Diffusion MRI simulation of realistic neurons with SpinDoctor and the Neuron Module

Van-Dang Nguyen\textsuperscript{a}, Chengran Fang\textsuperscript{b,c}, Demian Wassermann\textsuperscript{c}, Jing-Rebecca Li\textsuperscript{b,*}

\textsuperscript{a}Department of Computational Science and Technology, KTH Royal Institute of Technology, Sweden
\textsuperscript{b}INRIA Saclay, Equipe DEFI, CMAP, Ecole Polytechnique, 91128 Palaiseau Cedex, France
\textsuperscript{c}INRIA Saclay, Equipe Parietal, 1 Rue Honoré d’Estienne d’Orves, 91120 Palaiseau, France

Abstract

The diffusion MRI signal arising from neurons can be numerically simulated by solving the Bloch-Torrey partial differential equation. In this paper we present the Neuron Module that we implemented within the Matlab-based diffusion MRI simulation toolbox SpinDoctor. SpinDoctor uses finite element discretization and adaptive time integration to solve the Bloch-Torrey partial differential equation for general diffusion-encoding sequences, at multiple b-values and in multiple diffusion directions.

In order to facilitate the diffusion MRI simulation of realistic neurons by the research community, we constructed finite element meshes for a group of 36 pyramidal neurons and a group of 29 spindle neurons whose morphological descriptions were found in the publicly available neuron repository \textit{NeuroMorpho.Org}. These finite elements meshes range from having 15163 nodes to 622553 nodes. We also broke the neurons into the soma and dendrite branches and created finite elements meshes for these cell components. Through the Neuron Module, these neuron and components finite element meshes can be seamlessly coupled with the functionalities of SpinDoctor to provide the diffusion MRI signal attributable to spins inside neurons. We make these meshes and the source code of the Neuron Module available to the public as an open-source package.

To illustrate some potential uses of the Neuron Module, we show numerical examples of the simulated dMRI signals in multiple diffusion directions from whole neurons as well as from the soma and dendrite branches, include a comparison of the high b-value behavior between dendrite branches and whole neurons.

Keywords: Bloch-Torrey equation, diffusion magnetic resonance imaging, finite elements, simulation, neurons.

\textsuperscript{*}Corresponding author
Email address: jingrebecca.li@inria.fr

Preprint submitted to Elsevier October 18, 2019
1. Introduction

Diffusion magnetic resonance imaging is an imaging modality that can be used to probe the tissue micro-structure by encoding the incoherent motion of water molecules with magnetic field gradient pulses [1–3]. Using diffusion MRI to get tissue structural information in the mammalian brain has been the focus of much experimental and modeling work in recent years.

In terms of modeling, the predominant approach up to now has been adding the diffusion MRI signal from simple geometrical components and extracting model parameters of interest. Numerous biophysical models subdivide the tissue into compartments described by spheres (or ellipsoids), cylinders (or sticks), and the extra-cellular space. Such modeling work for the brain white matter can be found in [4–7] and for the gray matter in [8–13]. Some model parameters of interest include axon diameter and orientation, neurite density, dendrite structure, the volume fraction and size distribution of cylinder and sphere components and the effective diffusion coefficient or tensor of the extra-cellular space. More sophisticated mathematical models based on perturbations the intrinsic diffusion coefficient and homogenization can be found in [14, 15] and the references contained therein.

Numerical simulations can help deepen the understanding of the relationship between the cellular structure and the diffusion MRI signal and can play a significant role in the formulation and validation of appropriate models in order to answer the relevant biological questions. Some recent works that included numerical simulations of the diffusion MRI signal as part of model validation includes [16, 17]. They also can be used to investigate the effect of different pulse sequences and tissue features on the measured signal which can be used for the development, testing, and optimization of novel MRI pulse sequences, we refer to, for example, the work in [18–21]. In fact, given the recent availability of vastly powerful computational resources and computer memory, simulation frameworks have been began increasingly to be directly used as the model for tissue parameter estimation [12, 22].

Two main groups of approaches to the numerical simulation of diffusion MRI are 1) using random walkers to mimic the diffusion process in a geometrical configuration; 2) solving the Bloch-Torrey partial differential equation, which describes the evolution of the complex transverse water proton magnetization under the influence of diffusion-encoding magnetic field gradients pulses.

The first group is referred to as Monte-Carlo simulations in the literature and previous works include [12, 23–26]. A GPU-based acceleration of Monte-Carlo simulation was proposed in [27, 28]. Some software packages using this approach include

1. Camino Diffusion MRI Toolkit developed at UCL (http://cmic.cs.ucl.ac.uk/camino/);
2. DIFSIM developed at UC San Diego (http://csci.ucsd.edu/projects/simulation.html);
3. Diffusion Microscopist Simulator [24] developed at Neurospin, CEA.

The works on model formulation and validation for brain tissue diffusion MRI cited above [12, 16–22] all used Monte-Carlo simulations.

The second group of simulations, which up to now has been less often used in diffusion MRI, relies on solving the Bloch-Torrey partial differential equation (PDE) in a geometrical configuration. Numerical methods to solve the Bloch-Torrey equation with arbitrary temporal profiles have been proposed in [29, 32]. The computational domain is discretized either by a Cartesian grid [29, 30, 33] or finite elements [31, 32, 34–36]. The unstructured mesh of a finite element discretization appeared
to be better than a Cartesian grid in both geometry description and signal approximation \[31\]. For time discretization, both explicit and implicit methods have been used. The efficiency of diffusion MRI simulations is also improved by either a high-performance FEM computing framework \[37, 38\] for large-scale simulations on supercomputers or a discretization on manifolds for thin-layer and thin-tube media \[39\]. This simulation group could be seamlessly integrated with Cloud computing resources such as Google Colaboratory notebooks working on a web browser or with Google Cloud Platform with MPI parallelization \[40\]. Our previous work in PDE-based neuron simulations includes the simulation of neuronal dendrites using a tree model \[41\] and (using the techniques we introduce in this work) the demonstration that diffusion MRI signals reflect the cellular organization of cortical gray matter, these signals being sensitive to cell size and the presence of large neurons such as the spindle (von Economo) neurons \[42, 43\].

In a recent paper \[44\], we presented a MATLAB Toolbox called SpinDoctor that is a diffusion MRI simulation pipeline based on solving the Bloch-Torrey PDE using finite elements and an adaptive time stepping method. That first version of SpinDoctor focused on the brain white matter. It provides the following built-in functionalities:

1. placement of non-overlapping spherical cells (with an optional nucleus) of different radii close to each other;
2. placement of non-overlapping cylindrical cells (with an optional myelin layer) of different radii close to each other in a canonical configuration where they are parallel to the \(z\)-axis;
3. inclusion of an extra-cellular space that is enclosed either
   (a) in a tight wrapping around the cells; or
   (b) in a rectangular box;
4. deformation of the canonical configuration by bending and twisting;

and uses the following methods:

1. it generates a good quality surface triangulation of the user specified geometrical configuration by calling built-in MATLAB computational geometry functions;
2. it creates a good quality tetrahedra finite elements mesh from the above surface triangulation by calling Tetgen \[45\], an external package (executable files are included in the Toolbox package);
3. it constructs finite element matrices for linear finite elements on tetrahedra (P1) using routines from \[46\];
4. it adds additional degrees of freedom on the compartment interfaces to allow permeability conditions for the Bloch-Torrey PDE using the formalism in \[47\];
5. it solves the semi-discretized FEM equations by calling built-in MATLAB routines for solving ordinary differential equations.

It was shown in \[44\] that at equivalent accuracy, SpinDoctor simulations of the extra-cellular space in the white matter is 100 times faster than the Monte-Carlo based simulations of Camino (http://cmic.cs.ucl.ac.uk/camino/), and SpinDoctor simulations of a neuronal dendrite tree is 400 times faster than Camino. We refer the reader to \[44\] for the numerical validation of SpinDoctor simulations with regard to membrane permeability as well as extensions to non-standard pulse sequences and the incorporation of transverse relaxation.

In this paper, we present a new module of SpinDoctor called the Neuron Module that enables neuron simulations for a group of 36 pyramidal neurons and a group of 29 spindle neurons whose
morphological descriptions were found in the publicly available neuron repository *NeuroMorpho.Org* [43]. The key to making accurate simulation possible is the use of high quality meshes for the neurons. For this, we used licensed software from ANSA-BETA CEA Systems [49] to correct and improve the quality of the geometrical descriptions of the neurons. After processing, we produced good quality finite elements meshes for the 65 neurons. These finite elements meshes range from having 15163 nodes to 622553 nodes. They are used as input meshes in the Neuron Module, where they can be further refined if required using the built-in option in SpinDoctor. Currently, the simulations in the Neuron Module enforce homogeneous Neumann boundary conditions, meaning the spin exchange across the cell membrane is assumed to be negligible.

A recent direction for facilitating Monte-Carlo simulations is the generation of geometrical meshes that aim to produce ultra-realistic virtual tissues, see [50, 51]. Our work is similar in spirit, with a first step being providing high quality virtual meshes of realistic neurons for finite elements simulations. Through the Neuron Module, the neuron finite element meshes can be seamlessly coupled with the functionalities of SpinDoctor to provide the diffusion MRI signal attributable to spins inside neurons for general diffusion-encoding sequences, at multiple diffusion-encoding gradient amplitudes and directions.

2. Theory

Suppose the user would like to simulate the diffusion MRI signal due to spins inside a neuron, and assume that the spin exchange across the cell membrane is negligible for the requested simulations. Let \( \Omega \) be the 3 dimensional domain that describes the geometry of the neuron of interest and let \( \Gamma = \partial \Omega \) be the neuron cell membrane.

2.1. Bloch-Torrey PDE

In diffusion MRI, a time-varying magnetic field gradient is applied to the tissue to encode water diffusion. Denoting the effective time profile of the diffusion-encoding magnetic field gradient by \( f(t) \), and let the vector \( g \) contain the amplitude and direction information of the magnetic field gradient, the complex transverse water proton magnetization in the rotating frame satisfies the Bloch-Torrey PDE:

\[
\frac{\partial}{\partial t} M(x, t) = -I \gamma f(t) g \cdot x M(x, t) + \nabla \cdot (D_0 \nabla M(x, t)), x \in \Omega,
\]

where \( \gamma = 2.67513 \times 10^8 \text{ rad s}^{-1} \text{T}^{-1} \) is the gyromagnetic ratio of the water proton, \( I \) is the imaginary unit, \( D_0 \) is the intrinsic diffusion coefficient in the neuron compartment \( \Omega \). The magnetization is a function of position \( x \) and time \( t \), and depends on the diffusion gradient vector \( g \) and the time profile \( f(t) \).

Some commonly used time profiles (diffusion-encoding sequences) are:

1. The pulsed-gradient spin echo (PGSE) [2] sequence, with two rectangular pulses of duration \( \delta \), separated by a time interval \( \Delta - \delta \), for which the profile \( f(t) \) is

\[
f(t) = \begin{cases} 
1, & t_1 \leq t \leq t_1 + \delta, \\
-1, & t_1 + \Delta < t \leq t_1 + \Delta + \delta, \\
0, & \text{otherwise},
\end{cases}
\]

(2)
where \( t_1 \) is the starting time of the first gradient pulse with \( t_1 + \Delta > T_E/2 \), \( T_E \) is the echo time at which the signal is measured.

2. The oscillating gradient spin echo (OGSE) sequence \([52, 53]\) was introduced to reach short diffusion times. An OGSE sequence usually consists of two oscillating pulses of duration \( \sigma \), each containing \( n \) periods, hence the frequency is \( \omega = n\frac{2\pi}{\sigma} \), separated by a time interval \( \tau - \sigma \).

For a cosine OGSE, the profile \( f(t) \) is

\[
  f(t) = \begin{cases} 
  \cos \left( n\frac{2\pi}{\sigma} t \right), & t_1 < t \leq t_1 + \sigma, \\
  -\cos \left( n\frac{2\pi}{\sigma}(t - \tau) \right), & \tau + t_1 < t \leq \tau + t_1 + \sigma, \\
  0, & \text{otherwise},
  \end{cases}
\]

(3)

where \( \tau = T_E/2 \).

The PDE needs to be supplemented by interface conditions. For the neuron simulations within the Neuron Module, we assume negligible membrane permeability, meaning zero Neumann boundary conditions:

\[
  D_0 \nabla M(x, t) \cdot n = 0,
\]

where \( n \) is the unit outward pointing normal vector. The PDE also needs initial conditions:

\[
  M(x, 0) = \rho,
\]

where \( \rho \) is the initial spin density.

The diffusion MRI signal is measured at echo time \( t = T_E > \Delta + \delta \) for PGSE and \( T_E > 2D_0 \) for OGSE. This signal is the integral of \( M(x, T_E) \):

\[
  S := \int_{x \in \bigcup(\Omega)} M(x, T_E) \, dx.
\]

(4)

In a dMRI experiment, the pulse sequence (time profile \( f(t) \)) is usually fixed, while \( g \) is varied in amplitude (and possibly also in direction). The signal \( S \) is plotted against a quantity called the b-value. The b-value depends on \( g \) and \( f(t) \) and is defined as

\[
  b(g) = \gamma^2 ||g||^2 \int_0^{T_E} du \left( \int_0^u f(s) ds \right)^2.
\]

For PGSE, the b-value is \([2]\):

\[
  b(g, \delta, \Delta) = \gamma^2 ||g||^2 \delta^2 (\Delta - \delta/3).
\]

(5)

For the cosine OGSE with integer number of periods \( n \) in each of the two durations \( \sigma \), the corresponding b-value is \([29]\):

\[
  b(g, \sigma) = \gamma^2 ||g||^2 \frac{\sigma^3}{4n^2\pi^2} = \gamma^2 ||g||^2 \frac{\sigma^2}{\omega^2}.
\]

(6)

The reason for these definitions is that in a homogeneous medium, the signal attenuation is \( e^{-D_0 b} \), where \( D_0 \) is the intrinsic diffusion coefficient.
3. Method

3.1. Constructing finite element meshes of neurons

In the current version of the Neuron Module, we focus on a group of 36 pyramidal neurons and a group of 29 spindle neurons found in the anterior frontal insula (aFI) and the anterior cingulate cortex (ACC) of the neocortex of the human brain. These neurons constitute, respectively, the most common and the largest neuron types in the human brain [54, 55]. They share some morphological similarities such as having a single soma and dendrites branching on opposite sides. This group consists of 20 neurons for each type in the aFI, as well as 9 spindles and 16 pyramids in the ACC.

We started with the morphological reconstructions (SWC files) published in NeuroMorpho.Org [48], the largest collection of publicly accessible 3D neuronal reconstructions. These surface descriptions of the neurons cannot be used directly by SpinDoctor to generate finite elements meshes since they contain many self-intersections and proximities (see Figure 1 left). We used licensed software from ANSA-BETA CEA Systems [49] to manually correct and improve the quality of the neuron surface descriptions and produced new surface triangulations (see Figure 1 right) that are ready to be used for finite elements mesh generation. The new surface triangulations are passed into the software GMSH [56] to obtain volume tetrahedral meshes.

![Figure 1: Left: a surface description of a pyramidal neuron published in NeuroMorpho.Org [48] which contains many self-intersections and proximities; it cannot be used for finite elements mesh generation. Right: a new surface triangulation that fixes the self-intersections and proximities; it is ready to be used for finite elements mesh generation.](image)

In Figure 2 we summarize the pipeline that takes the SWC format files from NeuroMorpho.Org [48] to the volume tetrahedral meshes in the MSH format that the users of the Neuron Module will take as the input geometrical description to SpinDoctor. This pipeline is provided here for informational purposes, it is not needed to run diffusion MRI simulations in the Neuron Module.
To further study the diffusion MRI signal of neurons, we broke the neurons into disjoint geometrical components: namely, the soma and the dendrite branches. We manually rotated the volume tetrahedral mesh of a whole neuron so that upon visual inspection it lies as much as possible in the $x - y$ plane. In this orientation, we cut the volume tetrahedral mesh into sub-meshes of the soma and the dendrite branches. As an illustration, we show in Figure 3 the spindle neuron $03a_{spindle2aFI}$ volume tetrahedral mesh broken into sub-meshes of the soma and the two dendrite branches.

The complete set of the volume tetrahedral meshes of the whole neurons as well as the corresponding soma and dendrite branches for the 36 pyramidal neurons and the 29 spindle neurons are publicly available at

https://github.com/van-dang/RealNeuronMeshes

The names and sizes of the finite elements meshes of the 65 neurons is listed in Appendix A.
3.2. Diffusion MRI simulations using SpinDoctor and the Neuron Module

SpinDoctor [44] is a MATLAB-based diffusion MRI simulation toolbox. The Neuron Module takes the neuron volume tetrahedral meshes as the descriptions of the neurons and calls SpinDoctor routines to perform dMRI simulations on them. The finite elements mesh package Tetgen [45] contained in the release of SpinDoctor and the Neuron Module is used to refine the input volume tetrahedral meshes, if desired.

Information about running diffusion MRI simulations using the functionalities of SpinDoctor within the Neuron Module can be found in [Appendix B].

The documentation of both SpinDoctor and the Neuron Module are available at [https://github.com/jingrebeccali/SpinDoctor](https://github.com/jingrebeccali/SpinDoctor).

4. Numerical results

4.1. Validation

In this section, we validate our simulations by refining in space (making the finite elements smaller) and refining in time (decreasing the error tolerance of the ODE solver). The accuracy of the simulations was tuned using the following three simulation parameters:

1. $H_{tetgen}$ controls the finite element mesh size;
   (a) $H_{tetgen} = -1$ means the FE mesh size is determined automatically by the internal algorithm of Tetgen to ensure a good quality mesh (subject to the constraint that the radius to edge ratio of tetrahedra is no larger than 2.0).
   (b) $H_{tetgen} = h$ requests a desired FE mesh tetrahedra height of $h$ $\mu$m (in later versions of Tetgen, this parameter has been changed to the desired volume of the tetrahedra).
2. $rtol$ controls the accuracy of the ODE solver. It is the relative residual tolerance at all points of the FE mesh at each time step of the ODE solve;
3. $atol$ controls the accuracy of the ODE solver. It is the absolute residual tolerance at all points of the FE mesh at each time step of the ODE solve;

We ran the following three simulations with the simulation parameters:

1. $rtol = 10^{-2}$, $atol = 10^{-4}$, $H_{tetgen} = -1$;
2. $rtol = 10^{-3}$, $atol = 10^{-6}$, $H_{tetgen} = -1$;
3. $rtol = 10^{-3}$, $atol = 10^{-6}$, $H_{tetgen} = 0.2$; This simulation serves as the reference solution.

The diffusion MRI experimental parameters are the following:

- the diffusion coefficient is $2 \times 10^{-3}$ mm$^2$/s;
- the diffusion-encoding sequence is PGSE ($\delta = 10$ ms, $\Delta = 13$ ms);
- the $b$-values are $b = \{0, 1000, 4000\}$ s/mm$^2$;
- 11 gradient directions were simulated, they are uniformly distributed on the unit sphere;
The simulations were performed on the pyramidal neuron *25o_pyramidal18aFI* (Figure 4).

In Figure 5, we show the signal differences (in percent) between the first two simulations and the third simulation which serves as the reference solution, normalized by the reference signal at $b = 0$:

$$E(b) = \left| \frac{S_{Ref}(b) - S_{Simul}(b)}{S_{Ref}(b = 0)} \right| \times 100. \tag{7}$$

The finite elements mesh for the first two simulations contains 71209 nodes and 300698 elements. The finite elements mesh for reference signal contains 126108 nodes and 534981 elements. We see that the signal difference is less than 2.5% for $b = 4000 \text{s/mm}^2$ and less than 1% for $b = 1000 \text{s/mm}^2$ when the simulation parameters are:

$$rtol = 10^{-2}, atol = 10^{-4}, Htetgen = -1. \tag{8}$$

We will choose the above set of simulation parameters for the later simulations unless otherwise noted.
Figure 5: The signal differences between the reference signal and the simulated signals for the neuron 25a_pyramidal18aFI. The diffusion coefficient is $2 \times 10^{-3}$ mm$^2$/s; the diffusion-encoding sequence is PGSE ($\delta = 10$ms, $\Delta = 13$ms); Simul 1: $rtol = 10^{-2}$, $atol = 10^{-4}$, $Htetgen = -1$; Simul 2: $rtol = 10^{-3}$, $atol = 10^{-6}$, $Htetgen = -1$; Reference signal: $rtol = 10^{-3}$, $atol = 10^{-6}$, $Htetgen = 0.2$; The finite elements mesh for Simul 1 and Simul 2 contains 71209 nodes and 300098 elements. The finite elements mesh for reference signal contains 126108 nodes and 534981 elements.

4.2. High Angular Resolution Diffusion Imaging simulations

4.2.1. Diffusion directions distributed in three dimensions

In Figure 6 we show High Angular Resolution Diffusion Imaging (HARDI) simulation results of the spindle neuron 03a_spindle2aFI, the number of finite elements nodes and elements for the neuron are 38202 and 133422, respectively. The simulations were performed in 301 directions uniformly distributed in the unit sphere. The diffusion MRI signals are obtained by calling the BTPDE_HARDI function in SpinDoctor. We plot (Figure 6) the signal attenuation $|S^{\text{Simul}}(b) - S^{\text{Simul}}(b = 0)|$ in the 301 directions for $b = 1000$ s/mm$^2$ and $b = 4000$ s/mm$^2$. We see the signal attenuation shape in these 301 directions is ellipsoid at the lower b-value and the shape becomes more complicated at the larger b-value. At $b = 4000$ s/mm$^2$, there is more signal attenuation at the shorter diffusion time than at the higher diffusion time.
Figure 6: Top left: the finite elements mesh of the spindle neuron 03a_spindle2aFl, the unit is µm. The number of finite elements nodes and elements for the neuron are 38202 and 133422, respectively. The other three plots: HARDI simulations in 301 diffusion directions. The dots are the end points of the vectors in the diffusion-encoding direction multiplied the signal attenuation. The color gives the magnitude of the signal attenuation. The diffusion coefficient is $2 \times 10^{-3}$ mm$^2$/s. Top right: PGSE ($\delta = 10$ms, $\Delta = 13$ms), $b = 1000$ s/mm$^2$. Bottom left: PGSE ($\delta = 10$ms, $\Delta = 13$ms), $b = 4000$ s/mm$^2$. Bottom right: PGSE ($\delta = 10$ms, $\Delta = 73$ms), $b = 4000$ s/mm$^2$.

4.2.2. Diffusion directions distributed in two dimensions

We generated 90 diffusion directions uniformly distributed on the unit circle lying on the $x-y$ plane and computed the diffusion MRI signals in these 90 directions for three sequences:

- PGSE ($\delta = 2.5$ms, $\Delta = 5$ms);
- PGSE ($\delta = 10$ms, $\Delta = 43$ms);
- PGSE ($\delta = 10$ms, $\Delta = 433$ms);

For the whole neuron and the dendrite branches, the following simulation parameters were used:

$$rtol = 10^{-2}, atol = 10^{-4}, Htetgen = -1.$$  \hspace{1cm} (9)
For the soma, the following simulation parameters were used:

\[ rtol = 10^{-3}, atol = 10^{-6}, Htetgen = 0.1. \] (10)

The requested finite elements size was determined under the condition that the signal is computed to within an accuracy of 5%.

The results for the spindle neuron 03a_spindle2aFI are shown in Figure 7. We plot the non-normalized dMRI signals in the 90 diffusion directions on the \( x - y \) plane. The signals are not normalized by the value at \( b = 0 \) in order to show the influence of the volume fractions of the soma and the dendrite branches. The finite elements mesh of the geometry simulated is superimposed on the plots.

By visual inspection, the signals in the whole neuron are close to the volume weighted sums of the signals from the three geometrical components (the soma, the upper dendrite branch and the lower dendrite branch). It can be seen that the dendrite branch dMRI signals shape is more like an ellipse at \( b = 1000 \text{s/mm}^2 \), whereas at \( b = 4000 \text{s/mm}^2 \) the shape is non-convex. The signal shape of the soma is like ellipse for both \( b \)-values at the 3 diffusion times shown. At the two shorter diffusion times, the signal magnitudes at \( b = 4000 \text{s/mm}^2 \) is much reduced with respect to the magnitude at \( b = 1000 \text{s/mm}^2 \), in contrast to the dendrite branches, where the difference in the signal magnitudes between the two \( b \)-values is not nearly as significant.
Figure 7: The diffusion MRI signals in 90 directions lying on the $x-y$ plane, uniformly distributed on the unit circle. Top left: the whole neuron (finite elements mesh: 38202 nodes and 133422 elements). Top right: the soma (finite elements mesh: 8756 nodes and 44238 elements). Bottom left: the upper dendrite branch (finite elements mesh: 25033 nodes and 96080 elements). Bottom right: the lower dendrite branch (finite elements mesh: 17975 nodes and 70411 elements). The signals are not normalized by the value at $b = 0$ in order to show the influence of the volume fractions of the geometrical components. The signal attenuation is scaled by the volume of each geometrical model. The finite elements mesh of the simulated geometry is scaled so that it fits entirely onto the plots and then the mesh is superimposed on the signal attenuation plots. The volume of the two dendrite branches are 2347 $\mu m^3$ and 1930 $\mu m^3$, that of the soma is 13406 $\mu m^3$, making a total neuron volume of 17684 $\mu m^3$. The geometries come from the spindle neuron 03a_spindle2aFl. The simulated sequences are: Simul 1 = PGSE ($\delta = 2.5 ms, \Delta = 5 ms$), Simul 2 = PGSE ($\delta = 10 ms, \Delta = 43 ms$), Simul 3 = PGSE ($\delta = 10 ms, \Delta = 433 ms$). The b-values are $b = 1000 s/mm^2$ and 4000 $s/mm^2$. The diffusion coefficient is $2 \times 10^{-3} mm^2/s$.

We simulated another spindle neuron, 03b_spindle4aACC, and the diffusion MRI signals in 90 directions are shown in Figure 8. In this example, the signal shapes of the dendrite branches are
qualitatively similar to the previous neuron. For the soma, at the medium diffusion time, there
is not the large reduction in the signal magnitude between $b = 1000\, \text{s/mm}^2$ and $b = 4000\, \text{s/mm}^2$
that occurred for the previous neuron, and at $b = 4000\, \text{s/mm}^2$, the shape is not an ellipse like
for the previous neuron. Because the soma has a coarser discretization than the dendrites and
the whole neuron, the attenuation signal is not as smooth as for the other two types of geometries.
However, we see that at the shortest diffusion time, the large reduction in signal magnitude between
$b = 1000\, \text{s/mm}^2$ and $b = 4000\, \text{s/mm}^2$ does occur, as for the previous neuron.
Figure 8: The diffusion MRI signals in 90 directions lying on the $x - y$ plane, uniformly distributed on the unit circle. Top left: the whole neuron (finite elements mesh: 17370 nodes and 56163 elements). Top right: the soma (finite elements mesh: 4233 nodes and 20801 elements). Bottom left: the upper dendrite branch (finite elements mesh: 14298 nodes and 54089 elements). Bottom right: the lower dendrite branch (finite elements mesh: 5866 nodes and 22300 elements). The signals are not normalized by the value at $b = 0$ in order to show the influence of the volume fractions of the geometrical components. The signal attenuation is scaled by the volume of each geometrical model. The finite elements mesh of the simulated geometry is scaled so that it fits entirely onto the plots and then the mesh is superimposed on the signal attenuation plots. The volume of the two dendrite branches are 557 $\mu m^3$ and 414 $\mu m^3$, that of the soma is 3098 $\mu m^3$, making a total neuron volume of 4070 $\mu m^3$. The geometries comes from the spindle neuron 03b_spindle4aACC. The simulated sequences are: Simul 1 = PGSE ($\delta = 2.5 ms$, $\Delta = 5 ms$), Simul 2 = PGSE ($\delta = 10 ms$, $\Delta = 43 ms$), Simul 3 = PGSE ($\delta = 10 ms$, $\Delta = 433 ms$). The b-values are $b = 1000 s/mm^2$ and 4000 $s/mm^2$. The diffusion coefficient is $2 \times 10^{-3} mm^2/s$.

Finally, in Figure 9 we compare the diffusion MRI signals due to two different dendrite branches, one from 04b_spindle3aFI and one from 03b_spindle7aACC. The first branch has a single main
trunk whereas the second branch divides into two main trunks. We see at the higher b-value $b = 4000 \text{s/mm}^2$, at the longest diffusion time, the signal shape is more elongated (perpendicular to the main trunk direction) for the first dendrite branch than the second.

Figure 9: The diffusion MRI signals in 90 directions lying on the $x-\ y$ plane, uniformly distributed on the unit circle. Left: a dendrite branch from 04b_spindle3aFI, the volume of the dendrite branch is 3700 $\mu$m$^3$, the finite elements mesh contains 46857 nodes and 213338 elements. Right: a dendrite branch from 03b_spindle7aACC, the volume of the dendrite branch is 718 $\mu$m$^3$, the finite elements mesh contains 12389 nodes and 45072 elements. The signals are not normalized by the value at $b = 0$ in order to show the influence of the volumes of the geometrical components. The signal attenuation is scaled by the volume of each geometrical model. The finite elements mesh of the simulated geometry is scaled so that it fits entirely onto the plots and then the mesh is superimposed on the signal attenuation plots. The simulated sequences are: Simul 1 = PGSE ($\delta = 2.5\text{ms}, \Delta = 5\text{ms}$), Simul 2 = PGSE ($\delta = 10\text{ms}, \Delta = 43\text{ms}$), Simul 3 = PGSE ($\delta = 10\text{ms}, \Delta = 433\text{ms}$). The b-values are $b = 1000 \text{s/mm}^2$ and 4000 $\text{s/mm}^2$. The diffusion coefficient is $2 \times 10^{-3}\text{mm}^2/\text{s}$.

4.2.3. High b-value behavior

In [17] it was shown experimentally that the diffusion MRI signal of tubular structures such as axons exhibit a certain high b-value behavior, namely, the diffusion direction averaged signal, $S_{\text{ave}}(b)$, is linear in $\frac{1}{\sqrt{b}}$ at high b-values:

\[
S_{\text{ave}}(b) \equiv \int_{\|u_g\|=1} S_{u_g}(b) du_g \sim c_0 + c_1 \frac{1}{\sqrt{b}}. \tag{11}
\]

Because the dendrites of neurons also have a tubular structure, we test whether the diffusion direction averaged signal, $S_{\text{ave}}(b)$, of dendrite branches also exhibit the above high b-value behavior. We computed $S_{\text{ave}}(b)$ for two whole neurons as well as their associated dendrite branches (each neuron has two dendrite branches), averaged over 31 diffusion directions uniformly distributed in three dimensions. The results are shown in Figure 10. We see clearly the linear relationship between $S_{\text{ave}}(b)$ and $\frac{1}{\sqrt{b}}$ in the four dendrite branches for $b$-values in the range $2500 \text{s/mm}^2 \leq b \leq 20000 \text{s/mm}^2$. In contrast, in the whole neurons, due to the presence of the soma, such linear relationship is not exhibited.
Figure 10: The diffusion direction averaged signal, $S_{\text{ave}}(b)$, as a function of $1/\sqrt{b}$. The left figure is for the neuron 03a_spindle2aFI and the right figure is for the neuron 03b_spindle4aACC. The $S_{\text{ave}}(b)$ is the averaged over 31 diffusion directions, uniformly distributed in three dimensions. In addition, $S_{\text{ave}}(b)$ is normalized so that $S_{\text{ave}}(b = 0) = 1$. The linear relation is seen in the dendrites, but not in the whole neurons. The diffusion-encoding sequence is PGSE ($\delta = 10\text{ms}, \Delta = 43\text{ms}$). The diffusion coefficient is $2 \times 10^{-3} \text{mm}^2/\text{s}$. The b-values are $b = \{60000, 40000, 20000, 12000, 10000, 8000, 7000, 6000, 4000, 2500\} \text{s/mm}^2$.

4.3. Timing

In Figure 11 we give the average computational time per gradient direction to solve the Bloch-Torrey PDE for the examples shown in Figures 7 - 9. All simulations were performed on Intel(R) Xeon(R) E5-2630 v2 CPU @ 2.60 GHz, running Windows 7 Professional. The simulations run from 10 to 90 seconds per gradient direction.
Figure 11: Average computational time to solve the Bloch-Torrey PDE, per gradient direction, for the examples shown in Figures 7 - 9. The timings are the average computational time of 90 directions. The three sequences simulated are PGSE($\delta = 2.5\text{ms}, \Delta = 5\text{ms}$), PGSE($\delta = 10\text{ms}, \Delta = 43\text{ms}$), PGSE($\delta = 10\text{ms}, \Delta = 433\text{ms}$). The diffusion coefficient is $2 \times 10^{-3}\text{mm}^2/\text{s}$.

5. Discussion

In a previous publication [44], SpinDoctor, a MATLAB-based diffusion MRI simulation toolbox, was presented. SpinDoctor allows the easy construction of multiple compartment models of the brain white matter, with the possibility of coupling water diffusion between the geometrical compartments by permeable membranes. In [44], in addition to white matter simulations, SpinDoctor’s ability to import and use externally generated meshes provided by the user was illustrated with a neuronal dendrite branch simulation. This capability is expected to be very useful given the most recent developments in simulating ultra-realistic virtual tissues, typified by recent work such as [50, 51], which was meant to facilitate Monte-Carlo type simulations. In [44], the accuracy and computational time of SpinDoctor simulations were compared with Monte-Carlo simulations using Camino. It was shown that SpinDoctor offered a 400 times speed-up over Monte-Carlo based simulations in the neuronal dendrite tree. To compute the diffusion MRI signal at $8 b$-values: $b = \{0, 100, 500, 1000, 2000, 3000, 6000, 10000\} \text{s/mm}^2$, SpinDoctor took 109 seconds (using 24651 finite elements nodes), compared to 43918 seconds (using 4000 particles) of Camino. All the simulations were performed on a server computer with 12 processors (Intel (R) Xeon (R) E5-2667 @2.90 GHz), 192 GB of RAM, running CentOS 7. SpinDoctor was run using MATLAB R2019a.

In order to enrich the publicly available geometrical meshes that can be used for diffusion MRI simulations, we implemented the Neuron Module inside SpinDoctor. We have created high quality volume tetrahedral meshes for a group of 36 pyramidal neurons and a group of 29 spindle neurons.
In the current paper, we validated the accuracy and gave the computational times of some typical neuron simulations.

We illustrated the usefulness of the Neuron Module in providing simulated HARDI signals at various b-values and diffusion times. In addition to calculating the HARDI signals from whole neurons, we also simulated the HARDI signals of neuron components such as the soma and the dendrite branches. In particular, we compared the high b-value behavior of the diffusion MRI signals from dendrite branches and from whole neurons, confirming the linear relationship between the diffusion direction averaged signal and $\frac{1}{\sqrt{b}}$ for tubular structures.

In summary, we believe our work can add substantially to the understanding of the imaging of neuronal microstructure (neurite density, neurite orientation dispersion, neuronal morphology) [8, 9, 12, 13, 58-60]. Our work sets the stage for a systematic study of the diffusion MRI signal from different types of neurons by the diffusion MRI community (for preliminary results, see [12, 53]). We hope this work contributes to further understanding of the relationship between cell morphology and the resulting diffusion MRI signal and aids in better signal model formulation in the future. If there is sufficient interest from the modeling community, we may add finite elements meshes of other neurons in the future. In Appendix A we list the names and the finite element mesh sizes of the group of 36 pyramidal neurons and the group of 29 spindle neurons and the morphological characteristics of the neurons. A few of the neuron meshes are plotted there.

As this time, we have not implemented the Neuron Module for coupled compartments linked by permeable membranes. Rather, the diffusion MRI signal is computed with zero permeability on the compartment boundaries. The current emphasis of the Neuron Module is to show how the geometrical structure of the neurons affect the diffusion MRI signal. Thus, some of the input parameters related to multiple compartment models in SpinDoctor are not applicable in the current version of the Neuron Module. However, we have kept the exactly same input file formats as SpinDoctor in anticipation of the future development of the Neuron Module for permeable membranes.

In Appendix B we list the expected input files, as well as important functions relevant to the Matrix Formalism Module. Sample output figures are also provided there. The toolbox SpinDoctor and the Neuron Module are publicly available at:

https://github.com/jingrebeccali/SpinDoctor

The complete set of the volume tetrahedral meshes for the group of 36 pyramidal neurons and the group of 29 spindle neurons are publicly available at:

https://github.com/van-dang/RealNeuronMeshes

6. Conclusion

We presented the Neuron Module that we implemented in the Matlab-based diffusion MRI simulation toolbox SpinDoctor. We constructed high quality volume tetrahedral meshes for a group of 36 pyramidal neurons and a group of 29 spindle neurons. Using the Neuron Module, the neuron volume tetrahedral meshes can be seamlessly coupled with the functionalities of SpinDoctor to provide the diffusion MRI signal attributable to spins inside neurons for general diffusion-encoding sequences, at multiple diffusion-encoding gradient amplitudes and directions.
Acknowledgment

Van-Dang Nguyen was supported by the Swedish Energy Agency with the project ID P40435-1 and MSO4SC, the grant number 731063. We would like to thank ANSA from Beta-CAE Systems S. A., for providing an academic license.

Appendix A. Finite elements meshes of whole neurons and neuron components

Table A.1 lists the names and the finite element mesh sizes of the group of 36 pyramidal neurons and the group of 29 spindle neurons. Table A.2 shows the morphological characteristics for the neurons. The neuron models and the measurement data are from [48] and [61].
Table A.1: Names and sizes of all the neuron finite elements meshes generated by Tetgen with default settings ($H_{tetgen} = -1$). The number of FE elements (not shown) is approximately four times the number of FE nodes.

| Neuron ID         | Brain region          | Average diameter ($\mu m$) | Overall height ($\mu m$) | Soma volume ($\mu m^3$) | Total volume ($\mu m^3$) |
|-------------------|-----------------------|-----------------------------|---------------------------|--------------------------|--------------------------|
| 03a_spindle2aFI   | fronto-insula         | 1.74                        | 387.16                    | 13406.27                 | 17684.23                 |
| 03a_spindle6aFI   | fronto-insula         | 2.15                        | 532.30                    | 22557.27                 | 30189.28                 |
| 03b_spindle3aFI   | fronto-insula         | 2.15                        | 532.30                    | 22557.27                 | 30189.28                 |
| 03b_spindle5aFI   | fronto-insula         | 1.62                        | 381.56                    | 22475.52                 | 29804.96                 |
| 03b_spindle6aACC  | anterior cingulate    | 1.58                        | 363.08                    | 9065.56                  | 11579.71                 |
| 03b_spindle9aACC  | anterior cingulate    | 1.27                        | 404.85                    | 19701.05                 | 25639.61                 |
| 04a_pyramidal1aACC| anterior cingulate    | 1.58                        | 363.08                    | 9065.56                  | 11579.71                 |
| 04a_pyramidal1aFI | fronto-insula         | 1.58                        | 363.08                    | 9065.56                  | 11579.71                 |
| 05a_pyramidal1aACC| anterior cingulate    | 1.27                        | 404.85                    | 19701.05                 | 25639.61                 |
| 05a_pyramidal1aFI | fronto-insula         | 1.58                        | 363.08                    | 9065.56                  | 11579.71                 |

21
| 03a_spindle6aFI | fronto-insula | 1.66 | 501.47 | 33458.19 | 37812.72 |
|-----------------|--------------|------|--------|----------|----------|
| 03b_pyramidal2aACC | anterior cingulate | 1.48 | 189.29 | 2977.21 | 4487.44 |
| 03b_pyramidal3aACC | anterior cingulate | 1.14 | 188.45 | 6005.06 | 6891.04 |
| 03b_pyramidal3aFI | fronto-insula | 1.84 | 496.35 | 32510.62 | 46154.08 |
| 03b_pyramidal4aFI | fronto-insula | 1.92 | 430.06 | 15263.14 | 20532.57 |
| 03b_spindle4aACC | anterior cingulate | 1.43 | 363.33 | 3098.39 | 4079.19 |
| 03b_spindle5aACC | anterior cingulate | 1.33 | 398.36 | 496.35 | 6058.67 |
| 03b_spindle6aACC | anterior cingulate | 1.84 | 496.35 | 32510.62 | 46154.08 |
| 03b_spindle7aACC | anterior cingulate | 1.56 | 310.21 | 14718.05 | 21708.21 |
| 04a_pyramidal4aACC | anterior cingulate | 1.43 | 364.21 | 11004.30 | 21722.96 |
| 04b_pyramidal5aACC | anterior cingulate | 1.78 | 814.45 | 9911.07 | 11004.30 |
| 04b_pyramidal5aFI | fronto-insula | 2.18 | 281.02 | 7059.67 | 9911.07 |
| 04b_pyramidal6aACC | anterior cingulate | 1.66 | 361.45 | 32527.06 | 44679.46 |
| 04b_pyramidal7aACC | anterior cingulate | 1.56 | 361.45 | 32527.06 | 44679.46 |
| 04b_spindle3aFI | fronto-insula | 2.71 | 391.14 | 22569.99 | 28404.13 |
| 05a_pyramidal10aACC | anterior cingulate | 1.74 | 281.20 | 11925.78 | 20532.57 |
| 05a_pyramidal8aACC | anterior cingulate | 1.29 | 430.37 | 24709.77 | 29778.79 |
| 05b_pyramidal7aACC | anterior cingulate | 1.66 | 361.45 | 32527.06 | 44679.46 |
| 05b_pyramidal8aACC | fronto-insula | 1.56 | 310.21 | 14718.05 | 21708.21 |
| 05b_spindle5aACC | anterior cingulate | 1.56 | 650.60 | 23948.05 | 40014.54 |
| 05b_spindle5aFI | fronto-insula | 2.35 | 381.88 | 15383.08 | 18190.63 |
| 06a_pyramidal11aACC | anterior cingulate | 1.46 | 437.60 | 12178.08 | 15618.67 |
| 06b_pyramidal10aACC | fronto-insula | 1.92 | 421.68 | 11512.38 | 24230.94 |
| 06b_spindle6aACC | anterior cingulate | 1.92 | 365.18 | 43127.81 | 8738.01 |
| 07a_pyramidal13aACC | anterior cingulate | 1.37 | 325.73 | 6254.53 | 8738.01 |
| 07b_pyramidal14aACC | anterior cingulate | 1.67 | 350.40 | 16053.07 | 22722.96 |
| 07b_spindle9aACC | anterior cingulate | 1.70 | 438.77 | 21344.83 | 27307.48 |
| 08a_spindle15aACC | anterior cingulate | 1.74 | 814.45 | 9911.07 | 11004.30 |
| 08b_pyramidal11aACC | fronto-insula | 1.91 | 421.68 | 11512.38 | 24230.94 |
| 09a_spindle7aACC | fronto-insula | 2.90 | 472.87 | 22052.10 | 52179.53 |
| 09a_spindle8aACC | fronto-insula | 2.05 | 767.33 | 11923.76 | 24230.94 |
| 09b_pyramidal11aACC | anterior cingulate | 1.40 | 341.48 | 8522.11 | 10960.84 |
| 10a_spindle16aACC | anterior cingulate | 1.57 | 457.90 | 5895.17 | 7219.28 |
| 11a_pyramidal15aACC | anterior cingulate | 1.27 | 486.31 | 8807.01 | 12263.84 |
| 11o_pyramidal12aACC | fronto-insula | 1.91 | 369.34 | 70786.62 | 79516.92 |
| 12a_spindle19aACC | anterior cingulate | 2.05 | 431.22 | 12178.08 | 15618.67 |
| 12o_spindle9aACC | fronto-insula | 3.41 | 305.31 | 29983.79 | 36678.18 |
| 13o_spindle10aACC | fronto-insula | 2.69 | 516.92 | 39866.55 | 46022.15 |
| 15o_spindle12aACC | fronto-insula | 3.60 | 604.57 | 53192.65 | 79170.43 |
| 16o_spindle13aACC | fronto-insula | 2.17 | 364.66 | 17467.88 | 18888.13 |
| 17o_pyramidal15aACC | fronto-insula | 1.89 | 340.77 | 11004.30 | 21167.19 |
Table A.2: The morphological characteristics of the neurons.

As an illustration, in Figure A.12 we show the finite elements meshes of 4 spindle neurons and in Figure A.13 we show the finite elements meshes of 4 pyramidal neurons.
Figure A.12: The finite elements meshes of four spindle neurons. The unit is µm. Top left: 03a_spindle6aF1. Top right: 28a_spindle20aF1. Bottom left: 09a_spindle8aF1. Bottom right: 25a_spindle17aF1.
Appendix B. Solution of the Bloch-Torrey PDE with SpinDoctor

SpinDoctor [14] is a MATLAB-based diffusion MRI simulation toolbox. The Neuron Module follows the same workflow as SpinDoctor and builds upon the functionalities of SpinDoctor. To use the Neuron Module, it is necessary to read first the documentation of SpinDoctor. The documentation of both SpinDoctor and the Neuron Module are available at https://github.com/jingrebeccali/SpinDoctor.

The Neuron Module uses the same set of three input files as SpinDoctor. SpinDoctor allows the easy construction of multiple compartment models of the brain white matter, with the possibility of coupling water diffusion between the geometrical compartments by permeable membranes. As this time, we have not implemented the Neuron Module for coupled compartments linked by permeable membranes. Rather, the diffusion MRI signal is computed with zero permeability on the compartment boundaries. The current emphasis of the Neuron Module is to show how the geometrical structure of neurons affect the diffusion MRI signal. Thus, some of the input parameters related to multiple compartment models in SpinDoctor are not applicable in the current version of the Neuron Module. However, we have kept the exactly same input file formats as SpinDoctor in anticipation of the future development of the Neuron Module for permeable membranes. In particular, the various compartments in SpinDoctor have designations as IN, OUT, and ECS, and in the Neuron Module, the geometry defined by the finite element mesh is designated as the OUT compartment.
In this section, we list the input files, as well as important quantities and functions relevant to the Neuron Module, noting where relevant, the input parameters that are not applicable (marked by "na") to the current version of the Neuron Module. Sample output figures are also provided.

In SpinDoctor, there are three input files in which the user specifies the parameters of the desired simulations. They are

1. *params_cells.in*: contains cell parameters
2. *params_domain.in*: contains domain parameters
3. *params_experi.in*: contains experimental parameters

*Read cells parameters* The user provides an input file for the cell parameters, in the format described in Table B.3. To simulate the diffusion MRI signal of a neuron, the user chooses option 3 for the cell shape. The user specifies the name of the neuron to be simulated in line 2.

| Line | Variable name | Example | Explanation |
|------|---------------|---------|-------------|
| 1    | cell_shape    | 3       | 1 = spheres; 2 = cylinders; 3 = neuron; |
| 2    | fname         | 'msh_files/pyramidal/02b_pyramidal1aACC' | file name of neuron mesh |
| 3    | ncell         | 1       | number of cells |
| 4    | Rmin          | na      | min Radius |
| 5    | Rmax          | na      | max Radius |
| 6    | dmin          | na      | min (%) distance between cells |
| 7    | dmax          | na      | max (%) distance between cells |
| 8    | para_deform   | na      | $[\alpha \beta]$; $\alpha$ defines the amount of bend; $\beta$ defines the amount of twist |
| 9    | Hcyl          | na      | height of cylinders |

Table B.3: Input file containing cells parameters. "na" means not applicable to neuron simulation.

*Read simulation domain parameters* The user provides an input file for the simulation domain parameters, in the format described in Table B.4.
| Line | Variable name | Example | Explanation |
|------|---------------|---------|-------------|
| 1    | Rratio        | na      |             |
| 2    | include_ECS  | na      |             |
| 3    | ECS_gap       | na      |             |
| 4    | dcoeff_IN     | na      |             |
| 5    | dcoeff_OUT    | 0.002   | diffusion coefficient in OUT cmpt |
| 6    | dcoeff_ECS    | na      |             |
| 7    | ic_IN         | na      |             |
| 8    | ic_OUT        | 1       | initial spin density in OUT cmpt |
| 9    | ic_ECS        | na      |             |
| 10   | kappa_IN_OUT  | na      |             |
| 11   | kappa_OUT_ECS | na      |             |
| 12   | Htetgen       | -1      | Requested tetgen mesh size; -1 = Use tetgen default; |
| 13   | tetgen_cmd    | 'SRC/TETGEN/tetGen/win64/tetgen' | path to tetgen_cmd |

Table B.4: Input file of simulation domain parameters. "na" means not applicable to neuron simulation.

Read experimental parameters

The user provides an input file for the simulation experimental parameters, in the format described in Table B.5.
### Table B.5: Input file for simulation experiment parameters.

| Line | Variable name | Example | Explanation |
|------|---------------|---------|-------------|
| 1    | ngdir         | 300     | number of gradient direction; if $ngdir > 1$, the gradient directions are distributed uniformly on a sphere; if $ngdir = 1$, take the gradient direction from the line below; |
| 2    | gdir          | 1.0 0.0 0.0 | gradient direction; No need to normalize; |
| 3    | nexpri        | 1       | number of sequences to simulate; |
| 4    | sdeltavec     | 2500 10000 | small delta; |
| 5    | bdeltavec     | 5000 43000 | big delta; |
| 6    | seqvec        | 1 1 1   | diffusion sequence of experiment; 1 = PGSE; 2 = OGSEsin; 3 = OGSEcos; |
| 7    | nperevec      | 0 0 0   | number of period of OGSE; |
| 8    | solve_hadc    | na      | $0 = \text{do not solve HADC}; Otherwise solve HADC; |
| 9    | rtol,         | na      | $[r_{tol} \ a_{tol}]$: relative and absolute tolerance for HADC ODE solver; |
|      | atol,         |         | |
| 10   | solve_btpde   | 1       | $0: \text{do not solve BTPDE}; Otherwise solve BTPDE; |
| 11   | rtol_bt,      | 1e-2 1e-4 | $[r_{tol} \ a_{tol}]$: relative and absolute tolerance for BTPDE ODE solver; |
|      | atol_bt       |         | |
| 12   | nb            | 4       | number of b-values; |
| 13   | blimit        | 0       | $0 = \text{specify } bvec; 1 = \text{specify } [bmin, bmax]; 2 = \text{specify } [gmin, gmax]; |
| 14   | const_q       | 0       | $0: \text{use input } bvalues \text{ for all experiments}; 1: \text{take input } bvalues \text{ for the first experiment and use the same } q \text{ for the remaining experiments} |
| 15   | b-values      | 0 1000 2000 3000 | b-values or $[bmin, bmax]$ or $[gmin, gmax]$; depending on line 13; |

**Important functions of the Neuron Module** In Table B.6 we list important functions of the Neuron Module. For detailed information about them, including argument lists, please read the online documentation.
Table B.6: Some important Neuron Module functions.

| Function name          | Example  | Purpose                                               |
|------------------------|----------|-------------------------------------------------------|
| PLOT_FEMESH            | Figure B.14 | plots the finite elements mesh                        |
| PLOT_PDESOLUTION       | Figure B.15 | plots the PDE solution on the finite elements mesh    |
| BTPDE                  |           | computes the BTPDE signal in one diffusion-encoding direction |
| PLOT_SIGNAL            | Figure B.16 | displays the simulated signal in one diffusion-encoding direction |
| BTPDE_HARDI            |           | computes the BTPDE signal in multiple diffusion-encoding directions |
| PLOT_SIGNAL_HARDI      | Figure B.17 | displays the signal in multiple diffusion-encoding directions |

Example Output Figures from Neuron Module

Below we display figures from the functions PLOT_FEMESH, PLOT_PDESOLUTION, PLOT_SIGNAL, PLOT_SIGNAL_HARDI. The geometrical configuration is the spindle neuron 03b_spindle6aFI. The intrinsic diffusion coefficient is set to $D_0 = 2 \times 10^{-3} \text{mm}^2/\text{s}$, with nonpermeable membrane. We simulated 2 diffusion-encoding sequences:

Experiment 1: $f_1$ is PGSE ($\delta = 10.6\text{ms}, \Delta = 13\text{ms}$),
Experiment 2: $f_2$ is PGSE ($\delta = 10.6\text{ms}, \Delta = 73\text{ms}$).

![Figure B.14: The finite elements mesh of the neuron 03b_spindle6aFI. The unit is $\mu\text{m}$. (Using the command PLOT_FEMESH)](image-url)
Figure B.15: The PDE solution (magnetization) on the neuron 03b_spindle6aFI. The diffusion-encoding sequence is PGSE ($\delta = 10.6\text{ms}, \Delta = 13\text{ms}$). The b-value is $b = 4000\text{s/mm}^2$. (Using the command PLOT_PDESOLUTION)

Figure B.16: The (non-normalized) simulated signal of the neuron 03b_spindle6aFI in one diffusion-encoding direction. For Experi 1, the diffusion-encoding sequence is PGSE ($\delta = 10.6\text{ms}, \Delta = 13\text{ms}$). For Experi 2, the diffusion-encoding sequence is PGSE ($\delta = 10.6\text{ms}, \Delta = 73\text{ms}$). (Using the command PLOT_SIGNAL)
Figure B.17: The simulated signal in 180 diffusion-encoding directions. Left: The signal in 180 directions with PGSE ($\delta = 10.6\text{ms}, \Delta = 13\text{ms}$) and $b=4000\text{ s/mm}^2$. Right: The signal in 180 directions with PGSE ($\delta = 10.6\text{ms}, \Delta = 73\text{ms}$) and $b=4000\text{ s/mm}^2$. (Using the command PLOT_SIGNAL_HARDI)
References

[1] E. L. Hahn, Spin echoes, Phys. Rev. 80 (1950) 580–594.

[2] E. O. Stejskal, J. E. Tanner, Spin diffusion measurements: Spin echoes in the presence of a time-dependent field gradient, The Journal of Chemical Physics 42 (1) (1965) 288–292. doi:10.1063/1.1695690

[3] D. Le Bihan, E. Breton, D. Lallemand, P. Grenier, E. Cabanis, M. Laval-Jeantet, MR imaging of intravoxel incoherent motions: application to diffusion and perfusion in neurologic disorders. Radiology 161 (2) (1986) 401–407. URL http://radiology.rsna.org/content/161/2/401.abstract

[4] Y. Assaf, T. Blumenfeld-Katzir, Y. Yovel, P. J. Basser, Axcaliber: A method for measuring axon diameter distribution from diffusion MRI Magn. Reson. Med. 59 (6) (2008) 1347–1354. URL http://dx.doi.org/10.1002/mrm.21577

[5] D. C. Alexander, P. L. Hubbard, M. G. Hall, E. A. Moore, M. Ptito, G. J. Parker, T. B. Dyrby, Orientationally invariant indices of axon diameter and density from diffusion mri. NeuroImage 52 (4) (2010) 1374–1389. URL http://www.sciencedirect.com/science/article/pii/S1053811910007755

[6] H. Zhang, P. L. Hubbard, G. J. Parker, D. C. Alexander, Axon diameter mapping in the presence of orientation dispersion with diffusion mri. NeuroImage 56 (3) (2011) 1301–1315. URL http://www.sciencedirect.com/science/article/pii/S1053811911001376

[7] E. Panagiotaki, T. Schneider, B. Siow, M. G. Hall, M. F. Lythgoe, D. C. Alexander, Compartment models of the diffusion MR signal in brain white matter: A taxonomy and comparison. NeuroImage 59 (3) (2012) 2241–2254. URL http://www.sciencedirect.com/science/article/pii/S1053811911011566

[8] S. N. Jespersen, C. D. Kroenke, L. Astergaard, J. J. Ackerman, D. A. Yablonskiy, Modeling dendrite density from magnetic resonance diffusion measurements NeuroImage 34 (4) (2007) 1473–1486. URL http://www.sciencedirect.com/science/article/pii/S1053811906010950

[9] H. Zhang, T. Schneider, C. A. Wheeler-Kingshott, D. C. Alexander, Noddi: Practical in vivo neurite orientation dispersion and density imaging of the human brain. NeuroImage 61 (4) (2012) 1000–1016. URL http://www.sciencedirect.com/science/article/pii/S1053811912003539

[10] L. M. Burcaw, E. Fieremans, D. S. Novikov, Mesoscopic structure of neuronal tracts from time-dependent diffusion, NeuroImage 114 (2015) 18 – 37. doi:10.1016/j.neuroimage.2015.03.061

[11] M. Palombo, C. Linneul, J. Valette, Modeling diffusion of intracellular metabolites in the mouse brain up to very high diffusion-weighting: Diffusion in long fibers (almost) accounts for non-monoexponential attenuation, Magnetic Resonance in Medicine 77 (1) (2017) 343–350. doi:10.1002/mrm.26548
[12] M. Palombo, C. Ligneul, C. Najac, J. Le Douce, J. Flament, C. Escartin, P. Hantraye, E. Brouillet, G. Bonvento, J. Valette, New paradigm to assess brain cell morphology by diffusion-weighted MR spectroscopy in vivo, Proceedings of the National Academy of Sciences 113 (24) (2016) 6671–6676. arXiv:http://www.pnas.org/content/113/24/6671.full.pdf
doi:10.1073/pnas.1504327113

[13] B. Lampinen, F. Szczepankiewicz, J. Mårtensson, D. van Westen, P. C. Sundgren, M. Nilsson, Neurite density imaging versus imaging of microscopic anisotropy in diffusion mri: A model comparison using spherical tensor encoding, NeuroImage 147 (2017) 517–531. URL http://www.sciencedirect.com/science/article/pii/S105381191630670X

[14] D. S. Novikov, V. G. Kiselev, Effective medium theory of a diffusion-weighted signal, NMR in Biomedicine 23 (7) (2010) 682–697. doi:10.1002/nbm.1584

[15] L. Ning, E. Özarslan, C.-F. Westin, Y. Rathi, Precise inference and characterization of structural organization (picaso) of tissue from molecular diffusion, NeuroImage 146 (2017) 452 – 473. doi:10.1016/j.neuroimage.2016.09.057

[16] S. N. Jespersen, J. L. Olesen, A. Ianuș, N. Shemesh, Effects of nongaussian diffusion on “isotropic diffusion” measurements: An ex-vivo microimaging and simulation study Journal of Magnetic Resonance 300 (2019) 84–94. URL http://www.sciencedirect.com/science/article/pii/S1090780719300072

[17] J. Veraart, E. Fieremans, D. S. Novikov, On the scaling behavior of water diffusion in human brain white matter, NeuroImage 185 (2019) 379–387. URL http://www.sciencedirect.com/science/article/pii/S1053811918319475

[18] A. Ianuș, D. C. Alexander, I. Drobnjak, Microstructure imaging sequence simulation toolbox, in: S. A. Tsaftaris, A. Gooya, A. F. Frangi, J. L. Prince (Eds.), Simulation and Synthesis in Medical Imaging, Springer International Publishing, Cham, 2016, pp. 34–44.

[19] I. Drobnjak, H. Zhang, M. G. Hall, D. C. Alexander, The matrix formalism for generalised gradients with time-varying orientation in diffusion nmr, Journal of Magnetic Resonance 210 (1) (2011) 151 – 157. doi:10.1016/j.jmr.2011.02.022 URL http://www.sciencedirect.com/science/article/pii/S1090780711000838

[20] M. Mercredi, M. Martin, Toward faster inference of micron-scale axon diameters using monte carlo simulations, Magnetic Resonance in Medicine 31 (4) (2018) 511–530. doi:10.1007/s10334-018-0680-1 URL https://doi.org/10.1007/s10334-018-0680-1

[21] G. Rensonnet, B. Scherrer, S. K. Warfield, B. Macq, M. Taquet, Assessing the validity of the approximation of diffusion-weighted-mri signals from crossing fascicles by sums of signals from single fascicles, Magnetic Resonance in Medicine 79 (4) (2018) 2332–2345. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/mrm.26832 doi:10.1002/mrm.26832 URL https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.26832

[22] G. Rensonnet, B. Scherrer, G. Girard, A. Jankovski, S. K. Warfield, B. Macq, J.-P. Thiran, M. Taquet, Towards microstructure fingerprinting: Estimation of tissue properties from a dictionary of monte carlo diffusion mri simulations, NeuroImage 184 (2019) 964–980. URL http://www.sciencedirect.com/science/article/pii/S1053811918319487
[23] B. D. Hughes, Random walks and random environments, Clarendon Press Oxford : New York, 1995.

[24] C.-H. Yeh, B. Schmitt, D. Le Bihan, J.-R. Li-Schlittgen, C.-P. Lin, C. Poupon, Diffusion microscope simulator: A general monte carlo simulation system for diffusion magnetic resonance imaging, PLoS ONE 8 (10) (2013) e76626. doi:10.1371/journal.pone.0076626

[25] M. Hall, D. Alexander, Convergence and parameter choice for monte-carlo simulations of diffusion mri, Medical Imaging, IEEE Transactions on 28 (9) (2009) 1354 –1364. doi:10.1109/TMI.2009.2015756

[26] G. T. Balls, L. R. Frank, A simulation environment for diffusion weighted mri experiments in complex media, Magn. Reson. Med. 62 (3) (2009) 771–778. URL http://dx.doi.org/10.1002/mrm.22033

[27] K. V. Nguyen, E. H. Garzon, J. Valette, Efficient gpu-based monte-carlo simulation of diffusion in real astrocytes reconstructed from confocal microscopy, Journal of Magnetic Resonance (2018). doi:10.1016/j.jmr.2018.09.013. URL http://www.sciencedirect.com/science/article/pii/S1090780718302386

[28] C. A. Waudby, J. Christodoulou, Gpu accelerated monte carlo simulation of pulsed-field gradient mri experiments, Journal of Magnetic Resonance 211 (1) (2011) 67 – 73. doi:https://doi.org/10.1016/j.jmr.2011.04.004. URL http://www.sciencedirect.com/science/article/pii/S1090780711001376

[29] J. Xu, M. Does, J. Gore, Numerical study of water diffusion in biological tissues using an improved finite difference method, Physics in medicine and biology 52 (7) (Apr. 2007). URL http://view.ncbi.nlm.nih.gov/pubmed/17374905

[30] J.-R. Li, D. Calhoun, C. Poupon, D. L. Bihan, Numerical simulation of diffusion mri signals using an adaptive time-stepping method, Physics in Medicine and Biology 59 (2) (2014) 441. URL http://stacks.iop.org/0031-9155/59/i=2/a=441

[31] D. V. Nguyen, J.-R. Li, D. Grebenkov, D. Le Bihan, A finite elements method to solve the Bloch-Torrey equation applied to diffusion magnetic resonance imaging, Journal of Computational Physics 263 (0) (2014) 283–302. URL http://www.sciencedirect.com/science/article/pii/S002199911400308

[32] L. Beltrachini, Z. A. Taylor, A. F. Frangi, A parametric finite element solution of the generalised bloch–torrey equation for arbitrary domains, Journal of Magnetic Resonance 259 (2015) 126 – 134. doi:https://doi.org/10.1016/j.jmr.2015.08.008. URL http://www.sciencedirect.com/science/article/pii/S1090780715001743

[33] G. Russell, K. D. Harkins, T. W. Secomb, J.-P. Galons, T. P. Trouard, A finite difference method with periodic boundary conditions for simulations of diffusion-weighted magnetic resonance experiments in tissue, Physics in Medicine and Biology 57 (4) (2012) N35. URL http://stacks.iop.org/0031-9155/57/i=4/a=N35

[34] H. Hagslatt, B. Jonsson, M. Nyden, O. Soderman, Predictions of pulsed field gradient NMR echo-decays for molecules diffusing in various restrictive geometries. simulations of diffusion
35

propagators based on a finite element method, Journal of Magnetic Resonance 161 (2) (2003) 138–147. URL http://www.sciencedirect.com/science/article/pii/S1090780702000393

[35] N. Loren, H. Hagslatt, M. Nyden, A.-M. Hermansson, Water mobility in heterogeneous emulsions determined by a new combination of confocal laser scanning microscopy, image analysis, nuclear magnetic resonance diffusometry, and finite element method simulation, The Journal of Chemical Physics 122 (2) (2005) doi:10.1063/1.1830432

[36] B. F. Moroney, T. Stait-Gardner, B. Ghadirian, N. N. Yadav, W. S. Price, Numerical analysis of NMR diffusion measurements in the short gradient pulse limit, Journal of Magnetic Resonance 234 (0) (2013) 165–175. URL http://www.sciencedirect.com/science/article/pii/S1090780713001572

[37] V. D. Nguyen, A FEniCS-HPC framework for multi-compartment Bloch-Torrey models, Vol. 1, 2016, pp. 105–119, QC 20170509. URL https://www.eccomas2016.org/

[38] V.-D. Nguyen, J. Jansson, J. Hoffman, J.-R. Li, A partition of unity finite element method for computational diffusion mri, Journal of Computational Physics (2018). doi:10.1016/j.jcp.2018.08.039 URL http://www.sciencedirect.com/science/article/pii/S0021999118305709

[39] V.-D. Nguyen, J. Jansson, H. T. A. Tran, J. Hoffman, J.-R. Li, Diffusion mri simulation in thin-layer and thin-tube media using a discretization on manifolds, Journal of Magnetic Resonance 299 (2019) 176 – 187. doi:https://doi.org/10.1016/j.jmr.2019.01.002 URL http://www.sciencedirect.com/science/article/pii/S1090780719300023

[40] V.-D. Nguyen, M. Leoni, T. Dancheva, J. Jansson, J. Hoffman, D. Wassermann, J.-R. Li, Portable simulation framework for diffusion mri, Journal of Magnetic Resonance 309 (2019) 106611. doi:https://doi.org/10.1016/j.jmr.2019.106611 URL http://www.sciencedirect.com/science/article/pii/S1090780719302502

[41] D. V. Nguyen, D. Grebenkov, D. L. Bihan, J.-R. Li, Numerical study of a cylinder model of the diffusion mri signal for neuronal dendrite trees, Journal of Magnetic Resonance (0) (2015) doi:10.1016/j.jmr.2015.11.002 URL http://www.sciencedirect.com/science/article/pii/S1090780715000099

[42] D. Wassermann, D. V. Nguyen, G. Gallardo, J.-R. Li, W. Cai, V. Menon, Sensing Von Economo Neurons in the Insula with Multi-shell Diffusion MRI, International Society for Magnetic Resonance in Medicine, poster (2018). URL https://hal.inria.fr/hal-01807704

[43] V. Menon, G. Guillermo, M. A. Pinsk, V.-D. Nguyen, J.-R. Li, W. Cai, D. Wassermann, Quantitative modeling links in vivo microstructural and macrofunctional organization of human and macaque insular cortex, and predicts cognitive control abilities, bioRxiv (2019). arXiv:https://www.biorxiv.org/content/early/2019/06/06/662601.full.pdf doi:10.1101/662601 URL https://www.biorxiv.org/content/early/2019/06/06/662601
[44] J.-R. Li, V.-D. Nguyen, T. N. Tran, J. Valdman, C.-B. Trang, K. V. Nguyen, D. T. S. Vu, H. A. Tran, H. T. A. Tran, T. M. P. Nguyen, Spindoctor: A matlab toolbox for diffusion mri simulation, NeuroImage 202 (2019) 116120. doi:https://doi.org/10.1016/j.neuroimage.2019.116120
URL http://www.sciencedirect.com/science/article/pii/S1053811919307116

[45] H. Si, Tetgen, a delaunay-based quality tetrahedral mesh generator, ACM Trans. Math. Softw. 41 (2) (2015) 11:1–11:36. doi:10.1145/2629697
URL http://doi.acm.org/10.1145/2629697

[46] T. Rahman, J. Valdman, Fast matlab assembly of fem matrices in 2d and 3d: nodal elements, Applied Mathematics and Computation 219 (13) (2013) 7151–7158.

[47] D. V. Nguyen, J.-R. Li, D. Grebenkov, D. L. Bihan, A finite elements method to solve the bloch–torrey equation applied to diffusion magnetic resonance imaging, Journal of Computational Physics 263 (0) (2014) 283 – 302. doi:10.1016/j.jcp.2014.01.009
URL http://www.sciencedirect.com/science/article/pii/S0021999114000308

[48] G. A. Ascoli, D. E. Donohue, M. Halavi, Neuromorpho.org: A central resource for neuronal morphologies, Journal of Neuroscience 27 (35) (2007) 9247–9251. arXiv:http://www.jneurosci.org/content/27/35/9247.full.pdf, doi:10.1523/JNEUROSCI.2055-07.2007
URL http://www.jneurosci.org/content/27/35/9247

[49] BETA CAE Systems, ANSA pre-processor: The advanced CAE pre-processing software for complete model build up., https://www.beta-cae.com

[50] M. Palombo, D. C. Alexander, H. Zhang, A generative model of realistic brain cells with application to numerical simulation of the diffusion-weighted mr signal, NeuroImage 188 (2019) 391 – 402. doi:https://doi.org/10.1016/j.neuroimage.2018.12.025
URL http://www.sciencedirect.com/science/article/pii/S1053811918321694

[51] K. Ginsburger, F. Matuschke, F. Poupon, J.-F. Mangin, M. Axer, C. Poupon, Medusa: A gpu-based tool to create realistic phantoms of the brain microstructure using tiny spheres, NeuroImage 193 (2019) 10 – 24. doi:https://doi.org/10.1016/j.neuroimage.2019.02.055
URL http://www.sciencedirect.com/science/article/pii/S105381191930151X

[52] P. T. Callaghan, J. Stepianik, Frequency-domain analysis of spin motion using modulated-gradient NMR Journal of Magnetic Resonance, Series A 117 (1) (1995) 118–122.
URL http://www.sciencedirect.com/science/article/pii/S1064185885799597

[53] M. D. Does, E. C. Parsons, J. C. Gore, Oscillating gradient measurements of water diffusion in normal and globally ischemic rat brain, Magn. Reson. Med. 49 (2) (2003) 206–215. doi:10.1002/mrm.10385

[54] E. A. Nimchinsky, E. Gilissen, J. M. Allman, D. P. Perl, J. M. Erwin, P. R. Hof, A neuronal morphologic type unique to humans and great apes, Proceedings of the National Academy of Sciences 96 (9) (1999) 5268–5273.

[55] H. C. Evrard, T. Forro, N. K. Logothetis, Von economo neurons in the anterior insula of the macaque monkey, Neuron 74 (3) (2012) 482–489.
[56] C. Geuzaine, J. F. Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, International Journal for Numerical Methods in Engineering (2009).

[57] D. Miyamoto, swc2vtk, https://github.com/DaisukeMiyamoto/swc2vtk.git.

[58] D. S. Novikov, V. G. Kiselev, S. N. Jespersen, On modeling, Magnetic Resonance in Medicine 79 (6) (2018) 3172–3193. doi:10.1002/mrm.27101 URL https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.27101

[59] B. Lampinen, F. Szczepankiewicz, M. Novén, D. van Westen, O. Hansson, E. Englund, J. Märtensson, C.-F. Westin, M. Nilsson, Searching for the neurite density with diffusion mri: Challenges for biophysical modeling Human Brain Mapping 40 (8) (2019) 2529–2545. doi:10.1002/hbm.24542 URL https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.24542

[60] S. N. Jespersen, C. R. Bjarkam, J. R. Nyengaard, M. M. Chakravarty, B. Hansen, T. Vosegaard, L. Astergaard, D. Yablonskiy, N. C. Nielsen, P. Vestergaard-Poulsen, Neurite density from magnetic resonance diffusion measurements at ultrahigh field: Comparison with light microscopy and electron microscopy, Neurolmage 49 (1) (2010) 205–216. URL http://www.sciencedirect.com/science/article/pii/S1053811909009665

[61] K. K. Watson, T. K. Jones, J. M. Allman, Dendritic architecture of the von economo neurons, Neuroscience 141 (3) (2006) 1107–1112.