simulation, is available[27].

Apart from the obvious limitation of being ill-adapted to general geometries, another weakness of matrix-based simulations is that they become unfeasible in case where gradients are strong[28], as a large number of higher order moments are required for good accuracy. In fact, diffusion under the influence of strong gradients is an active area of dMRI research.

[26] The zeros are not well-known under Robin boundary conditions and must be solved numerically.
[27] https://pmc.polytechnique.fr/pagesperso/dg/MCF/MCF_e.htm
[28] The Taylor series 2.61 works well for small φ

Chapter 3

Basic Monte Carlo Simulation

3.1 Workflow

This chapter covers MC simulations of dMR images in the most general setting. The current section discuss the general framework of the simulation. Section 3.2 describes the algorithm for simulating diffusion encoding. Section 3.3 demonstrated the implementations and results in three prototypical geometries. The final section 3.4 is devoted to the analysis of simulation accuracy.

Prior to running simulations, some preparatory works need to be done. This include defining the sequences for both image- and diffusion encoding, as well as specifying the ground truth information of the sample of interest. As the image formation and diffusion encoding capture information at different length scales (recall section 2.9), we organize the ground truth in a two level hierarchy.

Global ground truth consists of the shape of the sample, a bounded 2-dimensional domain D and parameters for describing the macroscopic magnetization. For clinical applications, D corresponds to organ/tissue of interest. In simulations, D is rasterized /triangularized into a set of square or triangular meshes $\{K_i\}$ called *pixels*. The size of the meshes[1] indicates our desired resolution of the simulated image, which in turn imposes requirements on the design of imaging sequences (see [6] for datails). Typical resolutions are

[1]Defined by, for example, the maximum diameter of $\{K_i\}$.

around a millimeter and the thickness of selected slices are within similar ranges. Therefore, the signal at one pixel actually contains the information from a volumetric element, known as the *voxel*. The shape of a voxel can either be a cube or a triangular prism. The voxelwise-defined parameters $(\rho_i, T_{1,i}, T_{2,i})$ correspond to the the spin densities and time constants for relaxations.

Within each voxel, there is microstructure that restricts the diffusion of water molecules, which is modeled by a collection of disjoint open sets along with prescribed BCs on their boundaries. Intravoxel geometry serves as the 'playground' for the diffusing spins. The main algorithm for diffusion encoding is given in the next section.

As an analogy, readers may compare our diffusion simulation paradigm with the process of making an aquarium at home. One first sets up a fish tank (voxel) of a chosen size (determined by the k-space trajectory). Then one may proceed to add some decorations such as cemented castles or shipwrecks (microstructure). After filling up the tank with water and putting in some fish (diffusing spins), the owner now may sit back and enjoy watching the movement of fish (computing spin paths).

Therefore, to summarize, the main steps of the simulation are given below:

1. Specify the ground truth information of the sample at both global and local length scales;
2. Specify an imaging sequence augmented with diffusion encoding gradients;
3. Perform voxel-wise MC simulation for diffusion encoding and compute the diffusion signal
4. Multiply the DW factor by the spin density and relaxtion factors. Given a $k-$space trajectory, the MR image in associated with each k-space coordinate is computed according to the imaging equation.
5. Compute the image in physical space by taking inverse discrete Fourier transform.

3.2 Diffusion Encoding

Suppose that the domain for diffusion simulation is Ω of dimension d[2] and Initial spin positions follow the distribution $\rho(\boldsymbol{r}_0)$. Local Larmor frequencies are $\omega(\boldsymbol{r}, t)$, $\boldsymbol{r} \in \Omega$.

Let the number of sample paths be N_{walker} and the sequence duration is divided evenly into N_{time} time steps of size Δt. The statistical definition of diffusion signals naturally suggests the following MC simulation algorithm:

S.1 Sample initial positions $\{\boldsymbol{r}_0\}_{n=1}^{N_{walker}}$ from the initial distribution $\rho(\boldsymbol{r}_0)$;

S.2 For each given initial position, generate a sample path $\{\boldsymbol{r}_j\}_{j=1}^{N_{\text{time}}}$, where the position update rule at j-th iteration is:

(i) Generate a step vector, $\Delta\boldsymbol{r}_j$ with uniformly random direction and root mean square (rms) length of

$$ds = \sqrt{2dD\Delta t}; \tag{3.1}$$

(ii) Compute the naive update, $\boldsymbol{r}'_{j+1} = \boldsymbol{r}_j + \Delta\boldsymbol{r}_j$;

(iii) Checking for collision with the boundary, in other words, whether or not $\boldsymbol{r}'_{j+1} \in \Omega$. If there is no collision, the new position $\boldsymbol{r}_{j+1} = \boldsymbol{r}'_{j+1}$;

(iv) If collision does occur, take the appropriate action based on the boundary condition:

BC.1 If $\partial\Omega$ is absorbing, discard the current sample path;

BC.2 If $\partial\Omega$ is reflective, $\boldsymbol{r}_{j+1}$ is taken as $\boldsymbol{r}'_{j+1}$ reflected w.r.t. $\partial\Omega$;

BC.3 If $\partial\Omega$ is partially reflective, generate a Bernoulli random variable U with a prescribed probability p (explained later); If $U = 1$, go to S.1; otherwise go to S.2.

S.3 Compute the dephase φ accumulated along the sample path $\{\boldsymbol{r}_j\}$ by numerical integration. For example, the rectangular rule yields

$$\varphi = \sum_{j=0}^{N_{\text{time}}} \omega(\boldsymbol{r}_j, t_j)\Delta t, \tag{3.2}$$

where $t_j = j\Delta t$;

[2] In general d equals 3 but may be reduced be exploiting the symmetry of the domain

S.4 Repeat **S.2**, **S.3** for all spins to the dephase sample $\{\varphi_n\}$.

S.5 Calculate the diffusion signal,

$$S_{\text{sim}} = \overline{\exp(i\varphi_n)}, \tag{3.3}$$

where the overscore stands for the arithmetic mean.

The prescribed probability p in **BC.3** of **S.2** is called the *termination probability* and depends on the surface relaxivity h in Robin BCs (equation 2.38). Our choice of p is given by

$$p = 2h\sqrt{\frac{\Delta t}{\pi D}}. \tag{3.4}$$

This model for p assumes that

- the local boundary is approximately flat;
- the characteristic length scale of the domain is so much greater than ds that within one time step Ω can be treated as semi-infinite.

A full justification of equation 3.4 requires solving an initial boundary value problem, shown in Appendix D.

Although the natural MC algorithms are the most flexible, they rely on the accurate simulation of detailed Brownian trajectories. In order to make sure that $\{\boldsymbol{r}_j\}$ is a reasonable approximation of the discrete sample path, Δt needs to be chosen small. Coupled with the complexity of collision detection in complex geometries, this makes 'brute force' MC simulations highly expensive in general. Therefore, they are often used as a last resort.

3.3 Experiments

Here we illustrate the implementation of diffusion encoding under various confining geometries. All simulations in this work were conducted on a machine with two 8-core Intel Xeon E5-2640 CPUs and 256 GB RAM.

3.3.1 Free Space

We begin our discussion on the implementation of Monte Carlo simulation by considering the simplest case, diffusion in the free space.

As shown in Chapter 2, it suffices to fix the initial position at the origin and only consider the increment process. Therefore, the sampling of initial positions is skipped for free diffusion. Random positional increments, also referred to as 'jumps', are generated by the MATLAB function 'MakeSteps.mat' (Appendix F). The function allows users to specify the increment distribution and the root mean square (rms) length of the increments in each dimension, denoted as ds. In particular, the step size for uniform jumps is computed via

$$r = \sqrt{d+2}ds\sqrt[d]{U}, \tag{3.5}$$

where U follows Unif $[0,1]$ and $d = 2, 3$ is the dimension of the jump vector. We briefly show this is correct:
Let the maximum jump size be R. Since the area/volume element formula in d-dimension is proportional to r^{d-1}, the cumulative distribution for r should be $(r/R)^d$. Therefore, the variance of r satisfies:

$$dds^2 = \mathbb{E}\left[r^2\right] = \int_0^R r^2 d\frac{r^{d-1}}{R^d}dr = \frac{d}{d+2}R^2.$$

Solving for R yields $R = \sqrt{d+2}ds$.

The trajectory, as the time integral of the jumps, can be approximated using the trapezoidal rule. Subsequently, the same numerical integration technique is applied for approximating the dephase since it is the time integral of the trajectory (weighted by the discrete time-profile). For a single spin, a sample path of dephase time evolution is shown in Figure 3.1, which shows a clear deviation from the same process experienced by an immobile spin (recall the spin-echo experiment). The empirical dephase distribution for a large ensemble

of spins is summarized in a histogram (Figure 3.2). Unsurprisingly, the empirical dephase distribution is also Gaussian. Finally, it can be verified that the simulated signal is in agreement with the Stejskal-Tanner formula.

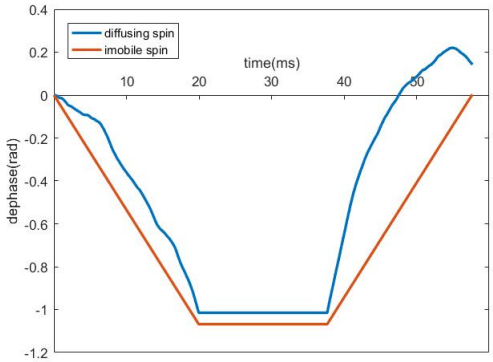


Figure 3.1: Dephase evolution experienced by a diffusing spin compared against an immobile one.

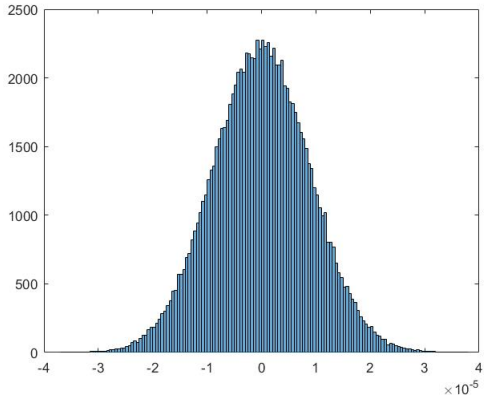


Figure 3.2: empirical dephase distribution under free diffusion.

3.3.2 Polygonal Domains

Our next step is to implement simulations in confined geometries. Boundaries specified by polygons and circular/spherical surfaces are considered. More general shapes can be constructed using splines or Bézier curves but are beyond the scope of this study. Let our confining domain in this section be a simple (non-self-intersecting) planar polygonal. Its vertex coordinates are given and fixed.

Initialization We first describe our method for initialization. Sampling of uniformly-distributed initial positions can be performed in a divide-and-conquer manner. First, the polygon is partitioned into a set of triangles (for example, the pentagon in Figure 3.3 is divided into three triangles by two of its diagonals). Second, one of the triangles is randomly selected with probability proportional to its area (for example it can be $\triangle V_1V_4V_5$). Finally, the desired initial position is taken as a random point inside the triangle.

How can we ensure that in our last step, the interior point is truly uniformly-distributed? Recall that for a triangle, the coordinate of any of its interior points can be written as a weighted average of its vertex coordinates[3]. For example in Figure 3.3, one has:

$$P = c_1V_1 + c_2V_4 + c_3V_5, \quad \text{for} \quad c_1, c_2, c_3 \geqslant 0, c_1 + c_2 + c_3 = 1. \tag{3.6}$$

The set of weight coefficients (c_1, c_2, c_3) is known as the standard $2-$simplex. The uniform distribution over the standard $2-$simplex is the *flat Dirichlet distribution*, which has pdf of

$$f(c_1, c_2, c_3) = 2c_1c_2c_3. \tag{3.7}$$

With a source of Gamma-distributed random variates, one can sample a random vector following Dirichlet distribution by simply dividing the aforementioned variates by their sum (see for [9] a proof). Furthermore, in the flat case, the Gamma variates in fact follows exponential distribution with parameter 1.

Therefore, we have the following recipe for position initialization (for the pentagon in Figure 3.3):

1. Partition the pentagon $V_1 \ldots V_5$ and calculate A_1, A_2, A_3, the area of $\triangle_1 = \triangle V_1V_2V_3$, $\triangle_2 = \triangle V_1V_3V_4$, $\triangle_3 = \triangle V_1V_4V_5$;

[3] This is known as the *barycentric coordinate system*.

2. $p_i = A_i / \sum_{i=1}^{3} A_i$ and let $I_1 = [0, p_1), I_2 = [p_1, p_1 + p_2), I_3 = [p_1 + p_2, 1]$;

3. Generate $V \sim$ Unif $[0, 1]$ and let (X, Y, Z) be the vertices of $\triangle_i$ if $V \in I_i, i = 1, 2, 3$;

4. Generate $U_j \overset{i.i.d}{\sim}$ Unif $[0, 1]$, $j = 1, 2, 3$ and let $x_j = -\log(U_j)$;

5. $c_j = x_j / \sum_{j=1}^{3} x_j$.

6. Compute $P = c_1 X + c_2 Y + c_3 Z$.

Neumann BCs We then describe the two-staged position-update rule. First, compute the 'naive update' (new position if there is no collision), $\boldsymbol{r}' = \boldsymbol{r} + \Delta \boldsymbol{r}$ and check if it is within the polygon. Take the pentagon in Figure 3.3 as an example, to check if the point P is inside, do:

1. Compute S_i, the sighted areas of $\triangle P V_i V_{i+1}, i = 1, .., 5 (P_5 = P_1)$ using the determinant method;

2. If all $S_i \geqslant 0$, conclude that P is inside; Otherwise, if $S_i < 0$, P needs to be reflected w.r.t to the side $V_i V_{i+1}$.

Whenever collision is confirmed, the position update follows the classical reflection law in optics (incident angles equal to reflection ones). Defensive programming is necessary to address possible multiple reflections within a single update (Figure 3.4). Failure to do so will result in spins escaping from the domain, leading to a loss of signal.

Results As an example, we let our domain be the unit triangle in the first quadrant. The numbers of spins and time steps are both chosen to be 10000 in the simulation. The whole simulation process took 0.28 seconds. The MATLAB codes for this example are supplied in Appendix F. A sample walker trajectory and the empirical dephase distribution are shown in Figure 3.5 and 3.6, respectively. Surprisingly, the empirical density function of the dephase also has roughly the shape of a triangle (!). A mathematical investigation will be conducted.

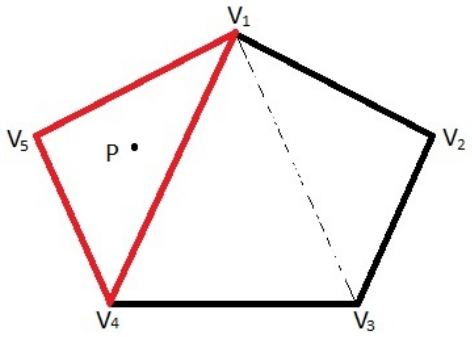


Figure 3.3: The partition of a polygon.

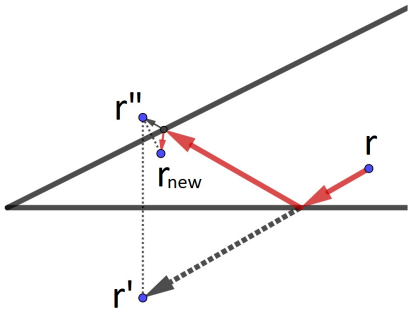


Figure 3.4: Multiple reflections.

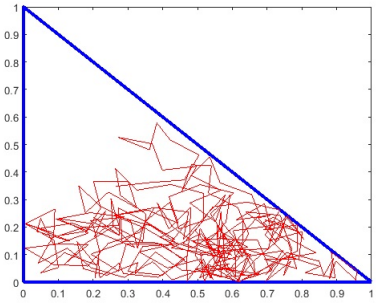


Figure 3.5: Random walk in a triangle: a sample path.

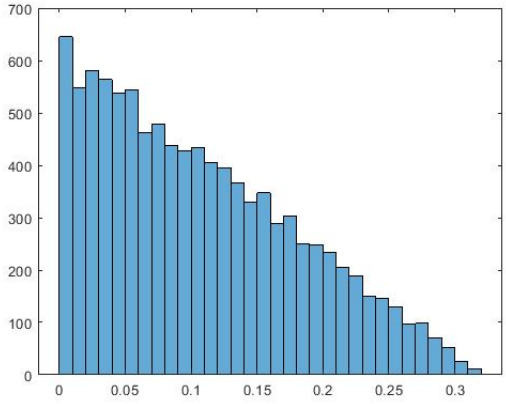


Figure 3.6: Empirical dephase distribution: triangular domain.

3.3.3 Single fiber Bundle

Nerve fibers or axons make up the most of the white matter. Nerve fibers connect gray matter areas (the locations of nerve cell bodies) of the brain to each other, and carry nerve impulses between neurons. The in-vivo detection of their geometrical and topological properties are of great interest.

Geometrical specification Within a voxel, the geometry of individual nerve fiber bundle can be modeled as equidistant, identical, parallel, infinite cylindrical shells with thickness (Figure 3.7). We further assume that no mobile water molecule is present within the

shell layer. Therefore, specifying the geometry requires the following parameters:

- $\hat{\boldsymbol{u}}$, a 3D unit vector along the fiber direction;
- d, halved fiber separation;
- R_{in}, fiber interior radius;
- R_{out}, fiber exterior radius.

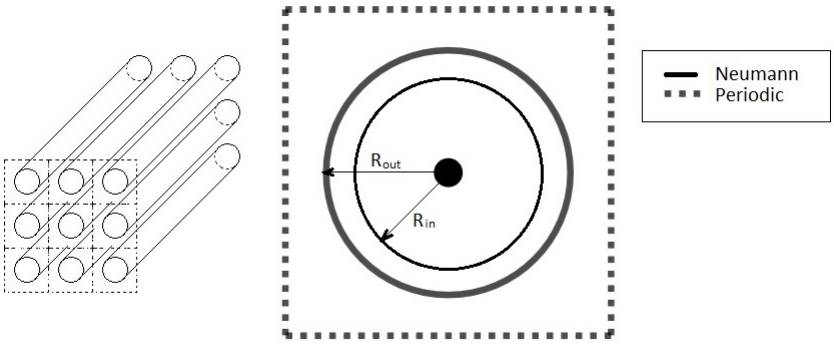


Figure 3.7: Model of one fiber bundle.

The simulation of Brownian trajectories under general orientation $\hat{\boldsymbol{u}}$ can be algebraically complicated. We avoid this problem by first running the simulation under the assumption that the fiber orientation aligns with the $z-$axis, and then applying a rotation transformation to the simulated trajectories so that $\hat{\boldsymbol{z}} = (0, 0, 1)$ becomes $\hat{\boldsymbol{u}}$.

Notice the structure of interest exhibits symmetry in two ways. First, translational invariance holds in the axial direction of the cylinders. This implies that the random walk can be orthogonally decomposed into two independent processes, namely the restricted motion in the transverse plane and the free diffusion in the longitudinal direction. Second, the transverse cross section shows periodic pattern. Therefore, to simulate the 2D random walks in the cross-section, the problem can be reformulated inside a minimal cell, refereed to as a lattice [4]. The side length of the square-shaped lattice equals to the fiber separation. A single ring is centrally placed inside the lattice. Boundary conditions required include periodic BCs imposed on the lattice boundary, along with Neumann BCs on both circles of the ring.

[4]similar to a crystal lattice in chemistry.

Initialization Initial positions are generated in a straight forward acceptance-rejection (A-R) fashion. A candidate point is uniformly sampled within the lattice square and is accepted unless it is within the cylindrical shell. Such action is repeat until a desired number of initial positions (N_{walker}) are sampled. Since typically the shell layer occupies only a small fraction of the lattice volume, the A-R sampling is efficient.

Intra-axonal updates Position updates of the 2D restricted random walk are are treated separately for intra-axonal and extra-axonal cases. At any iteration, let the current position be $\boldsymbol{r}$ and the jump vector be $\Delta\boldsymbol{r}$. In the intra-axonal case, at each step, one only needs to check for collision against the inner circle. Whenever the 'naive' update, $\boldsymbol{r\prime} = \boldsymbol{r} + \Delta\boldsymbol{r}$, has norm greater than R_{in}, one needs to solve an equation to find the collision point:

$$\|r + t\Delta\boldsymbol{r}\| = R, \tag{3.8}$$

where t is a scalar and $R = R_{\text{in}}$. This leads to a quadratic equation:

$$\|\Delta\boldsymbol{r}\|^2 t^2 + 2\boldsymbol{r} \cdot \Delta\boldsymbol{r}t + \|r\|^2 - R^2 = 0. \tag{3.9}$$

The positive root is chosen (Figure 3.8) for obvious reason. The collision point is then treated as the new starting position and the remainder of the initial jump, $(1-t)\Delta\boldsymbol{r}$, is taken as the new jump after reflection with respect to the tangent line. Similar to the triangle example, the problem of multiple reflections[5] needs to be addressed. Figure 3.3.3 shows a sample trajectory for the intra-axonal case.

Extra-axonal updates For the extra-axonal case, a walker may transit across several lattices. There are multiple possible implementations of the periodic BC. Here we describe our solution. First, locate the lattice containing the current position, represented by an index. Then, check for collision against the outer layer of the ring in the lattice.

The lattice containing all the initial positions is the region $\{(x, y) :\mid x \mid< d, \mid y \mid< d\}$ and its index is designated as $(0, 0)$. For an arbitrary point (x, y), it is located in the $(i, j)^{-th}$ lattice $(i, j \in \mathbb{Z})$ if its coordinates satisfy inequalities:

$$\{(x, y) :\mid x - 2id \mid< d, \mid y - 2jd \mid< d\}. \tag{3.10}$$

It can be easily verified that

$$(i, j) = \left\lfloor \frac{(x, y)/d + 1}{2} \right\rfloor, \tag{3.11}$$

[5]Multiple reflections can in fact occur in any convex domain.

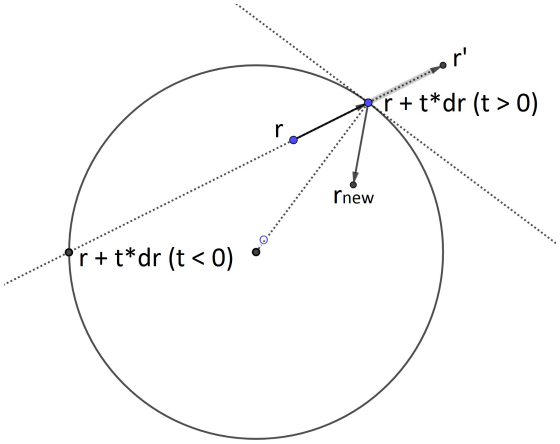


Figure 3.8: Interior collision: the larger root of equation 3.9 is chosen.

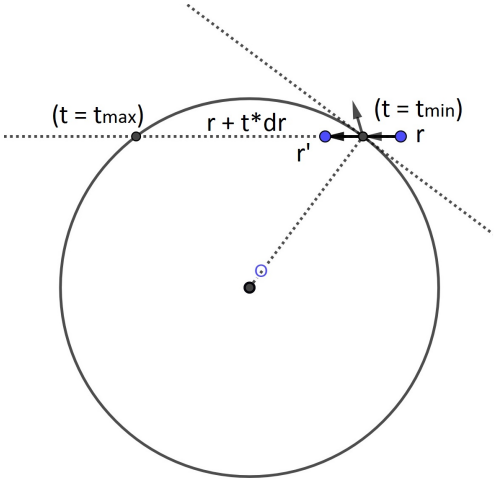


Figure 3.9: Exterior collision: the smaller root of equation 3.9 is chosen.

where $\lfloor \cdot \rfloor$ is the floor function. The local coordinates of the walker (and the 'naive' update) are then computed w.r.t to the lattice center:

$$\boldsymbol{r}_{\text{loc}} = \boldsymbol{r} - 2d(i, j). \tag{3.12}$$

Collision the local coordinates is handled almost identically to the treatment for the intra-axonal case, with two exceptions. First, the smaller root of the equation **??** is taken for computing the collision point (Figure 3.9). Second, multiple collision is impossible.

Adjusting for fibre axes Finally, as promised, the simulated trajectories need to be adjusted in accordance to the prescribed fiber orientation $\hat{\boldsymbol{u}}$. One way of achieving this is using the Rodrigues' rotation formula[6]. It states the following: If $\boldsymbol{v} \in \mathbb{R}^3$ and $\hat{\boldsymbol{k}}$ is a unit vector describing an axis of rotation about which v rotates by an angle θ according to the right hand rule, the rotated vector $\boldsymbol{v}_{\text{rot}}$ can written as

$$\boldsymbol{v}_{\text{rot}} = \boldsymbol{v}\cos\theta + (\hat{\boldsymbol{k}} \times \boldsymbol{v})\sin\theta + \hat{\boldsymbol{k}}(\hat{\boldsymbol{k}} \cdot \boldsymbol{v})(1 - \cos\theta). \tag{3.13}$$

[6]Other methods include using quaternions, Euler angles etc.

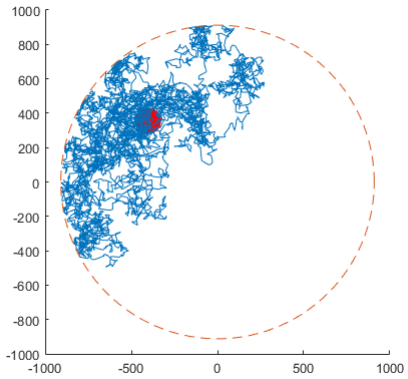


Figure 3.10: Intra-axonal diffusion: a sample path.

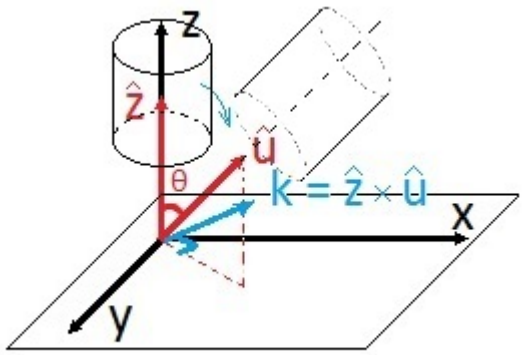


Figure 3.11: Rotation: adjusting trajectories for fiber orientation.

Since our desired transformation rotates $\hat{\boldsymbol{z}}$ into $\hat{\boldsymbol{u}}$, the (pre-normalized) rotation axis and the rotation angle are determined by (Figure 3.11):

$$\begin{aligned} \boldsymbol{k} &= \hat{\boldsymbol{z}} \times \hat{\boldsymbol{u}}, \\ \cos\theta &= \hat{\boldsymbol{z}} \cdot \hat{\boldsymbol{u}}. \end{aligned}$$

Notice that $\|\boldsymbol{k}\| = \sin\theta$, the second term on the RHS of equation 3.13 becomes $\boldsymbol{k} \times \boldsymbol{v}$.

Rewriting equation 3.13 into a matrix form is convenient for computer programming. Using the matrix notation of taking cross-product:

$$\boldsymbol{f} \times \boldsymbol{v} = [k]_\times \boldsymbol{v}, \tag{3.14}$$

where $[\boldsymbol{k}]_\times = \begin{bmatrix} 0 & -k_z & k_y \\ k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{bmatrix}$, and the fact that since $\boldsymbol{k}$ itself is a cross product, $[\boldsymbol{k}]_\times$ can be written as[7]

$$[\boldsymbol{k}]_\times = \hat{\boldsymbol{u}}\hat{\boldsymbol{z}}^T - \hat{\boldsymbol{z}}\hat{\boldsymbol{u}}^T.$$

Furthermore, simplifying the third term of the RHS of equation 3.13 yeilds

$$\hat{\boldsymbol{k}}(\hat{\boldsymbol{k}} \cdot \boldsymbol{v})(1 - \cos\theta) = (1 - \cos\theta)\hat{\boldsymbol{k}}(\hat{\boldsymbol{k}}^T \boldsymbol{v}) = \frac{(1 - \cos\theta)}{\sin^2\theta}(\boldsymbol{k}\boldsymbol{k}^T)\boldsymbol{v} = \frac{1}{1 + \cos\theta}(\boldsymbol{k}\boldsymbol{k}^T)\boldsymbol{v}.$$

[7] https://en.wikipedia.org/wiki/Cross_product#Alternative_ways_to_compute_the_cross_product

Therefore, the desired rotation matrix can be written as

$$\boldsymbol{R}_{\boldsymbol{k}} = \cos\theta \mathbb{1}_3 + (\hat{\boldsymbol{u}}\hat{\boldsymbol{z}}^T - \hat{\boldsymbol{z}}\hat{\boldsymbol{u}}^T) + \frac{1}{1+\cos\theta}(\boldsymbol{k}\boldsymbol{k}^T), \tag{3.15}$$

where $\mathbb{1}_3$ is the 3×3 identity matrix. The desired transform is simply

$$\boldsymbol{v}_{\text{rot}} = \boldsymbol{R}_{\boldsymbol{k}}\boldsymbol{v}. \tag{3.16}$$

Results As an example, we consider a geometry with specifications $\hat{\boldsymbol{u}} = \hat{\boldsymbol{x}}$, $(R_{\text{in}}, R_{\text{out}}, d) = (2.4, 2.5, d)\mu$m, where $\hat{\boldsymbol{x}} = (1,0,0)^T$. Again we used simulation parameters $N_{\text{walker}} = N_{\text{time}} = 10000$ and the whole simulation took 288s. In Appendix F, the reader can find the core MATLAB codes for simulation in such geometry. As one can see from the outputs, the dephase distribution in the transverse direction ($yz-$plane) shows a non-Gaussian profile and has much a smaller variance than the one in the longitudinal direction. Measured diffusion signals and their corresponding dephase distributions are shown in Table 4.1 and Figure 3.12.

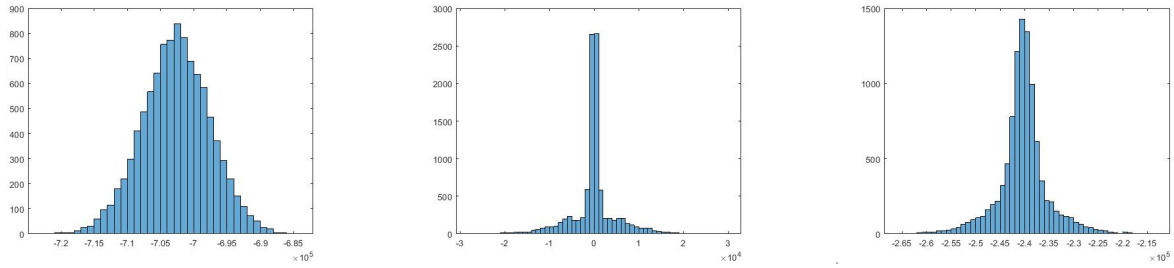


Figure 3.12: Dephase distribution ($\hat{\boldsymbol{g}} = \hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, (\cos(70°), \sin(70°), 0)^T$, from left to right).

Gradient Direction	Diffusion Signal
$(1,0,0)^T$	0.0019
$(0,1,0)^T$	0.0065
$(\cos(70°), \sin(70°), 0)^T$	0.0051

Table 3.1: Diffusion signals (single fiber bundle).

3.4 Timestepping Error

In this section, the theoretical framework for analysing the error due to timestepping is laid out[8]. It suffices to consider a scalar-valued position process although the same calculation steps apply to vector-valued processes as well.

Let $\{X_t\}$ be a partially-reflected Brownian motion in domain $\Omega \subset \mathbb{R}$ with initial distribution $\rho_0(x)$ and propagator $G_t(x, x')$. A linear diffusion-weighing gradient is present during a time interval $[0, T]$ with strength g and temporal profile $f(t)$. Therefore, the total dephase at $t = T$ is given by

$$\varphi = \gamma g \int_0^T X_t f(t) dt. \tag{3.17}$$

In basic Monte Carlo simulations, the time interval is divided into a fine mesh as follows:

$$0 = t_0 < t_1 < \cdots < t_k = T, \tag{3.18}$$

which typically has uniform time steps $\Delta t = \frac{T}{k} = t_j - t_{j-1}, j = 0, 1, \cdots$. To quantify the discretization error, let us first assume that the realizations of at the time nodes are given, denoted by

$$\{X_j\} = X_0, X_1, X_2, \cdots, X_k, \text{ where } X_j = X_{t_j}. \tag{3.19}$$

This means that for now, the sample paths of $\{X_t\}$ are pinned at locations[9] $\{X_j\}_{j=0}^k$. If we approximate the time integral for φ by the right rectangular rule, the simulated dephase is given by

$$\hat{\varphi} = \gamma g \sum_{j=1}^{k} X_j f(t_j)(t_j - t_{j-1}). \tag{3.20}$$

The error, given that the skeleton trajectory is $\{x_j\}$, is the difference between the extact integral and its numerical approximation,

$$(\varphi - \hat{\varphi})|\{X_j = x_j\} = \gamma g \sum_{j=1}^{k} \int_{t_{j-1}}^{t_j} (X_t^{(j)} f(t) - x_j f(t_j)) dt, \tag{3.21}$$

[8] As the name sugguests, the root of this error is purely because of discretization. In other words, this is the error we get if we collect infinitely many sample trajectories, which needless to say, is always lower than the actual error.

[9] In the analysis we assume that X_j's are simulated exactly. In other words given that initial position is x_0, X_1 is sampled from the exact distribution $G_{t_1}(x_0, x_1)$ (with the proper normalisation constant), etc.

where $\{X_t^{(j)}\}_{t=t_{j-1}}^{t_j}$, the $j-$th segment of the position process, satisfies initial and final conditions $X_{t_{j-1}}^{(j)} = x_{j-1}, X_{t_j}^{j} = x_j$. The accuracy of the simulation is ultimately quantified by the distribution of the random variable $(\varphi - \hat{\varphi})$. For that, we would like to evaluate its moments.

Due to the Markov property of $\{X_t\}$'s, all of its segments are mutually independent and consequently so are the time integrals in the summation above. In addition, the density of $X_t^{(j)}$ is equal to

$$f^{(j)}(x,t) = \frac{G_{t-t_{j-1}}(x_{j-1},x)G_{t_j-t}(x,x_j)}{G_{t_j-t_{j-1}}(x_{j-1},x_j)}, \tag{3.22}$$

where the denominator (normalisation constant) is determined from the Chapman-Kolmogorov equation. Densities of arbitrary orders can be expressed in similar fashion. As a result, the mean and auto-covariance functions of $X_t^{(j)}$ can be expressed in terms of the propagator:

$$\begin{aligned} m^{(j)}(t) &= \mathbb{E}(X_t^{(j)}) = \int_\Omega x \frac{G_{t-t_{j-1}}(x_{j-1},x_j)G_{t_j-t}(x,x_j)}{G_{t_j-t_{j-1}}(x_{j-1},x_j)}dx, \quad t_{j-1} < t < t_j, \\ C^{(j)}(t,t') &= \mathbb{E}\left[(X_t^{(j)} - m^{(j)}(t))(X_{t'}^{(j)} - m^{(j)}(t'))\right] \\ &= \iint_{\Omega^2} (x - m^{(j)}(t))(x' - m^{(j)}(t')) \frac{G_{t-t_{j-1}}(x_{j-1},x)G_{t'-t}(x,x')G_{t_j-t'}(x',x_j)}{G_{t_j-t_{j-1}}(x_{j-1},x_j)}dxdx', \\ &\quad t_{j-1} < t \leqslant t' < t_j. \end{aligned} \tag{3.23}$$

Therefore given the skeleton trajectory, the timestepping error has conditional mean

$$\mathbb{E}\left((\varphi - \hat{\varphi})|\{X_j = x_j\}\right) = \gamma g \sum_{j=1}^{k} \int_{t_{j-1}}^{t_j} (m^{(j)}(t)f(t) - x_j f(t_j))dt \tag{3.24}$$

and conditional variance

$$\begin{aligned} \mathrm{Var}\left((\varphi - \hat{\varphi})|\{X_j = x_j\}\right) &= \gamma^2 g^2 \sum_{j=1}^{k} \mathbb{E}\left(\int_{t_{j-1}}^{t_j} \left(X_t^j - m^{(j)}(t)\right) f(t)dt\right)^2 \\ &= 2\gamma^2 g^2 \sum_{j=1}^{k} \int_{t_{j-1}}^{t_j} dt f(t) \int_t^{t_j} dt' f(t') C^{(j)}(t,t'). \end{aligned} \tag{3.25}$$

The *bias* due to time discretization is defined as the mean of the error. By the law of total expectation, it equal to the conditional mean averaged over all possible skeletons:

$$\begin{aligned}\text{Bias} &= \mathbb{E}(\varphi - \hat{\varphi}) = \mathbb{E}\left[\mathbb{E}\left((\varphi - \hat{\varphi})|\{X_j\}\right)\right] \\ &= \int_\Omega dx_0 \rho_0(x_0) \int \cdots \int_{\Omega^k} dx_1 \cdots dx_k G_{t_1}(x_0, x_1) G_{t_2 - t_1}(x_1, x_2) \\ &\quad \times \cdots G_{t_k - t_{k-1}}(x_{k-1}, x_k) \mathbb{E}\left((\varphi - \hat{\varphi})|\{X_j = x_j\}_{j=0}^k\right).\end{aligned} \tag{3.26}$$

The $(k+1)$-fold integral arises from the joint density of the skeleton. Finally, the (unconditional) variance of the discretization error is given by the law of total variance:

$$\text{Var}(\varphi - \hat{\varphi}) = \text{Var}\left[\mathbb{E}\left((\varphi - \hat{\varphi})|\{X_j\}\right)\right] + \mathbb{E}\left[\text{Var}\left((\varphi - \hat{\varphi})|\{X_j\}\right)\right], \tag{3.27}$$

where both terms are computed using the joint density of $\{X_j\}_{j=0}^k$, similar to the computation of the bias.

3.4.1 Application: free diffusion

We would like to apply the described general paradigm to assess our simulation of the free diffusion. In this case, it suffices to let $\{X_t\}$ be its increment process $\{X_t - X_0\}$ since the position process is spatially homogeneous and the initial position has zero contribution to the dephase. The increment process is the scaled Wiener process. Therefore, $\{X_t^{(j)}\}$ is recognized as the generalised Brownian bridge,

$$X^{(j)}(t) = x_{j-1} + \Delta x_j \tau + \sqrt{2D(t_j - t_{j-1})} B(\tau), \tag{3.28}$$

where $\tau = \frac{t - t_{j-1}}{t_j - t_{j-1}}$ is the normalised time and $\Delta x_j = x_j - x_{j-1}$. Here $\{B(\tau)\}$ is the standard Brownian bridge defined by

$$B(\tau) = W_\tau | (W_1 = 0), \quad 0 \leqslant \tau \leqslant 1, \tag{3.29}$$

where $\{W_t\}$ is the standard Wiener process.

It is well-known that the Brownian bridge has mean and auto-covariance functions

$$\begin{aligned}\mathbb{E}\left(B(\tau)\right) &\equiv 0; \\ \mathbb{E}\left(B(\tau)B(\tau')\right) &= \tau(1 - \tau'), \quad \tau < \tau'.\end{aligned} \tag{3.30}$$

Thus we obtain functions $m^{(j)}(t)$ and $C^{(j)}(t,t')$ without using the propagator of X_t:

$$\begin{aligned} m^{(j)}(t) &= x_{j-1} + \Delta x_j \tau, \\ C^{(j)}(t,t') &= 2D(t_j - t_{j-1})\tau(1-\tau'), \quad t < t', \end{aligned} \tag{3.31}$$

where $\tau' = \frac{t'-t_{j-1}}{t_j - t_{j-1}}$. Assume $[0,T]$ is discretized by a uniform grid of unit spacing $\Delta t = \frac{T}{k}$ and denote $q = \gamma g \Delta t$, $s = \sqrt{2D\Delta t}$. The conditional mean and variance of the error become

$$\begin{aligned} \mathbb{E}\left((\varphi - \hat{\varphi})|\{X_j = x_j\}\right) &= \gamma g \sum_{j=1}^{k} \int_0^1 \left((x_{j-1} + \Delta x_j \tau) f(t_{j-1} + \Delta t\tau) - (x_{j-1} + \Delta x_j) f(t_j)\right) \Delta t d\tau \\ &= q \sum_{j=1}^{k} (a_j x_{j-1} + b_j \Delta x_j), \\ \mathrm{Var}\left((\varphi - \hat{\varphi})|\{X_j = x_j\}\right) &= 2q^2 s^2 \sum_{j=1}^{k} c_j, \end{aligned} \tag{3.32}$$

where the coefficients are

$$\begin{aligned} a_j &= \int_0^1 (f(t_{j-1} + \Delta t\tau) - f(t_j)) d\tau; \\ b_j &= \int_0^1 (\tau f(t_{j-1} + \Delta t\tau) - f(t_j)) d\tau; \\ c_j &= \int_0^1 d\tau \tau f(t_{j-1} + \Delta t\tau) \int_\tau^1 d\tau' f(t_{j-1} + \Delta t\tau')(1-\tau'). \end{aligned} \tag{3.33}$$

In our simulation, $x_0 = 0$, $\Delta x_j = s z_j$, $x_j = x_{j-1} + \Delta x_j$, where z_j's are independently drawn from the standard Gaussian distribution. Therefore, $\mathbb{E}(X_j) = 0$ for all j and it is established that $\hat{\varphi}$ is an unbiased estimate for φ (no systematic error):

$$\begin{aligned} \text{Bias} &= \mathbb{E}\left[\mathbb{E}\left((\varphi - \hat{\varphi})|\{X_j\}\right)\right] \\ &= q \sum_{j=1}^{k} \left(a_j \mathbb{E}(X_{j-1}) + b_j \mathbb{E}(\Delta X_j)\right) = 0. \end{aligned} \tag{3.34}$$

Lastly, to compute the variance of the error, we first calculate the variance of the conditional

mean:

$$\begin{aligned}\mathrm{Var}\left[\mathbb{E}\left((\varphi-\hat{\varphi})|\{X_j\}\right)\right] &= q^2\sum_{j=1}^{k}\mathrm{Var}\left(a_jX_{j-1}+b_j\Delta X_j\right)+2q^2 \\ &\times\sum_{1\leqslant i<j\leqslant k}\sum\mathbb{E}\left(a_ia_jX_{i-1}X_{j-1}+a_ib_jX_{i-1}\Delta X_j+b_ia_jX_{j-1}\Delta X_i+b_ib_j\Delta X_i\Delta X_j\right) \\ &= q^2\sum_{j=1}^{k}\left(a_j^2s^2(j-1)+b_j^2s^2\right)+2q^2\sum_{1\leqslant i<j\leqslant k}\sum\left(a_ia_js^2(i-1)+b_ia_js^2\right).\end{aligned} \tag{3.35}$$

The relations invoked are

$$\mathbb{E}\left(X_iX_j\right)=s^2\min\left(i,j\right),\quad \mathbb{E}\left(\Delta X_i\Delta X_j\right)=s^2\delta_{ij},\quad \mathbb{E}(X_i\Delta X_j)=s^2\mathbb{1}_{i\geqslant j}, \tag{3.36}$$

where δ_{ij} and $\mathbb{1}$ are Kronecker delta and indicator functions respectively.

Since the conditional variance is position-independent, its average over $\{X_j\}$ is trivial. Combining both terms in the total variance formula, one gets

$$\begin{aligned}\mathrm{Var}(\varphi-\hat{\varphi}) &= 2q^2s^2\sum_{j=1}^{k}c_j+q^2s^2\sum_{j=1}^{k}\left(a_j^2(j-1)+b_j^2\right) \\ &+2q^2s^2\sum_{1\leqslant i<j\leqslant k}\sum\left(a_ia_j(i-1)+b_ia_j\right).\end{aligned} \tag{3.37}$$

Since we know that in the case of free diffusion, both φ and $\hat{\varphi}$ are Gaussian, the error itself must be Gaussian. Therefore, we conclude that

$$(\varphi-\hat{\varphi})\sim N(0,\mathrm{Var}(\varphi-\hat{\varphi})). \tag{3.38}$$

Assuming a constant temporal profile, i.e $f(t)\equiv 1$, a direct computation yields $a_j=0, b_j=-\frac{1}{2}, c_j=\frac{1}{24}$. Thus, the variance of the error is reduced to

$$\mathrm{Var}(\varphi-\hat{\varphi})=\frac{1}{3}q^2s^2k=\frac{2}{3}D\gamma^2g^2T\Delta t^2\propto\Delta t^2. \tag{3.39}$$

Therefore, in the case of free diffusion, the discretization error converges linearly in time step (in L_2 sense).

3.4.2 Effect of Reflective BCs

It turns out that the presence of a reflective boundary substantially degrades the accuracy. Consider a reflected Brownian motion on the non-negative real line denoted as $\{X_t\}$. Its skeleton trajectory is sample as follows: Draw a uniform random point on the non-negative axis X_0. The random jump size and the position updates are given by

$$\Delta X_j = \sqrt{2D\Delta t} Z_j, \quad X_j = |X_{j-1} + \Delta X_j|, \quad j = 0, 1, \cdots, k, \tag{3.40}$$

where Z_j's are independent standard Gaussian random variables. [3] states that the 'global truncation error' of such scheme is approximately proportionate to the square-root of the grid size, (in L_1 sense). In our notation, this result reads

$$\mathbb{E}|X_t^{(j)} - X_j| \simeq \text{const}\sqrt{\Delta t}, \quad t_{j-1} \leqslant t \leqslant t_j. \tag{3.41}$$

As a result, assuming a constant temporal profile, the discretization error of the dephase has the following L_1-estimate:

$$\mathbb{E}|\varphi - \hat{\varphi}| \leqslant \gamma g \sum_{j=1}^{k} \int_{t_{j-1}}^{t_j} \mathbb{E}|X_t^{(j)} - X_j| dt \simeq \text{const} \times \gamma g k \Delta t \sqrt{\Delta t} \propto \sqrt{\Delta t}. \tag{3.42}$$

This slow convergence rate is both interesting and disappointing compared to the result established in the free diffusion case. This signifies the intrinsic difficulty of Monte Carlo methods.

Chapter 4

Improved Monte Carlo Method

4.1 Objective

As demonstrated in the last chapter, MC simulations of diffusion weighted MR signal can be computationally expensive. Roughly speaking, the simulation time is proportional to the product of the number of independent trajectories N_{walker} and the number of time steps R_{time}. Unfortunately, little can be done to reduce N_{walker}. It is however, possible to reduce the number of time steps by considering some 'maximum free space' surrounding the particle (the meaning of which will be elaborate later). This leads to the geometrically adaptive fast random walk (GAFRW) algorithm, first introduced in [14].

First, let us consider an analogy in numerical integration (for deterministic integrands). Newton-Cotes methods[1] use equispaced grids, compared to Gaussian quadrature rules where ingeniously designed, uneven grids are used. While Newton-Cotes methods are more general, Gaussian quadrature rules are much more efficient when integrands can be well approximated by polynomials.

In our problem 2.59, the integrand is stochastic. The MC schemes we showed in Chapter 3 uses uniform time steps regardless of the nature of the path. If somehow we know that the diffusion in certain areas is more 'free' then others, analytical results can then be employed to avoid simulating detailed trajectories.

[1] Trapezoidal rules, simpson rules are all examples of Newton-Cotes methods

The original algorithm simulates the diffusion signal under the FID sequence. In this chapter, GAFAW is generalized for sequences with piecewise constant temporal profiles such as FID, rectangular PGSE and CPMG sequences. The experiment in this chapter uses the PGSE sequence.

The structure of the present chapter is the following: Section 4.2 introduces the idea of position-dependent time steps. Section 4.4 covers how phase updates are calculated in. Section 4.5 explains how certain functions defined in previous sections are tabulated. Section 4.6 demonstrates an experiment using the improved scheme for crossing fibers. Lastly, section 4.7 briefly covers a potential generalization to the Robin case, which will soon be finalized.

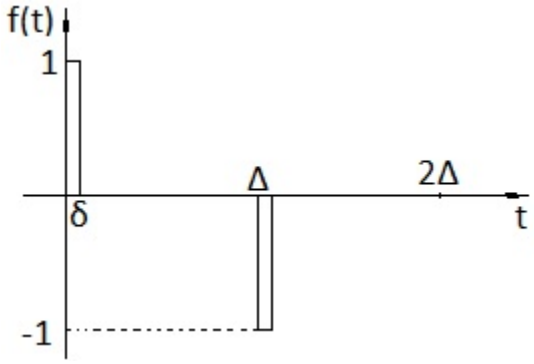


Figure 4.1: PGSE sequence under NPA

4.2 Geometrically adaptive time steps

Let Ω be the confining domain with a reflective boundary $\partial\Omega$. For pedagogical reasons, we first consider an important limiting temporal profile called the *narrow pulse approximation* (NPA). As we shall see, the advantage of spatially adaptive time steps is evident.

As its name suggests, NPA is valid when the pulse duration is much smaller than the diffusion time (Figure 4.1),

$$\delta \ll \Delta, \tag{4.1}$$

while the 'impulse' $\boldsymbol{q} = \boldsymbol{g}\delta$ is non-negligible[2]. Physically, NPA is interesting since it simplifies the calculation for phase accumulation since spins can be treated as immobile during short pulses. For the PGSE sequences under NPA, the dephase only depends on the total displacement travelled during the diffusion time,

$$\varphi = \int_0^T dt f(t)\boldsymbol{g}\cdot\boldsymbol{r}(t) = \delta\boldsymbol{g}\cdot(\boldsymbol{r}(\Delta)-\boldsymbol{r}(0)) = \boldsymbol{q}\cdot(\boldsymbol{r}(\Delta)-\boldsymbol{r}(0)), \tag{4.2}$$

where $\boldsymbol{r}(t)$ is a sample path of RBM.

Suppose that the initial positions are sampled (uniformly), only the final positions are needed to calculate the dephases. Doing so by using the paradigm in Chapter 3, one has no choice but to generate the entire trajectory. Therefore, our goal is to bypass as many intermediate positions as possible.

The FRW algorithm determines the time and position update based on whether or not the position at the current step is near the boundary. In addition, the generation of the final position requires a special treatment. The instructions for all cases are provided below.

4.2.1 Non-boundary case

Boundary layer thickness In GAFRW algorithm, positions that are not within a 'layer' centered at $\partial\Omega$ are considered 'far away' from the boundary. To establish a concrete criteria, the layer thickness, ε, is introduced as a numerical parameter. ε is chosen to be

[2]Under NPA, numerically solving the BT equation is difficult as δ needs to be finely discretized

extremely small compared to to L, the characteristic length of the domain.

The distance function computing the Euclidean distance between an interior point of Ω and $\partial\Omega$ is defined as

$$d_{\partial\Omega} : \Omega \mapsto \mathbb{R}^+,\ d_{\partial\Omega}(\boldsymbol{r}) = \inf_{\boldsymbol{r}' \in \partial\Omega} \|\boldsymbol{r} - \boldsymbol{r}'\| . \tag{4.3}$$

A point $\boldsymbol{r}$ is considered to be far away from $\partial\Omega$ if

$$d_{\partial\Omega}(\boldsymbol{r}) \geqslant \varepsilon, \tag{4.4}$$

For the time being, the initial position $\boldsymbol{r}_0$ is assume to satisfy the inequality 4.4.

Time steps In this case, let S_0 be the sphere centered at $\boldsymbol{r}_0$ with radius d. Clearly, S_0 is tangent to $\partial\Omega$. As long as the diffusing particle stays inside S_0, there is no chance for it to collide with the boundary. After a random amount of time t_1, the Brownian particle reaches the surface of S_0 for the first time. The exit point $\boldsymbol{r}_1$ is uniformly random on ∂S_0 due to radial symmetry. By solving a first exit problem, the exact cumulative distribution function of t_1 can be derived (in the next section). The cdf of the dimensionless first exit time is given by[3]

$$F_\tau(t) = 1 - 2\sum_{n=1}^{\infty}(-1)^{n+1} e^{-n^2\pi^2 t} \quad t \geqslant 0. \tag{4.5}$$

Therefore, t_1 can be generated via the inverse CDF technique and taken as our first time step. The numerical evaluation of both $F_\tau(t)$ and its inverse is covered in section 4.5. In fact, the tabulated values of $F_\tau(t)$ and all other 'special functions' in terms of infinite series are computed once and forever to save the cost of the Monte Carlo simulation.

Non-final updates If t_1 has not exceeded $T_{\max}$, a prescribed maximum time, the particle will continue its random walk until the time runs out. Since the random walk is Markov, $\boldsymbol{r}_1$ can be treated as the initial position and the same calculations are repeated to generate the new time steps and new positions, producing a finite sequence $\{(\boldsymbol{r}_n, T_n)\}_{n\geqslant 0}$, where $T_n = t_1 + t_2 + \cdots + t_n$. This 'spherical process' is illustrated in Figure 4.2.

[3] $F_\tau\left(\pi\sqrt{\frac{t}{2}}\right)$ is in fact the cdf of the Kolmogorov distribution, used in the Kolmogorov-Smirnov test in statistics.

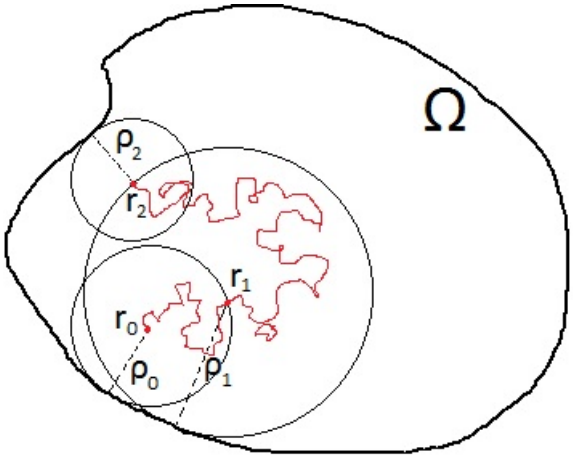


Figure 4.2: In the geometrically adaptive fast random walk algorithm, we only simulate the positions at time points when collision with the boundary is possible. The random first exit times for leaving spheres $S_0, S_1, \ldots, S_n$ are chosen as time steps.

Final update It is expected that for some n, $T_n < T_{\max} < T_{n+1}$, which implies that the particle does not have enough time to escape the sphere S_n. Therefore, the final position can only to take interior points of S_n (Figure 4.3). Therefore, the final time update is just

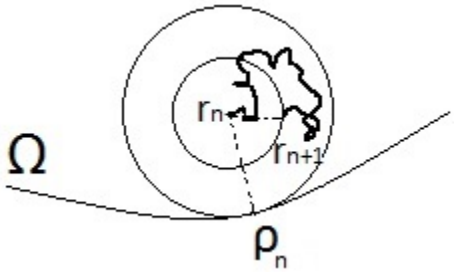


Figure 4.3: The final position update.

$t_{n+1} = T_{\max} - T_n$ and the average displacement is zero due to symmetry. We therefore want to study the conditional distribution of the final position if we are to provide a non-trivial position update rule.

Since the exact conditional CDF of the displacement size Δr is time-dependent, using the inverse CDF technique in general would be highly inefficient since tabulated values need to be stored for a large number of time steps, say t's. We avoid this problem by consider t's in different regimes. If t_{n+1} is small, intuitively, the walker can barely 'feel'

the boundary of S_n at all so the diffusion is approximately free (Gaussian). For large t's, it turns out that the conditional pdf is time-independent and we can generate Δr using the inverse CDF technique. Lastly, in the intermediate regime, the jump size is taken as its time-dependent mean,

$$R(t) = 1 - \frac{8M(t)}{S_0(t)}, \tag{4.6}$$

where $S_0(t) = 1 - F_\tau(t)$ and

$$M(t) = \sum_{n=1,3,5,\ldots} \frac{e^{-n^2\pi^2 t}}{n^2\pi^2}. \tag{4.7}$$

Overall, for a given final time step, we have the following function for generating the dimensionless final jump size,

function: FinalJump (t)

Case.i If $t < t_{\text{small}}$, generate a standard 3D Gaussian random vector $\boldsymbol{R}$ and keep computing

$$\Delta r = \sqrt{2t}\,\|\boldsymbol{R}\| \tag{4.8}$$

until $\Delta r_n \leqslant 1$ (fail-safe);

Case.ii If $t > t_{\text{large}}$, generate U following Unif $[0, 1]$ and take Δr as the solution to the transcendental equation

$$U = F_R(r) \equiv \frac{\sin(\pi r) - \pi r \cos(\pi r)}{\pi}; \tag{4.9}$$

Case.iii Otherwise, compute

$$\Delta r = R(t). \tag{4.10}$$

end

Overall, given that the current position $\boldsymbol{r}$ is not close to the boundary $\partial\Omega$. The time and position updates are performed according to the following scheme:

Scheme 1 Inputs: $\boldsymbol{r}$, $t_{\max}$ (time remaining); output: $\boldsymbol{r}_{\text{new}}$, t (geometrically adaptive time step).

S.1 Compute the distance between the boundary and the current position,

$$\rho = d_{\partial\Omega}(\boldsymbol{r}); \tag{4.11}$$

S.2 Generate U following Unif $[0, 1]$ and then compute the time step

$$t = F_{\tau}^{-1}(U)\frac{\rho^2}{D}, \tag{4.12}$$

where D is the diffusion coefficient;

S.3 Generate a uniformly random direction $\hat{\boldsymbol{u}}$ in $\mathbb{R}^3$, if

Case.i $t < t_{\max}$, compute

$$\boldsymbol{r}_{\text{new}} = \boldsymbol{r} + \hat{\boldsymbol{u}}\rho; \tag{4.13}$$

Case.ii $t > t_{\max}$, rewrite t as $t_{\max}$ and compute

$$\boldsymbol{r}_{\text{new}} = \boldsymbol{r} + \hat{\boldsymbol{u}} \cdot \rho \times \textbf{FinalJump}(t). \tag{4.14}$$

end

4.2.2 Boundary case

Release radius If the current position $\boldsymbol{r}$, fails to satisfy inequality 4.4, the particle is considered to be close to the boundary.

If the boundary is absorbing (Dirichlet BCs), the random walk simply gets terminated. For reflective BCs however, it would be inefficient to generate the same 'spherical processes' described in the non-boundary case. This is because the spheres generated for the next several steps would approximately be of size ε (Figure 4.4, (a)), which leads to simulations of detailed trajectories near the boundary.

To overcome this issue, the release radius R_{rls}, another numerical parameter, is introduced, which satisfies

$$\varepsilon \ll R_{\text{rls}} \ll L, \tag{4.15}$$

where L is the local radius of curvature of $\partial\Omega$ evaluated at the point on $\partial\Omega$ closest to the current position. The right inequality ensures that the *base*, the region $\partial\Omega \cap S_n$ is approximately flat (Figure 4.4 (b)), where $S_n = \{\boldsymbol{r} : \|\boldsymbol{r} - \boldsymbol{r}_n\| < R_{\text{rls}}\}$.

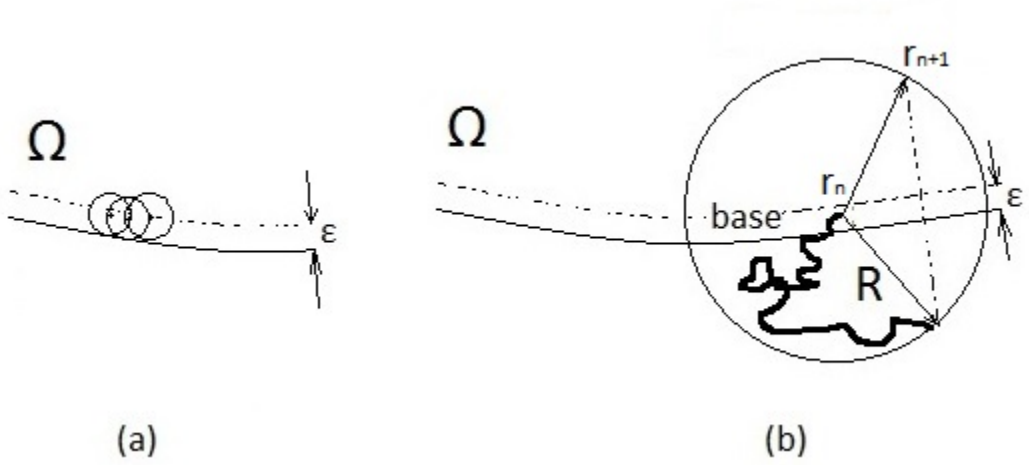


Figure 4.4: (a) When the newly updated position is close to the boundary, the regular position update method would generate tiny jumps in the next couple of steps, thus slowing down the simulation. (b) Instead, the jump size is set to be equal to the release radius R. If the uniformly sampled new position is outside Ω, it simply gets reflected.

Neumann BC It is known that the components of a 3D Brownian motion in all orthogonal directions are mutually independent. Thus, the random walk in tangent directions of the base are unaffected by the boundary.

On the other hand, the normal component of the Brownian motion is subject to a reflective boundary at the center of the base. Recall the Reflection Principle of the Wiener process (subsection 2.7.2) and we can implement the Neumann BC in the following way: Assuming for now that the reflective boundary is absent, we let the random walk continue freely inside S_n until it reaches ∂S_n. A uniformly random point on ∂S_n is taken as updated position if it is inside Ω. Otherwise, the reflection of the aforementioned point w.r.t. the base is chosen instead.

Such choice is justified as probabilistically, the reflected trajectory of free random walk is equivalent to random walk subject to Neumann BCs. The method for time update is identical to the non-boundary case and it is assumed that within the final time step t, the walker is always able to reach ∂S_n. Since R_{rls} is chosen to be small, t is small as well so that the accumulated time is extremely unlikely to exceed the maximum time. Therefore, it is safe not to include a special treatment addressing the final update for the boundary case.

Overall, we have the following scheme for the time and position update for the boundary case:

Scheme 2 Input: $\boldsymbol{r}$ (current position); output: $\boldsymbol{r}_{\text{new}}$, t (time step).

S.1 Compute $\hat{\boldsymbol{n}}$, the interior normal unit vector of $\partial\Omega$ at $\boldsymbol{r}$ (approximately);

S.2 Generate a uniformly random direction $\hat{\boldsymbol{u}}$ in $\mathbb{R}^3$ and if $\hat{\boldsymbol{u}} \cdot \hat{\boldsymbol{n}} < 0$, overwrite $\hat{\boldsymbol{u}}$ by[4]

$$\hat{\boldsymbol{u}} = \hat{\boldsymbol{u}} - 2(\hat{\boldsymbol{u}} \cdot \hat{\boldsymbol{n}})\hat{\boldsymbol{n}}; \tag{4.16}$$

S.3 Compute

$$\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + \hat{\boldsymbol{u}} R_{\text{rls}}; \tag{4.17}$$

S.4 Generate U following Unif $[0, 1]$ and compute

$$t = F_\tau^{-1}(U)\frac{R_{\text{rls}}^2}{D}. \tag{4.18}$$

end

4.2.3 Interlude: q-space Imaging

NPA naturally leads to an imaging technique called the *q-space imaging* (QSI), which has the potential of overcoming certain weakness of traditional DTI.

The key foundation of QSI is that NPA implies the Fourier duality between the diffusion signal and the 'average' propagator. To see this, Rename the initial and final positions in equation 4.2 as $\boldsymbol{r}$, $\boldsymbol{r}'$ respectively and plug into equation 2.60, we obtain the diffusion signal,

$$\begin{aligned} S(\boldsymbol{q}, \Delta) &= \mathbb{E}\left[e^{i\varphi}\}\right] = \mathbb{E}\left[e^{i\boldsymbol{q}\cdot(\boldsymbol{r}'-\boldsymbol{r})}\right] \\ &= \int_\Omega d\boldsymbol{r} \int_\Omega d\boldsymbol{r}' f(\boldsymbol{r}, \boldsymbol{r}'; \Delta) e^{i\boldsymbol{q}\cdot(\boldsymbol{r}'-\boldsymbol{r})}, \end{aligned} \tag{4.19}$$

[4] It is known as the Householder transformation

where $f(\boldsymbol{r}, \boldsymbol{r}'; \Delta)$ is the joint pdf of random positions $\boldsymbol{r}$ and $\boldsymbol{r}'$, which can be expressed as

$$f(\boldsymbol{r}, \boldsymbol{r}'; \Delta) = \rho(\boldsymbol{r}) G_\Delta(\boldsymbol{r}, \boldsymbol{r}'), \tag{4.20}$$

provided that the process is Markov. Plugging equation 4.20 into equation 4.19, the signal becomes

$$S(\boldsymbol{q}, \Delta) = \int_{\mathbb{R}^3} d\boldsymbol{r_1} \overline{G}_\Delta(\boldsymbol{r}_1) e^{-i\boldsymbol{q}\cdot\boldsymbol{r}_1}, \tag{4.21}$$

where $\boldsymbol{r}_1 = \boldsymbol{r}' - \boldsymbol{r}$ is the displacement vector and $\overline{G}_\Delta(\boldsymbol{r}_1)$ is the *ensemble average propagator* (EAP) defined as

$$\overline{G}_\Delta(\boldsymbol{r}_1) = \int_\Omega d\boldsymbol{r} G_\Delta(\boldsymbol{r}, \boldsymbol{r} + \boldsymbol{r}_1). \tag{4.22}$$

Equation 4.21 states that the signal is the Fourier transform of the EAP, which is analogous to the imaging equation 2.22. Similar to k-space, *q-space* consists of all possible values of $\boldsymbol{q}$.

EAP, obtained by taking the inverse Fourier transform of 4.21, is expected to carry spatial information about the confining geometry[5]. Qualitatively speaking, since the diffusion propagator is the likelihood of transition, its values increases in directions in which the diffusion is less restricted. Hence, q-space imaging has the potential of detecting multiple fiber orientation within voxels (Figure[6]4.5).[23] shows that QSI can be used to infer geometric parameters of brain cells.

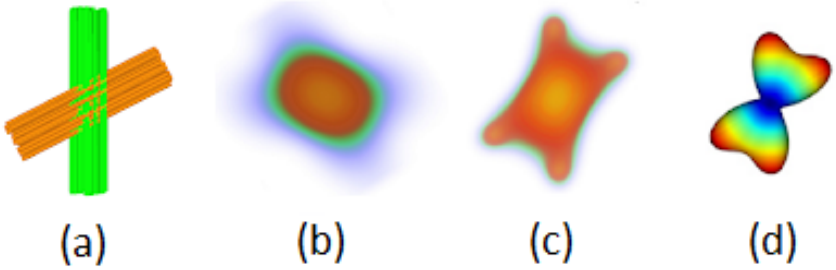


Figure 4.5: q-space imaging: (a) crossing fibers (the ground truth); (b) Diffusion signal in q-space; (c) EAP recovered by Fourier inversion; (d) directional EAP.

[5]In applications of QSI, NPA may not hold

[6] Source of the image:http://dsi-studio.labsolver.org/course/q-space-imaging-1

4.2.4 Summary

Combining all cases covered so far, the GAFRW algorithm for time and position update is summarized below.

Algorithm 4.1 Inputs: T (maximum diffusion time), $\boldsymbol{r}_{\text{ini}}$ (initial position); output: $\boldsymbol{r}_{\text{fin}}$ (final position).

S.1 Set $n = 0$, $\boldsymbol{r}_n = \boldsymbol{r}_{\text{ini}}$, $T_n = 0$;

S.2 While $T_n \leqslant T$, do

Case.i If $\boldsymbol{r}_n$ is far from $\partial\Omega$, execute Scheme 1;

Case.ii If $\boldsymbol{r}_n$ is close to $\partial\Omega$, execute Scheme 2;

S.3 return $\boldsymbol{r}_{\text{fin}} = \boldsymbol{r}_n$.

end

Under NPA, the following steps calculate the simulated diffusion signal:

Algorithm 4.2 Inputs: N_{walker}, $\boldsymbol{q}$ (sequence impulse); output: S.

S.1 Generate a uniformly random point in Ω, $\boldsymbol{r_0}$;

S.2 Apply Algorithm 4.1 with inputs $\Delta, \boldsymbol{r_0}$ to obtain final position $\boldsymbol{r}(\Delta)$;

S.3 Compute the dephase φ according to equation 4.2;

S.4 Repeat S.1 to S.3 N_{walker}, obtain a sequence $\{\varphi_k\}$.

S.5 Compute the diffusion signal

$$S = \overline{\exp\left(i\varphi_k\right)}. \tag{4.23}$$

end

4.3 Derivation of relevant distributions

4.3.1 First exit problem in unit sphere

Given that a random walker starts from the origin, we are interested in τ, the random amount of time need for the walker to exit the unit sphere,

$$\mathbb{B} = \{\boldsymbol{r} : \|r\| < 1\}. \tag{4.24}$$

In other words,

$$\tau = \min\{t : \boldsymbol{r}(t) \in \partial\mathbb{B}\}. \tag{4.25}$$

Figure 4.6 shows a sample path of the described process.

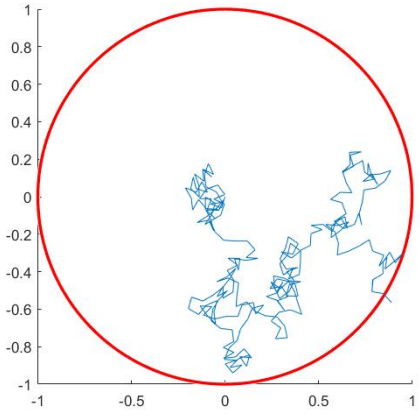


Figure 4.6: A simulated Brownian trajectory inside a unit disk ($dt = 0.001$, $D = 1$, $ds = \sqrt{4Ddt}$. First exit occurs at $\tau = 0.2620$.

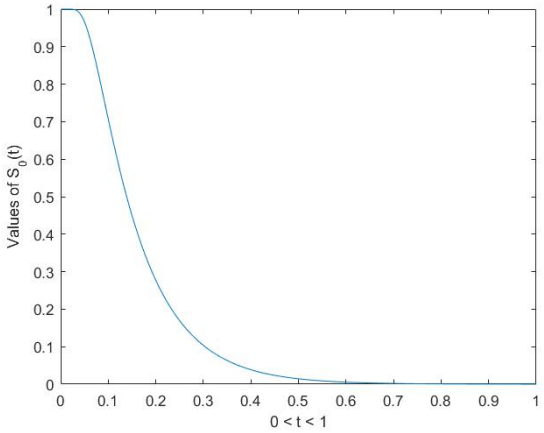


Figure 4.7: Graph of $S_0(t)$ for $0.001 < t < 1$, we can see that $S_0(1)$ decays extremely fast and is almost zero even for $t = 1$.

Position distribution Let $u(\boldsymbol{r}, t)$ be the pdf of the random position of the walker $\boldsymbol{r}$ at time t. In other words, $u(\boldsymbol{r}, t) = G_t(0, \boldsymbol{r})$, where $G_t(\boldsymbol{r}, \boldsymbol{r'})$ is the Green's function of the diffusion equation in the unit ball subject to absorbing boundary condition. Notice that the solution for $u(\boldsymbol{r}, t)$ should not depend on angular coordinates θ and ϕ due to the radial symmetry of the initial and boundary conditions. Therefore, denote $u(\boldsymbol{r}, t)$ as $u(r, t)$, $r \geqslant 0$

which satisfies the initial boundary problem (IBVP),

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{1}{r^2}\frac{\partial}{\partial r}(r^2\frac{\partial u}{\partial r}) \\ u(r,0) &= \delta_3(r) \\ u(1,t) &= 0 \quad \text{for} \quad t \geqslant 0, \end{aligned} \tag{4.26}$$

where $\delta_3(r)$ satisfies

$$1 = \int_0^\infty 4\pi r^2 \delta_3(r) dr. \tag{4.27}$$

Introduce a change of variable

$$u = \frac{p}{r} \quad \Rightarrow \quad u_t = \frac{p_t}{r}, \quad \frac{\partial}{\partial r}(r^2\frac{\partial u}{\partial r}) = p_{rr} r.$$

The PDE for $p(r,t)$ is equivalent to the heat equation in a slab. The IBVP for p can be written as

$$\begin{aligned} p_t &= p_{rr} \\ p(r,0) &= \frac{\delta_3(r)}{r} \\ p(1,t) &= 0 \quad \text{for} \quad t \geqslant 0. \end{aligned} \tag{4.28}$$

Applying the standard separation of variables procedure and multipling the solution of $p(r,t)$ by r, we acquire the solution for $u(\boldsymbol{r},t)$,

$$u(\boldsymbol{r},t) = \frac{\pi}{2}\sum_{n=1}^{\infty} e^{-n^2\pi^2 t}\text{sinc}(n\pi r).$$

where $\text{sinc}(x) = \frac{\sin(x)}{x}$. Integrating u over θ and ϕ (which is simply multiplying u by $4\pi r^2$) yields the radial distribution function,

$$f(r,t) = 2\sum_{n=1}^{\infty} e^{-n^2\pi^2 t} n\pi r \sin(n\pi r). \tag{4.29}$$

4.3.2 First Exit Time distribution

Inverse CDF technique As mentioned previously, the time steps GAFRW are random first exit times. We show that τ can be generated by inverting its *cumulative distribution function* (CDF),

$$F_\tau(t) = \mathbb{P}(\tau \leqslant t). \tag{4.30}$$

To see why it works, suppose that a random variable U follows Unif $[0,1]$ and let $T = F_\tau^{-1}(U)$, where F_τ^{-1} is the inverse of F_τ. It can be shown that T follows the same distribution of τ:

$$\mathbb{P}(T \leqslant t) = \mathbb{P}(F_\tau^{-1}(U) \leqslant t) = \mathbb{P}(U \leqslant F_\tau(t)) = F_\tau(t).$$

CDF of τ $F_\tau(t)$ can be calculated through its probabilistic meaning:

$$\begin{aligned} F_\tau(t) &= \mathbb{P}(\tau \leqslant t) = 1 - \mathbb{P}(\tau > t) \\ &= 1 - \mathbb{P}(t < \text{the shortest time it takes for } \boldsymbol{r}(t') \text{ to reach } \partial\mathbb{B}) \\ &= 1 - \mathbb{P}(\boldsymbol{r}(t') \in \mathbb{B} \text{ for } \forall\ 0 \leqslant t' \leqslant t) \\ &= 1 - S_0(t), \end{aligned} \tag{4.31}$$

where $S_0(t)$ is the *survival probability*, the probability that the random walker remains inside the unit sphere during the whole duration of $[0,t]$. It is equal to the integral of $u(\boldsymbol{r},t)$ over $\mathbb{B}$:

$$S_0(t) = \int_\Omega u(\boldsymbol{r},t)d\boldsymbol{r} = \int_0^1 f(r,t)dr. \tag{4.32}$$

The integration above is elementary and we have,

$$S_0(t) = 2\sum_{n=1}^{\infty}(-1)^{n+1}e^{-n^2\pi^2 t}. \tag{4.33}$$

In the long-time limit, the series is dominated by the first term:

$$S_0(t) \approx 2e^{-\pi^2 t}, \quad \text{for} \quad t \to \infty. \tag{4.34}$$

The graph of $S_0(t)$ is shown in Figure 4.7.

Short-time behaviour However, series 4.33 is ill-suited evaluating $S_0(t)$ when t is small (In fact if we set $t = 0$ we get $2\sum_{n=1}^{\infty}(-1)^n$, a divergent series!). Therefore, an alternative expression for $S_0(t)$ is needed. Taking the Laplace Transform of $S_0(t)$, we get

$$\mathcal{L}\{S_0\}(s) = 2\sum_{n=1}^{\infty}\frac{(-1)^{n+1}}{s+n^2\pi^2}, \tag{4.35}$$

which can be evaluated explicitly by using the residue summation technique described in Appendix C,

$$\mathcal{L}\{S_0\}(s) = \frac{1}{s} - \frac{1}{\sqrt{s}\sinh(\sqrt{s})}. \tag{4.36}$$

The initial value theorem of the Laplace transform states that the short-time behaviour of $S_0(t)$ depends on $\mathcal{L}\{S_0\}(s)$ when s is large. Applying geometric series, we have the expansion,

$$\begin{aligned}\frac{1}{\sqrt{s}\sinh(\sqrt{s})} &= \frac{2}{\sqrt{s}(e^{\sqrt{s}}-e^{-\sqrt{s}})} = \frac{2}{\sqrt{s}e^{\sqrt{s}}}\frac{1}{1-e^{-2\sqrt{s}}} \\ &= \frac{2}{\sqrt{s}e^{\sqrt{s}}}\sum_{n=0}^{\infty} e^{-2n\sqrt{s}} = 2\sum_{n=0}^{\infty}\frac{e^{-(2n+1)\sqrt{s}}}{\sqrt{s}}.\end{aligned} \tag{4.37}$$

The expansion is valid since $e^{-\sqrt{s}} \ll 1$ when s is large. Taking the inverse Laplace transform of 4.37, we get

$$S_0(t) = 1 - 2\sum_{n=0}^{\infty}\frac{1}{\sqrt{\pi t}}e^{-\frac{(2n+1)^2}{4t}}, \tag{4.38}$$

in which the relation

$$\mathcal{L}^{-1}\left\{\frac{e^{-\beta\sqrt{s}}}{\sqrt{s}}\right\} = \frac{1}{\sqrt{\pi t}}e^{-\frac{\beta^2}{4t}} \tag{4.39}$$

was invoked. The general terms in series 4.38 are extremely small when t approaches zero. Therefore, the short-time limit of $S_0(t)$ is given by

$$S_0(t) \approx 1 - \frac{2}{\sqrt{\pi t}}e^{-\frac{1}{4t}}, \quad \text{for} \quad t \to 0. \tag{4.40}$$

4.3.3 Conditional radial distribution

Given that the particle stays inside the unit ball for the whole duration of $[0, t]$, what would be an 'educated guess' for its final position at time t? This is the problem encountered in the final position update (subsection 4.2.1).

Due to radial symmetry, the direction of the jump should be uniformly random. On the other hand, the distribution of the jump size should be the conditional radial distribution given that first exit has not yet occurred at time t,

$$f_t(r|\boldsymbol{r}(t') \in \mathbb{B} \text{ for } \forall\ 0 \leqslant t' \leqslant t) = \frac{f(r,t)}{S_0(t)}, \tag{4.41}$$

where $f(r,t)$ is given by series 4.29. As discussed earlier, we would like to investigate the behaviour of 4.41 in different regimes of t. Figure 4.8 shows that the conditional radial pdf shifts to the right as time evolves.

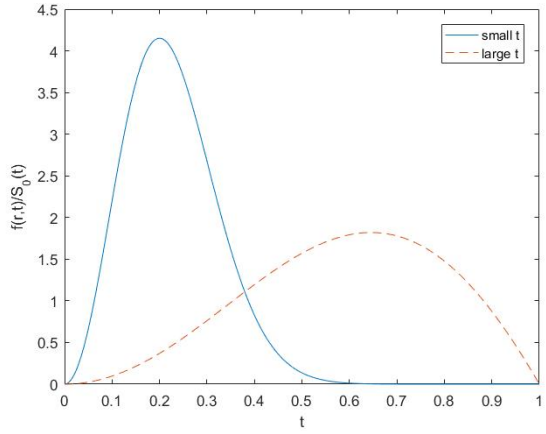


Figure 4.8: The pdfs of the conditional radial distribution for small and large t.

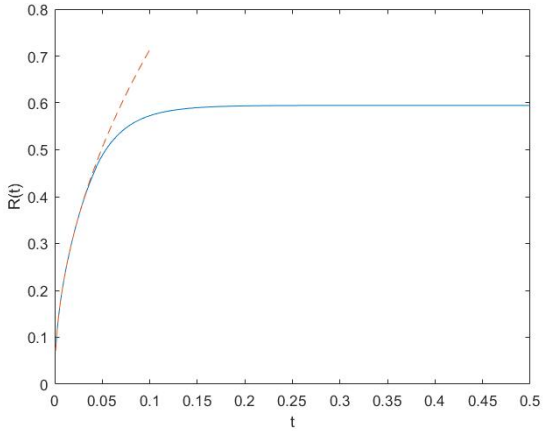


Figure 4.9: Graph of $R(t)$. The dotted red curve is the short-time limit, $4\sqrt{\frac{t}{\pi}}$. $R(t)$ approximately plateaus after reaching $t = 0.15$.

Long-time regime For large values of t, $f(r,t)$ has the approximation

$$f(r,t) \approx 2e^{-\pi^2 t}\pi r \sin(n\pi r), \tag{4.42}$$

which has the same decay rate as $S_0(t)$ does (4.40). Therefore, in the long-time limit, the conditional radial distribution becomes time-independent,

$$\frac{f(r,t)}{S_0(t)} \approx \pi r \sin(n\pi r). \tag{4.43}$$

Integrating the asymptotic pdf above, one arrive at the corresponding conditional CDF $F_R(r)$ in equation 4.9.

Intermediate regime The conditional CDF for intermediate values of t admits no simple expression. We therefore consider taking the conditional mean as our educated guess for r,

$$R(t) = \frac{1}{S_0(t)} \int_0^1 r f(r,t) dr. \tag{4.44}$$

Using elementary methods for integration, one arrives at equation 4.6. Figure 4.9 shows the graph of $R(t)$.

Short-time regime We witnessed the power of Laplace transform in terms analysing short-time asymptotics of $S_0(t)$. We wish to do the same to $R(t)$.

The Laplace transform of $M(t)$ given by 4.7 is given by

$$\mathcal{L}\{M(t)\} = \sum_{n=1,3,5,\ldots} \frac{1}{n^2\pi^2(s+n^2\pi^2)}, \tag{4.45}$$

which can be explicitly summed by using the summation formula C.15[7],

$$\mathcal{L}\{M(t)\} == \frac{1}{8s} - \frac{\tanh\left(\frac{\sqrt{s}}{2}\right)}{4s^{\frac{3}{2}}}. \tag{4.46}$$

Taking the inverse Laplace transform and multiplying by 8, we get

$$\begin{aligned} 8M(t) &= 1 - 2\mathcal{L}^{-1}\left\{\frac{1}{s^{\frac{3}{2}}}\frac{1-e^{-\sqrt{s}}}{1+e^{-\sqrt{s}}}\right\} \\ &= 1 - 2\mathcal{L}^{-1}\left\{\frac{1}{s^{\frac{3}{2}}}(1-2e^{-\sqrt{s}}+o(e^{-2\sqrt{s}}))\right\} \\ &= 1 - 4\sqrt{\frac{t}{\pi}} + 4\mathcal{L}^{-1}\left\{\frac{1}{s^{\frac{3}{2}}}(e^{-\sqrt{s}}+o(e^{-2\sqrt{s}}))\}\right\}. \end{aligned} \tag{4.47}$$

The relation $\mathcal{L}^{-1}\{s^{-\beta}\} = \frac{t^{\beta-1}}{\Gamma(\beta)}$ is used in the last step, where $\Gamma(z)$ is the Gamma function. It is also known that[8]

$$\mathcal{L}^{-1}\left\{\frac{e^{-k\sqrt{s}}}{s^{\frac{3}{2}}}\right\} = 2\sqrt{\frac{t}{\pi}}e^{-\frac{k^2}{4t}} - k\left(\frac{k}{2\sqrt{t}}\right), \tag{4.48}$$

where k is a positive constant and (x) is the complementary error function. the asymptotic relation

$$x = \frac{e^{-x^2}}{x\sqrt{\pi}}\left(1 - o(x^{-1})\right)$$

holds when x is large. Thus when $t \to 0$, we have

$$\mathcal{L}^{-1}\left\{\frac{e^{-k\sqrt{s}}}{s^{\frac{3}{2}}}\right\}(t) = o(t)e^{-\frac{k^2}{4t}}$$

[7]Choose $y(z) = \cos(z/2)$ and $G(z) = y'(z)/y(z)$

[8]Formula 29.3.85 of Abramowitz and Stegan, 1964.

Therefore,

$$M(t) = 1 - 4\sqrt{\frac{t}{\pi}} + o(t)e^{-\frac{k^2}{4t}}. \tag{4.49}$$

Combining with $\frac{1}{S_0(t)} = 1 + o(t)$ (a very conservative estimate), we obtain the short-time asymptotic,

$$\boxed{R(t) = 4\sqrt{\frac{t}{\pi}} + o(t).} \tag{4.50}$$

The square root of t dependency in the short-time limit of $R(t)$ agrees with the rms displacement of Brownian motions: Let $\boldsymbol{r} = (X, Y, Z)$ be a random vectors whose components are independent Gaussian random variables with mean zero and variance $\sigma^2 = 2t$. $\boldsymbol{r}$ then has pdf

$$\frac{1}{(2\pi)^{\frac{3}{2}}\sigma^3} \exp\left(-\frac{x^2 + y^2 + z^2}{2\sigma^2}\right) = \frac{1}{(2\pi)^{\frac{3}{2}}\sigma^3} \exp -\frac{r^2}{2\sigma^2}. \tag{4.51}$$

Multiplying the pdf by $4\pi r^2$ yields the marginal distribution of the length r,

$$f(r) = \sqrt{\frac{2}{\pi}} \frac{r^2}{\sigma^3} e^{-\frac{r^2}{2\sigma^2}}, \tag{4.52}$$

whose mean is given by

$$\mathbb{E}\{r\} = \int_0^\infty \sqrt{\frac{2}{\pi}} \frac{r^3}{\sigma^3} e^{-\frac{r^2}{2\sigma^2}} dr = 2\sigma\sqrt{\frac{2}{\pi}} = 4\sqrt{\frac{t}{\pi}}. \tag{4.53}$$

The Gaussian integral formula,

$$\int_0^\infty x^{2n+1} e^{-ax^2} dx = \frac{n!}{2a^{n+1}},$$

was invoked in the calculation above.

4.4 Phase Update

Let us now return to the case where a general PGSE sequence is applied (δ no longer small).
To simulate the signal, it is natural to partition the diffusion encoding duration $[0, \Delta + \delta]$ into intervals where the time profile are constants, namely $[0, \delta), [\delta, \Delta), [\Delta, \Delta + \delta]$. In each interval, time and position updates are identical to Algorithm 4.1. For intervals with non-zero time profile however, the GAFRW algorithm now needs to include phase updates.

Contrary to the basic Monte Carlo method, phases can no longer be calculated via rectangular/trapezoidal rules as the time steps can potentially be large. The phase increments are instead nonlinear functions of the time step, which are derived by using the matrix formalism in [14] and will not be reproduced here. The original derivation is terse so we provide supplementary explanation for the mathematics involved in Appendix B and C. Further extension the GAFRW algorithm requires a solid understanding of the derivation.

As usual, the phase increment $\Delta\varphi$ are calculated separately for non-boundary and boundary cases.

4.4.1 Non-boundary Case

Non-final updates Let $\{\boldsymbol{X}(t')\}_{t'=0}^{t}$ be the underlying position process during a non-final update in Scheme 1, which by design satisfies

Cond.1 initial condition $\boldsymbol{X}(0) = \boldsymbol{r}$;

Cond.2 final condition $\boldsymbol{X}(t) = \boldsymbol{r}_{\text{new}}$;

Cond.3 $\|\boldsymbol{X}(t') - \boldsymbol{r}\| < \rho$, $0 \leqslant t' < t$.

The random phase increment is therefore given by the time integral,

$$\Delta\tilde{\varphi} = \int_0^t dt' \boldsymbol{X}(t') \cdot \boldsymbol{g}, \tag{4.54}$$

where $\boldsymbol{g}$ is the diffusion gradient times the gyromagnetic ration γ.

The exact distribution for the phase increment during each update is analytically intractable. The main idea is to use moments under given conditions as approximations[9]. Although the moment calculations are still challenging, it is a considerable simplification. The integral 4.54 is approximated by its first moment given conditions **Cond.1** to **Cond.3**,

$$\boxed{\Delta\varphi = t\boldsymbol{r}\cdot\boldsymbol{g} + \frac{\rho^2}{D}\Phi_1\left(\frac{Dt}{\rho^2}\right)(\boldsymbol{r}_{\text{new}} - \boldsymbol{r})\cdot\boldsymbol{g},} \tag{4.55}$$

where the function $\Phi_1(\tau)$ is the normalized first moment [10], given by

$$\Phi_1(\tau) = \frac{1}{-2S_0'(\tau)}\left(\frac{3}{2}S_0(\tau) - \sum_{k=0}^{\infty}(-1)^k\sqrt{\lambda_{1k}+1}e^{-\lambda_{1k}\tau}\right), \tag{4.56}$$

where the set $\{\lambda_{nk}\}_{n,k\geqslant 0}$ is the Dirichlet spectrum of the Laplacian associated with the unit sphere $\mathbb{B}$.

It is well-known that the eigenvalues λ_{nk} are squares of the positive zeros of spherical Bessel function of the first kind,

$$\lambda_{nk} = \alpha_{nk}^2, \quad j_n(\alpha_{nk}) = 0, \tag{4.57}$$

where for a fixed n, α_{nk}'s increase w.r.t. k. In particular, $\alpha_{0k} = (k+1)\pi$, $k \geqslant 0$

Same as functions given by infinite series before, values of $\Phi_1(t)$ are tabulated once and forever and are loaded prior to the Monte Carlo simulation to save computation time. The graph of $\Phi_1(t)$ is shown in Figure 4.11.

Remarks regarding formula 4.55 :

- The second term can be thought of the adjustment to the rectangular rule used in basic Monte Carlo methods, given by the first term;
- Two orthogonal components of $\{\boldsymbol{X}(t)\}$ contribute independently to the phase increment. In other words,
$$\boldsymbol{X}(t) = X_{\parallel}(t)\hat{\boldsymbol{g}} + \boldsymbol{X}_{\perp}(t), \tag{4.58}$$
where $\{\boldsymbol{X}_{\perp}(t)\}$ is orthogonal to the gradient direction $\hat{\boldsymbol{g}}$ (Figure 4.10).

[9] This is a common approach in statistical inference known as the 'method of moments'.
[10] 'normalized' means that the pdf used for calculating the moment is normalized

1. Due to the initial condition $\boldsymbol{X}(0) = \boldsymbol{r}$, $\{\boldsymbol{X}_\perp(t)\}$ stays in the plane,

$$\{\boldsymbol{r'} : \ (\boldsymbol{r'} - \boldsymbol{r}) \cdot \boldsymbol{g} = 0\}, \tag{4.59}$$

   on which the gradient field is of constant strength $\boldsymbol{r} \cdot \boldsymbol{g}$. Therefore, the phase contribution due to the component $\{\boldsymbol{X}_\perp(t)\}$ is simply $t\boldsymbol{r} \cdot \boldsymbol{g}$.

2. On the other hand, it is unsurprising that the contribution due to $X_\parallel(t)\hat{\boldsymbol{g}}$ is proportional to the projection of the displacement vector onto the gradient vector, which is the second, nonlinear term.

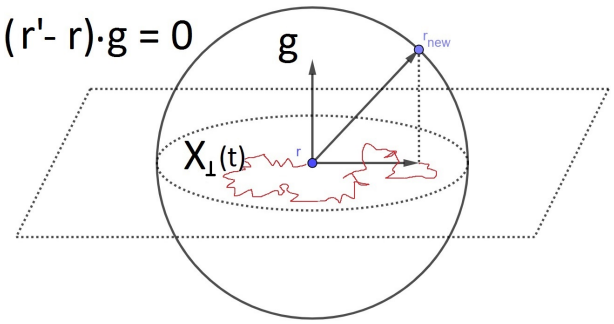


Figure 4.10: The component of $\boldsymbol{X}(t)$ orthogonal to the gradient direction.

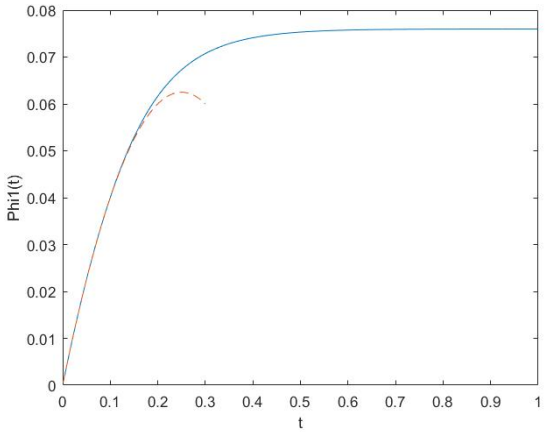


Figure 4.11: Graph of $\Phi_1(t)$ along with its short-time approximation $\Phi_1(t) \approx \frac{1}{2}t(1-2t)$.

Final update If $\{\boldsymbol{X}(t')\}_{t'=0}^{t}$ is the underlying position process during the final update. According to our analysis in subsection 4.2.1, $\{\boldsymbol{X}(t')\}_{t=0}^{t}$ only satisfies **Cond.1** and **Cond.3** (with a modification $0 \leqslant t' \leqslant t$).

The phase increment defined by integral 4.54 has zero conditional mean since $\mathbb{E}\{\boldsymbol{X}(t)\} = 0$ for all t due to radial symmetry. A non-trivial estimate of the phase increment is given via Gaussian phase approximation. More specifically, $\Delta\varphi$ is treated as a Gaussian random variable with zero mean and variance of to the conditional second moment of integral 4.54[11].

[11]The Gaussian phase approximation is valid when the gradient strength is not particularly high. See [11] for a detailed discussion on Gaussian phase approximation.

The final phase increment is given by

$$\boxed{\Delta\varphi = t\boldsymbol{r}\cdot\boldsymbol{g} + \tilde{Z}\times\frac{\rho^2}{D}\Phi_2\left(\frac{Dt}{\rho^2}\right)\boldsymbol{g}\cdot(1,1,1)^T,} \tag{4.60}$$

where T stands for matrix transpose and $\tilde{Z}$ is a standard Gaussian random variable. The normalized standard deviation of the dephase, $\Phi_2(\tau)$ (Figure 4.12), is defined as ,

$$\Phi_2(\tau) = \sqrt{\frac{S_2(\tau)}{S_0(\tau)}}, \tag{4.61}$$

where $S_2(t)$ is the pre-normalized second moment,

$$\begin{aligned} S_2(\tau) =& \frac{1}{24}\sum_{k=0}^{\infty}(-1)^k(-\lambda_{0k}^{-1} + 17\lambda_{0k}^{-2} - 174\lambda_{0k}^{-3})e^{-\lambda_{0k}\tau} + \frac{5}{12}\sum_{k=0}^{\infty}\frac{(-1)^k\sqrt{\lambda_{1k}+1}}{\lambda_{1k}^2}e^{-\lambda_{1k}\tau} \\ &+ \frac{\tau}{12}\sum_{k=0}^{\infty}(-1)^k(4\lambda_{0k}^{-1} + 3\lambda_{0k}^{-2})e^{-\lambda_{0k}\tau}. \end{aligned} \tag{4.62}$$

Notice that in the last phase increment, contrary to formula 4.55, the nonlinear component is independent of the final position. This is because we assume that the final position $\boldsymbol{r}_{\text{fin}} = \boldsymbol{X}(t)$ is unspecified and is therefore integrated over the sphere $B_\rho(\boldsymbol{r})$ in the derivation of the second moment (Appendix B).

4.4.2 Boundary Case

As the updated position in the boundary case is on a hemisphere of prescribed radius, a rigorous consideration for the phase update analogous to the non-boundary case would consists of the following steps. First, find the Laplacian eigenvalues and eigenfunctions of the hemispherical domain subject to mixed boundary conditions, which are Dirichlet BC on the hemisphere and Neumann BC on its base. Second, compute matrix elements of $\mathcal{B}, U, \tilde{U}$ in subsection 2.8.3. Lastly, impose initial and final conditions on the position process and calculate the first conditional moment. However, this approach turns out to be unwieldy cumbersome.

Once again, we look at the Reflection Principle for inspiration. Let $\{\boldsymbol{X}(t')\}_{t'=0}^{t}$ be the underlying position process during updates of boundary case, excluding the reflection. In other words, $\{\boldsymbol{X}(t')\}_{t'=0}^{t}$ is simply the free diffusion inside S_n, satisfying **Cond.1** to **Cond.3**.

If we are to use $\boldsymbol{X}(t)$ to compute the equivalent phase increment due the reflected position process, the spatial profile of the effective magnetic field inside S_n needs to be symmetric w.r.t. the base [12]. This effective (or 'reflected') spatial profile is therefore no longer linear and its piecewise form is given by

$$B_{\text{eff}}(\boldsymbol{r'}) = \begin{cases} \boldsymbol{r'} \cdot \boldsymbol{g} & \text{if } (\boldsymbol{r'} - \boldsymbol{r}) \cdot \hat{\boldsymbol{n}} > 0; \\ (\boldsymbol{r'} - 2((\boldsymbol{r'} - \boldsymbol{r}) \cdot \hat{\boldsymbol{n}}))\hat{\boldsymbol{n}}) \cdot \boldsymbol{g} & \text{if } (\boldsymbol{r'} - \boldsymbol{r}) \cdot \hat{\boldsymbol{n}} \leqslant 0, \end{cases} \tag{4.63}$$

where $\hat{\boldsymbol{n}}$ is the interior unit normal vector of Ω, evaluated near $\boldsymbol{r}$.

Orthogonal decomposition of $\boldsymbol{g}$ The gradient $\boldsymbol{g}$ can be decomposed to directions that are orthogonal or tangent to the base,

$$\boldsymbol{g} = \boldsymbol{g}_{\perp} + \boldsymbol{g}_{\parallel}, \tag{4.64}$$

where $\boldsymbol{g}_{\perp} \equiv (\boldsymbol{g} \cdot \hat{\boldsymbol{n}})\hat{\boldsymbol{n}}$, $\boldsymbol{g}_{\parallel} \equiv \boldsymbol{g} - (\boldsymbol{g} \cdot \hat{\boldsymbol{n}})\hat{\boldsymbol{n}}$[13]. With some algebra, equation 4.63 can be rewritten in terms of $\boldsymbol{g}_{\perp}$ and $\boldsymbol{g}_{\parallel}$,

$$B_{\text{eff}}(\boldsymbol{r}) = \underbrace{(\boldsymbol{r}_{n-1} \cdot \boldsymbol{g}_{\perp} + \boldsymbol{r} \cdot \boldsymbol{g}_{\parallel})}_{\text{linear gradient}} + \underbrace{|(\boldsymbol{r} - \boldsymbol{r}_{n-1}) \cdot \hat{\boldsymbol{n}}|(\boldsymbol{g} \cdot \hat{\boldsymbol{n}})}_{\text{'V-shaped' gradient}}, \quad \boldsymbol{r} \in S_n. \tag{4.65}$$

[12] Effective magnetic field strength at positions symmetric about the base must be the same.

[13] Treat the subscripts $\parallel$ and $\perp$ as operators. It is easy to verify that they are both Hermitian operators, i.e. for $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^3$, $\boldsymbol{a}_{\perp} \cdot \boldsymbol{b} = \boldsymbol{a} \cdot \boldsymbol{b}_{\perp}$, $\boldsymbol{a}_{\parallel} \cdot \boldsymbol{b} = \boldsymbol{a} \cdot \boldsymbol{b}_{\parallel}$.

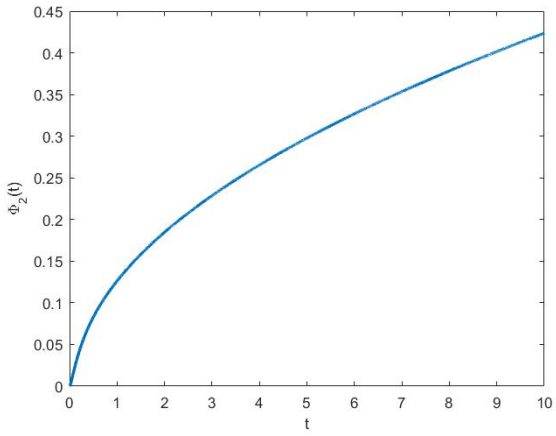


Figure 4.12: The graph of the standard deviation function $\Phi_2(t)$.

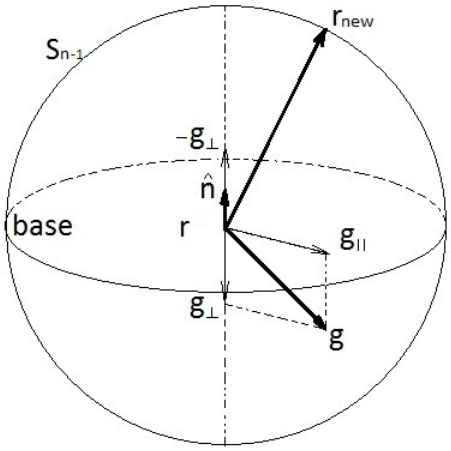


Figure 4.13: B_{eff} at a Neumann boundary: a superposition of a linear gradient field ($\boldsymbol{g}_\parallel$) and a 'V-shaped' field ($\pm\boldsymbol{g}_\perp$ on either side of the base).

This representation suggests that $B_{\text{eff}}(\boldsymbol{r})$ is equivalent to the superposition of two fields (Figure 4.13). They are:

1. a linear field with gradient $\boldsymbol{g}_\parallel$;
2. a 'V-shaped' field [14] with opposite gradients, $\pm\boldsymbol{g}_\perp$ on either side of the base. Its strength varies linearly on the distance towards the base.

Since the dephase is a linear functional of the magnetic field, the total dephase be written as the sum of the contributions from the two aforementioned fields. Let us look into the phase contribution from each field in detail.

1. The contribution from the linear field is estimated by its first conditional moment given the final condition $\boldsymbol{X}(t) = \boldsymbol{r}_{\text{new}}$. This is identical to the cased covered by formula 4.55 under the gradient $\boldsymbol{g}_\parallel$,

$$\Delta\varphi_\parallel = t\boldsymbol{r}\cdot\boldsymbol{g}_\parallel + \frac{R_{\text{rls}}^2}{D}\Phi_1\left(\frac{D}{R_{\text{rls}}^2}\right)(\boldsymbol{r}_{\text{new}} - \boldsymbol{r})\cdot\boldsymbol{g}_\parallel. \tag{4.66}$$

[14] due to the shape of the absolute value function

2. The contribution from the 'V-shaped' field is estimated by its first conditional moment under a weaker final condition: $\boldsymbol{X}(t) \in \partial S_n$. The accuracy should not suffer 'too much' as we do not expect a large number of iterations are of the boundary case. Therefore $\boldsymbol{r}_{\text{new}}$ is integrated over S_n in the derivation. Under the weakened final condition, the conditional mean can be greatly simplified and is in fact a linear function (!) of t,

$$\Delta\varphi_\perp = t\boldsymbol{r}\cdot\boldsymbol{g}_\perp + \frac{1}{4}R_{\text{rls}}t(\hat{\boldsymbol{n}}\cdot\boldsymbol{g}_\perp). \tag{4.67}$$

Finally, combining the two contributions, we obtain the total phase increment during the reflection step:

$$\boxed{\Delta\varphi = t\boldsymbol{r}\cdot\boldsymbol{g} + \frac{1}{4}R_{\text{rls}}t\hat{\boldsymbol{n}}\cdot\boldsymbol{g} + \frac{R_{\text{rls}}^2}{D}\Phi_1\left(\frac{Dt}{R^2}\right)(\boldsymbol{r}_{\text{new}}-\boldsymbol{r})_\parallel\cdot\boldsymbol{g}.} \tag{4.68}$$

Remark It is worth noticing that $\boldsymbol{g}$ can be factored out in all formulae for the dephase increments. This means that we can perform all dot product operations with $\boldsymbol{g}$ after our simulation. This can be advantageous in certain situations. For example, to simulated a q-space image, DW signal needs to be calculated for a large number of $\boldsymbol{q}$ vectors to recover the EAP. In Monte Carlo simulations the signal calculation is trivial once the sequence of position and time pairs are obtained.[15] The solutions of BT equation on the other hand need to be recalculated for each gradient direction since the PDE changes.

4.4.3 Summary: full GAFRW algorithm

We conclude this section with a summary combining all cases. Suppose that a constant diffusion encoding gradient $\boldsymbol{g}$ is present during a time interval $[0, T]$ and the initial position of a walker is $\boldsymbol{r}_0$.

[15]Given a fixed geometry, basic Monte Carlo method generates a large sample of detailed trajectories. From that the signal can be obtained for varying spatial and temporal profile (for example in a PGSE squence, changing while keeping $\Delta+\delta$ fixed). For the GAFRW algorithm however, for a new time profile, a brand new simulation is needed.

Algorithm 4.3 Inputs $T, \boldsymbol{g}, \boldsymbol{r}_{\text{ini}}, \varphi_{\text{ini}}$; outputs: $\boldsymbol{r}_{\text{fin}}, \varphi_{\text{fin}}$.

S.1 Set $n = 0$, $T_n = 0$, $\boldsymbol{r}_n = \boldsymbol{r}_{\text{ini}}$, $\varphi_n = \varphi_{\text{ini}}$;

S.2 While $T_n \leqslant T$, do

(a) If $\boldsymbol{r}_n$ is far from $\partial\Omega$, execute Scheme 1 and compute 4.55 or 4.60;

(b) If $\boldsymbol{r}_n$ is close to $\partial\Omega$, execute Scheme 2 and compute 4.68;

S.3 return $\boldsymbol{r}_{\text{fin}} = \boldsymbol{r}_n$, $\varphi_{\text{fin}} = \varphi_n$.

end

Under PGSE sequences, the steps for simulating diffusion signals summarized given below.

Algorithm 4.4 Inputs: N_{walker}, $\delta/\Delta/\boldsymbol{g}$ (sequence parameters); output: S

S.1 Generate a uniformly random point in Ω, $\boldsymbol{r_0}$;

S.2 Apply Algorithm 4.3 with inputs δ, $\boldsymbol{g}$, $\boldsymbol{r_0}$, 0, obtain outputs $\boldsymbol{r}(\delta)$, φ_+;

S.3 Apply Algorithm 4.1 with inputs $(\Delta - \delta)$, $\boldsymbol{r}(\delta)$, obtain outputs $\boldsymbol{r}(\Delta)$;

S.4 Apply Algorithm 4.2 with inputs δ, $-\boldsymbol{g}$, $\boldsymbol{r}(\Delta)$, φ_+, obtain outputs $\boldsymbol{r}(\delta + \Delta), \varphi$;

S.5 Repeat S.1 to S.4 N_{walker} times, obtain a sequence $\{\varphi_k\}$.

S.6 Compute the diffusion signal,

$$S = \overline{\exp(i\varphi_k)}. \tag{4.69}$$

end

4.5 Numerical Evaluation

Numerical evaluations of the functions defined in terms of infinite series are calculated mainly based on finite truncation. Whenever such basic approach proves to be numerically unstable or there exists convenient approximations, asymptotic formulae are derived instead.

The input range for tabulation $[t_{\min}, t_{\max}]$ is chosen to be sufficiently wide so that the probability of t falling outside of such range is negligible. For untabulated inputs, the function outputs are taken as the interpolation of their neighboring tabulated values.

4.5.1 $S(t)$

Both series representations of $S_0(t)$, series 4.38 and 4.33 are used for evaluating $S_0(t)$.

Let $S^N_{\text{long}}(t)$ be the partial sum of series 4.33,

$$S^N_{\text{long}}(t) = 2\sum_{n=1}^{N}(-1)^{n+1}e^{-n^2\pi^2 t}. \tag{4.70}$$

According to the alternating series estimation theorem, an upper bound for the truncation error is simply the absolute value of $(N+1)$-th term,

$$|S_0(t) - S^N_{\text{long}}(t)| \leqslant 2e^{-(N+1)^2\pi^2 t}. \tag{4.71}$$

On the other hand, the truncation error of using the partial sum of the short-time representation, 4.38, is given by

$$|S_0(t) - S^N_{\text{short}}| \leqslant \frac{2e^{-\frac{2N+2}{t}}}{\sqrt{\pi t}(1-e^{-\frac{2}{t}})}, \tag{4.72}$$

where

$$S^N_{\text{short}}(t) = 1 - \frac{2}{\sqrt{\pi t}}\sum_{n=0}^{N} e^{-\frac{(2n+1)^2}{4t}}. \tag{4.73}$$

To obtain the upper bound in 4.72, we use the inequality $(2n+1)^2 \geqslant 8n$ and geometric series:

$$|S_0(t) - S^N_{\text{short}}(t)| = \frac{2}{\sqrt{\pi t}}\sum_{n=N+1}^{\infty} e^{-\frac{(2n+1)^2}{4t}} \leqslant \frac{2}{\sqrt{\pi t}}\sum_{n=N+1}^{\infty} e^{-\frac{2n}{t}} \leqslant \frac{2e^{-\frac{2N+2}{t}}}{\sqrt{\pi t}(1-e^{-\frac{2}{t}})}.$$

For $N = 0$ specifically, we have

$$|S_0(t) - S^0_{\text{short}}(t)| \leqslant \frac{2e^{-\frac{2}{t}}}{\sqrt{\pi t}(1 - e^{-\frac{2}{t}})}.$$

Numerical calculation shows that for $t \leqslant 0.1155$, the approximation $S^0_{\text{short}}(t) = 1 - \frac{2}{\sqrt{\pi t}} e^{-\frac{1}{4t}}$ has error less than 10^{-7}. For $t > 0.1155$, $S^N_{\text{long}}(t)$ is used for $N \leqslant 3$.

4.5.2 F_τ^{-1}

Once $S_0(t)$ is tabulated within a desired accuracy, we can subsequently compute $F_\tau(t) = 1 - S_0(t)$ and its inverse. When U is close to 0 or 1, F_τ has vanishing derivatives. To avoid numerical division by zero, closed-form asymptotics of

$$U = F_\tau(t) \tag{4.74}$$

are used for the short- and long-time regimes.

When U is small, the short-time limit of 4.74 is the transcendental equation,

$$U = \frac{2e^{-\frac{1}{4t}}}{\sqrt{\pi t}}, \tag{4.75}$$

which can be rearranged to

$$f(t) = t \ln(t) + A = 0, \tag{4.76}$$

where $A = 2 \ln \frac{U\sqrt{\pi}}{2} + \frac{1}{2}$. Applying Newton's method with initial guess $t = 1$, we have the scheme,

$$\begin{aligned} & t_0 = 1; \\ \textbf{do} \quad & t_{n+1} = t_n - \frac{t_n \ln t_n + A}{\ln t_n + 1}, \quad n \geqslant 0, \\ \textbf{until} \quad & \epsilon = 10^{-4} > |t_{n+1} - t_n|. \end{aligned} \tag{4.77}$$

On the other hand, when U is close to 1, the equation 4.74 is reduced to its long-time limit,

$$U = 1 - 2e^{-\pi^2 t}. \tag{4.78}$$

It can be solved algebraically and

$$t = -\frac{1}{\pi^2} \ln\left(\frac{1 - U}{2}\right). \tag{4.79}$$

For intermediate values of U, $F_\tau^{-1}(U)$ are evaluated by a simple lookup table.

4.5.3 $R(t)$

The tabulation of $R(t)$ (4.6) is only required for intermediate values of t. Since we have computed the values of $S_0(t)$ only the values of $M(t)$ (4.7) are to be tabulated.

The partial sum of 4.7 and its truncation error of are given by

$$M_N(t) = \sum_{l=1}^{N} \frac{e^{-(2l-1)^2\pi^2 t}}{(2l-1)^2\pi^2}, \quad M(t) - M_N(t) = \sum_{l=N+1}^{\infty} \frac{e^{-(2l-1)^2\pi^2 t}}{(2l-1)^2\pi^2}, \tag{4.80}$$

respectively. The truncation error for $N \geqslant 2$ has upper bound,

$$|M(t) - M_N(t)| \leqslant \min\left\{ \frac{1}{2\pi^2(2N-1)}, \frac{e^{-(2N+1)^2\pi^2 t}}{(2N+1)^2\pi^2(1-e^{-4(2N+1)\pi^2 t})} \right\}. \tag{4.81}$$

To show this we notice that, on one hand, by the integral test for convergence,

$$|M(t) - M_N(t)| \leqslant \sum_{l=N+1}^{\infty} \frac{1}{(2l-1)^2\pi^2} \leqslant \int_N^{\infty} \frac{1}{(2x-1)^2\pi^2} dx = \frac{1}{2\pi^2(2N-1)}.$$

On the other hand, by geometric series,

$$\begin{aligned} |M(t) - M_N(t)| &\leqslant \frac{1}{(2N+1)^2\pi^2} \sum_{l=N+1}^{\infty} e^{-(2l-1)^2\pi^2 t} = \frac{1}{(2N+1)^2\pi^2} \sum_{n=0}^{\infty} e^{-(2n+2N+1)^2\pi^2 t} \\ &\leqslant \frac{e^{-(2N+1)^2\pi^2 t}}{(2N+1)^2\pi^2} \sum_{n=0}^{\infty} e^{-4(2N+1)n\pi^2 t} = \frac{e^{-(2N+1)^2\pi^2 t}}{(2N+1)^2\pi^2(1-e^{-4(2N+1)\pi^2 t})}. \end{aligned}$$

We used the inequality,

$$(2n+2N+1)^2 - (2N+1)^2 = 4(2N+1)n + 4n^2 \geqslant 4(2N+1)n.$$

Hence the error bound is established. In our computation we always keep at least the first two terms in the partial sum.

4.5.4 F_R^{-1}

When solving the transcendental equation 4.9, special treatments are required when U is close to 0 or 1 once again due to vanishing derivatives. We employ appropriate Taylor

expansions to avoid this issue.

Since $\sin(x) - x\cos(x) = \frac{1}{3}x^3 + o(x^3)$, for small U, equation 4.9 approximately reduces to the simple equation, $U = \frac{\pi^2}{3}r^3$, which has the root,

$$r = \left(\frac{3U}{\pi^2}\right)^{\frac{1}{3}}. \tag{4.82}$$

In addition $\sin(x) - x\cos(x)$ admits the expansion at $x = \pi$,

$$\sin(x) - x\cos(x) = \pi - \frac{1}{2}\pi(x-\pi)^2.$$

Hence, for U close to 1, equation 4.9 approximately reduces to

$$U = 1 - \frac{1}{2}\pi^2(1-r)^2,$$

which has the root,

$$r = 1 - \sqrt{\frac{2(1-U)}{\pi}}. \tag{4.83}$$

For intermediate values of U, we applying Newton's method with initial guess 0.5:

$$\begin{aligned} & x_0 = 0.5; \\ \textbf{do} \quad & x_{n+1} = x_n - \frac{\sin x_n - x_n \cos x_n - \pi U}{x_n \sin(x_n)},\ n \geqslant 0 \\ \textbf{until} \quad & \epsilon > |x_{n+1} - x_n|, \\ \textbf{return} \quad & r = \frac{x_n}{\pi}. \end{aligned} \tag{4.84}$$

4.5.5 $\{\alpha_{1k}\}$

The MATLAB function for finding approximate roots of $j_1(z)$, written by Greg von Winckel, is available online[16]. The function uses simple equations to approximate the first three zeros. The forth and higher zeros are obtained from Halley's method.

[16] https://www.mathworks.com/matlabcentral/fileexchange/6794-bessel-function-zeros

4.5.6 $\Phi_1(t)$

When tabulating $\Phi_1(t)$ based on equation 4.56, the upper bound of truncation error is hard to estimate due to the fact it is a ratio of two infinite series. Moreover, equation 4.56 performs poorly for small t's sincere both the numerator and denominator are extremely small so the division is numerically unstable[17].

Therefore, for small t[18], we employ the short-time asymptotic,

$$\boxed{\Phi_1(t) \approx \frac{1}{2}t(1-2t),} \tag{4.85}$$

as $t \to 0$. The quadratic asymptotic time-dependency makes sense heuristically. To see this, given that the Brownian particle has to exit the unit sphere in a short time t, its trajectory (continuous with probability 1) should almost be a straight line segment connecting the initial and exit points. As a result, the accumulated phase is an integral of a linear function of time ($B(\boldsymbol{r}(t))$ is linear since both $B(\boldsymbol{r})$ and $\boldsymbol{r}(t)$ are linear), which has to be quadratic in t.

Now we derive 4.85 by again using Laplace transform. For the denominator in Φ_1, termwise differentiating the series 4.38 yields

$$2S_0'(t) = \frac{1}{\sqrt{\pi t^5}}\sum_{k=0}^{\infty} e^{-\frac{(2k+1)^2}{4t}}(2t-(2k+1)^2) \approx \frac{1}{\sqrt{\pi t^5}}e^{-\frac{1}{4t}}(2t-1). \tag{4.86}$$

On the other hand, denote the numerator as

$$S_1(t) = \frac{3}{2}S_0(t) - I(t), \tag{4.87}$$

where

$$I(t) = \sum_{k=0}^{\infty}(-1)^k\sqrt{\alpha_{1k}^2+1}e^{-\alpha_{0k}^2 t}. \tag{4.88}$$

Taking the Laplace transform and applying the residume summation technique, we have

$$\mathcal{L}\{S_1\}(s) = \frac{\sqrt{s}}{2(\sqrt{s}\cosh(\sqrt{s}) - \sinh(\sqrt{s}))} - \frac{3}{2\sqrt{s}\sinh(\sqrt{s})}. \tag{4.89}$$

[17] As one can observe from Figure 4.7, the denominator $S_0'(t)$ (the slope of $S_0(t)$) is almost zero as t gets close to 0.

[18] In our calculation, the threshold is taken where $S_0'(t)$ is lower than 10^{-10}.

Since $\sinh(x) \approx \cosh(x) \approx \frac{e^x}{2}$ as $x \to \infty$, for large s, we have the approximation,

$$\mathcal{L}\{S_1\}(s) \approx e^{-\sqrt{s}} \frac{1}{1-\frac{1}{\sqrt{s}}} - 3e^{-\sqrt{s}} = e^{-\sqrt{s}}(1+\frac{1}{\sqrt{s}}+o(\frac{1}{\sqrt{s}})) - 3\frac{e^{-\sqrt{s}}}{\sqrt{s}} = e^{-\sqrt{s}}(1-\frac{2}{\sqrt{s}}+o(\frac{1}{\sqrt{s}}))). \tag{4.90}$$

Thus as $t \to 0$, an approximation for $S_1(t)$ is

$$S_1(t) = \frac{e^{-\frac{1}{4t}}}{2\sqrt{\pi}t^{\frac{3}{2}}}(1-4t+o(t)). \tag{4.91}$$

The desired short-time asymptotic is obtained by taking the ratio of equation 4.91 and 4.86,

$$\begin{aligned} \Phi_1(t) = \frac{S_1(t)}{-2S_0'(t)} &\approx \frac{t(1-4t+o(t))}{2(1-2t)} \\ &= \frac{1}{2}t(1-4t+o(t))(1+2t+o(t)) \\ &= \frac{1}{2}t(1-2t+o(t)), \end{aligned} \tag{4.92}$$

where geometric series is used in the second last step. On the other hand, Simply keeping only leading exponential terms in equations 4.33 and 4.88 yields the long-time approximation,

$$\Phi_1(t) = \frac{3}{4\pi^2} - \frac{\sqrt{\alpha_{10}^2+1}}{4\pi^2} e^{-(\alpha_{10}^2-\pi^2)t}. \tag{4.93}$$

4.5.7 $\Phi_2(t)$

We only need to compute $S_2(t)$ as the $S_0(t)$ is tabulated earlier. Although all three series in the expression of $S_2(t)$ look complicated, their evaluation is actually a simple matter due to their fast convergence speed (even for small values of t). Since they are all alternating series, an upper bound of the truncation error is easy to obtain and is omitted here.

4.6 Experiment: orthogonal fiber bundles

In applications, one voxel may contain more than one bundle of fibers (Figure 4.14). In diffusion MRI reconstruction and tractography, this presents a fundamental challenge as currently there exists no method capable of mapping multiple neural paths consistently. We would like to simulate diffusion encoding processes in such environment by applying the GAFAW algorithm. Let us introduce the infinite cylinder model established in section 3.3.3 with an additional bundle of orthogonally placed fibers.

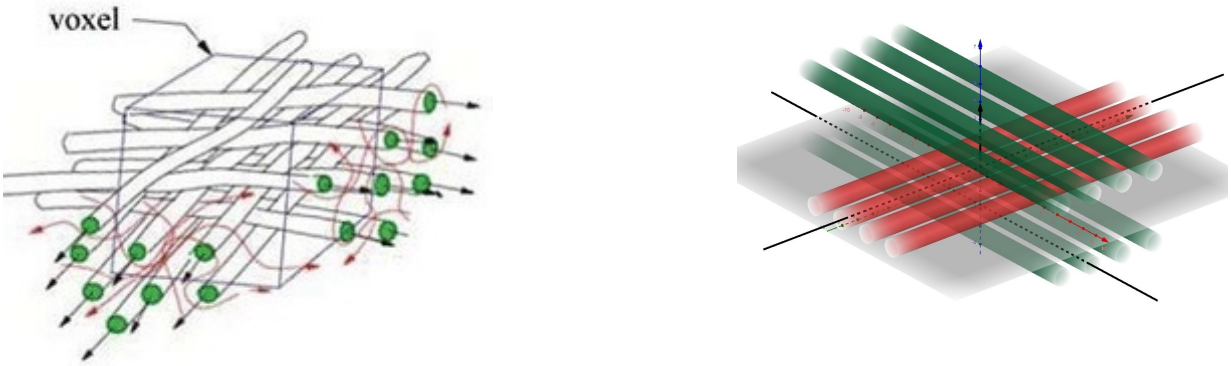


Figure 4.14: Crossing fibers: real microstructure and its geometric idealization.

Here, we assume that for each group of fibers, the horizontal separation is $2d$. The vertical separation of horizontal 'layers' of fibers is also $2d$. The fibers regardless of orientations all have inner and outer radius of $R_{\text{in}}, R_{\text{out}}$ respectively. Recognizing the periodicity of the structure, we formulate the simulation problem in a cube-shaped minimal lattice (Figure 4.15). Let us place the center of the lattice at the origin. Inside the lattice, we have four (quartered) finite cylindrical surfaces. The axes of the quartered cylinders coincide with four of the lattice sides. The positions of all axes can be easily read from Figure 4.15. For example, axis x_1 has vertices $d(1,-1,-1), d(-1,-1,-1)$.

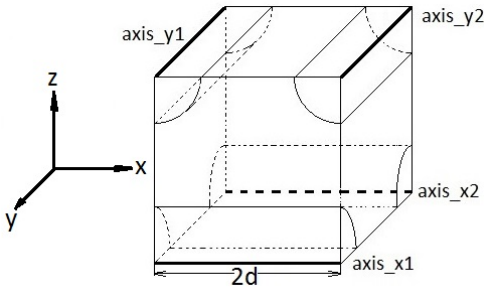


Figure 4.15: Lattice for orthogonal fibers (thickness not shown).

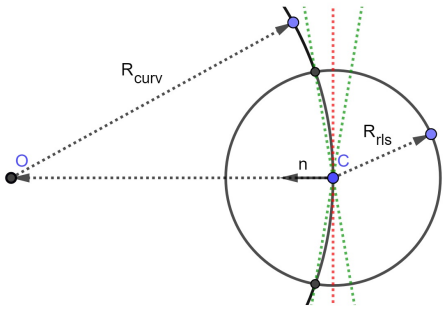


Figure 4.16: Exclude jump directions falling between the two green dotted lines.

4.6.1 Implementation and Results

Assuming Neumann boundary condition on the cylinders, the whole simulation procedure can be divided into intra-axonal and extra-axonal cases, two independent procedures. Therefore, we define a geometry class for each domain equipped with its own functions for calculating distance, surface normal vector and uniformly-sampled initial positions.

Due to translation and rotational symmetry, intra-axonal diffusion can be simulated within a single infinite cylinder with a default orientation (such as z-axis). The computed dephase sample is then transformed according to the Rodrigues' rotation formula described in section 3.3.3. The MATLAB implementation for the extra-axonal domain can be found in section F.6.1. Notice that, as before, the structural periodicity is handled compactly by using the floor function in the distance calculation.

Position initialization is performed as following: We first specify N_{walker}, the total number of spins. The number of intra-axonal walkers, N_{in} is propertional to the volume fraction, $VF = \frac{\pi R_{\text{out}}^2}{4d^2}$. The extra-axonal walkers have a population size of $(N_{\text{walker}} - N_{\text{in}})$ and are sampled by using the acceptance-rejection (A-R) method[19]. Lastly, N_{in} initial positions are generated inside a $z-$oriented infinite cylinder with radius R_{in}.

The MATLAB function for the full fast random walk algorithm is shown in section F.6. We make one extra comment on the treatment for the boundary case. For boundary points

[19] A-R methods can be inefficient when VF is large due to large number of rejected samples. However, as of now, a more efficient scheme does not seem available.

whose associated local curvature vector pointing towards the interior of the confining domain, we want to avoid jumps whose direction are 'almost tangent' to the boundary. As an example, consider Figure 4.16. Starting from a boundary point C, the position update, as described earlier, should be a point on the release sphere with prescribed radius R_{rls} (reflected when necessary). Notice that if the jump vector falls between the two green dotted lines, even after reflection, the new position would still be outside of the domain. Therefore, the aforementioned jumps are prohibited in the simulation. Since the angle is typically small, it is reasonable to assume that no significant bias is caused.

In our experiment, geometric parameters are set as $(d, R_{\text{out}}, R_{\text{in}}) = (2, 1.7, 1.6)\mu$m. For both intra- and extra-axonal domains, their boundary layer thickness and release radius are defined as $(\varepsilon, R_{rls}) = R(10^{-4}, 0.05)$, where $R = R_{\text{in}}, R_{\text{out}}$ respectively. Sequence parameters, except for directions, are the same as the ones in Chapter 3. The size of the simulation N_{walker} equals to 1000 and $(N_{\text{in}}, N_{\text{out}}) = (433, 567)$. The extra- and intra-axonal simulation lasted 7848s and 4168s respectively.

The simulated signals for gradient direction being $\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}$ are shown in Table. The dephase distributions in all directions exhibit significant non-Gaussian behavior (Figure 4.17).

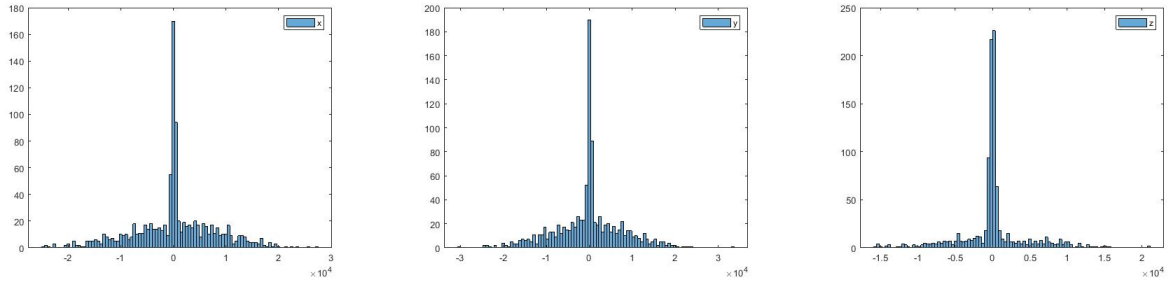


Figure 4.17: Dephase distribution ($\hat{\boldsymbol{g}} = \hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}$), from left to right).

Gradient Direction	Diffusion Signal
$(1, 0, 0)^T$	0.0193
$(0, 1, 0)^T$	0.0214
$(0, 0, 1)^T$	0.0143

Table 4.1: Diffusion signals (orthogonal fiber bundles).

4.7 Possible Extension for Robin BCs

In applications, an important source of magnetization loss is due to surface relaxation, modeled by Robin BCs. Therefore, it is valuable to incorporate its implementation into GAFRW. The basic idea appears to be theoretically sound and is presented here.

Release Cylinder Suppose that the Robin BC (equation 2.38) holds on a boundary $\partial\Omega$. Similar to our treatment of Neumann BC, numerical parameters ε and R_{rls} are introduced to avoid detailed simulations near the boundary. We introduce another length scale $\rho = \min\left(\frac{D}{h}, R_{\text{rls}}\right)$[20].

Consider a point P close to the boundary with coordinate $\boldsymbol{r}_P$ (In the sense that inequality 4.4 is false.). Construct a cylinder (Figure 4.18), Σ, with the height ρ; the axis $\boldsymbol{n}$, where $\boldsymbol{n}$ is the local interior unit normal vector of the boundary; the lower base,

$$S = \{\boldsymbol{r} : \ \|\boldsymbol{r} - \boldsymbol{r}_P\| \leqslant R_{\text{rls}}, \ (\boldsymbol{r} - \boldsymbol{r}_P) \cdot \boldsymbol{n} = 0\}; \tag{4.94}$$

and the upper base, which is just S translated in direction $\boldsymbol{n}$ by distance ρ. We refer this

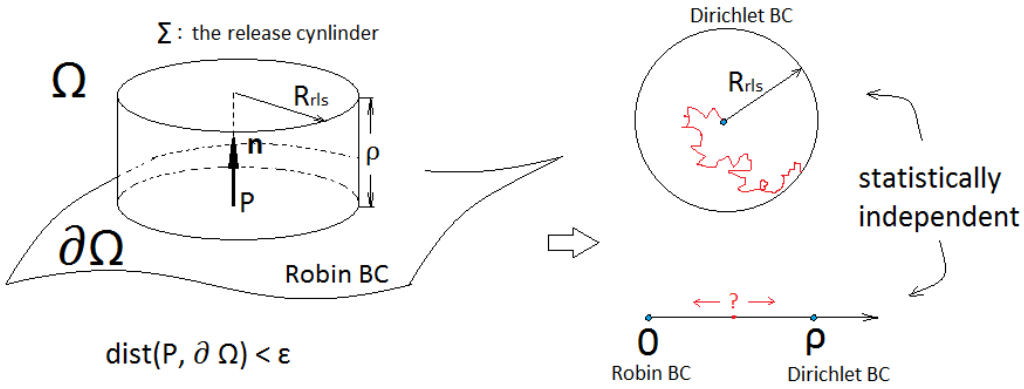


Figure 4.18: First exit problems inside the release cylinder.

[20] $\frac{D}{h}$ is the distance a spin should travel near the boundary before the surface relaxivity reduces its expected magnetic moment[13].

cylinder as the release cylinder and consider the following first exit problem: Given that S is semi-absorbing and the rest portion of Σ's boundary is fully absorbing. How long would it take for a random walker to exit Σ if its initial position is P?

The exit point can either be located on the side of Σ, or on any of its two bases. By orthogonal decomposition of Brownian motions, the original first exit time t is in fact the minimum of the following two independent first exit times:

1. Given that $\boldsymbol{r}_S(t)$ is a 2D random walk starting from the origin, t_S is the minimum amount of time s.t. $\|S(t_S)\| = R_{\text{rls}}$;
2. Consider the interval $I_\rho = [0, \rho]$ whose left and right end points are semi-absorbing and fully absorbing, respectively. t_I is the first exit time for a 1D random walk $X(t)$ to leave I_ρ ($X(0) = 0$).

Both t_S and t_I have analytically tractable CDFs and thus can be generated via the Inverse CDF technique.

Position and phase updates Since $t = \min(t_S, t_I)$ is our time step, the rule for position update will depend on which of the aforementioned first exit events occurs first.

Case 1 If $t_S = \min(t_S, t_I)$, the exit point will be on the side of Σ. Thus, the updated position is taken as,

$$\boldsymbol{r}_{\text{new}} = \boldsymbol{r}_P + \hat{\boldsymbol{u}} R_{\text{rls}} + \boldsymbol{n} \times \mathbb{E}\left[X(t_S) | t_S < t_I\right], \tag{4.95}$$

where $\hat{\boldsymbol{u}}$ is a random unit vector in $\mathbb{R}^2$. As we can see, the magnitude of the vertical displacement is taken as the conditional first moment of $X(t)$, a value within I.

Case 2 If $t_I = \min(t_S, t_I)$, we need to decide which base the exit point will be located on. If $X(t_I) = 0$, it implies that the random walker is absorbed by the semi-absorbing boundary and the random walk gets terminated. Generate a Bernoulli variate U with parameter $p_0(t_S)$, where $p_0(t)$ is the time-dependent termination probability. If $U = 1$, discard the current iteration; if $U = 0$, the updated position is taken as,

$$\boldsymbol{r}_{\text{new}} = \boldsymbol{r}_P + \hat{\boldsymbol{u}} \sqrt{\mathbb{E}\left[\|\boldsymbol{r}_S(t_I)\|^2 | t_I < t_S\right]} + \boldsymbol{n}\rho. \tag{4.96}$$

As we can see, similar to our treatment of the final update, the size of the tangential displacement is estimated by its rms value. The conditional moments (dimensionless) as functions of time (dimensionless) will be derived and tabulated once and forever.

As for the phase increment, the contribution due to the random walk in S is derived in [14]. For the contribution due to the 1D random walk, consider the random time integral,

$$\int_0^t X(t')dt'. \tag{4.97}$$

For case 1, take the rms value of integral 4.97; for case 2 where $U = 0$, take its first conditional moment instead.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This section summarizes the main achievements of this thesis.

- As a pedagogical contribution, Chapter 2 of this thesis is aimed at concisely covering the mathematical physics of dMRI for new researchers. Readers familiar with undergraduate-level differential equations and probability theory should find the material accessible.
- Numerically, both the basic and improved algorithms are programmed and tested (Chapter 3 and 4). Careful considerations are taken for accurate evaluation of functions defined in terms of infinite series. Numerical experiments showed that the computational cost saved by choosing the improved algorithm over the basic one is only moderate in typical geometries arise from dMRI of white matter.
- Methodologically, using the proposed error analysis arguments, we showed that the basic Monte Carlo method is of first order or worse (Chapter 3).
- Mathematically, the residue method for evaluating infinite series is modified and applied to problems of moment calculation.

5.2 Future Work

This section discusses the prospect of further research to come based on this study.

5.2.1 Software optimization

The computer programs in this study are written in MATLAB for purpose of high readability and short development time. For large scale simulation, however, languages such as C++ or Fortran should be more efficient. Employing techniques such as multithreading can also be helpful for performance improvement.

5.2.2 Implementation of Robin BCs

GAFRW so far are only implemented for Dirichlet and Neumann BCs. In future studies, an extension addressing Robin BCs will be worked out in details based on the main idea discussed in section 4.7. As usual, analytical formulae for calculating escape probabilities and conditional moments can be derived using the MCF formalism.

5.2.3 Inverse problems

This thesis focus entirely on the direct problem in dMRI, namely the question, 'how to compute diffusion signals given a confining geometry'. The holy grail however, is the inverse problem, 'to what extend can a general confining geometry be inferred by the diffusion signal?'. In this subsection, we discuss recent advancement of the inverse problem. In addition, a weaker inverse problem is proposed in hopes of bringing new light to the topic.

The spectral formalism (section 2.8.3) inevitably begs the comparison between the inverse problem in dMRI and the celebrated question posed by Kac[21]. Although the original Kac question has a negative answer[10], the situation in dMRI is slightly different due to the augmented complex 'reaction' term in the BT equation[1]. Recent studies have shown that at least for closed domains, it is possible to recovery of their geometries completely (e.g. see [24]). The key observation is that sequences with anti-symmetrical time profiles

[1]The MCF approach express the solution to the direct probelm entirely in terms of diffusion propagators (equation 2.88) and can be therefore thought as a perturbation approach of solving the BT equation (w.r.t to pure diffusion)

(such PGSE) are in fact ill-suited for shape estimation[2]. An experimental imaging technique called *diffusion pore imaging* (DPI) uses a weak, long pulse followed by a strong, short pulse in the opposite direction and has shown great promises[25]. In future studies we wish to explore this phenomena through MC simulation.

Real microstructures of white matter is way more complex than what can be described as closed pores. In the foreseeable future, determining their 'exact' geometries still faces great challenges. We consider an less ambitious approach inspired by the particle in a box model in introductory quantum mechanics.

Mathematically, the particle in a box problem can be formulated by the Schrödinger equation with free Hamiltonian, subject to Dirichlet BCs. On the other hand, the 'hard boundary', namely the interval containing the quantum particle, can be represented by an infinite potential well. This means that, formally, BCs can be removed if the potential term can be incorporated into the Hamiltonian (although mathematically the equation is ill-defined).

Returning to our discussion of restricted spin diffusion in magnetic fields, it is logical to model geometric restrictions as some potential landscape $V(\boldsymbol{r})$. In order to have a well-defined 'Hamiltonian' to work with, $V(\boldsymbol{r})$ needs to be smooth[3]. This implies that we essentially give up the notion of 'confining geometry' in its rigorous sense (defined by BCs) and instead define our problem in free spaces instead. Quantitatively, $V(\boldsymbol{r})$ should be 'steep' at positions close to the physical boundary.

In the absence of magnetic fields, diffusion under the influence of a deterministic potential is described by the Smoluchowski equation[22]. Applying the similar heuristic argument[4] used in section 2.9, we model the time evolution of magnetization density $m(\boldsymbol{r}, t), (\boldsymbol{r} \in \mathbb{R}^3)$ by an augmented BT equation,

$$\frac{\partial m}{\partial t} = D\nabla^2 m + \nabla \cdot (\nabla V(\boldsymbol{r}) m) - i\omega(\boldsymbol{r}, t) m, \quad \text{for} \quad \boldsymbol{r} \in \mathbb{R}^d, \tag{5.1}$$

[2]Phase information is cancelled under PGSE. One may imagine attempting to reconstruct images only from power spectrum densities. Unsurprisingly, the result will be poor.

[3]or at least differentiable in some weak sense

[4]We interpreted BT equation as 'free diffusion plus spin precession'. Here in equation 5.1, we have 'diffusion with drift plus spin precession'

where $d = 1, 2, 3$. The three terms on the RHS of equation 5.1 represent the influence due to free diffusion, boundary and magnetic field encoding respectively. The resulting signal is the total magnetization,

$$S[\omega] = \int_{\mathbb{R}^d} m(\boldsymbol{r}, TE) d\boldsymbol{r}, \tag{5.2}$$

where TE is the echo time. Notice that in our notation above, we purposefully treat the signal as a functional of the input function $\omega(\boldsymbol{r}, t)$.

We now formulate the weaker problem in terms of the following PDE inverse coefficient problem.

Inverse Problem Suppose that $m(\boldsymbol{r}, t)$ satisfies equation 5.1 and D, $\omega(\boldsymbol{r}, t)$ are known, can $V(\boldsymbol{r})$ be recovered by measuring $S[\omega]$, defined by equation 5.2?

If the answer is yes[5], we wish to answer a second inverse problem,

Optimization problem What kind of $\omega(\boldsymbol{r}, t)$'s are the most 'effective' in terms of their ability to recover $V(\boldsymbol{r})$?

Notice that the answer to the second inverse problem will depend on the choice of cost function. Meaningful choices for the cost function may address factors such as reconstruction accuracy, hardware limitations, etc.

Since for given ground truth, MC simulators should produce corresponding diffusion signals. The simulation schemes and synthesized data presented in this work can be useful for investigating the proposed inverse problems.

[5] And chances are it will be, considering the effectiveness of DPI for simple domains.

References

[1] Lars Ahlfors. *Complex Analysis*. McGraw Hill, third edition, 1979.

[2] Andrew L. Alexander, Jee Eun Lee, Mariana Lazar, and Aaron S. Field. Diffusion Tensor Imaging of the Brain. *Neurotherapeutics*, 4(3):316 – 329, 2007. Advances in Neuroimaging/Neuroethics.

[3] Soren Asmussen, Peter Glynn, and Jim Pitman. Discretization error in simulation of one-dimensional reflecting brownian motion. *Ann. Appl. Probab.*, 5(4):875–896, 11 1995.

[4] Leandro Beltrachini, Zeike A. Taylor, and Alejandro F. Frangi. A parametric finite element solution of the generalised bloch–torrey equation for arbitrary domains. *Journal of Magnetic Resonance*, 259:126 – 134, 2015.

[5] Jorge Zavala Bojorquez, Stéphanie Bricq, Clément Acquitter, François Brunotte, Paul Michael Walker, and Alain Lalande. What are normal relaxation times of tissues at 3 T? *Magnetic resonance imaging*, 35:69–80, 2017.

[6] Robert W. Brown, Yu-Chung N. Cheng, E. Mark Haacke, Michael R. Thompson, and Ramesh Venkatesan. *Magnetic Resonance Imaging: Physical Principles and Sequence Design*. John Wiley and Sons, Hoboken, New Jersey, second edition, 2014.

[7] Paul T. Callaghan. *Principles of Nuclear Magnetic Resonance Microscopy*. Oxford University Press, New York, 1991.

[8] John Crank. *The Mathematics of Diffusion*. Oxford University Press, New York, second edition, 1975.

[9] Bela A. Frigyik, Amol Kapila, and Maya R. Gupta. Introduction to the Dirichlet Distribution and Related Processes. 2010.

[10] C. Gordon. You can't hear the shape of a drum. *American Scientist*, 84(1):46–55, 1966.

[11] Denis S. Grebenkov. NMR survey of reflected Brownian motion. *Review of Modern Physics*, 79(3):1077, 2007.

[12] Denis S. Grebenkov. Nuclear magnetic resonance restricted diffusion between parallel planes in a cosine magnetic field: An exactly solvable model. *The Journal of Chemical Physics*, 126(10):104706, 2007.

[13] Denis S. Grebenkov. Analytical solution for restricted diffusion in circular and spherical layers under inhomogeneous magnetic fields. *The Journal of chemical physics*, 128 13:134702, 2008.

[14] Denis S. Grebenkov. A fast random walk algorithm for computing the pulsed-gradient spin-echo signal in multiscale porous media. *Journal of Magnetic Resonance*, 208(2):243–255, 2010.

[15] E. L. Hahn. Spin Echoes. *Physical Review*, 80(4):580, 1950.

[16] Matt G. Hall and Daniel C. Alexander. Convergence and parameter choice for Monte-Carlo simulations of diffusion mri. *IEEE Transactions on Medical Imaging*, 28(9):1354, 2009.

[17] Heather Haynes and William Holmes. *The Emergence of Magnetic Resonance Imaging (MRI) for 3D Analysis of Sediment Beds*. 12 2013.

[18] Magnus Herberthson, Evren Özarslan, Hans Knutsson, and Carl-Fredrik Westin. Dynamics of local magnetization in the eigenbasis of the bloch-torrey operator. *The Journal of Chemical Physics*, 146(12):124201, 2017.

[19] Noah A. Hughes. Infinite Series and the Residue Theorem. May 2013.

[20] A.V. Skorokhod I.I. Gikhman. *The Theory of Stochastic Processes II*. Springer-Verlag, Berlin, Heidelberg, New York, 2004.

[21] M. Kac. Can one hear the shape of a drum? *The American Mathematical Monthly*, 73(4):1–23, 1966.

[22] N.G. Van Kampen. *Stochastic Processes in Physics and Chemistry*. Elsevier, Amsterdam, Netherlands, 3rd edition, 2007.

[23] Martin D. King, John Houseman, Simon A. Roussel, Nicholas Van Bruggen, Stephen R. Williams, and David G. Gadian. q-space imaging of the brain. *Magnetic Resonance in Medicine*, 32(6):707–713, 1994.

[24] Frederik Bernd Laun, Tristan Anselm Kuder, Wolfhard Semmler, and Bram Stieltjes. Determination of the defining boundary in nuclear magnetic resonance diffusion experiments. *Phys. Rev. Lett.*, 107:048102, Jul 2011.

[25] Frederik Bernd Laun, Tristan Anselm Kuder, Andreas Wetscherek, Bram Stieltjes, and Wolfhard Semmler. NMR-based diffusion pore imaging. *Phys. Rev. E*, 86:021906, Aug 2012.

[26] Syeda Maryam, Laura McCrackin, Mark Crowley, Yogesh Rathi, and Oleg Michailovich. Application of probabilistically weighted graphs to image-based diagnosis of Alzheimer's disease using diffusion MRI. In Samuel G. Armato III and Nicholas A. Petrick, editors, *Medical Imaging 2017: Computer-Aided Diagnosis*, volume 10134, pages 620 – 629. International Society for Optics and Photonics, SPIE, 2017.

[27] Dang Van Nguyen, Jing-Rebecca Li, Denis Grebenkov, and Denis Le Bihan. A finite elements method to solve the bloch–torrey equation applied to diffusion magnetic resonance imaging. *Journal of Computational Physics*, 263:283 – 302, 2014.

[28] Khieu-Van Nguyen, Edwin Hernández-Garzón, and Julien Valette. Efficient GPU-based Monte-Carlo simulation of diffusion in real astrocytes reconstructed from confocal microscopy. *Journal of Magnetic Resonance*, 296:188 – 199, 2018.

[29] Bernt Oksendal. *Stochastic Differential Equations: An Introduction with Applications*. 1998.

[30] William S. Price. *NMR Studies of Translational Motion: Principles and Applications*. Cambridge University Press, New York, 2009.

[31] Inbar Seroussi, Denis S. Grebenkov, Ofer Pasternak, and Nir Sochen. Microscopic interpretation and generalization of the bloch-torrey equation for diffusion magnetic resonance. *Journal of Magnetic Resonance*, 277:95 – 103, 2017.

[32] 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*, 42(1):288, 1965.

[33] R.K. Michael Thambynayagam. *The Diffusion Handbook: Applied Solutions for Engineers*. US: McGraw-Hill Professional, 2011.

[34] H. C. Torrey. Bloch equations with diffusion terms. *Physical Review*, 104(3):563, 1956.

[35] C W.Gardiner. *Stochastic Methods: A Handbook for the Natural and Social Sciences*. Springer, Berlin, Germany, 2009.

[36] Chun-Hung Yeh, Benoît Schmitt, Denis Le Bihan, Jing-Rebecca, Li-Schlittgen, Ching-Po Lin, and Cyril Poupon. Diffusion Microscopist Simulator: A General Monte Carlo Simulation System for Diffusion Magnetic Resonance Imaging. *PLOS ONE*, 8(10):e76626, 2013.

APPENDICES

Appendix A

Generating Uniformly Random Directions

In this appendix, we provide a scheme for generating uniformly random points on $\mathbb{S}^2$, the unit sphere in $\mathbb{R}^3$.
One may be tempted to draw 3D standard Gaussian variates and then normalize them, which would be theoretically correct. However, there is a significant chance that the Gaussian variates have short length and the problem of numerical division by zero arises. Notice that the joint pdf of standard Gaussian distribution satisfies radial symmetry,

$$f(x,y,z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} = \frac{1}{(2\pi)^{\frac{3}{2}}} e^{-\frac{r^2}{2}}. \tag{A.1}$$

The normalization condition of $f(x,y,z)$ in spherical coordinates can be written as

$$\begin{aligned} 1 &= \int_{\mathbb{R}^3} f(x,y,z)dxdydz \\ &= \int_{\mathbb{R}^3} \frac{1}{(2\pi)^{\frac{3}{2}}} e^{-\frac{r^2}{2}} r^2 \sin\theta dr d\theta d\phi. \end{aligned}$$

The integrand in the last equation is of the form of the product of single-variable functions of coordinates r, θ (inclination angle) and ϕ (azimuth angle), which implies that random variables r, θ and ϕ are mutually independent. Therefore, the pdfs of θ and ϕ are propor-

tional to 1 and $\sin\theta$, respectively. After normalizing, one gets pdfs

$$f_1(\phi) = \frac{1}{2\pi} \quad \text{for} \quad 0 \leqslant \phi \leqslant 2\pi, \tag{A.2}$$

$$f_2(\theta) = \frac{1}{2}\sin\theta \quad \text{for} \quad 0 \leqslant \theta \leqslant \pi. \tag{A.3}$$

It is straightforward to show that $z = \cos\theta$, the z-component of a point on $\mathbb{S}^2$, follows the uniform distribution over $[-1, 1]$:

$$\begin{aligned}\mathbb{P}(Z \leqslant z) &= \mathbb{P}(\cos\theta \leqslant z) = \mathbb{P}(\theta \geqslant \arccos(z)) \\ &= \int_{\arccos(z)}^{\pi} \frac{1}{2}\sin\theta d\theta = \frac{1}{2}(1+z).\end{aligned}$$

Therefore, we have the following scheme for generating a uniformly distributed 3D random unit vector:

1. Generate $U_1, U_2 \quad \text{iid} \quad U[0,1]$;
2. $Z = 2U_1 - 1$;
3. $(X, Y) = \sqrt{1 - Z^2}(\cos(2\pi U_2), \sin(2\pi U_2)$;
4. Return (X, Y, Z).

Appendix B

Derivation of Moments

The conditional 1st and 2nd moments of the dephase used in GAFRW are derived by using the matrix-formalism in [14]. More details of their derivations are provided in the present and next appendix as they are crucial for any further extension of the fast random walk algorithm. Guiding ideas of the derivation are explained in this appendix.

First and the second order MCFs Recall the non-dimensionalized first exit problem stated in subsection 4.3.1.

First moments are used in all but the last update. For the non-boundary case, the final condition is that the walker exited the sphere at time t at position r. On the other hand, for the boundary case, at time t the walker still exit the unit sphere yet the exact exit point is unspecified. Using the general MCF in its matrix form (2.96), we can formally write down the first order MCF,

$$\mathbb{E}\left[B(\boldsymbol{r}(t_1))\right] = Ue^{-\Lambda t_1}\mathcal{B}e^{-\Lambda(t-t_1)}V, \tag{B.1}$$

where matrices U consists of the generalized Fourier coefficients of the Dirac delta function centered at 0 for both cases; and the entries of V are the normal directional derivative of the eigenfunctions

- evaluated at the exit point, for the boundary case;
- averaged over the unit sphere, for the non-boundary case.

The definition of V is justified by our discussion on BCs for the propagator (subsection 2.7.1).

The second moment is used in the last update, in which the final condition is that the walker has not yet exit the sphere by the time t. The 2nd order MCF is therefore,

$$\mathbb{E}\left[B(\boldsymbol{r}(t_1))B(\boldsymbol{r}(t_2))\right] = Ue^{-\Lambda t_1}\mathcal{B}e^{-\Lambda(t_2-t_1)}\mathcal{B}e^{-\Lambda(t-t_2)}\tilde{U}^*, \tag{B.2}$$

where U has the same definition in equation B.1 and $\tilde{U}*$ is defined in 2.93.

Time average As established in section 2.8, the moments of φ are the time average of MCFs. In GAFRW, at each update the gradient stays constant (FID sequence), thus the 1st and 2nd order time average operators for our problem are

$$\begin{aligned}\langle h\rangle_1 &= \int_0^t dt_1 h(t_1);\\ \langle h\rangle_2 &= \int_0^t dt_1 \int_{t_1}^t dt_2 h(t_1,t_2).\end{aligned} \tag{B.3}$$

MCFs B.1, B.2 can then be expanded in the same manner as series 2.92. Applying the operators $\langle\cdot\rangle_1$, $\langle\cdot\rangle_2$ to the expanded MCF of corresponding order, one may notice some cancellation which reduces the number of sums to be evaluated.

Matrix $\mathcal{B}$ Recall that the matrix $\mathcal{B}$ is the coordinate representation of the gradient field in the eigenbasis.

It is obvious that the choice of coordinate system does not affect the amount of spin dephase. In the computation of $\mathcal{B}$, we used the spherical coordinate system whose zenith direction (z-axis) is the gradient direction. This choice is made solely for simplifying the calculation of integrals and should not be confused with the convention in NMR literature of z-axis being the direction of the main field. Consequently, the linear spatial profile of the DW gradient has the form

$$B(\boldsymbol{r}) = r\cos\theta, \tag{B.4}$$

where θ is the inclination angle. The following sections discuss the quantities and techniques required for calculating matrix $\mathcal{B}$ in detail.

It is worth noticing that, the final analytical results in terms of infinite series may still be ill-suited for numerical evaluation, especially for small values of t. For that reason, asymptotic formulae are derived.

B.1 Normalized Eigenfunctions

We briefly present the well-known solution of the Laplacian eigenvalue problem in the unit sphere $\mathbb{B} = \{\boldsymbol{r} : \|\boldsymbol{r}\| < 1\}$.

The azimuth angle, φ, is omitted since the magnetic field spatial profile does not dependent on it. The Helmholtz's equation $\nabla^2 u + \lambda u = 0$ becomes

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial u}{\partial r}\right) + \lambda r^2 + \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\theta\frac{\partial u}{\partial\theta}\right) = 0, \tag{B.5}$$

subject to the Dirichlet BC,

$$u(1,\theta) = 0, \quad \text{for } 0 < \theta < \pi. \tag{B.6}$$

Applying the method of separation of variables, the eigenfunctions are of the form

$$u_{nk}(r,\theta) = \beta_{nk} j_n(\alpha_{nk} r) P_n(\cos\theta) \quad \text{for} \quad n,k = 0,1,2,\ldots \tag{B.7}$$

where

- $j_n(z)$ is the n-th order spherical bessel function of the 1st kind;
- α_{nk}'s are the positive zeros of $j_n(z)$;
- $P_n(x)$ is the n-th order Legendre polynomial;
- β_{nk} is the normalization constant.

To calculate β_{nk}, we need to evaluate the integral

$$\int_\Omega u_{nk}^2(\boldsymbol{r})d\boldsymbol{r} = 2\pi\beta_{nk}^2 \int_0^1 j_n^2(\alpha_{nk}r)r^2 dr \int_0^\pi P_n^2(\cos\theta)\sin\theta d\theta. \tag{B.8}$$

The angular integral is equal to $\frac{2}{2n+1}$ due to the orthogonality condition:

$$\int_{-1}^{1} P_m(x)P_n(x)dx = \frac{2\delta_{mn}}{2n+1},$$

The radial integral is obtained by using the following technique: First by using the fact that spherical Bessel functions are related to the standard Bessel functions via relation:

$$j_n(z) = \sqrt{\frac{\pi}{2z}} J_{n+\frac{1}{2}}(z), \tag{B.9}$$

one can rewrite the radial integral into

$$\int_0^1 j_n^2(\alpha_{nk} r) r^2 dr = \frac{\pi}{2\alpha_{nk}^3} \int_0^{\alpha_{nk}} J_{n+\frac{1}{2}}^2(z) z dz \tag{B.10}$$

Secondly, $J_{n+\frac{1}{2}}$ is a solution of the Bessel DE:

$$z^2 J'' + zJ' + (z^2 - l^2)J = 0, \quad \text{where} \quad l = n + \frac{1}{2},$$

and we then manipulate the DE so that the integrand of the RHS of equation B.10 can be expressed as a derivative. By multiplying $2J'$, one gets

$$\begin{aligned} [(J')^2 z^2]' + z^2 (J^2)' - l^2 (J^2)' = 0, \\ \Rightarrow [(J')^2 z^2]' + (z^2 J^2)' - 2zJ^2 - l^2 (J^2)' = 0. \end{aligned}$$

As as result, one has

$$2z J_{n+\frac{1}{2}}^2 = [(J'_{n+\frac{1}{2}})^2 z^2 + z^2 J_{n+\frac{1}{2}}^2 - l^2 (J_{n+\frac{1}{2}}^2)]'. \tag{B.11}$$

Rewrite the right hand side above in terms of $j_n(z)$ then integrate and use $j_n(\alpha_n k) = 0$, one gets

$$\begin{aligned} \int_0^{\alpha_n k} 2z J_{n+\frac{1}{2}}^2 dz &= \frac{2}{\pi} \left(\frac{1}{2} z j_n(z) + z^{\frac{3}{2}} j'_n(z) \right)^2 + (z^2 - l^2) \frac{2z}{\pi} j_n^2(x) \Big|_0^{\alpha_{nk}} \\ &= \frac{2}{\pi} \alpha_{nk}^3 (j'_n(\alpha_{nk}))^2. \end{aligned} \tag{B.12}$$

Combine the radial and angular integral, we have

$$\beta_{nk} = \sqrt{\frac{2n+1}{2\pi}} \frac{1}{j'_n(\alpha_{nk})}. \tag{B.13}$$

B.2 Elements of the Matrix $\mathcal{B}$

The integral defining the elements of the matrix $\mathcal{B}$ for the linear gradient field $B(\boldsymbol{r}) = r\cos\theta$ expressed in spherical coordinates is expressed as

$$\mathcal{B}_{nk,n'k'} = 2\pi\beta_{nk}\beta_{n'k'}\int_0^1 j_n(\alpha_{nk}r)j_{n'}(\alpha_{n'k'}r))r^3dr\int_0^\pi P_n(\cos\theta)P_{n'}(\cos\theta)\cos\theta\sin\theta d\theta. \tag{B.14}$$

B.2.1 Integral over θ

The angular integral in easy to evaluate. Using the recurrence relations of Legendre polynomials,

$$(2n+1)xP_n(x) = (n+1)P_{n+1}(x) + nP_{n-1}(x),$$

and the orthogonality condition,

$$\int_{-1}^1 P_m(x)P_n(x) = \frac{2\delta_{mn}}{2n+1},$$

where δ_{mn} is the Kronecker delta, one gets

$$\begin{aligned}
&\int_0^\pi P_n(\cos\theta)P_{n'}(\cos\theta)\cos\theta\sin\theta d\theta \\
&= \int_{-1}^1 P_n(x)P_{n'}(x)xdx \\
&= \frac{1}{2n'+1}\left((n'+1)\int_{-1}^1 P_n(x)P_{n'+1}(x)dx + n'\int_{-1}^1 P_n(x)P_{n'-1}(x)dx\right) \\
&= \frac{2}{2n'+1}\left(\frac{(n'+1)\delta_{n,n'+1}}{2(n'+1)+1} + \frac{n'\delta_{n,n'-1}}{2(n'-1)+1}\right).
\end{aligned} \tag{B.15}$$

Writing out the results for $n = n'+1$ and $n = n'-1$ separately and combining them with a single Kronecker delta, one gets

$$\int_0^\pi P_n(\cos\theta)P_{n'}(\cos\theta)\cos\theta\sin\theta d\theta = \delta_{n,n\pm1}\frac{n+n'+1}{(2n+1)(2n'+1)}. \tag{B.16}$$

B.2.2 Integral over r

It is much more difficult to evaluate the radial integral in equation B.14. See [13] for details for this computation. The basic idea is to once again exploit the fact that $v_{nk}(r) = j_n(\alpha_{nk}r)$ satisfying the underlying DE,

$$(r^2v')' + (\alpha_{nk}^2 r^2 - n(n+1))v = 0 \quad \text{for } k = 0, 1, 2, \ldots \tag{B.17}$$

subject to BCs

$$v(0) < \infty, \quad v(1) = 0. \tag{B.18}$$

Consider the integral

$$(\alpha_{nk}^2 - \alpha_{n'k'}^2)^2 \int_0^1 r^3 v_{nk}(r) v_{n'k'}(r) dr. \tag{B.19}$$

Using the DE in B.17 to isolate $\alpha_{nk}^2 r^2 v_{nk}(r)$ and its primed counterparts. we can remove the derivatives in the 'integrand' via integration by parts. Under the condition $n = n' \pm 1$ the radial integral has a closed form result:

$$\int_0^1 j_n(\alpha_{nk}r) j_{n'}(\alpha_{n'k'}r)) r^3 dr = \frac{j_n'(\alpha_{nk}) j_{n'}'(\alpha_{n'k'}) \alpha_{nk}\alpha_{n'k'}}{(\alpha_{nk}^2 - \alpha_{n'k'}^2)^2}. \tag{B.20}$$

Appendix C

Residue Summation Technique

Analytical results regarding the conditional moments of dephases are crucial to GAFRW. Explicit evaluations of infinite series involving zeroes of Bessel functions are frequently encountered. Techniques from elementary calculus are mostly ineffective for computing those summations. Fortunately, with the help of Residue and Mittag-Leffler theorem in complex analysis, we are able to derive closed-form results for those series. The techniques demonstrated in the present appendix are heavily inspired by [19], [11].

In this appendix, we first briefly review some fundamentals of complex analysis. Section C.2 sketches the main idea of the summation procedure. Section C.3 shows how specific moments are calculated.

C.1 Some complex analysis

We first briefly review some basics concepts and results about complex functions. A detailed treatment can be found in [1].

Definition of a Residue Let $f(z)$ be a meromorphic function a punctured disk centred at $z = z_0$. It then admits a Laurent series expansion,

$$f(z) = \sum_{n=-\infty}^{\infty} c_n (z - z_0)^n. \tag{C.1}$$

The *residue* of $f(z)$ at z_0 is defined as the coefficient c_{-1},

$$\text{Res}(f; z_0) = c_{-1}. \tag{C.2}$$

The notation for residue can be abbreviated to $\text{Res}(z_0)$ if there is no ambiguity regarding the function of interest.

Residues at poles If the function

$$(z - z_0)^n f(z) \tag{C.3}$$

is holomorphic and non-zero in the neighborhood of z_0, z_0 is called a pole of order n. For calculating residues at poles, apart from directly computing the Laurent series, another common method is to use the following formula,

$$\text{Res}(z_0) = \lim_{z \to z_0} \frac{1}{(n-1)!} \frac{d^{n-1}}{dz^{n-1}} [(z - z_0)^n f(z)]. \tag{C.4}$$

In particular, when n=1,

$$\text{Res}(z_0) = \lim_{z \to z_0} (z - z_0) f(z), \tag{C.5}$$

in which z_0 is referred to as a *simple* pole.

Residue Theorem Residues are significant due to their roles in calculating contour integrals. Let Γ is a positively oriented, simple closed contour. The celebrated *Residue theorem* states that

$$\oint_\Gamma f(z) dz = 2\pi i \sum_k \text{Res}(z_k), \tag{C.6}$$

where z_k's are poles of $f(z)$ enclosed by Γ.

Mittag-Leffler Theorem When studying rational functions, it is often useful to rewrite a rational function in terms of a sum of polynomials of simpler fractions, a technique called partial fraction decomposition. The Mittag-Leffler Theorem in complex analysis is concerned with an analogous procedure for meromorphic functions, known as the *pole expansion.*

Let $D \subset \mathbb{C}$ be an open set and $E \subset D$ a closed discrete subset. For each $a \in E$, let p_a be a polynomial with a zero constant term.

The Mittag-Leffler Theorem states that there exists a meromorphic function f on D s.t. for each $a \in E$, the function

$$f(z) - p_a((z-a)^{-1}) \tag{C.7}$$

only has a removable singularity at a. Furthermore, if E is unbounded, $f(z)$ admits the expansion[1],

$$f(z) = h(z) + \sum_{a \in E} \left[p_a((z-a)^{-1}) + g_a(z) \right], \tag{C.8}$$

where $h(z)$ is an entire function and for each a, $g_a(z)$ is a polynomial.

In particular, the rational function

$$p_a((z-a)^{-1}) \tag{C.9}$$

is referred to as the *principal part* of $f(z)$ at $z = a$, which in other words is the negative-power portion of the Laurent series of $f(z)$ about $z = a$.

The Residue and Mittag-Leffler Theorem serve as the foundation for our summation technique. Although the technique is quite general, meticulous justifications are needed for individual cases.

C.2 Main idea

Summation problem Let $f(z)$ be meromorphic function with non-removable singularities at $\{p_i\}$ and $\{z_k\}$ be a prescribed discrete set of complex numbers with increasing moduli. Assuming that $\{p_i\} \cap \{z_k\} = \varnothing$ [2], compute the sum,

$$\sum_k f(z_k). \tag{C.10}$$

[1] Here the equal sign should be understood as uniform convergence for arbitrary compact set $K \subset D$.
[2] otherwise certain terms in equation C.10 may blow up

Basic idea Suppose that there exists a meromorphic function $G(z)$ s.t.

P.1 $G(z)$ has simple poles at precisely the points $\{z_k\}$;

P.2 The residues of $G(z)$ at $\{z_k\}$ are all equal to 1.

Let $\{\Gamma_N\}$ be a sequence of simple, closed, positively oriented contours so that Γ_N encloses exactly $z_1, \ldots, z_N$ out of all z_k's. [3]. Consider the contour integral

$$\oint_{\Gamma_N} f(z)G(z)dz \tag{C.11}$$

and apply Residue Theorem, we get

$$\begin{aligned}\oint_{\Gamma_N} f(z)G(z)dz &= 2\pi i \sum \text{Res}(f(z)G(z);\ \text{poles of } f(z)G(z) \text{ enclosed by } \Gamma_N) \\ &= 2\pi i \sum \text{Res}(f(z)G(z); p_i) + 2\pi i \sum_{k=1}^{N} \text{Res}(f(z)G(z); z_k).\end{aligned} \tag{C.12}$$

The second sum in the last step equals to

$$\sum_{k=1}^{N} f(z_k) \tag{C.13}$$

since

$$\text{Res}(G(z)f(z); z_k) = \lim_{z \to z_k} (z - z_k)G(z)f(z) = \text{Res}(G(z); z_k)f(z_k) = f(z_k). \tag{C.14}$$

If the contour integral on the LHS of the equation C.12 vanishes as N approaches infinity, we have the following *Residue Summation Formula*, given by

$$\sum_k f(z_k) = -\sum_i \text{Res}(G(z)f(z); p_i), \tag{C.15}$$

where $G(z)$ satisfies **P.1**, **P.2**.

[3] For example , $\{\Gamma_N\}$ can be concentric circles centred at the origin with radius $R_N \to \infty$ as $N \to \infty$

Existence and construction of $G(z)$ However, questions remain whether the function $G(z)$ exists and if so how we can construct it. In cases where the set $\{z_k\}$ is finite, both questions are trivial and we simply choose

$$G(z) = \sum_{k=1}^{n} \frac{1}{z - z_k}. \tag{C.16}$$

In cases where $\{z_k\}$ is infinite however, there is both good news and bad news.

The good news is that $G(z)$ always exists. The Mittag-Leffler Theorem ensures that it is always possible to construct a meromorphic function with prescribed poles and principal parts at those poles, provided $\{z_k\}$ is unbounded.

However, the bad news is that in general, the construction of $G(s)$ can be difficult. If one naively choose

$$G(z) = \sum_{k=1}^{\infty} \frac{1}{z - z_k}, \tag{C.17}$$

there is guarantee that equation C.17 will converge. In the present appendix, we limit our discussion within a special case where z_k's are simple zeroes of some entire function. For our application this turns out to be sufficient.

A special case Let $y(z)$ be an entire function having only simple zeroes at precisely $\{z_k\}$. Furthermore, $y(z)$ is assumed to admit the following infinite product form,

$$y(z) = \prod_k (z - z_k). \tag{C.18}$$

Then its logarithmic derivative can be written in terms of pole expansion:

$$\frac{y'(z)}{y(z)} = \sum_k \frac{1}{z - z_k}. \tag{C.19}$$

The left hand side is our desired $G(z)$. The convergence can be assured if $\{z_k\}$ is unbounded and $\sum_k \frac{1}{|z_k|} = \infty$.

Thus Residue Summation Formula C.15 is then written as

$$\sum_k f(z_k) = -\sum_i \mathrm{Res}\left(\frac{y'(z)}{y(z)} f(z); p_i\right). \tag{C.20}$$

Lastly, one needs to check that as $R_N \to \infty$,

$$\left| \oint_{\Gamma_N} f(z)G(z)dz \right| \to 0, \tag{C.21}$$

which is done by using an estimation lemma.

C.3 Application to moment calculations

Summations involving zeros of spherical Bessel functions $j_n(z)$ are common in computing moments of dephase φ. Let $\{\alpha_{nk}\}$ be the positive zeroes of $j_n(z)$, $n = 0, 1$. We shall examine how the Residue Summation Formula applies for both values of n.

Sums involving $\{\alpha_{0k}\}$ Since j_0 is in fact the sinc function,

$$j_0(z) = \frac{\sin z}{z}, \tag{C.22}$$

we know that $\{\alpha_{0k}\} = \{(k+1)\pi, k = 0, 1, 2, \dots\}$, which are in turn positive zeroes of $y(z) = \sin z$. We thus choose

$$G(z) = \frac{y'(z)}{y(z)} = \cot z. \tag{C.23}$$

Since the zero set of $y(z)$ is [4]

$$y^{-1}[\{0\}] = \{0\} \cup \{\pm\alpha_{0k}\} \tag{C.24}$$

Formally, the Residue Summation Formula (equation C.15) reads

$$\begin{aligned} \sum_{z_k \in y^{-1}(0)} f(z_k) &= -\sum_i \mathrm{Res}(\cot(z) f(z); p_i) \\ \Rightarrow f(0) + \sum_{k=0}^{\infty} [f(-\alpha_{0k}) + f(\alpha_{0k})] &= -\sum_i \mathrm{Res}(\cot(z) f(z); p_i). \end{aligned} \tag{C.25}$$

The validity of equation C.25 depends on the choice of $f(z)$. In similar fashion, another useful formula can be derived for evaluate alternating series of the form,

$$\sum_{z_k \in y^{-1}[\{0\}]} (-1)^k f(z_k). \tag{C.26}$$

[4] $f^{-1}[\mathcal{A}]$ stands for the inverse image of the set $\mathcal{A}$.

Noticing that $\sec(\alpha_{0k}) = (-1)^{k+1}$, we apply equation C.25 to function $\sec(z)f(z)$,

$$\begin{aligned} f(0) + \sum_{k=0}^{\infty}(-1)^{k+1}[f(-\alpha_{0k}) + f(\alpha_{0k})] &= -\sum_i \text{Res}(\cot(z)\sec(z)f(z); p_i) \\ &= -\sum_i \text{Res}(\csc(z)f(z); p_i). \end{aligned} \tag{C.27}$$

Sums involving $\{\alpha_{1k}\}$ The positive zeros of the function

$$j_1(z) = \frac{\sin z}{z^2} - \frac{\cos z}{z} \tag{C.28}$$

are the positive roots of a transcendental equation. More specifically,

$$\{\alpha_{1k}\} = x > 0 : \quad \tan(x) = x. \tag{C.29}$$

We would like to evaluate sums of the form

$$\sum_{z_k \in j_1^{-1}[\{0\}]} f(z_k) \tag{C.30}$$

by using equation C.15. By choosing $G(z)$ as the logarithmic derivative of $j_1(z)$, which can be written as

$$G(z) = \frac{j_1'(z)}{j_1(z)} = \frac{j_0(z)}{j_1(z)} - \frac{2}{z}. \tag{C.31}$$

Since $j_1(z)$ is an odd function, its full zero set also includes 0 the opposites of α_{1k}. Applying the summation formula C.15, we arrive at

$$f(0) + \sum_{k=0}^{\infty}[f(-\alpha_{1k}) + f(\alpha_{1k})] = -\sum_i Res\left(\left(\frac{j_0(z)}{j_1(z)} - \frac{2}{z}\right) f(z); p_i\right). \tag{C.32}$$

In the next section, we proceed to use the tools developed to evaluate a concrete example.

C.4 An Example

In subsection 4.5.6, we wish evaluate the series **??**. The numerator of the general term may seem daunting at first glance since it involves a radicals which is not meromorphic. One can avoid this nuisance can be avoided by noticing that α_{1k}'s satisfies $\tan(x) = x$. In addition, since $\{\sin \alpha_{1k}\}_k$ alternates its sign w.r.t the index k, we have the following relation

$$\csc(\alpha_{1k}) = (-1)^{k+1} \frac{\sqrt{1+\alpha_{1k}^2}}{\alpha_{1k}} \quad \text{for} \quad k = 0, 1, \ldots, \tag{C.33}$$

which not removes not only the radical but also the alternating sign $(-1)^k$ as well. We thus have the following simplified problem:

Evaluate

$$\tilde{I}(s) = \sum_{k=0}^{\infty} \frac{\alpha_{1k} \csc(\alpha_{1k})}{s + \alpha_{1k}^2}. \tag{C.34}$$

Solution Let $f(z) = \frac{z \csc(z)}{s+z^2}$. Since this is a series involving $\{\alpha_{1k}\}$, the Residue Summation Formula C.32 will be used, which calls for the residues of the function

$$\frac{j_1'(z)}{j_1(z)} f(z) = \left(\frac{j_0(z)}{j_1(z)} - \frac{2}{z} \right) \frac{z \csc(z)}{s+z^2} = \left(\frac{1}{j_1(z)} - 2\csc(z) \right) \frac{1}{s+z^2} \tag{C.35}$$

at the poles of $f(z)$, which are

$$\{\pm i\sqrt{s}\} \cup \{n\pi : n \in \mathbb{Z} \setminus \{0\}\}. \tag{C.36}$$

The residues at the two imaginary poles can be calculated by using formula C.5:

$$\begin{aligned}
\operatorname{Res}(\pm i\sqrt{s}) &= \lim_{z \to \pm i\sqrt{s}} \frac{z \mp i\sqrt{s}}{s+z^2} \left(\frac{1}{j_1(z)} - 2\csc(z) \right) \\
&= \begin{cases} \frac{1}{2i\sqrt{s}} \left(\frac{1}{j_1(i\sqrt{s})} - 2\csc(i\sqrt{s}) \right) & \text{if } + \\ \frac{1}{-2i\sqrt{s}} \left(\frac{1}{j_1(-i\sqrt{s})} - 2\csc(-i\sqrt{s}) \right) & \text{if } - \end{cases} \\
&= \frac{1}{2j_1(i\sqrt{s})i\sqrt{s}} - \frac{\csc(i\sqrt{s})}{i\sqrt{s}} \\
&= \frac{\sqrt{s}}{2(\sinh\sqrt{s} - \sqrt{s}\cosh\sqrt{s})} + \frac{1}{\sqrt{s}\sinh(\sqrt{s})}.
\end{aligned} \tag{C.37}$$

In the last step, relations between hyperbolic and trigonometric functions

$$\sinh(x) = -i\sin(ix), \qquad \text{and} \qquad \cosh(x) = \cos(ix) \tag{C.38}$$

are invoked to eliminate the imaginary unit. Similarly, we obtain the residues at the real poles:

$$\begin{aligned}
\mathrm{Res}(n\pi) &= \lim_{z\to n\pi} \frac{z-n\pi}{s+z^2}\left(\frac{1}{j_1(z)} - 2\csc(z)\right) \\
&= -2\lim_{z\to n\pi} \frac{z-n\pi}{\sin(z)(s+z^2)} \\
&= \frac{-2}{s+z^2}\lim_{z\to n\pi}\frac{1}{\cos(z)} = \frac{2(-1)^{n+1}}{s+n^2\pi^2}.
\end{aligned} \tag{C.39}$$

We are now ready to write down both sides of the formula C.32. Noticing that $f(z)$ is an even function and $f(0) = \frac{1}{s}$, we have

$$\begin{aligned}
\frac{1}{s} + \tilde{I}(s) &= -(\mathrm{Res}(-i\sqrt{s}) + \mathrm{Res}(i\sqrt{s}) + \sum_{n\neq 0}\mathrm{Res}(n\pi)) \\
&= -\frac{\sqrt{s}}{\sinh\sqrt{s} - \sqrt{s}\cosh\sqrt{s}} - \frac{2}{\sqrt{s}\sinh(\sqrt{s})} - 2\sum_{n\neq 0}\frac{(-1)^{n+1}}{s+n^2\pi^2}.
\end{aligned} \tag{C.40}$$

The alternating series on the RHS involves $\{\alpha_{0k}\}$,

$$\sum_{n\neq 0}\frac{(-1)^n}{s+n^2\pi^2} = 2\sum_{k=0}^{\infty}\frac{(-1)^{k+1}}{s+\alpha_{0k}^2}. \tag{C.41}$$

Using equation C.27, we have

$$\frac{1}{s} + 2\sum_{k=0}^{\infty}\frac{(-1)^{k+1}}{s+\alpha_{0k}^2} = \sum_{\pm}\mathrm{Res}\left(\frac{\csc(z)}{s+z^2}; \pm i\sqrt{s}\right). \tag{C.42}$$

Both residues of the RHS are equal to $\frac{\csc(i\sqrt{s})}{2i\sqrt{s}}$. Applying relations C.38 once again to eliminate complex numbers, C.42 can be rearranged into,

$$\sum_{n\neq 0}\frac{(-1)^n}{s+n^2\pi^2} = -\frac{1}{s} + \frac{1}{\sqrt{s}\sinh(\sqrt{s})} \tag{C.43}$$

Plugging equation C.43 into equation C.40, we arrive at the closed-form expression of C.34:

$$\tilde{I}(s) = \frac{\sqrt{s}}{2(\sinh(\sqrt{s}) - \sqrt{s}\cosh(\sqrt{s}))} + \frac{3}{2s}. \tag{C.44}$$

Both applications of the Residue Summation Formula (C.35, C.42) are justified. Here we only show the first one. The proof is a modification of the proof for Lemma 3.1 in [19].

Claim There exists a family of contours $\{\Gamma_N\}$ s.t. the contour integral

$$\oint_{\Gamma_N} \left(\frac{1}{j_1(z)} - 2\csc(z)\right) \frac{1}{s+z^2} dz \to 0, \tag{C.45}$$

as $N \to \infty$.

Proof Let Γ_N (Figure C.1) be a positively oriented square with vertices

$$A_1 = R(1+i), \quad A_2 = R(-1+i), \quad A_3 = R(-1-i), \quad A_4 = R(1-i), \tag{C.46}$$

where $R = (N + \frac{1}{4})\pi$, $N = 1, 2, \ldots$.
We shall split the contour integral of interest C.45 into two parts,

$$\oint_{\Gamma_N} \frac{1}{j_1(z)(s+z^2)} dz + 2\oint_{\Gamma_N} \frac{csc(z)}{s+z^2} dz. \tag{C.47}$$

Second integral The second part of C.47 has the following estimate,

$$\left| \oint_{\Gamma_N} \frac{\csc z}{s+z^2} dz \right| \leqslant \max_{z \in \Gamma_N} \frac{1}{|\sin z|} \frac{1}{|s+z^2|} 8R. \tag{C.48}$$

Let N to be sufficiently large so that Γ_N encloses $\pm\sqrt{i}$, the two imaginary poles. Since $|z| \geqslant R^2$ on $Gamma_N$, using the triangle inequality, we have

$$|s+z^2| \geqslant |z|^2 - |s| \geqslant R^2 - |s|. \tag{C.49}$$

We wish to establish a lower bound for $|\sin z|$ for $z \in \Gamma_N$. Since our contour is a square, we consider expressing the complex sine function in its rectangular form. Plugging $z = x + iy$ $(x, y \in \mathbb{R})$ into

$$\sin z = \frac{e^{iz} - e^{-iz}}{2i} \tag{C.50}$$

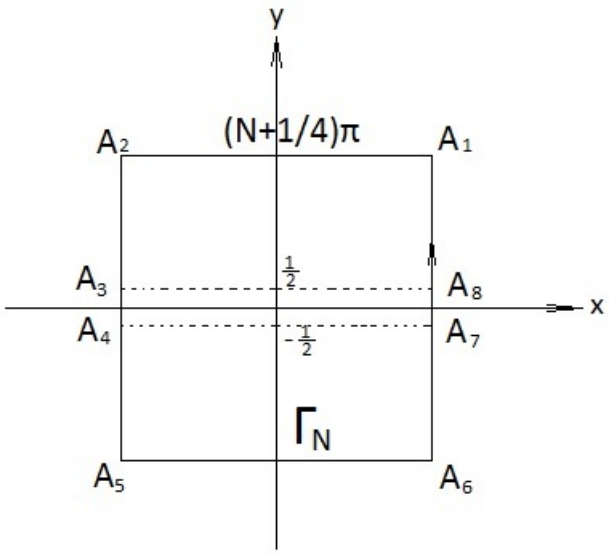


Figure C.1: Γ_N: an illustration

and using Euler's relation $e^{\theta} = \cos\theta + i\sin\theta$, we have

$$\sin z = \sin x \cosh y + i \cos x \sinh y. \tag{C.51}$$

Noticing that when $x = \pm R$, $\sin^2 x = \cos^2 x = \frac{1}{2}$, thus the modulus of $\sin z$ can be simplified as,

$$\begin{aligned} |\sin z|^2 &= |\sin x \cosh y + i \cos x \sinh y|^2 \\ &= \sin^2 x \cosh^2 y + \cos^2 x \sinh^2 y \\ &= \frac{1}{2}(\cosh^2 y + \sinh^2 y) = \frac{1}{2}\cosh 2y. \end{aligned} \tag{C.52}$$

Since for real arguments, hyperbolic cosines have lower bound of 1, one the two vertical sides of Γ_N, $\sin z$ has the estimate,

$$|\sin z| \geqslant \frac{1}{\sqrt{2}}. \tag{C.53}$$

One the horizontal sides of Γ_N, $y = \pm R$ and $\sin z$ has the estimate $|\sin z| \geqslant \sinh R$ since

$$\begin{aligned} |\sin z|^2 &= \sin^2 x \cosh^2 y + \cos^2 x \sinh^2 y \\ &\geqslant (\sin^2 x + \cos^2 x)\sinh^2 y \geqslant \sinh^2 y. \end{aligned} \tag{C.54}$$

As $R \to \infty$,

$$|\sin z| \geqslant \min(1/\sqrt{2},\ \sinh R) = \sinh R, \tag{C.55}$$

for $z \in \Gamma_N$. Therefore, by combining inequalities C.48, C.49, C.55, we have the convergence result for the second part of C.47,

$$\left| \oint_{\Gamma_N} \frac{\csc z}{s + z^2} dz \right| \leqslant \frac{8R}{\sinh R(R^2 - |s|)} \to 0, \quad \text{for } R \to \infty. \tag{C.56}$$

First integral For the first part of C.47, one notices that $\frac{\cos z}{z}$ should dominate $j_1(z)$ as $|z|$ gets large. To express this idea rigorously, we rewrite

$$\frac{1}{j_1(x)} = \frac{1}{\frac{\sin z}{z^2} - \frac{\cos z}{z}} = -\frac{z}{\cos z} \frac{1}{\left(1 - \frac{\tan z}{z}\right)}. \tag{C.57}$$

Applying the complex geometric series, one can write

$$\frac{1}{\left(1 - \frac{\tan z}{z}\right)} = 1 + o(1), \quad \text{for } z \in \Gamma_N. \tag{C.58}$$

This is true because $\frac{|\tan z|}{|z|} < 1$ for $z = x + iy \in \Gamma_N$. To see this is true, we once again investigate bounds on vertical and horizontal parts of Γ_N individually. The rectangular form of $\cos z$ is given by

$$\cos z = \cos x \cosh y - i \sin x \sinh y. \tag{C.59}$$

- If $x = \pm R$, then

$$\begin{aligned} \tan z &= \frac{\sin x \cosh y + i \cos x \sinh y}{\cos x \cosh y - i \sin x \sinh y} = \frac{\tan x \cosh y + i \sinh y}{\cosh y - i \tan x \sinh y} \\ &= \frac{\tan\left(\pm(N + \frac{1}{4})\pi\right) \cosh y + i \sinh y}{\cosh y - i \tan\left(\pm(N + \frac{1}{4})\pi\right) \sinh y} \\ &= \frac{\pm \cosh y + i \sinh y}{\cosh y \mp i \sinh y} \\ \Rightarrow |\tan z| &= 1. \end{aligned}$$

- If $y = \pm R$, then

$$\begin{aligned} |\tan z| &= \frac{|e^{iz} - e^{-iz}|}{|e^{iz} + e^{-iz}|} = \frac{|e^{ix-y} - e^{-ix+y}|}{|e^{ix-y} + e^{-ix+y}|} \\ &\leqslant \frac{|e^{ix-y}| + |e^{-ix+y}|}{||e^{ix-y}| - |e^{-ix+y}||} = \frac{e^{-y} + e^{y}}{|e^{-y} - e^{y}|} = |\coth y| \\ \Rightarrow |\tan z| &\leqslant \coth R. \end{aligned}$$

Combing both cases, we have $|\tan z| \leqslant \max(1,\ \coth R)$ and thus as $R \to \infty$,

$$\frac{|\tan z|}{|z|} \leqslant \frac{\coth R}{R} < 1.$$

Hence, we obtained the following rigorous estimate for the reciprocal of $j_1(z)$:

$$\exists C > 0 \text{ s.t.} \quad \left|\frac{1}{j_1(z)}\right| \leqslant C|z \sec z| \quad \text{for} \quad z \in \Gamma_N. \tag{C.60}$$

Hence, the second contour integral has the following estimate :

$$\left|\oint_{\Gamma_N} \frac{1}{j_1(z)(s+z^2)} dz\right| \leqslant C \max_{z\in\Gamma_N} \frac{|z|}{|s+z^2|} \left|\oint_{\Gamma_N} \sec z dz\right| \leqslant \frac{C\sqrt{2}R}{R^2 - |s|} \left|\oint_{\Gamma_N} \sec z dz\right|. \tag{C.61}$$

The fraction in the RHS of equation C.61 goes to zero. Unfortunately, the contour integral of $\sec z$ does not vanishes as Γ_N grows. The reason is that while on the most part of Γ_N $\sec z$ is negligible, it is not the case for points close to the x axis.

It turns out that the contour integral of $\sec z$ is bounded by a non-zero constant instead. To show this, we need to pay special attention to the sections of Γ_N within the strip between $y = \pm\frac{1}{2}$. Let $\{A_3, A_4, A_7, A_8\} = \{z = x + iy \in \Gamma_N :\ |y| = \frac{1}{2}\}$. On Γ_N, results similar to C.52 and C.54 can be derived for the modulus of $\cos z$, which are

$$\begin{aligned} |\cos z|^2 = \frac{1}{2}\cosh 2y, \quad &\text{for} \quad |x| = R; \\ |\cos z| \geqslant \sinh R, \quad &\text{for} \quad |y| = R; \end{aligned} \tag{C.62}$$

Case: horizontal sides Similar to estimating the first contour integral when $\mathrm{Im}(z) = \pm R$, we have

$$\left|\int_{A_1}^{A_2} \sec z dz\right| \leqslant \frac{2R}{\sinh R}. \tag{C.63}$$

The same upper bound holds for the integral from A_5 to A_6.

Case: vertical sides ($|\mathbf{Im}(z)| \geqslant \frac{1}{2}$) We use the inequality

$$|\cos z|^2 = \frac{1}{2}\cosh 2y \geqslant \frac{1}{4}e^{2y} \Rightarrow |\sec z| \leqslant 2e^{-|y|}. \tag{C.64}$$

Thus,

$$\left|\int_{A_8}^{A_1} \sec z dz\right| \leqslant \int_{A_8}^{A_1} |\sec z| dy \leqslant \int_{1/2}^{R} 2e^{-y} dy = 2(e^{-\frac{1}{2}} - e^{-R}). \tag{C.65}$$

The same upper bound holds for the integral from A_2 to A_3, from A_4 to A_5 and from A_6 to A_7.

Case: vertical sides ($|\mathbf{Im}(z)| < \frac{1}{2}$) We use a more conservative estimate of $\cos z$,

$$|\cos z|^2 = \frac{1}{2}\cosh 2y \geqslant \frac{1}{2} \Rightarrow |\sec z| \leqslant \sqrt{2}, \tag{C.66}$$

which implies

$$\left|\int_{A_7}^{A_8} \sec z dz\right| \leqslant \sqrt{2} \cdot 1. \tag{C.67}$$

The same upper bound holds for the integral from A_3 to A_4.

Combining inequalities C.64, C.65, C.67, one has

$$\left|\oint_{\Gamma_N} \sec z dz\right| \leqslant 2 \cdot 2 \cdot \frac{2R}{\sinh R} + 4 \cdot 2(e^{-\frac{1}{2}} - e^{-R}) + 2 \cdot \sqrt{2} \cdot 1 \to \text{const} \tag{C.68}$$

as $R \to \infty$. As a result, the first part of the integral C.47 vanishes as well. And we are done.

C.5 More summations

C.5.1 High Order Poles

In applications, it is sometimes required to compute residues at high orders. To do so, it often more convenient to directly compute the Laurent series expansion than using formula C.4. An example is given below.

Evaluate

$$S_k = \sum_{k'=0}^{\infty} \frac{\alpha_{0k'}}{j_1(\alpha_{0k'})} \frac{1}{(\alpha_{1k}^2 - \alpha_{0k'}^2)^3}. \tag{C.69}$$

This infinite series is encountered in the derivation of the first moment of the dephase (Appendix A.3 of [14]).

Solution Plugging $\alpha_{0k'} = (k'+1)\pi$ into equation C.28 yields the result,

$$j_1(\alpha_{0k'}) = \frac{(-1)^{k'}}{\alpha_{0k'}}, \quad k = 0, 1, 2, \ldots. \tag{C.70}$$

The series of interest turns out to be an alternating series involving $\{\alpha_{0k}\}$,

$$S_k = \sum_{k'=0}^{\infty} \frac{(-1)^{k'} \alpha_{0k'}^2}{(a^2 - \alpha_{0k'}^2)^3}, \tag{C.71}$$

where $a = \alpha_{1k}$. Naturally, our plan is to apply formula C.27.

Let $f(z) = \frac{z^2}{(a^2-z^2)^3}$. The residue summation technique calls for the residues of the function,

$$h(z) = \csc(z) f(z) = \frac{\csc(z) z^2}{(a^2 - z^2)^3} \tag{C.72}$$

at the poles of $f(z)$, $z = \pm a$, which are poles of order 3. Obviously, differentiating $(z \pm a)^3 h(z)$ twice can lead to cumbersome expressions. Instead, we aim to use its Laurent series expansion. Introducing a change of variable $t = z - a$, we have

$$h(z) = \frac{\csc(z) z^2}{(a+z)^3 (a-z)^3} = \frac{\csc(t+a)(t+a)^2}{-t^3(t+2a)^3} \equiv g(t). \tag{C.73}$$

The Laurent series of $g(t)$ about $t = 0$ can be obtained by multiplying simpler series. The following taylor expansions are calculated by the software Maple:

- $$\csc(t+a) = b_0 + b_1 t + b_2 t^2 + o(t^3), \tag{C.74}$$
  where $b_0 = \csc a, b_1 = -\cot a \csc a, b_2 = \frac{\csc^3 a}{4}(\cos 2a + 3)$;
- $$\frac{(t+a)^2}{(t+2a)^3} = c_0 + c_1 t + c_2 t^2 + o(t^3), \tag{C.75}$$
  where $c_0 = \frac{1}{8a}, c_1 = \frac{1}{16a^2}, c_2 = -\frac{1}{16a^3}$.

Multiplying the two series as well as the factor $-1/t^3$, we get

$$h(z) = -\frac{1}{t^3}(b_0 + b_1 t + b_2 t^2 + o(t^3))(c_0 + c_1 t + c_2 t^2 + o(t^3)). \tag{C.76}$$

By definition, the desired residue at $z = \frac{a}{\pi}$ is the coefficient of t^{-1} in the series above. A direct computation yeilds

$$\mathrm{Res}(a) = -(b_0c_2 + b_1c_1 + b_2c_0). \tag{C.77}$$

The exact same technique can be used to compute the residue at $z = -a$ and it turns out that both residues are equal,

$$\mathrm{Res}(a) = \mathrm{Res}(-a). \tag{C.78}$$

Applying formula C.27 and noticing that $f(z)$ is even and $f(0) = 0$, we have

$$2S_k = -(\mathrm{Res}(-a) + \mathrm{Res}(a)) = 2(b_0c_2 + b_1c_1 + b_2c_0). \tag{C.79}$$

The closed-form expression of S_k is now obtained. The only work left is to plugging in the b's and c's to simplify the result:

$$\begin{aligned} S_k &= b_0c_2 + b_1c_1 + b_2c_0 \\ &= -\frac{\csc a}{16a^3} - \frac{\cot a \csc a}{16a^2} + \frac{\csc^3 a(\cos(2a) + 3)}{32a} \\ &= \frac{\csc a}{32a}\left(-\frac{2}{a^2} - \frac{2\cot a}{a} + \csc^2 a(4 - 2\sin^2 a)\right). \end{aligned} \tag{C.80}$$

Recall that a satisfies $\tan a = a$. In the long bracket of the RHS, we can replace trigonometric functions by algebraic one by using relations,

$$\cot a = \frac{1}{a}, \quad \csc^2 a = \frac{1}{\sin^2 a} = \frac{1}{a^2} + 1. \tag{C.81}$$

Therefore,

$$S_k = \frac{\csc a}{16a} = (-1)^{k+1}\frac{\sqrt{1 + \alpha_{1k}^2}}{16\alpha_{1k}^2}. \tag{C.82}$$

The alternating sign stems from the fact that the sign of $\sin(\alpha_{1k})$ alternates w.r.t the parity of k: when k's are even, $\sin(\alpha_{1k})$'s are negative; when k's are odd, $\sin(\alpha_{1k})$'s are positive.

The Reside Summation Formula is valid for our problem due to Corollary 5.1 in [19] since $|f(z)| = O(\frac{1}{|k|^4})$ for $|k|$ sufficiently large.

C.5.2 Summation via Partial Fraction

An alternative to the residue summation technique for evaluating certain series involving Laplacian eigenvalues $\{\lambda_k\}_{k\geqslant 0}$ is to use partial fraction decomposition. We assume that the series of interest has the form,

$$S = \sum_k \frac{P(\lambda_k)}{Q(\lambda_k)}, \tag{C.83}$$

in which $P(z)$, $Q(z)$ are polynomials and the general term is of irreducible form. Furthermore, it is assumed that the series converges. Here we explain the recipe and provide one more example arising from moment calculation.

Let the eigenvalues $\lambda_k = \alpha_k^2$, where $\{\alpha_k\}$ are the positive zeros of an even or odd entire function $y(z)$. The key idea is, if the pole expansion of the logarithmic derivative of $y(z)$ is valid then we can derive a basic series involving λ_k's.

$z = 0$ is a zero of $y(z)$ Applying equation C.19, we have

$$\frac{y'(z)}{y(z)} = \sum_k \frac{1}{z+\alpha_k} + \frac{1}{z} + \sum_k \frac{1}{z-\alpha_k} = \frac{1}{z} + \sum_k \frac{2z}{z^2-\alpha_k^2}. \tag{C.84}$$

Replacing z by $\sqrt{z}$, we have the following basic series:

$$\eta(z) = \sum_k \frac{1}{z-\lambda_k} = \frac{y'(\sqrt{z})}{2y(\sqrt{z})\sqrt{z}} - \frac{1}{2z}. \tag{C.85}$$

$z = 0$ is not a zero of $y(z)$, we have

$$\eta(z) = \sum_k \frac{1}{z-\lambda_k} = \frac{y'(\sqrt{z})}{2y(\sqrt{z})\sqrt{z}}. \tag{C.86}$$

If $Q(z)$ can be written as a polynomial of the simple fraction $(z-\lambda_k)^{-1}$ using partial fraction decomposition, $\eta(z)$ and its derivatives

$$\eta^{(n)}(z) = \sum_k \frac{(-1)^n n!}{(z-\lambda_k)^{n+1}} \tag{C.87}$$

can serve as building blocks of the closed-form expression of S. An example is given below.

Evaluate

$$S_k = \sum_{k'=0}^{\infty} \frac{\alpha_{1k'}^2}{(\alpha_{0k}^2 - \alpha_{1k'}^2)^3}. \tag{C.88}$$

This is another series encountered in the derivation of the first moment (Appendix A.3 of [14]).

Solution Denote $z = \alpha_{0k}^2$ and $\lambda_k = \alpha_{1k'}^2$. This is a series involving $\{\alpha_{1k}\}$. We therefore consider the basic series,

$$\begin{aligned} \eta(z) = \sum_k \frac{1}{z - \lambda_k} &= \frac{j_1'(\sqrt{z})}{2j_1(\sqrt{z})\sqrt{z}} - \frac{1}{2z} \\ &= \frac{\sin\sqrt{z}}{2(\sin\sqrt{z} - \sqrt{z}\cos\sqrt{z})} - \frac{3}{2z}. \end{aligned} \tag{C.89}$$

Assuming that the general term admits partial fraction decomposition,

$$\frac{\lambda_k}{(z-\lambda_k)^3} = \frac{A}{z-\lambda_k} + \frac{B}{(z-\lambda_k)^2} + \frac{C}{(z-\lambda_k)^3}, \tag{C.90}$$

the coefficients can solved using standard methods:

$$A = 0, \quad B = -1, \quad C = z. \tag{C.91}$$

Therefore, S_k can be written in closed form:

$$S_k = -\eta'(z) + \frac{z}{2}\eta''(z)\Big|_{z=\alpha_{0k}^2}. \tag{C.92}$$

A tedious but routine calculation yields

$$S_k = -\frac{3}{16\alpha_{0k}^2}. \tag{C.93}$$

Appendix D

Survival Probability under Robin BCs

This appendix demonstrate a probabilistic interpretation of the relaxation rate h. It also provides a justification for the implementation of Robin BCs in section 3.2.

Propagator Consider the partially-reflected Brownian motion on the positive real line, where the random walk can either be absorbed or reflected at $x = 0$. Let the initial position be x_0. The propagator of the RBM $g(x,t)$ solves the following initial boundary value problem (IBVP),

$$\begin{cases} \frac{\partial g}{\partial t}(x,t) &= D\frac{\partial^2 g}{\partial x^2}(x,t) \quad x,t>0 \\ g(x,0) &= \delta(x-x_0) \\ hg(0,t) &= D\frac{\partial g}{\partial x}(0,t) \\ g(\infty,t) &= 0 \end{cases}, \tag{D.1}$$

where $\delta(x)$ is the Dirac delta function and h, D are positive constants. We are interested in the survival probability at time t, defined as

$$P_{x_0}(t) = \int_0^\infty g(x,t)dx, \tag{D.2}$$

which is the probability of the random walker still 'alive' at time t.

To solve the IBVP, we consider taking the Laplace transform of $g(x,t)$ with respect to t,

$$\mathcal{L}_t\{g(x,t)\} = \int_0^\infty g(x,t)e^{-st}dt \equiv G(x,s). \tag{D.3}$$

To simplify notations, we omit the dependence of G on s temporarily. Employ the initial condition of $g(x,t)$ yields the following ODE,

$$\frac{d^2G}{dx^2}(x) - \frac{s}{D}G(x) = -\frac{1}{D}\delta(x-x_0). \tag{D.4}$$

It is well-known that the associated homogeneous ODE has two linearly independent solutions $e^{\pm\sqrt{\frac{s}{D}}x}$. To simplify calculations for our particular problem, we define basis solutions,

$$G_1(x) = \sinh\left(\sqrt{\frac{s}{D}}(x-x_0)\right), \quad G_2(x) = \cosh\left(\sqrt{\frac{s}{D}}(x-x_0)\right), \tag{D.5}$$

which are both linear combinations of $e^{\pm\sqrt{\frac{s}{D}}x}$.

Since the delta inhomogeneous term vanishes whenever $x \neq x_0$, the general solution to the ODE is

$$G(x) = \begin{cases} A_1G_1(x) + B_1G_2(x) & \text{if } x < x_0 \\ A_2G_1(x) + B_2G_2(x) & \text{if } x > x_0 \end{cases}, \tag{D.6}$$

where the constants A_1, A_2, B_1, B_2. To determine them, we first notice that at $x = x_0$, G is continuous and G' experience a jump by $-\frac{1}{D}$ thanks to the delta inhomogeneous term, the following matching conditions hold,

$$\begin{cases} G(x_0^+) & = G(x_0^-) \\ G'(x_0^+) & = G'(x_0^-) - \frac{1}{D} \end{cases}. \tag{D.7}$$

Plugging equation D.6 into D.7 and exploiting these facts,

$$G_1(x_0) = 0, \quad G_2(x_0) = 1, \quad G_1'(x) = \sqrt{\frac{s}{D}}G_2(x), \quad G_2'(x) = \sqrt{\frac{s}{D}}G_1(x), \tag{D.8}$$

one arrives at relations below,

$$\begin{aligned} B_2 &= B_1 \equiv B \\ \sqrt{\frac{s}{D}}A_2 &= \sqrt{\frac{s}{D}}A_1 - \frac{1}{D} \end{aligned} \tag{D.9}$$

Denoting $A \equiv A_2$ and plugging relations D.9 into D.6, $G(x)$ now has only two unknown constants A, B,

$$G(x) = \begin{cases} \left(A + \frac{1}{\sqrt{Ds}}\right) G_1(x) + BG_2(x) & \text{if } x < x_0 \\ AG_1(x) + BG_2(x) & \text{if } x > x_0 \end{cases}. \tag{D.10}$$

On the other hand, we also have boundary conditions at $x = 0, \infty$,

$$\begin{cases} hG(0) &= DG'(0) \\ G(\infty) &= 0 \end{cases}. \tag{D.11}$$

The second equation of D.11 implies

$$A = -B, \tag{D.12}$$

since the positive exponents in $G_1(x)$ and $G_2(x)$ must cancel. Thus, $G(x)$ now only has one unknown constant and D.10 becomes

$$G(x) = \begin{cases} \left(-B + \frac{1}{\sqrt{Ds}}\right) G_1(x) + BG_2(x) & \text{if } x < x_0 \\ B \exp\left[-\sqrt{\frac{s}{D}}(x - x_0)\right] & \text{if } x > x_0 \end{cases}. \tag{D.13}$$

B can now be solved by the Robin BC at $t = 0$, which reads

$$h\left[\left(-B + \frac{1}{\sqrt{Ds}}\right) G_1(0) + BG_2(0)\right] = D\left[\left(-B + \frac{1}{\sqrt{Ds}}\right)\sqrt{\frac{s}{D}} G_2(0) + B\sqrt{\frac{s}{D}} G_1(0)\right]. \tag{D.14}$$

Rearranging D.14 yields

$$B = \frac{\frac{h}{\sqrt{Ds}} \sinh\left(\sqrt{\frac{s}{D}} x_0\right) + \cosh\left(\sqrt{\frac{s}{D}} x_0\right)}{(h + \sqrt{Ds}) \exp\left[\sqrt{\frac{s}{D}} x_0\right]}. \tag{D.15}$$

Therefore, $G(x)$ is now fully solved.

Bring the argument s back into $G(x)$, the solution for $g(x, t)$ can be obtained by taking the inverse Laplace transform of $G(x, s)$,

$$g(x, t) = \mathcal{L}^{-1}\{G(x, s)\}, \tag{D.16}$$

where $G(x, s)$ is give by equations D.13, D.15.

Neumann case The exact expression of $g(x,t)$ for general values of h is intractable. For the special case where $h = 0$ (pure reflection) however, $g(x,t)$ does have a closed form solution. Setting $h = 0$ in equation D.15 yields

$$B|_{h=0} = \frac{\cosh\left(\sqrt{\frac{s}{D}}x_0\right)}{\sqrt{Ds}e^{\sqrt{\frac{s}{D}}x_0}}, \tag{D.17}$$

and $G(x,s)$ becomes

$$\begin{aligned} G(x,s)|_{h=0} &= \begin{cases} \frac{\cosh\left(\sqrt{\frac{s}{D}}x\right)}{\sqrt{Ds}e^{\sqrt{\frac{s}{D}}x_0}} & \text{if } x < x_0 \\ \frac{\cosh\left(\sqrt{\frac{s}{D}}x_0\right)}{\sqrt{Ds}e^{\sqrt{\frac{s}{D}}x}} & \text{if } x > x_0 \end{cases} \\ &= \frac{1}{2\sqrt{Ds}}\left[e^{-\sqrt{\frac{s}{D}}(x+x_0)} + e^{-\sqrt{\frac{s}{D}}|x-x_0|}\right]. \end{aligned} \tag{D.18}$$

Inverse Laplace transform shows that the propagator is a sum of two Gaussian pdfs[1],

$$g(x,t)|_{h=0} = \frac{1}{2\sqrt{D\pi t}}\left[e^{-\frac{(x+x_0)^2}{4Dt}} + e^{-\frac{(x-x_0)^2}{4Dt}}\right], \quad x \geqslant 0. \tag{D.19}$$

The recovery of the solution under Neumann BC convinces us that our solution for $g(x,t)$ is indeed correct, despite the fact that a direct verification is impossible.

Survival probability In particular, we would like to investigate the survival probability when $x_0 = 0$. In such case, $G(x,s)$ becomes

$$G(x,s) = \frac{1}{(h+\sqrt{Ds})}e^{-\sqrt{\frac{s}{D}}x}, \quad \text{for } x > 0. \tag{D.20}$$

The Laplace transform of the desired survival probability is therefore

$$\mathcal{L}_t\{P_0(t)\}(s) = \int_0^\infty G(x,s)dx = \frac{1}{(h+\sqrt{Ds})}\sqrt{\frac{D}{s}}. \tag{D.21}$$

When t is small, the asymptotic expression for $P_0(t)$ can be derived from its Laplace transform when s is large:

$$\mathcal{L}_t\{P_0(t)\}(s) = \frac{1}{s}\frac{1}{1+\frac{h}{\sqrt{Ds}}} \approx s^{-1} - \frac{h}{\sqrt{D}}s^{-\frac{3}{2}}. \tag{D.22}$$

[1]This result is often demonstrated as an example of solving PDEs using the method of 'images'.

Inverse Laplace transform yields

$$P_0(t) \approx 1 - 2h\sqrt{\frac{t}{\pi D}}, \quad \text{for } t \to 0, \tag{D.23}$$

in which the relation $\mathcal{L}^{-1}\left\{2\sqrt{\frac{t}{\pi}}\right\} = s^{-\frac{3}{2}}$ is invoked.

Probabilistic interpretation of h and its significance to simulations Therefore, equation D.23 shows that in the short-time regime, the survival probability at the boundary decreases linearly with respect to the relaxation rate h, which gives the relaxation rate h a probabilistic interpretation.

Moreover, equation D.23 sugguets a possible implementation for simulation of random walk under Robin BCs. By definition, $1 - P_0(t)$ is the termination probability. Given a sufficiently small time step Δt, the random variable U satisfying

$$U \sim \text{Bernoulli}\left(2h\sqrt{\frac{\Delta t}{\pi D}}\right) \tag{D.24}$$

serves as the decision maker on whether or not the current sample path should be discarded.

Other asymptotics of $P_0(t)$ In addition, it is worth noting that survival probability decreases extremely quickly w.r.t to t in the short-time regime (infinity at $t = 0$ in fact). This behaviour agrees with our physical intuition: given that the initial position is at to the boundary, the walker is likely to collide against the boundary multiple times during a short period of time, thus increasing the chance of absorption.

It is in fact possible to obtain an exact expression for $P_0(t)$ (Figure D.1),

$$P_0(t) = \exp\left(\frac{h^2 t}{D}\right)\text{erfc}\left(h\sqrt{\frac{t}{D}}\right), \quad t \geqslant 0, \tag{D.25}$$

where $\text{erfc}(x)$ is the complementary error function. We make several concluding remarks:

- The survival probability under Neumann and Dirichlet boundary conditions are recovered is recovered by setting $h = 0$ and $h \to \infty$ respectively:

$$P_0(t) \equiv \begin{cases} 1 & \text{if } h = 0; \\ 0 & \text{if } h \to \infty \end{cases} \quad t \geqslant 0. \tag{D.26}$$

- Using the asymptotic expression of (x), we obtain the following long-time behaviour of the survival probability:

$$P_0(t) \approx \frac{1}{h}\sqrt{\frac{D}{\pi t}}, \quad \text{for } t \to \infty. \tag{D.27}$$

This implies that the first exit time follows a heavy-tail distribution.

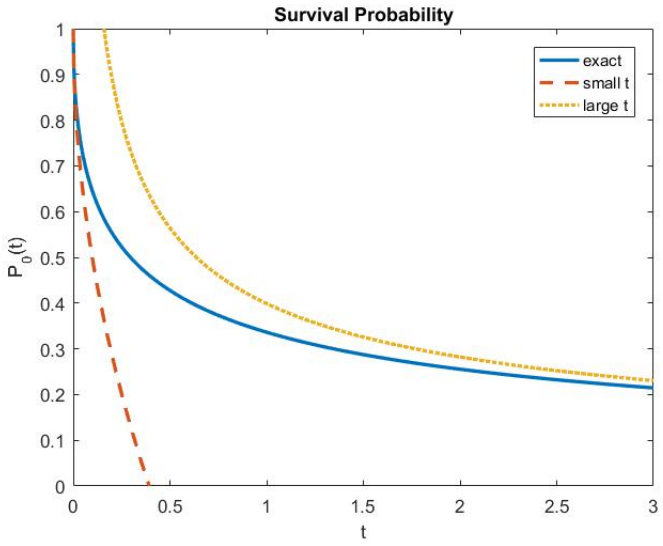


Figure D.1: The graph of $P_0(t)$ (set time unit $\frac{D}{h^2} = 0.5s$).

Appendix E

Mesh-wise Fourier Integral

To form an MR image, the total transverse magnetization of a sample is required (equation 2.22). This is in contrast to diffusion signals which are computed for individual voxels. Let $\{(k_x, k_y)\}$ be a k-space trajectory (recall section 2.3) and Fourier integrals need to be computed for each (k_x, k_y)[1]. To do this, we adopt a strategy inspired by the preprocessing step in finite element methods: first partitioning the sample into meshes and then compute mesh-wise Fourier integrals.

Meshifying 2D domains is a well studied problem in computational geometry and can be solved the Delaunay triangulation algorithm[2]. Let us suppose our sample (the selected slice) $\Omega \subset \mathbb{R}^2$ is partitioned into N triangular meshes, $\{K_j\}_{j=1}^N$ and each mesh has uniform spin density of ρ_j.

We now calculate mesh-wise contributions to the total Fourier integral. Let K be a triangle with vertices $\boldsymbol{q}_i = (x_i, y_i)$ for $i = 1, 2, 3$ and we wish to evaluate

$$S_K(k_x, k_y) = \int_K e^{i(k_x x + k_y y)} dx dy. \tag{E.1}$$

Consider $\hat{K}$, the unit triangle on the ξ, η plane with vertices $(0, 0), (1, 0)$ and $(0, 1)$. We would like to express the integral E.1 as a integral over $\hat{K}$. To do this, we define an affine

[1] Assuming that slice selection is done

[2] Documentation of DT (the MATLAB function for Delaunay triangulation): https://www.mathworks.com/help/matlab/math/delaunay-triangulation.html

transformation T,

$$T : \hat{K} \mapsto K,\ T\left(\begin{bmatrix}\xi\\ \eta\end{bmatrix}\right) = \begin{bmatrix}x_1\\ y_2\end{bmatrix} + \begin{bmatrix}x_2 - x_1 & x_3 - x_1\\ y_2 - y_1 & y_3 - y_1\end{bmatrix}\begin{bmatrix}\xi\\ \eta\end{bmatrix}. \tag{E.2}$$

Introducing vector notations for the wave vector, two sides of K starting from $\boldsymbol{q}_1$, an arbitrary point in $\hat{K}$,

$$\boldsymbol{k} = \begin{bmatrix}k_x\\ k_y\end{bmatrix},\ \boldsymbol{a} = \begin{bmatrix}x_2 - x_1\\ y_2 - y_1\end{bmatrix},\ \boldsymbol{b} = \begin{bmatrix}x_3 - x_1\\ y_3 - y_1\end{bmatrix},\ \boldsymbol{\xi} = \begin{bmatrix}\xi\\ \eta\end{bmatrix}; \tag{E.3}$$

and the Jacobian matrix of T,

$$\boldsymbol{J}_T = [\boldsymbol{a}\ \boldsymbol{b}]. \tag{E.4}$$

The desired Fourier integral E.1 can be written as

$$S_K(\boldsymbol{k}) = \det(\boldsymbol{J}_T) e^{i\langle \boldsymbol{k}, \boldsymbol{q}_1\rangle} I(\alpha, \beta), \tag{E.5}$$

where $\alpha = i\langle \boldsymbol{k}\cdot\boldsymbol{a}\rangle$, $\beta = i\langle \boldsymbol{k}\cdot\boldsymbol{b}\rangle$ and $I(\alpha,\beta)$ is a Fourier integral over $\hat{K}$,

$$I(\alpha,\beta) = \int_{\hat{K}} d\xi d\eta e^{i\langle \boldsymbol{k}, \boldsymbol{J}\boldsymbol{\xi}\rangle} = \int_0^1 d\xi e^{\alpha\xi} \int_0^{1-\xi} d\eta e^{\beta\eta}. \tag{E.6}$$

A direct computation yields

$$I(\alpha,\beta) = \begin{cases} \frac{1}{2}, & \text{if } \beta = \alpha = 0,\\ \frac{e^\alpha - 1}{\alpha^2} - \frac{1}{\alpha}, & \text{if } \beta = 0,\ \alpha \neq 0\\ \frac{1}{\beta}\left(\frac{e^\beta - 1}{\beta} - 1\right), & \text{if } \beta \neq 0,\ \alpha = 0\\ \frac{1}{\beta}\left(\frac{e^\beta(\beta-1)}{\beta} + \frac{1}{\beta}\right), & \text{if } \beta \neq 0,\ \alpha = \beta\\ \frac{1}{\beta}\left(\frac{e^\alpha - e^\beta}{\alpha - \beta} - \frac{e^\alpha - 1}{\alpha}\right), & \beta \neq 0,\ \alpha \neq 0 \text{ and } \alpha \neq \beta \end{cases}. \tag{E.7}$$

In addition, we notice that the Jacobian of T is in fact twice of the signed area of K (denoted as $|K|$),

$$\det(\boldsymbol{J}_T) = \begin{vmatrix} x_1 & y_1 & 1\\ x_2 & y_2 & 1\\ x_3 & y_3 & 1\end{vmatrix} = 2|K|. \tag{E.8}$$

Hence, mesh-wise contributions can be calculated by analytical formulae. Combing all contributions, we obtain the total Fourier integral,

$$S_\Omega(\boldsymbol{k}) = 2\sum_{j=1}^{N} \rho_j |K_j| e^{i\langle \boldsymbol{k}, \boldsymbol{q}_{1,j}\rangle} I(\alpha_j, \beta_j), \tag{E.9}$$

referred to as the discrete imaging equation. To incorporate diffusion effect, each term in E.9 needs to be multiplied by the voxel-wise diffusion signal computed via MC simulations.

Furthermore, if transverse relaxation also needs to be incorporated, mesh-wise Fourier intergals are simply multiplied by their corresponding T_2 decay factors, $e^{-\frac{TE}{T_{2,j}}}$'s.

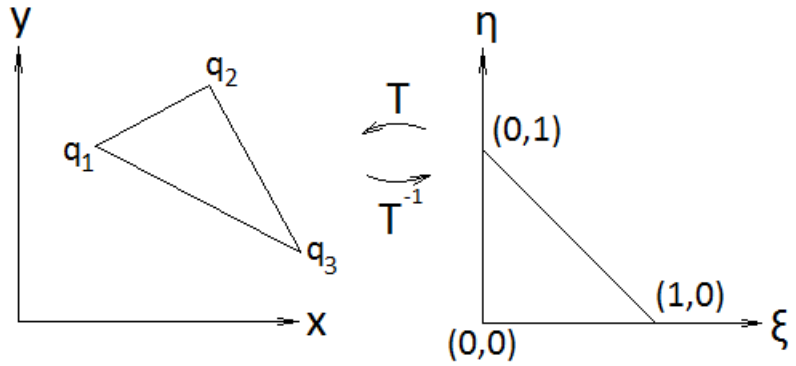


Figure E.1: T maps the unit triangle to a physical 'element'.

Appendix F

Selected MATLAB Codes

The author is aware that there is room for improvement for these codes in terms of optimization, style and documentation and is continuously refining them. All programs in this work will eventually be made public on Github [1].

F.1 Preparation

F.1.1 Loading constants

```
classdef const
    properties (Constant)
        D = 2.3                 % mu_m^2/ms
        gamma = 2.675e5;        % rad/ms/T
    end
end
```

F.1.2 Defining sequences

```
classdef STsequence
    % Diffusion gradient : Stejskal-Tanner Pulsed-Gradient Spin
        Echo
    properties                  % Sequence Parameters
```

[1] https://github.com/hilroy?tab=repositories

```
4           Delta;              % unit = ms, separation of the two
                pulses
5           delta;              % unit = ms, duration of the gradient
6           Ttot;
7           strength;           % typical value ~ e-8T/mu_m
8           direction;          % list of gradient directions, colums
                are unit vector
9           bvalue;             % Stejskal-Tanner formula (without gamma
                )
10      end
11
12      methods
13          function dgradient = STsequence(Delta,delta,strength,
                direction,const)
14              dgradient.Delta     = Delta;
15              dgradient.delta     = delta;
16              dgradient.Ttot      = Delta + delta;
17              dgradient.strength  = strength;
18              dgradient.direction = direction;
19              dgradient.bvalue    = (const.gamma*strength*delta)
                    ^2*(Delta-delta/3);
20          end
21
22          function [seq_discrete, dt, ds] = time_discretize(dgrad,
                N_time,const)
23              % discretized time profile, time step, rms jump size
24              dt = dgrad.Ttot/N_time;
25              ds = sqrt(2*const.D*dt);
26              seq_discrete = zeros(1, N_time);
27              N_d = round(dgrad.delta/dt);
28              N_D = round(dgrad.Delta/dt);
29              seq_discrete(1 : N_d) = 1;
30              seq_discrete(N_D:N_time) = -1;
31          end
32
33          function [dephase, DWsignal] = CalcSignal(Ph,dgrad,const)
34              % calculate diffusion-weighted signal
35              dephase = const.gamma*dgrad.strength*dgrad.direction
```

```
                    '*Ph;
36                  DWsignal = abs(real(mean(exp(1i * dephase),2)));
37              end
38          end
39  end
```

F.2 Random steps generation

```
1   function steps = MakeSteps(ds, dim, N_time, N_walker, type)
2       % ds: rms jump size
3       % dim: dimension of the step vectors (1,2,3);
4       % type: gaussian, uniform, equisized
5       % steps: size = dim * N_time * N_walker
6       % first generate directions
7       N_steps = N_time * N_walker;
8       U = rand(1, N_steps);
9       if dim == 1
10          dir = U;
11          dir(dir < 0.5) = -1;
12          dir(dir >= 0.5) = 1;
13      elseif dim == 2
14          theta = exp(2*1i*pi*U);
15          dir = [real(theta); imag(theta)];
16      elseif dim == 3
17          % mathematically proven in Appendix A
18          Z = U;
19          r_xy = sqrt(1 - Z.^2);
20          U = rand(1, N_steps);
21          XY = r_xy.*exp(2*1i*pi*U);
22          dir = [real(XY); imag(XY); Z];
23      end
24      % then multiply dir by appropriate jump size
25      if strcmp(type, 'equisized') == 1
26          rho = ds * ones(1, N_steps);
27      elseif strcmp(type, 'uniform') == 1
28          U = rand(1, N_steps);
29          rho = sqrt(dim+2) * ds * nthroot(U,dim);
```

```
    elseif strcmp(type, 'gaussian') == 1
        U = chi2rnd(dim,[1 N_steps]);
        rho = ds * sqrt(U);
    end
    steps = bsxfun(@times,dir,rho);
    steps = reshape(steps, [dim N_time N_walker]);
end
```

F.3 Case: free space

```
function Ph = RW_free(steps, dt, seq_dis)
   % free diffusion, no need for collision checking
   % steps : dim * N_time * N_walker
    traj = trapz(steps,2);
    dph = dt * bsxfun(@times,traj,seq_dis);
    Ph = trapz(dph,2);
    Ph = reshape(Ph, [size(steps,1), size(steps,3)]);
end
```

F.4 Case: unit triangle

F.4.1 Main script

```
% define geometry
p1 = [1;0];
p2 = [0;1];
p3 = [0;0];
trig = triangle(p1,p2,p3);

% generate initial positions
N_walker = 10000;

r_ini = unif(trig, N_walker);

% define diffusion gradient
Delta = 20;
delta = 10;
```

```
strength = 4e-4; % this is a bit higher than typical values
direction = [0;1];
dgrad = STsequence(Delta,delta,strength,direction);

% discretization
N_time = 10000;
[dgrad_dis, dt, ds] = time_discretize(dgrad, N_time);

steps = ds * randn(2,N_time,N_walker); % huge!!!

% compute phase gain
tic
Ph_sample = BasicMC(trig, r_ini, steps, dt, dgrad_dis);
toc

% calculate signal
load('PhysicalConstants.mat')
dephase = gamma * dgrad.strength * dgrad.direction' * Ph_sample;
signal = real(mean(exp(1i * dephase)));
histogram(dephase)
```

F.4.2 Domain

```
classdef triangle
    properties
        V1;V2;V3;% vertices, 2 dim
        R1;R2;R3; % Householder reflection matrices for each side
            , 2 by 2
    end

    methods
        function trig = triangle(p1,p2,p3)
            % constructor
            trig.V1 = p1;
            trig.V2 = p2;
            trig.V3 = p3;
            u = (p2-p1)/norm(p2-p1); % unit tangent vector of an
                edge
```

```
14              u = [0 -1;1 0]*u;           % normal vector
15              trig.R3 = eye(2) - 2*u*u';
16              u = (p3-p2)/norm(p3-p2);
17              u = [0 -1;1 0]*u;
18              trig.R1 = eye(2) - 2*u*u';
19              u = (p1-p3)/norm(p1-p3);
20              u = [0 -1;1 0]*u;
21              trig.R2 = eye(2) - 2*u*u';
22          end
23
24          function r_ini = unif(trig, N_walker)
25              % generate initial positions
26              V = [trig.V1 trig.V2 trig.V3];
27              U = log(rand(3, N_walker));
28              C = U./sum(U); % flat Dirichlet distribution
29              r_ini = V*C;
30          end
31
32          function [r_new, ph_new] = update(trig, r, ph, dr, isgrad
                )
33              p1 = trig.V1; p2 = trig.V2; p3 = trig.V3;
34              H1 = trig.R1; H2 = trig.R2; H3 = trig.R3;
35              r = r + dr;
36              [r_new,ph_new] = triangle_refl(p1,p2,p3,H1,H2,H3,r,ph
                    ,isgrad);
37          end
38
39          function Ph_sample = BasicMC(trig, r_ini, steps, dt,
                seq_dis)
40              N = size(r_ini,2);
41              Ph_sample = zeros(2,N);
42              T = size(seq_dis);
43              for n = 1 : N
44                  r = r_ini(:,n);
45                  ph = Ph_sample(:,n);
46                  for t = 1: T
47                      [r, ph] = update(trig, r, ph, steps(:,t,n),
                            seq_dis(t));
```

```
                end
                Ph_sample(:,n) = ph * dt;
            end
        end
    end
end
```

F.4.3 Helper function: Neumann BC

```
function [r_new,ph_new] = triangle_refl(p1,p2,p3,H1,H2,H3,r,ph,
    isgrad)
    % check for interactions against all sides
    A3 = det([r' 1; p1' 1; p2' 1]);
    % A3 = signed area of triangle r p1 p2;
    % A3 < 0 means that r is on the exterior side of p1p2
    A1 = det([r' 1; p2' 1; p3' 1]);
    A2 = det([r' 1; p3' 1; p1' 1]);
    while A3 < 0 || A1 < 0 || A2 < 0
        % keep reflecting r until it reaches the inside
        if A3 < 0
            r = H3*(r - p1) + p1;
        end
        if A1 < 0
            r = H1*(r - p2) + p2;
        end
        if A2 < 0
            r = H2*(r - p3) + p3;
        end
        A3 = det([r' 1; p1' 1; p2' 1]);
        A1 = det([r' 1; p2' 1; p3' 1]);
        A2 = det([r' 1; p3' 1; p1' 1]);
    end
    r_new = r;
    ph_new = ph + (r + r_new)*isgrad/2; % trapezoidal integration
end
```

F.5 Case: single fiber bundle

F.5.1 Helper function: Neumann BC for the interior cylinder

```
function [r_new, ph_new] = FiberBundle_update_in(r,ph,dr,R,
    isgradient)
    % case: current position r is inside cyinder with radius R
    % update spin position and dephase
    r_input = r;
    r_prime = r + dr;

    if norm(r_prime) < R
        % no reflection
        r_new = r_prime;
    else
        % to find the collision point, need to solve a quadratic
            equation
        coeff = [dot(dr,dr) 2*dot(r,dr) dot(r,r) - R^2]; %
            equation (*)
        % take the positive root
        t = roots(coeff);
        t = t (t > 0);
        % overwrite: r = collision point, dr = remaining step (
            reflected)
        r = r + t * dr;
        normal = - r / R;
        Refl = eye(2) - 2 * (normal * normal');
        dr = Refl * ((1 - t) * dr);
        r_prime = r + dr;
        while norm(r_prime) > R   % multiple collision confirmed
            % for subsequent collisions, (*) reduced to a linear
                equation
            t = - 2*dot(r, dr) / dot(dr,dr);
            % repeat previous steps
            r = r + t * dr;
            normal = - r / R;
            Refl = eye(2) - 2 * (normal * normal');
```

```
            dr = Refl * ((1 - t) * dr);
            r_prime = r + dr;
        end
        r_new = r_prime;
    end
    ph_new = ph + (r_input + r_new) * isgradient/2; % phase update
        : trapezoidal rule
end
```

F.6 GAFRW

```
function spin_fin = FastRandomWalk(geometry,spin,N_walker,t_max,
    isgrad,const)
    % Fast Random Walk algorithm, producing the final spin states
    % geometry: an object (methods: dist, unif)
    % spin: array of structures (fields: pos, phi), length =
        N_walker
    % t_max: max duration of a constant pulse
    % isgrad: 1,0,-1;
    % const: tabulated values of moment functions
    spin_fin = spin;
    for n_w = 1 : N_walker
        t_rem = t_max;
        r = spin(n_w).pos;
        ph = spin(n_w).phi;
        while t_rem > 0
            [d,nv] = dist(geometry,r);
            tau = gen_tau(const);
            u = RandDir;
            if d > geometry.d_bd % non-boundary case
                t_unit = d^2/const.D;
                dt = t_unit * tau;
                if  dt < t_rem
                    t_rem = t_rem - dt;      % r_new is on the
                        inscribed sphere
                    r = r + d * u;
                    if isgrad ~= 0               % 'nl' stands for
```

```
                    nonlinear
24                  dph_nl = evalPhi1(const, tau)* d * t_unit
                        * u;
25                  ph = ph + (dt * r + dph_nl) * isgrad;
26              end
27          else
28              dt = t_rem;
29              t_rem = 0;
30              tau = t_rem / t_unit;
31              r = r + d*gen_r(tau,const)*u; % r_new is
                    inside the inscribed sphere
32              if isgrad ~= 0
33                  % dphi_nl ~ N(0,sigma), where sigma is
                        proportional to Phi2
34                  dph_nl = evalPhi2(const, tau)* randn * d *
                        t_unit * ones(3,1);
35                  ph = ph + (dt * r + dph_nl) * isgrad;
36              end
37          end
38      else % boundary case
39          R_rls = geometry.R_rls;
40          t_unit = R_rls^2/const.D;
41          if geometry.iscvx == true(1)
42              while abs(dot(u,nv)) < 1/20 % avoid spins '
                    leaking' out!
43                  u = RandDir;
44              end
45          end
46          % As R_rls << R_in, we assume the walker is always
                able to reach the
47          % release hemisphere.
48          dt = tau * t_unit;
49          t_rem = t_rem - dt;
50          cosangle = dot(u,nv);
51          tgt = u - cosangle * nv;
52          tgt = tgt / norm(tgt);
53          if isgrad ~= 0
54              dph_nl = (nv*tau/4+tgt*evalPhi1(const, tau)) *
```

```
                        R_rls * t_unit;
55                      ph = ph + (dt * r + dph_nl) * isgrad;
56                  end
57                  if cosangle < 0
58                      u = u - 2*cosangle*nv;
59                  end
60                  r = r + u*R_rls;
61              end
62          end
63          spin_fin(n_w).pos = r;
64          spin_fin(n_w).phi = ph;
65          if rem(n_w,10) == 0
66              fprintf('%d out of %d iterations completed.\n', n_w,
                    N_walker)
67          end
68      end
69  end
```

F.6.1 Orthogonal fibres: extra-axonal geometry

```
1   classdef CrossFiber
2       % orthogonally-crossing Fibers
3       properties
4           iscvx;      % is the geometry convex?
5           d_lat;      % halved lattice side length
6           axis;       % 4 axes(see schematics), represented by
                segment end points
7           R_out;      % outer radius
8           VF_out      % volume fraction of the extra-axonal space
9           d_bd;       % boundary layer thickness
10          R_rls;      % release radius
11      end
12
13      methods
14          function lattice = CrossFiber(d, R_out)
15              lattice.iscvx = false(1);
16              lattice.d_lat = d;
17              A = zeros(3,2,4);
```

```
18              A(:,:,1) = d*[1 -1; 1 1; -1 -1];   % x1
19              A(:,:,2) = d*[1 -1; -1 -1; -1 -1]; % x2
20              A(:,:,3) = d*[-1 -1; 1 -1; 1 1];   % y1
21              A(:,:,4) = d*[1 1; 1 -1; 1 1];     % y2
22              lattice.axis    = A;
23              lattice.R_out   = R_out;
24              lattice.VF_out = 1 - (pi/4)*(R_out/d)^2;
25              lattice.d_bd    = R_out * 1e-4;
26              lattice.R_rls   = R_out * 0.05;
27          end
28
29          function [d_out, n_out] = dist(lat, pt)
30              % dist(pt, an extra-axonal point, boundary)
31              % here pt are the local coordinates wrt to the
                    lattice containing
32              % the spin, if dist is 'small', return a nonzero
                    normal vector
33              ax = lat.axis;
34              d  = lat.d_lat;
35              index = floor(((pt/d)+1)/2); % addressing periodicity
36              pt = pt - 2*d*index;
37              sides = pt - reshape(ax,[3,8]);
38              side1 = sides(:,1:2:end-1);
39              side2 = sides(:,2:2:end);
40              DistAll = vecnorm(cross(side1,side2))/(2*d) - lat.
                    R_out;
41              d_out = min(DistAll);
42              n_out = zeros(3,1);   % default value
43              if 0 <= d_out && d_out < lat.d_bd
44                  closest = find(DistAll == d_out);
45                  axdir = diff(ax(:,:,closest),1,2);
46                  side  = pt - ax(:,1,closest);
47                  n_out = side - dot(side,axdir)* axdir/(4*d^2);
48                  n_out = n_out/norm(n_out);
49              end
50          end
51
52          function [r_ini_out, N_out, N_in] = unif(lat, N_walker)
```

```
53                % generate initial positions
54                d = lat.d_lat;
55                N_out = round(N_walker * lat.VF_out);
56                r_ini_out = zeros(3,1,N_out);
57                n_out = 0;
58                while n_out < N_out  % A-R scheme.
59                    r = d * (2*rand(3,1)-1);
60                    [d_out, ~] = dist(lat,r);
61                    if d_out > 0
62                        n_out = n_out + 1;
63                        r_ini_out(:,1,n_out) = r;
64                    end
65                    % expect a lot of unwanted samples ...
66                end
67                N_in = N_walker - N_out;
68            end
69        end
70    end
```