



HAL
open science

Efficient GPU-based Monte-Carlo simulation of diffusion in real astrocytes reconstructed from confocal microscopy

Khieu-Van Nguyen, Edwin Hernández-Garzón, Julien Valette

► To cite this version:

Khieu-Van Nguyen, Edwin Hernández-Garzón, Julien Valette. Efficient GPU-based Monte-Carlo simulation of diffusion in real astrocytes reconstructed from confocal microscopy. *Journal of Magnetic Resonance*, 2018, 296, pp.188-199. 10.1016/j.jmr.2018.09.013 . cea-02155389

HAL Id: cea-02155389

<https://cea.hal.science/cea-02155389>

Submitted on 13 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Efficient GPU-based Monte-Carlo simulation of diffusion in real astrocytes reconstructed from confocal microscopy

Khieu-Van Nguyen*, Edwin Hernández-Garzón, Julien Valette

Molecular Imaging Research Center (MIRcen), Commissariat l'Energie Atomique, Fontenay-aux-Roses, France



ARTICLE INFO

Article history:

Received 13 April 2018

Revised 14 September 2018

Accepted 27 September 2018

Available online 27 September 2018

Keywords:

Octree structure

Binary marker

GPU

Monte-Carlo simulation

Diffusion-weighted

Nuclear magnetic resonance (NMR)

ABSTRACT

The primary goal of this work is to develop an efficient Monte-Carlo simulation of diffusion-weighted signal in complex cellular structures, such as astrocytes, directly derived from confocal microscopy. In this study, we first use an octree structure for spatial decomposition of surface meshes. Octree structure and radius-search algorithm help to quickly identify the faces that particles can possibly encounter during the next time step, thus speeding up the Monte-Carlo simulation. Furthermore, we propose to use a three-dimensional binary marker to describe the complex cellular structure and optimize the particle trajectory simulation. Finally, a GPU-based version of these two approaches is implemented for more efficient modeling. It is shown that the GPU-based binary marker approach yields unparalleled performance, opening up new possibilities to better understand intracellular diffusion, validate diffusion models, and create dictionaries of intracellular diffusion signatures.

© 2018 Elsevier Inc. All rights reserved.

1. Introduction

Diffusion-weighted nuclear magnetic resonance (DW-NMR), including diffusion-weighted magnetic resonance imaging (DW-MRI) and diffusion-weighted magnetic resonance spectroscopy (DW-MRS), allows measuring the diffusive motion of molecules (water or metabolites, respectively) along the applied magnetic field gradient direction [1].

In biological tissues, the diffusion of molecules is usually restricted (e.g. by cell membranes), hence molecular diffusion is sensitive to underlying tissue microstructure [2,3]. In particular, the mean squared displacement of molecules is smaller than in the case of free diffusion (i.e. smaller than $\sqrt{2dD_0T}$ [4,5], where d is the number of dimensions considered for the displacement, D_0 is the intrinsic diffusion coefficient, and T is the diffusion time).

Linking DW-NMR signal to the underlying microstructure is essential to non-invasively study biological tissues. Numerous authors have been using simple geometrical models (such as spheres, cylinders, for which analytical expressions describing diffusion exist [6]) to estimate relevant structural parameters such as axonal diameter and density based on DW-MRI [7–10] or neuronal and glial fiber diameter based on DW-MRS [11]. However, due to the complexity of realistic brain cells, analytical expressions do not always exist. Instead, numerical study of molecular diffusion

can help to better understand the relationship between diffusion and microstructure. Many numerical methods exist for that purpose, such as solving the Bloch-Torrey equation [12] by using finite difference method [13–16] or finite element method [17,18]. Another method relies on Monte-Carlo techniques [19] to simulate the diffusive motion of molecules [20–24], which was for example recently used to analyze DW-MRS data at long diffusion times and extract long-range brain cell fiber structures [25].

One of the most critical limitations of numerical approaches is that they can be highly time-consuming, in a way that scales with the complexity of the geometry and the diffusion time. Some works already investigate Monte-Carlo simulations in complex models of biological tissue. For example, in [26], due to memory limitation, the authors down-sampled high-resolution confocal images and built the triangular mesh models of green asparagus tissue based on these low-resolution images. Then they explored the effect of mesh resolution on simulated signal to select the resolution minimizing computational time. However, geometry models reconstructed from down-sampled images ignore small microstructural features and thus alter important characteristics such as the surface-to-volume ratio. In [27], the authors used another technique to reduce the computational time, where the simulation domain was split into uniform subvolumes using a 3D grid. Each subvolume contained a subset of surface meshes and diffusing particles. Therefore, each diffusing particle needs to test the interaction with the surface elements of the local subvolume rather than with all surface elements. Like in [27], a polygon cache

* Corresponding author.

E-mail address: van.nguyen@cea.fr (K.-V. Nguyen).

was used in [28] to speed up the estimation of the closest surface elements likely to interact with particles. In addition, the simulation times can be reduced by efficient parallelization on a CPU cluster. However, it might be difficult for many research groups to access large CPU clusters able to reduce computation times by several orders of magnitudes.

In this paper, we propose two methods for accelerating Monte-Carlo simulations of diffusion. First, we use the octree structure [29,30] to manage and accelerate the interaction of particles with the surface. Octree structure and fixed radius search algorithms allow quickly identifying triangular surface elements that particles can possibly encounter (and undergo specular reflection on) during the next time step. In the second method, instead of conventional mesh-based structure representation, we use a 3D binary marker to describe the complex cellular structure. This new approach is expected to be much more efficient for handling collision with membranes, as only one equality test is required. In this latter method, the simple rejection method is used, like in some previous works [21,24,31]. GPU-based versions are finally implemented for acceleration of both methods.

2. Methods

2.1. Generation of astrocytic binary markers and astrocyte's surface triangular meshes

Brain slices from mice perfused intracardially with 4% paraformaldehyde were used to acquire images of GFP expressing astrocytes. Briefly, after perfusion, the brain was sliced on a Leica razor vibratome, with 100- μm thickness. The slices were mounted in Vectashield mounting medium for one day. The slices were then scanned using a Leica SP8 confocal system equipped with a broadband white-light laser with an oil immersion objective 60x, 1.4 numerical aperture (NA). The white laser power was settled to 70%, the acquisition speed was 600 Hz, and the final image was the result of an average of 4 laser scans. The confocal image resolution is approximately $0.074 \times 0.074 \times 0.299 \mu\text{m}^3$. One cortical protoplasmic astrocyte and one striatal fibrous astrocyte were selected. Subvolume image of single astrocytes were cropped from the confocal microscopy images. These volume images were used for generating binary marker matrix based on the signal intensity with dual threshold (lower and upper) level and an eighteen-connectivity neighborhood. This segmentation method, called hysteresis [32], was implemented in Matlab (MathWorks Inc., Natick, MA, United States). In the end, the structure is described by a 3D matrix filled with 1 and 0, where 1 indicates that corresponding voxel is inside the cell, while 0 indicates a voxel outside the cell.

The volume binary image above was also used for generating the triangular surface mesh by using an open-source meshing toolbox "iso2Mesh" [33] adapted with the *cgalmesh* [34] library via volume-to mesh (*vol2mesh*) procedure. The typical astrocytic surface consists of $5 \times 10^5 - 1 \times 10^6$ surface elements.

2.2. Octree structure: spatial representation of astrocyte's surface triangular mesh

To manage and accelerate the interaction of particles with astrocytic surface, we used an octree structure [29,30] to identify faces that particles can encounter during the next time step. Octree is a hierarchical data structure based on a recursive spatial space decomposition of a 3D data. The root node is represented by a cube (or rectangle box) containing whole data. Each node comprising more than 32 data points (or some other value, depending on data size) is divided into eight child-nodes. The circumcenter points and the circumradii describe the astrocyte's surface triangular mesh.

The circumcenter points are used as data to build the octree structure, and the circumradii information are used in the octree radius search algorithm. More detail about the octree structure and relevant information can be found on [30,35,36] and references therein.

2.3. Binary matrix representation: conversion between voxel and real world coordinates

The transformation from a position (i, j, k) in voxel coordinates (volume image space, i.e. indices in the 3D binary matrix) to real world coordinates (anatomical space) (x, y, z) is an affine transformation defined by

$$x = i \times r_x + x_0,$$

$$y = j \times r_y + y_0,$$

$$z = k \times r_z + z_0,$$

where r_x , r_y and r_z represents the image resolutions; (x_0, y_0, z_0) is the voxel origin's position in real world space. For simplicity, we can use $(x_0, y_0, z_0) = (0, 0, 0)$. The inverse transformation from a real world coordinates system position (x, y, z) to a voxel coordinates system position (i, j, k) can be determined by

$$\begin{aligned} i &= \text{round}\left(\frac{x}{r_x}\right), \\ j &= \text{round}\left(\frac{y}{r_y}\right), \\ k &= \text{round}\left(\frac{z}{r_z}\right). \end{aligned} \quad (1)$$

2.4. Simulating DW-NMR signal in astrocytes

Throughout this paper, we focus on the DW-NMR signal from isolated astrocytes by ignoring molecular diffusion in the extracellular space. To compute that signal, we are using three-dimensional Monte-Carlo simulation, as performed in previous works [20–22,24,31,37]. The algorithm for generating trajectories and the GPU implementation can be found in more detail in [24]. For the handling of the restriction during the particle random walking, we are describing two different methods.

2.4.1. Method 1: in the octree structure representation

A set of N particles is initialized randomly inside the astrocyte using point in polyhedron algorithm [38,39]. Monte-Carlo simulation of the diffusion is performed very similar to the Camino simulator [22], i.e. at each time step each particle randomly moves over a constant distance $\sigma = \sqrt{6D_0d_t}$ along one random direction, where d_t is the time-step size. In case of interaction with a surface element, the particle undergoes multiple specular reflections until no further surface intersections are detected. The main difference with Camino is the use of octree structure and radius search algorithm (and eventually GPU implementation) for acceleration.

2.4.2. Method 2: in the 3D binary markers representation

Here we focus on the use of binary markers to accelerate the Monte-Carlo simulation. First, a set of N particles is initialized randomly inside the astrocyte. While the position of each particle is randomly generated in real world coordinates system, it is converted to voxel coordinates (voxel index) according to Eq. (1), which then allows testing if the particle is inside the astrocyte (i.e. the value of astrocytes' binary marker for this voxel index being equal to 1). If not, a new initial position is drawn. Then, for each time step, the particle randomly moves over a constant distance $\sigma = \sqrt{6D_0d_t}$ along one random direction, where d_t is the

time-step size as mentioned before. Again, the real world coordinates position of the particle is transformed to voxel coordinates and is then simply tested against astrocyte's binary marker to determine whether the new position is accepted (particle still inside the astrocyte) or rejected (particle is outside the astrocyte). If the position is rejected, then the particle's movement during this time step is canceled. This "rejection sampling" method was used in previous works, such as [21,24,31]. As mentioned in these works, the moving step size σ must be taken much smaller than the smallest geometric features. The principle of the Monte-Carlo simulation in a binary marker is illustrated in 2D in Fig. 1.

In both methods, the DW-NMR signal was computed using the phase accumulation approach. Briefly, the echo signal is calculated according to

$$S = \frac{1}{N} \sum_{j=1}^N e^{i\gamma\theta_j} \quad (2)$$

where $i = \sqrt{-1}$, γ is the gyromagnetic ratio, and

$$\theta_j = \sum_{k=1}^{N_t} (G_x^k x_j^k + G_y^k y_j^k + G_z^k z_j^k) d_t \quad (3)$$

where N_t is the number of time steps, (G_x^k, G_y^k, G_z^k) and (x_j^k, y_j^k, z_j^k) are the gradient intensity and the position of the particle j^{th} at the time-step t_k , respectively.

Particle trajectories can be simulated *de novo* for each diffusion parameters set of gradient duration δ , gradient separation Δ , and b-value. Alternatively, we can generate particle trajectories for a given random walk duration $(\delta + \Delta)$ and then use these trajectories to simulate DW-NMR signals for multiple b-values obtained by varying gradient strength [23], as we did for this study. This is one advantage of the Monte-Carlo simulation method in compar-

ison to the other methods, such as finite element or finite difference methods by solving the Bloch-Torrey equation.

All codes were implemented in C++ using the CUDA v8.0 library to interface with NVIDIA GPUs (Tesla K40c) and performed on an HP workstation (Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20 GHz, 16 GB DDR4 RAM) on Windows 7 professional. The pure CPU version and GPU version of the code were both implemented for comparison purposes.

2.5. Validation and performance of synthetic DW-NMR signal with simple geometries

The analytical expressions for diffusion in simple geometries like parallel planes, cylinders, and spheres are known [40]. In this work, a box (dimension of $L_x \times L_y \times L_z = 4.02 \mu\text{m} \times 9.02 \mu\text{m} \times 14.02 \mu\text{m}$) is used for initial validation and comparison of computational times between CPU and GPU versions. Fig. 2 presents the triangular mesh of the box (240 surface elements) used in this study. This box was also voxelized into the 3D binary marker representation with the resolution of $0.035 \mu\text{m}^3$ and used as a geometry input for method 2. Moreover, we also simulate the DW-NMR signal with the same parameters using Camino toolkit (<http://camino.cs.ucl.ac.uk>) for reference in terms of computation time. More precisely, particle trajectories were simulated using Camino, and the DW-NMR signals were then computed in Matlab for multiple b-values based on these trajectories.

Besides, we consider finite-length cylinders (radius $R = 2 \mu\text{m}$ and length $L = 20 \mu\text{m}$) to validate the numerically simulated DW-NMR signal in the case of gradient directions perpendicular and parallel to cylinder direction.

Thirty randomly oriented cylinders were used to validate and test the simulation, with gradient direction rotated accordingly always to simulate signal along parallel and perpendicular directions (see Fig. 3a). The rationale for doing this was to account for possible errors due to too coarse discretization of the surface for some oblique orientations when using the binary matrix representation.

Each of these cylinders was voxelized with five different isotropic spatial resolutions of 0.28^3 , 0.21^3 , 0.14^3 , 0.07^3 , $0.035^3 \mu\text{m}^3$ that

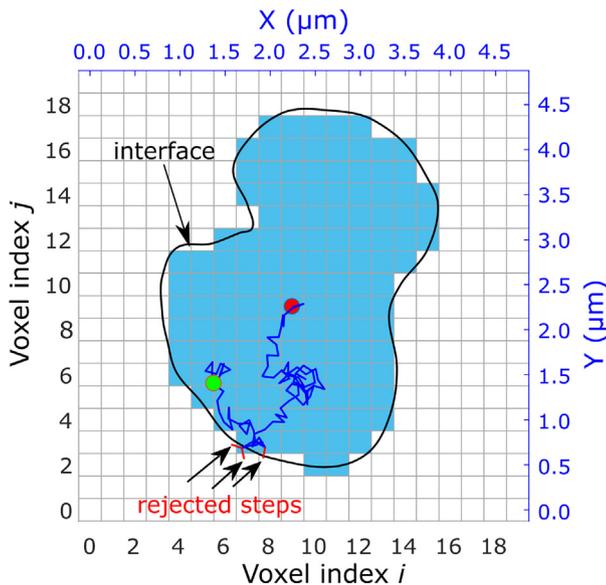


Fig. 1. Example of one particle's trajectory in 2D using binary marker representation. The cell (interface represented as a smooth black line) is voxelized, with the cyan region standing for the intracellular space and the white region standing for extracellular space. The particle starts at the red point $((2.25 \mu\text{m}, 2.25 \mu\text{m})$ in real world coordinates system, corresponding to $(9, 9)$ in voxel coordinates system), and then randomly moves over a constant distance $\sigma = 0.125 \mu\text{m}$. The blue color segments represent accepted steps. The red color segments represent rejected steps. The particle stops after 100 steps (green point). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

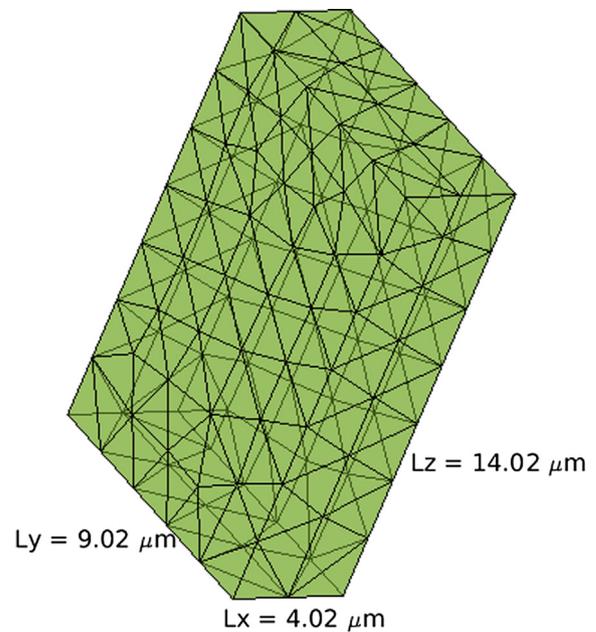


Fig. 2. The triangular mesh of the box dimension $L_x \times L_y \times L_z = 4.02 \mu\text{m} \times 9.02 \mu\text{m} \times 14.02 \mu\text{m}$. The number of surface elements is 240.

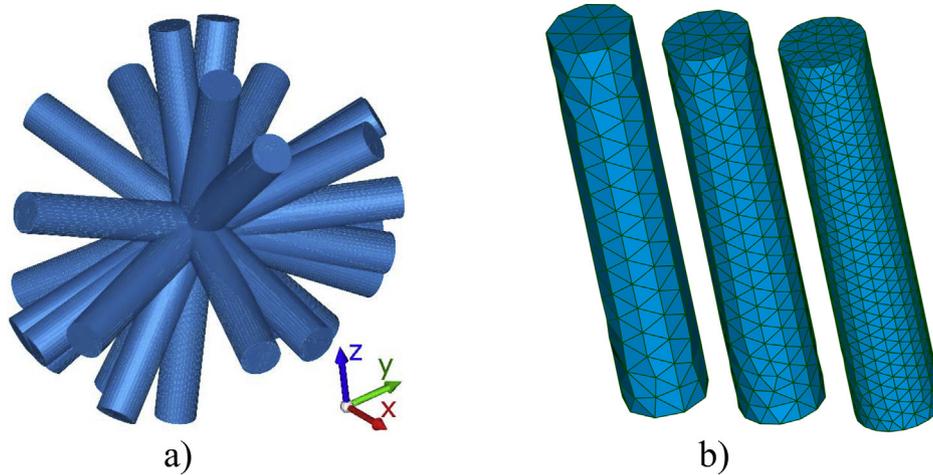


Fig. 3. (a) Thirty randomly oriented finite-length cylinders with radius $R = 2 \mu\text{m}$ and length $L = 20 \mu\text{m}$ used to validate and test the simulation. (b) Three different cylinder meshes used for Camino simulation. From left to right, the number of surface elements is 344, 630 and 1292, respectively.

were named “resolution-1” to “resolution-5”, respectively. The CPU-based and GPU-based Monte-Carlo simulations were performed on each resolution for the thirty cylinders.

Moreover, one oblique cylinder was used to generate the surface triangular mesh to simulate DW-NMR signal using the Octree approach as well as Camino for reference. Due to the dependency of the simulation time and accuracy on the number of surface elements, three different meshes were created for simulation (see Fig. 3b).

The relative difference l^2 -norm (RE) determined by

$$RE = \frac{\sqrt{\sum_{b=0}^{60} (S_{\text{simul}}(b) - S_{\text{analytic}}(b))^2}}{\sqrt{\sum_{b=0}^{60} (S_{\text{analytic}}(b))^2}} \quad (4)$$

was used to estimate the relative errors between the simulated signal S_{simul} and the analytic signal S_{analytic} , over 61 b -values which were increased linearly from 0 to $60 \text{ ms}/\mu\text{m}^2$. The mean and standard deviation of RE, and the simulation time over 30 cylinders (the same spatial resolution) were used to quantify the accuracy and performance of the proposed simulation method (both CPU and GPU versions).

3. Results

3.1. Validation of simulated DW-NMR signal in simple geometries

For the validation purposes, the intrinsic diffusion coefficient $D_0 = 0.5 \mu\text{m}^2/\text{ms}$, gradient duration $\delta = 0 \text{ ms}$ and gradient separation $\Delta = [50; 250] \text{ ms}$ were chosen for simulation. We follow the recommendation of Ford and Hackney in [21] and Waudby and Christodoulou in [24] about the selection of time-step size considering the smallest structural features. We also follow the recommendation of Hall and Alexander in [22] that the simulation complexity (product of the number of particles by the number of time steps) should be more than 10^8 – 10^9 to get a good approximation of simulated DW-NMR signal. In the following, we demonstrate the validity of the two proposed accelerated methods. Let *Octree + CPU/GPU* denotes the simulation method that uses octree structure representation of traditional triangular mesh, and *Bin3D + CPU/GPU* denotes the simulation method that uses 3D binary marker representation. The suffix *CPU* or *GPU* means the simulation performed on CPU or GPU respectively.

3.1.1. Case study 1: Simulated DW-NMR signal in the box

We set the number of particles to $N = 2^{18}$ and the time-step size $d_t = 0.05 \text{ ms}$ for $\Delta = 50 \text{ ms}$ and $d_t = 0.25 \text{ ms}$ for $\Delta = 250 \text{ ms}$ (corresponding to a fixed number of time steps to $N_t = 1000$ for Camino simulation used as reference for computational time) for both CPU and GPU Monte-Carlo simulation, resulting in a complexity larger than 10^8 . Fig. 4 shows the simulated and analytical DW-NMR signals for gradient directions aligned with box axes. As shown in Fig. 4, the simulated DW-NMR signals generated by *Octree + CPU/GPU* method and *Bin3D + CPU/GPU* method are in good agreement with the analytical solutions.

3.1.2. Case study 2: Looking for a potential bias due to discretization of oblique surfaces when using binary marker representation

In this case, we set the number of particles to $N = 2^{17}$ and time-step size to $d_t = 0.005 \text{ ms}$, resulting in a complexity larger than 10^9 .

The mean and standard deviation of the log of simulated DW-NMR signal (over 30 randomly oriented cylinders) at $\Delta = 50 \text{ ms}$ and $\Delta = 250 \text{ ms}$ for the five different spatial resolutions is shown in Fig. 5. The RE was calculated according to Eq. (4) for the simulated DW-NMR signals along gradient directions perpendicular and parallel to cylinder axes. The mean and standard deviation of RE over 30 randomly oriented cylinders under the same spatial resolution was calculated and is shown in Table 1.

As shown in Fig. 5, the DW-NMR signals obtained by *Bin3D + CPU* and *Bin3D + GPU* Monte-Carlo simulations are the same. In addition, given a fixed number of particles and time-step size, the highest spatial resolution yields simulated DW-NMR signal closest to the analytical signal, and reaches the smallest standard deviation. More specifically, for the gradient direction perpendicular to cylinder principal axes, the simulated DW-NMR signals for the five spatial resolutions are all very close to the analytic approximate DW-NMR signal (the REs were smaller than 1% as shown in Table 1). However, for the gradient direction parallel to cylinder principal axis, the lower spatial resolution ($0.28^3 \mu\text{m}^3$) gets substantial errors at high b -values; the higher spatial resolutions, from 0.21^3 to $0.035^3 \mu\text{m}^3$, obtained better approximations of the analytical signal as the RE were smaller than 2%, as shown in Table 1. In addition, by increasing the number of particles to $N = 2^{20}$, which remains perfectly manageable using GPU (see Section 3.2 below), while maintaining the time-step size $d_t = 0.005 \text{ ms}$, we can decrease the standard deviation of the simulated signals as shown in Fig. 6 and Table 1.

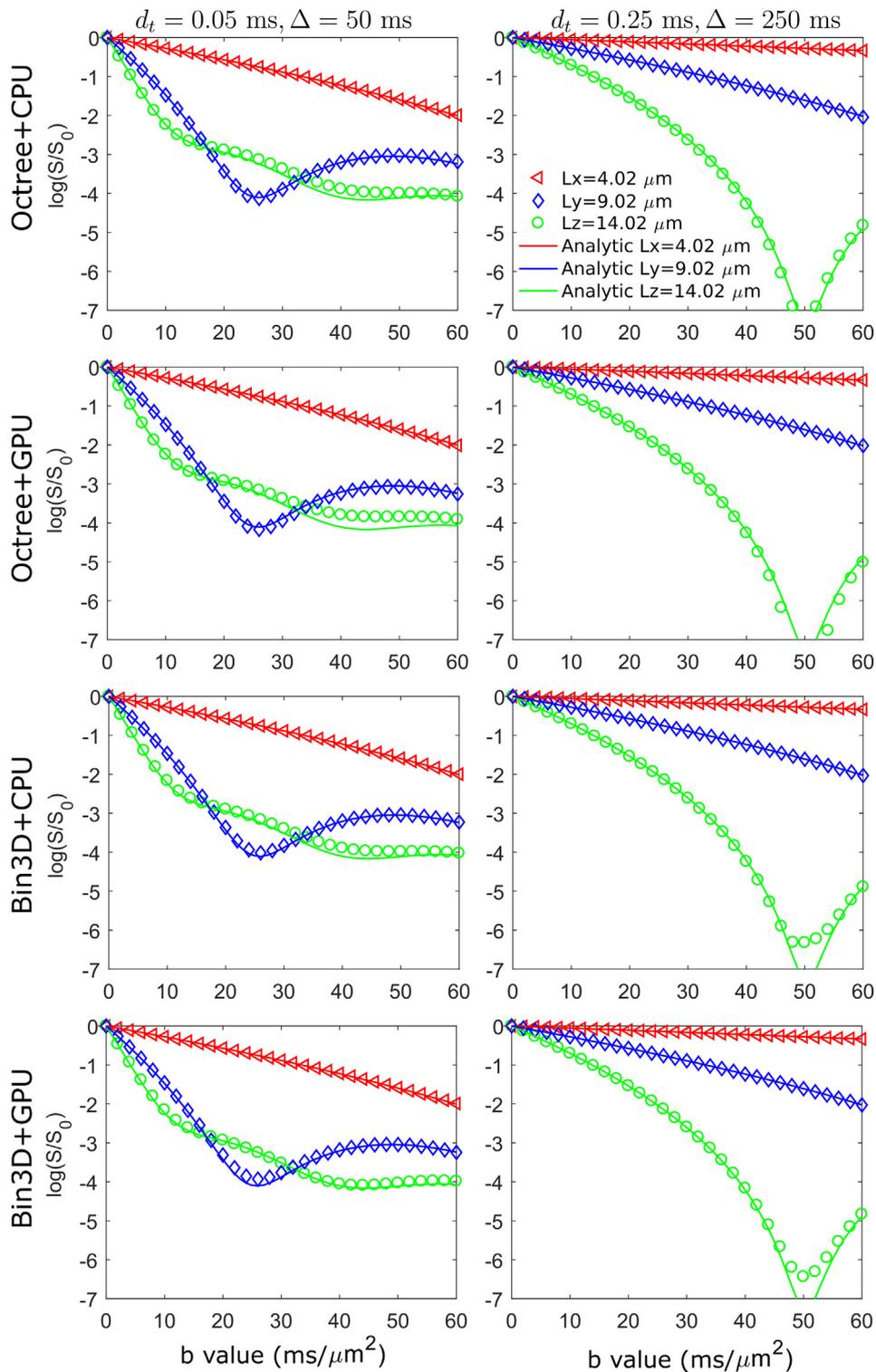


Fig. 4. The simulated DW-NMR signal (marker) in the box dimension $L_x \times L_y \times L_z = 4.02 \mu\text{m} \times 9.02 \mu\text{m} \times 14.02 \mu\text{m}$ and analytic approximate DW-NMR signal (solid line) as a function of b -values at $\Delta = 50$ ms (left) and $\Delta = 250$ ms (right) in gradient direction aligned to the box axes (using $N = 2^{18}$ particles, $d_t = 0.05$ ms for $\Delta = 50$ ms and $d_t = 0.25$ ms for $\Delta = 250$ ms). The first and second rows represent the simulated DW-NMR signal using Octree + CPU and Octree + GPU methods, respectively. The two last rows represent the simulated DW-NMR signal using Bin3D + CPU and Bin3D + GPU methods, respectively. In each figure, the red, blue and green colors represent the simulated signal in gradient direction aligned to the box axes $L_x = 4.02 \mu\text{m}$, $L_y = 9.02 \mu\text{m}$, and $L_z = 14.02 \mu\text{m}$, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.2. Performance

To assess the performance of proposed methods, we used the previous configurations (Section 3.1) for CPU and GPU Monte-Carlo simulations.

First, the computational time of DW-NMR simulation in the box for $\Delta = 50$ ms and $\Delta = 250$ ms, using Camino as a reference tool, Octree + CPU/GPU method and Bin3D + CPU/GPU method as described in the previous subsection, is presented in Fig. 7. Although the box mesh is very simple and the number of surface

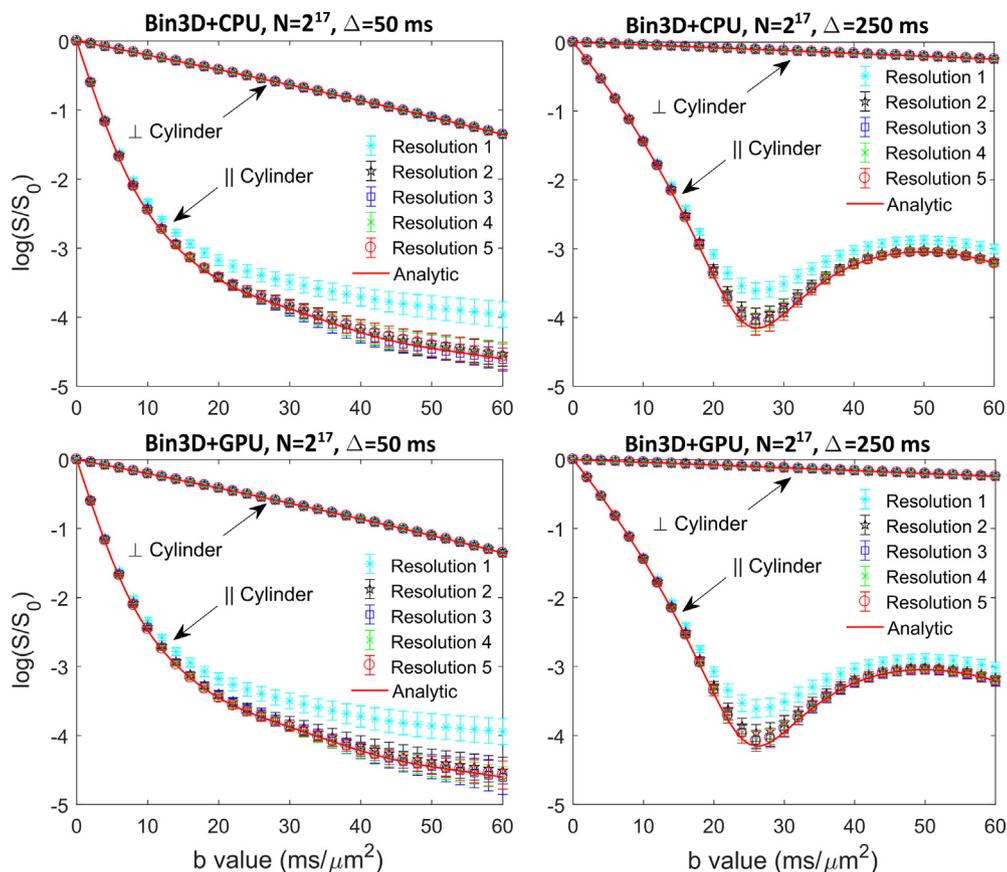


Fig. 5. Mean and standard deviation of the log of CPU (top) and GPU (bottom) simulated DW-NMR signals using the binary marker representation, and analytic DW-NMR signal (solid line) as a function of b-values at $\Delta = 50$ ms (left) and $\Delta = 250$ ms (right) in gradient direction parallel and perpendicular to the cylinder.

Table 1

RE (represented as mean \pm std) of the DW-NMR signals simulated using the binary marker approach by CPU and GPU Monte-Carlo simulation at $\Delta = 50$ ms and $\Delta = 250$ ms for the gradient direction parallel and perpendicular to the cylinder principal axes. The mean and std are calculated over 30 randomly oriented cylinders.

		Perpendicular to cylinder: RE (%)		
		Bin3D + CPU (N = 2 ¹⁷)	Bin3D + GPU (N = 2 ¹⁷)	Bin3D + GPU (N = 2 ²⁰)
$\Delta = 50$ ms	Resolution 1	0.657 \pm 0.380	0.737 \pm 0.397	0.735 \pm 0.375
	Resolution 2	0.195 \pm 0.159	0.164 \pm 0.115	0.092 \pm 0.074
	Resolution 3	0.194 \pm 0.135	0.206 \pm 0.119	0.073 \pm 0.042
	Resolution 4	0.143 \pm 0.124	0.203 \pm 0.118	0.075 \pm 0.056
	Resolution 5	0.167 \pm 0.106	0.166 \pm 0.126	0.086 \pm 0.057
$\Delta = 250$ ms	Resolution 1	0.124 \pm 0.086	0.133 \pm 0.075	0.129 \pm 0.073
	Resolution 2	0.044 \pm 0.037	0.037 \pm 0.026	0.024 \pm 0.014
	Resolution 3	0.032 \pm 0.027	0.037 \pm 0.030	0.016 \pm 0.011
	Resolution 4	0.038 \pm 0.020	0.032 \pm 0.027	0.015 \pm 0.011
	Resolution 5	0.048 \pm 0.028	0.038 \pm 0.027	0.017 \pm 0.013
		Parallel to cylinder: RE (%)		
		Bin3D + CPU (N = 2 ¹⁷)	Bin3D + GPU (N = 2 ¹⁷)	Bin3D + GPU (N = 2 ²⁰)
$\Delta = 50$ ms	Resolution 1	5.964 \pm 1.323	5.934 \pm 1.454	5.890 \pm 1.406
	Resolution 2	2.097 \pm 0.375	2.126 \pm 0.405	2.028 \pm 0.176
	Resolution 3	1.654 \pm 0.354	1.765 \pm 0.363	1.485 \pm 0.138
	Resolution 4	1.443 \pm 0.414	1.291 \pm 0.382	1.098 \pm 0.109
	Resolution 5	1.456 \pm 0.407	1.175 \pm 0.220	1.057 \pm 0.137
$\Delta = 250$ ms	Resolution 1	3.933 \pm 1.006	3.949 \pm 1.129	3.973 \pm 0.981
	Resolution 2	1.196 \pm 0.462	1.302 \pm 0.553	1.012 \pm 0.170
	Resolution 3	0.831 \pm 0.333	0.966 \pm 0.325	0.719 \pm 0.175
	Resolution 4	0.838 \pm 0.439	0.799 \pm 0.366	0.456 \pm 0.117
	Resolution 5	0.776 \pm 0.342	0.848 \pm 0.530	0.494 \pm 0.159

elements is small, Camino takes more than 1 h for simulation, whereas Octree + CPU method only needs ~ 7 min when performed in CPU and less than 2 min when performed in GPU (Octree + GPU).

Bin3D + CPU takes less than min, whereas Bin3D + GPU only takes less than 2 s. Note that the simulation time mentioned in Bin3D + CPU/GPU does not account for ~ 20 s for loading the binary mar-

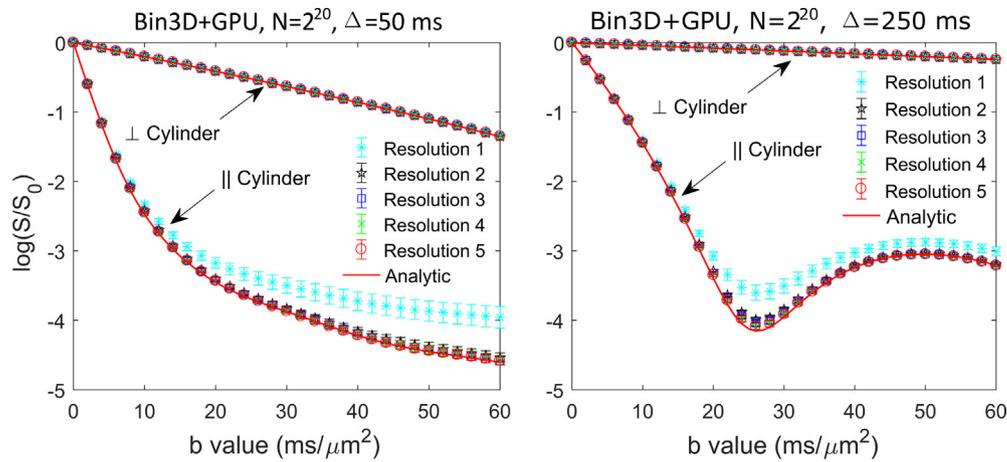


Fig. 6. Mean and standard deviation of the log of simulated DW-NMR signals using the binary marker representation, and analytic DW-NMR signal as a function of b-values at $\Delta = 50$ ms (left) and $\Delta = 250$ ms (right) in gradient direction parallel and perpendicular to the cylinder. The number of particles used for simulation is $N = 2^{20}$.

ker ascii file (containing $352 \times 465 \times 450$ values of 0 and 1) at the beginning. It can be reduced by saving the binary marker file as binary format instead of ascii format. Table 2 shows the speedup factor between Octree + CPU, Octree + GPU, Bin3D + CPU and Bin3D + GPU methods in comparison with Camino simulation time.

For in-depth performance test, especially for the method using 3D binary geometry marker presentation, we performed the DW-NMR simulation in 30 randomly oriented finite-length cylinder that described in Section 2.5 and measured the simulation time at $\Delta = 50$ ms. The mean and standard deviation of RE over 30 randomly oriented cylinders under the same spatial resolution was calculated and plotted as a function of the averaged simulation time (see Fig. 8). Increased simulation time was associated with increased spatial resolution. Fig. 8c, e, and f show that the highest spatial resolution gives the lowest averaged RE values for both Bin3D + CPU and Bin3D + GPU simulations. Moreover, when increasing the number of particles up to $N = 2^{20}$ in Bin3D + GPU Monte-Carlo simulation, the averaged RE did not decrease substantially (see Fig. 8e and f). The standard deviation, however, was smaller presumably due to the larger number of particles (see Fig. 8e and f). The Bin3D + GPU implementation reduced the simulation time from hundreds of seconds (in Bin3D + CPU implementation) to seconds. Moreover, the GPU simulation time is increasing linearly with the number of particles.

We also use Octree + CPU/GPU and Camino simulation toolkit to generate the reference simulated DW-NMR signals in three cylinders with a different number of surface elements (see Fig. 3b),

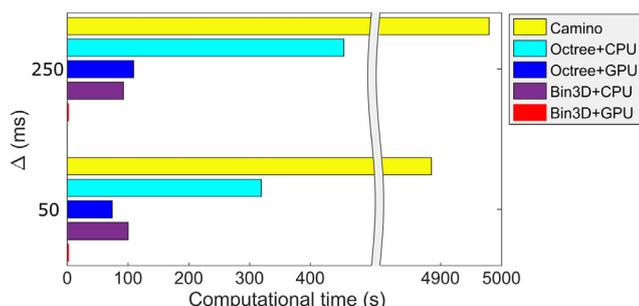


Fig. 7. The measured computational times of DW-NMR simulation in Camino toolkit (yellow), Octree + CPU (cyan), Octree + GPU (blue), Bin3D + CPU (violet) and Bin3D + GPU (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

using the same parameters as for Bin3D + CPU and Bin3D + GPU Monte-Carlo simulations, except for time-step size that we degraded to $d_t = 0.05$ ms (resulting in complexity $\sim 10^8$) to maintain reasonable computation times in Camino. As shown in Fig. 8a, b, and Fig. 8d, using the same simulation parameters, Octree helps reducing the computational time from ~ 5 h in Camino to 4 min in Octree + CPU and 2 min in Octree + GPU while keeping the same relative error level. Moreover, the larger the number of surface elements, the more efficient the Octree method in comparison with Camino. The RE of simulated DW-NMR signals using Camino, in the case of gradient direction perpendicular to the cylinder principal-axis, was decreased when refining the mesh. However, when the gradient direction parallels the cylinder principal-axis, the measured RE of simulated DW-NMR signals remains stable $\sim 1\%$ because the representation of cylinder's extremities remains accurate for all mesh sizes (see Fig. 3b). Although the number of surface elements remains relatively small (e.g. as compared to complex cellular surface meshes) and the time-step used in Camino simulations is ten times longer than in Bin3D + CPU/GPU Monte-Carlo simulations, Camino simulation time remains very long (see Fig. 8d). The reason is due to at each time-step, the intersection between the displacement vector (from current to new particle's position) and all triangular elements of the mesh should be checked to make sure that the particle does not cross the surface. Segment-triangle intersection algorithms need some cross-products and some comparison operators [41], thus driving the computation-cost when the number of surface elements is large. Instead, when using the binary marker representation, we only need one conversion from real world coordinates to voxel coordinates, and one Boolean comparison to determine whether a given particle crosses the surface or not, whatever the complexity of the geometry. This explains the efficiency of the binary marker representation used in the present work as opposed to the more traditional mesh-based representation.

3.3. Selection of time-step size or number of time steps for the binary marker approach

The choice of an adequate time-step is particularly important for the binary marker approach, because in this approach particle interaction with membranes is treated using the "rejection method", which requires that the elementary jump size is much smaller than typical distance between membranes. To carefully evaluate the effect of time-step size or number of time-steps on the DW-NMR signal, different time-steps $d_t = [0.5; 0.1; 0.05; 0.01; 0.005;$

Table 2

The acceleration factor of Octree + CPU, Octree + GPU, Bin3D + CPU, and Bin3D + GPU in comparison with Camino simulator considered as a reference simulation tool.

	Octree + CPU	Octree + GPU	Bin3D + CPU	Bin3D + GPU
$\Delta = 50$ ms	15.3×	66.1×	48.8×	2177×
$\Delta = 250$ ms	10.9×	45.6×	53.8×	2311×

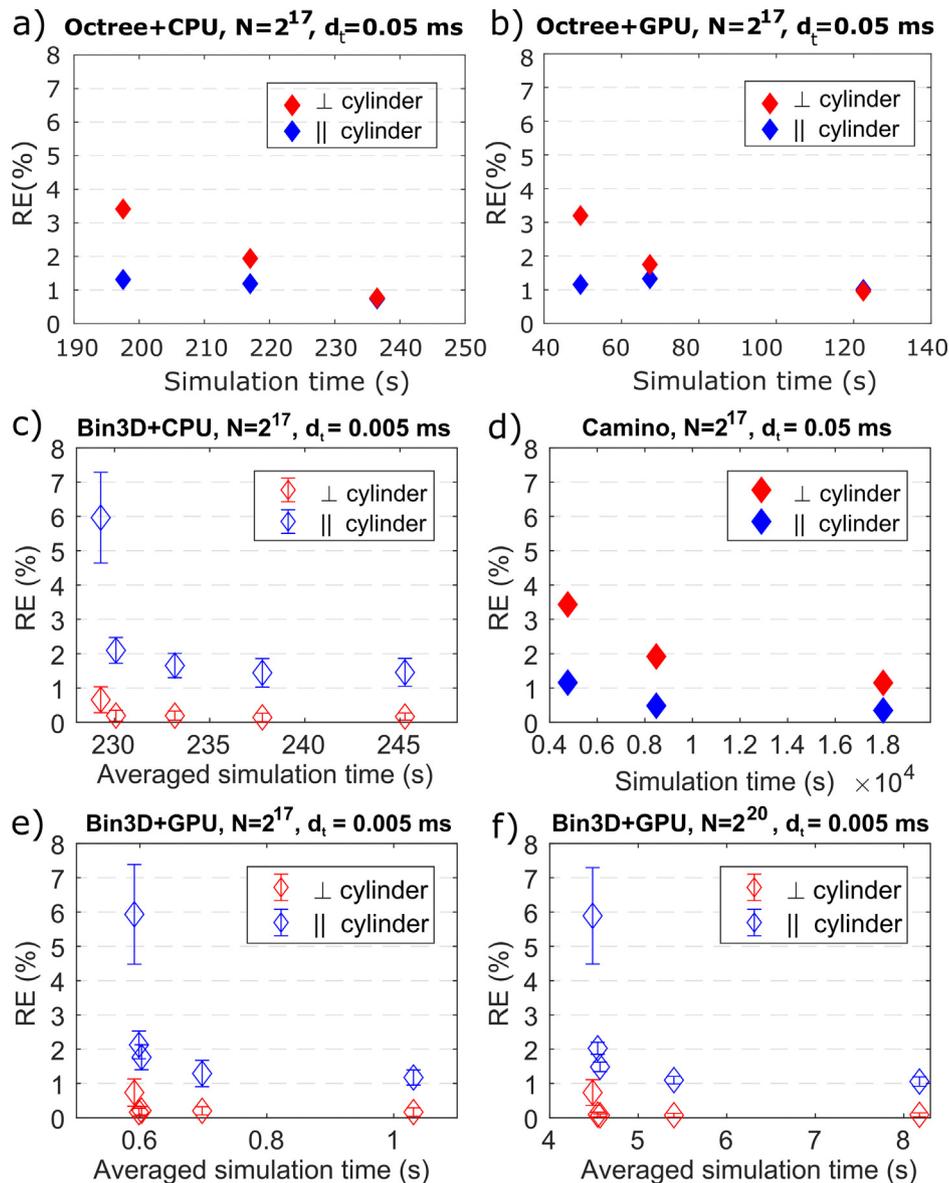


Fig. 8. The RE and simulation time of Octree + CPU (a), Octree + GPU (b) and Camino (d) simulations in cylinder, for three different numbers of surface elements (left: 344, middle: 630 and right: 1292 elements) at time-step size $d_t = 0.05$ ms. The performance of Bin3D + CPU (c) and Bin3D + GPU (e, f) Monte-Carlo simulations at time-step size $d_t = 0.005$ ms for five different spatial resolutions (corresponding to the 5 markers for a given color/orientation). In subfigures c,e,f: the simulation time increases with increased spatial resolution; For each spatial resolution, the mean and standard deviation of RE were calculated over 30 random orientations. In all sub-figures, the blue and red colors represent the measured RE of the simulated DW-NMR signal parallel and perpendicular to the cylinder main axis, respectively. The number of particles used in all subfigures is $N = 2^{17}$, except $N = 2^{20}$ in subfigure f. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

0.001; 0.0005; 0.0003] ms (with fixed number of particles $N = 2^{17}$) were used, for three finite-length cylinders: radius $R = 0.5 \mu\text{m}$, length $L = 5 \mu\text{m}$; $R = 1 \mu\text{m}$, $L = 10 \mu\text{m}$; and $R = 2 \mu\text{m}$, $L = 20 \mu\text{m}$. These cylinders were voxelized with the resolution of $0.07 \times 0.07 \times 0.07 \mu\text{m}^3$ (identical to the confocal resolution along X and Y). The intrinsic diffusion and gradient duration were

selected as in the previous section, $D_0 = 0.5 \mu\text{m}^2/\text{ms}$ and $\delta = 0$ ms, respectively. Three gradient separations $\Delta = 5$ ms, $\Delta = 50$ ms and $\Delta = 250$ ms were used in this test.

The RE between the simulated signal and analytical signal in the direction parallel and perpendicular to the principal cylinder axes was calculated according to Eq. (4) and is shown in Fig. 9.

As shown in Fig. 9, the RE between the simulated and analytical signal in the direction perpendicular to the cylinder axis is very small ($RE < 1\%$ for all three different diffusion time) when the time-step size is smaller than 0.05 ms. In theory, smaller time-step size should reduce the error arising from the “rejection method” when handling the interaction between particles and the surface. However, too small time-step can lead to another kind of error, due to the accumulation of truncation errors during the diffusion of molecules when the jump size is very small. The above reasons explained the increase of RE when the time-step is too short ($d_t = 0.0003$ ms) as shown in Fig. 9 (arrows). Therefore, the time-step should be carefully chosen for Monte-Carlo simulations of diffusion, and extraordinary GPU performances should not be misused to reduce the time-step below some limit. Once this limit has been reached, further gains in GPU performances should rather be used to increase the number of particles. With our current implementation, we recommend a lower-bound on time-step of ~ 0.0005 ms, with an optimum around ~ 0.05 – 0.001 ms (as apparent in Fig. 9), corresponding to jump size $\sigma \approx 0.387$ – 0.055 μm .

Moreover, in the case of gradient direction perpendicular to the cylinder’s principal axis, it is surprising that even when the time-step size is large ($d_t = 0.5$ or $d_t = 0.1$ ms, corresponding to a jump size $\sigma \approx 1.22$ μm or $\sigma \approx 0.55$ μm , respectively), the RE remains very low ($< 1\%$) for $\Delta = 50$ and $\Delta = 250$ ms, but not for $\Delta = 5$ ms (see Fig. 9 - red line colors). Thus, the jump size seems to have less impact on the simulated signal in small confining structures at long diffusion time. In contrast, in the short diffusion time regime, the simulated signal is more sensitive to the jump size (or time-step). In the case of the gradient direction parallel to the cylinder principal axis, the RE values were higher than in the case of the gradient direction perpendicular to the cylinder principal axis

(see Fig. 9 - blue line colors). For the acceptable $RE < 5\%$, the time-step size should be 5×10^{-4} – 5×10^{-2} ms.

3.4. Simulation of DW-NMR signal in astrocytes

Two astrocytes (one protoplasmic astrocyte and one fibrous astrocyte, see Fig. 10) were selected and extracted from confocal microscopy images to test simulations signals in two different realistic shapes. The astrocytic surface triangular meshes and the 3D binary markers were then generated for simulation as described in Section 2.1.

We used $\delta = 0$ ms, $\Delta = 50$ ms, a large number of particles ($N = 2^{20}$) to reduce the standard deviation of the simulated signal (especially at high b-value), 61 b-values linearly increased from 0 to 60 $\text{ms}/\mu\text{m}^2$. Six different time-step sizes from $d_t = 0.01$ ms (complexity $\sim 6 \times 10^9$) to $d_t = 0.0001$ ms (complexity $\sim 10^{11}$), corresponding to number of time steps ranging from 5000 to 500,000, were evaluated to test the impact of d_t on the simulation in these cells. The simulated DW-NMR signal along three gradient directions (x, y, and z-direction) using the GPU-based implementation of the 3D binary marker method is shown in Fig. 11. In addition, we also generated the DW-NMR signal (gradient direction along x, y, and z-direction) using the GPU-based implementation of the octree structure representation method at time-step $d_t = 0.001$ ms and number of particles $N = 2^{18}$ resulting in complexity of $\sim 10^{10}$, thus exceeding the requirement upon complexity to get a good approximation of simulated DW-NMR signal as recommended in [22]. These simulated signals were then used as the “ground-truth signal”, as the specular reflection implemented in the octree approach can be considered as more conventional and less prone to errors than the rejection method (note that Camino, which also

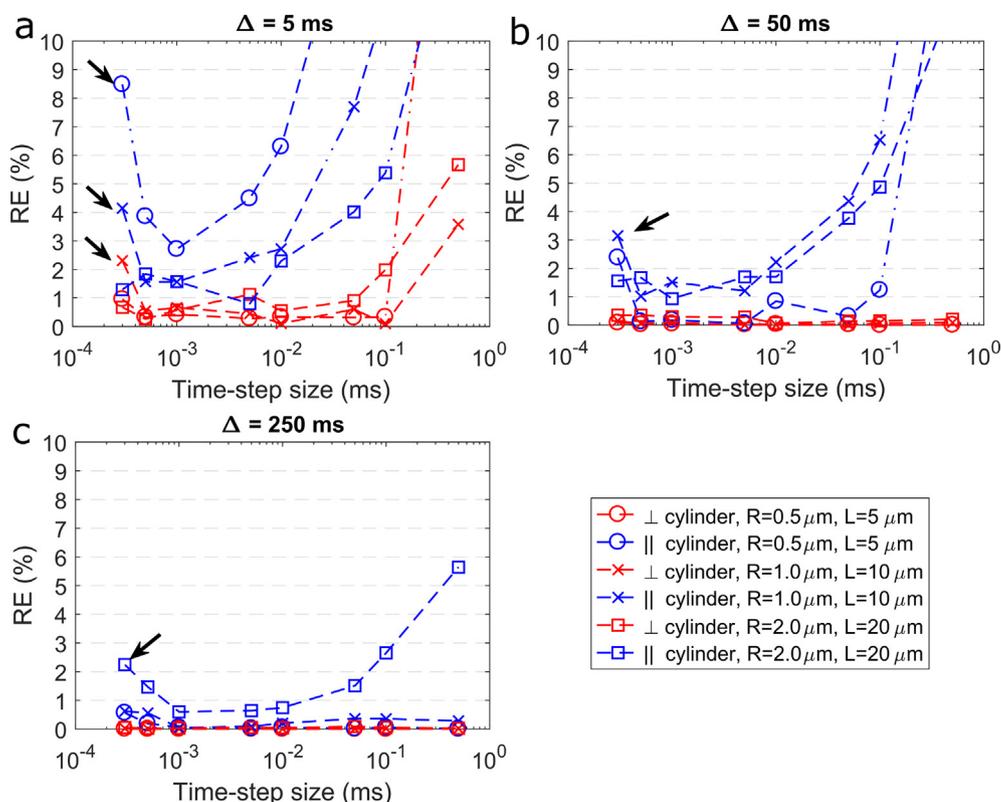


Fig. 9. RE of simulated DW-NMR signal relative to analytical signal in the direction perpendicular (red colors) and parallel (blue colors) to the cylinder principal axes at $\Delta = 5$ ms (a), $\Delta = 50$ ms (b), and $\Delta = 250$ ms (c), respectively. The circle, cross and squared marker represent the cylinder (radius $R = 0.5$ μm , length $L = 5$ μm), cylinder ($R = 1.0$ μm , $L = 10$ μm), and cylinder ($R = 2.0$ μm , $L = 20$ μm), respectively. Arrows point to the increasing of RE when the time-step is too small, $d_t = 0.0003$ ms. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

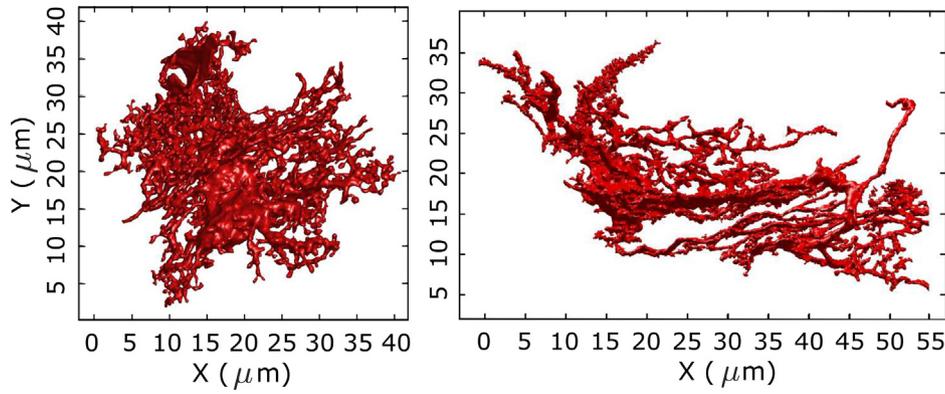


Fig. 10. X-Y view of two surface meshes from real astrocytes used for test the simulation of DW-NMR signal in astrocytes. Left: Protoplasmic astrocytes, Right: Fibrous astrocytes.

relies on specular reflection, ran too slowly for these complex astrocytic structures).

Because the distribution of astrocytic branches is almost identical in the x and y -directions for the protoplasmic astrocyte, the simulated intracellular DW-NMR signals in x and y -direction are similar (see Fig. 11a). In contrast, the fibrous astrocyte extends preferentially along x rather than y -direction, explaining the different behavior of the simulated DW-NMR signal presented in Fig. 11b. Moreover, as shown in this figure, changing time-step seems to have only a moderate effect. RE was calculated between the simulated signals and the ground-truth signal (from octree approach). The averaged RE in x , y , and z -directions is shown in Fig. 12. As shown in this figure, when $d_t < 0.003$ ms (corresponding to $\sigma < 0.1 \mu\text{m}$), the averaged RE was very small (<3%). This threshold may be used in other diffusion simulation studies in astrocytes with the 3D binary marker method. Note that, the confocal microscopy image resolution in z -direction is limited to $\sim 0.28 \mu\text{m}$, so that reconstructed confocal microscopy images seem stretched in the z -direction. Therefore, even using interpolation to oversample the confocal image in the z -direction to $0.07 \mu\text{m}$ as in x and y -direction, the simulated DW-NMR signal in z -direction always exhibits more substantial attenuation than in x and y -direction (see Fig. 11).

Although the number of particles is enormous and the time-step is small, the GPU Monte-Carlo simulation time remains short, as shown in Fig. 13. At an acceptable time-step $d_t = 0.002$ ms, the

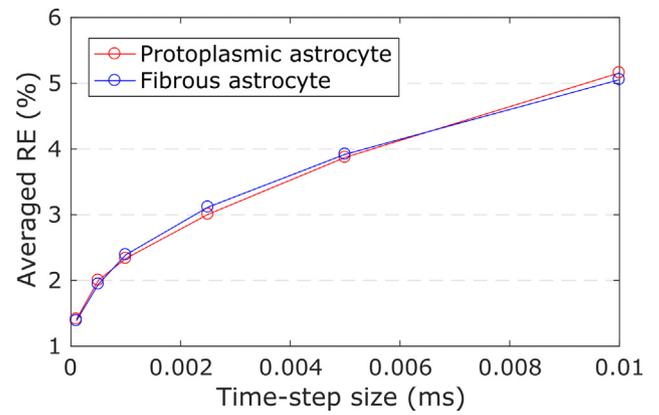


Fig. 12. Averaged RE of the simulated signal (Bin3D + GPU) relative to the ground-truth simulated signal (Octree + GPU) in the two astrocytes.

simulation time is approximately 50–70 s. In contrast, simulation of the ground-truth signal using the octree approach required 7 h 20 min for protoplasmic astrocyte, and 18 h 40 min for fibrous astrocyte. The GPU-based implementation of the 3D binary marker method thus opens the possibility to simulate the DW-NMR signal in complex cellular structures at high b -values and long diffusion times within reasonable times.

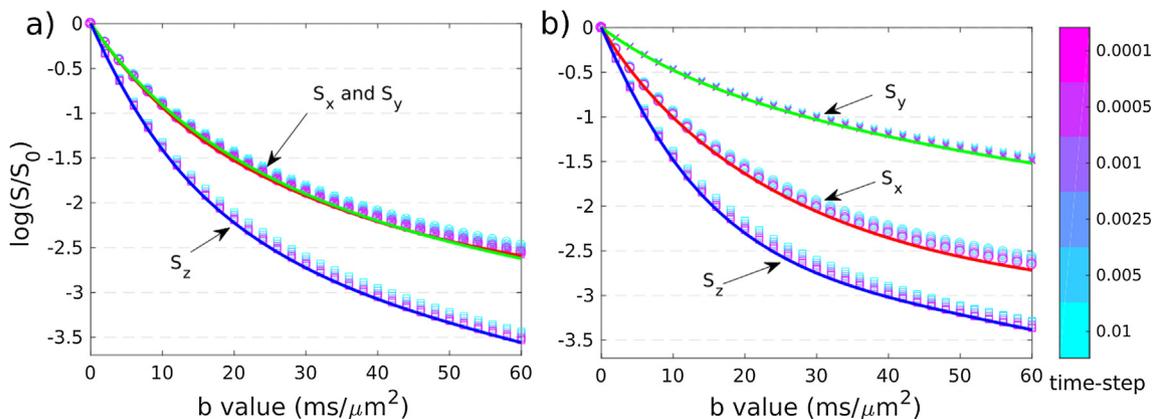


Fig. 11. Simulated DW-NMR signals inside protoplasmic astrocytes (a) and fibrous astrocytes (b). The simulated DW-NMR signals generated by the GPU-based implementation of the 3D binary marker presentation method is presented in markers. The different colors represent the different time-step which showed in the color bar. Red, green and blue solid line represents the ground-truth signals of gradient direction aligned in x , y , and z -direction, respectively, as computed using the Octree + GPU method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

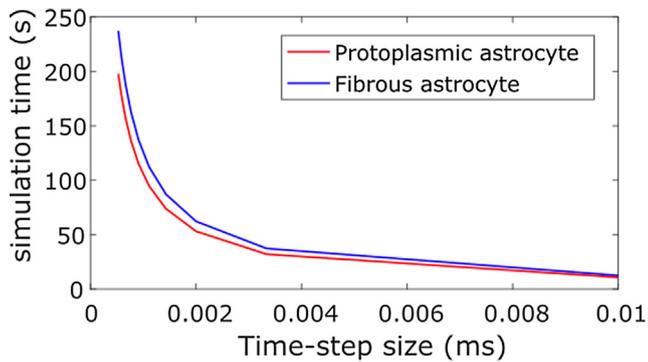


Fig. 13. Simulation time for the two astrocytes using the Bin3D + GPU method.

4. Discussion and conclusion

Based on the results about the accuracy of the simulated DW-NMR signal on finite length cylinders, a $0.07 \mu\text{m}$ spatial resolution appears sufficient to get an accurate simulated DW-NMR signals, even in small structures like cylinders of $0.5 \mu\text{m}$ radius. It substantiates the idea that confocal microscopy image resolution of $\sim 0.07 \times 0.07 \times 0.28 \mu\text{m}^3$ is good enough for using astrocytes directly extracted from confocal microscopy image as an input geometry for DW-NMR signal simulation, at least along x and y directions.

The strategy of using real world coordinates for particles' position allows decoupling the moving step size σ (and therefore the time-step size d_t) from the voxel size (resolution) of the astrocyte's binary marker. Then, using one conversion from real world coordinates to voxel coordinates, and one Boolean comparison to determine whether a given position is accepted or rejected at each time-step, allows drastically minimizing the computational burden as compared to mesh-based approaches (even when using an octree structure), thus minimizing the running time of Monte-Carlo simulation in complex geometries.

Although infinitely short gradient duration, $\delta = 0 \text{ ms}$, was used in this work, the method described in this paper allows simulating DW-NMR signal in various gradient durations and using various pulse sequences.

The effect of membrane permeability was not considered here but could be interesting to study and might be incorporated in future works. One possibility to treat permeability would be to reject motion with some probability smaller than 1 when particles encounter membranes. If motion is not rejected, the particle would enter the membrane, which in the binary representation would have some "thickness" of at least one pixel. Thus, we would have different options to describe the passage of the particle through the membrane: a first option, easy to implement, would be that the particle could diffuse inside the membrane and then exit again with some non-zero probability at some point, when it encounters a new interface. Such an approach might offer a realistic description of passive diffusion through membranes. Another option that would require slightly adapting the current version of the code, would be to have the particle travel through the membrane and exit on the other side, at the shortest distance from its entry point. This would be a realistic description of transport via membranes pores or active transporters.

In the case of wall relaxivity, each time a particle's jump is rejected due to interaction with membrane, it would retain some memory of this interaction (e.g. by incrementing an integer number associated with this particle), and at the end of the simulation it would be trivial to weight this particle's contribution according to the number of interactions it underwent [24,42].

Other phenomena might affect molecular diffusion, such as active transport over long distances (e.g. flow of the extracellular fluid, or vesicular trafficking inside the cells). Although in theory nothing prevents incorporating such phenomena into modeling tools, we think that the main difficulty would not arise from adapting algorithms to account for these phenomena, but rather from the very limited knowledge we have about their magnitude and the laws governing them.

All source code and geometry models will be released soon on the author's GitHub address: <https://github.com/khieunguyen/GPUMCdMRI>.

Acknowledgments

This work was funded by the European Research Council (grant number 336331 INCELL project). The NVIDIA Tesla K40c was generously donated by NVIDIA Corporation.

References

- [1] P.J. Basser, J. Mattiello, D. LeBihan, MR diffusion tensor spectroscopy and imaging, *Biophys. J.* 66 (1994) 259–267.
- [2] D.G. Norris, The effects of microscopic tissue parameters on the diffusion weighted magnetic resonance imaging experiment, *NMR Biomed.* 14 (2001) 77–93.
- [3] C. Beaulieu, The basis of anisotropic water diffusion in the nervous system – a technical review, *NMR Biomed.* 15 (2002) 435–455.
- [4] H.C. Berg, *Random Walks in Biology*, Princeton University Press, 1993.
- [5] J. Zhong, J.C. Gore, Studies of restricted diffusion in heterogeneous media containing variations in susceptibility, *Magnetic Reson. Med.* 19 (1991) 276–284.
- [6] D.S. Grebenkov, NMR survey of reflected Brownian motion, *Rev. Mod. Phys.* 79 (3) (2007) 1077–1137, 8.
- [7] Y. Assaf, T. Blumenfeld-Katzir, Y. Vovel, P.J. Basser, Axcaliber: a method for measuring axon diameter distribution from diffusion MRI, *Magnetic Reson. Med.* 59 (2008) 1347–1354.
- [8] D.C. Alexander, P.L. Hubbard, M.G. Hall, E.A. Moore, M. Ptito, G.J.M. Parker, T.B. Dyrby, Orientationally invariant indices of axon diameter and density from diffusion MRI, *NeuroImage* 52 (2010) 1374–1389.
- [9] H. Zhang, P.L. Hubbard, G.J.M. Parker, D.C. Alexander, Axon diameter mapping in the presence of orientation dispersion with diffusion MRI, *NeuroImage* 56 (2011) 1301–1315.
- [10] 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 (2012) 1000–1016.
- [11] M. Palombo, C. Ligneul, 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 Reson. Med.* 77 (2017) 343–350.
- [12] H.C. Torrey, Bloch Equations with Diffusion Terms, *Phys. Rev.* 104 (3) (1956) 563–565, 11.
- [13] S.N. Hwang, C.-L. Chin, F.W. Wehrli, D.B. Hackney, An image-based finite difference model for simulating restricted diffusion, *Magnetic Reson. Med.* 50 (2003) 373–382.
- [14] J. Xu, M.D. Does, J.C. Gore, Numerical study of water diffusion in biological tissues using an improved finite difference method, *Phys. Med. Biol.* 52 (2007) N111.
- [15] 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, *Phys. Med. Biol.* 57 (2012) N35.
- [16] J.-R. Li, D. Calhoun, C. Poupon, D.L. Bihan, Numerical simulation of diffusion MRI signals using an adaptive time-stepping method, *Phys. Med. Biol.* 59 (2014) 441.
- [17] 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, *J. Comput. Phys.* 263 (2014) 283–302.
- [18] L. Beltrachini, Z.A. Taylor, A.F. Frangi, A parametric finite element solution of the generalised Bloch-Torrey equation for arbitrary domains, *J. Magnetic Reson.* 259 (2015) 126–134.
- [19] B.D. Hughes, *Random Walks and Random Environments*, Clarendon Press, Oxford University Press, 1995.
- [20] A. Szafer, J. Zhong, J.C. Gore, Theoretical model for water diffusion in tissues, *Magnetic Reson. Med.* 33 (1995) 697–712.
- [21] J.C. Ford, D.B. Hackney, Numerical model for calculation of apparent diffusion coefficients (ADC) in permeable cylinders—comparison with measured ADC in spinal cord white matter, *Magnetic Reson. Med.* 37 (1997) 387–394.
- [22] M.G. Hall, D.C. Alexander, Convergence and parameter choice for Monte-Carlo simulations of diffusion MRI, *IEEE Trans. Med. Imaging* 28 (9) (2009) 1354–1364.

- [23] D.S. Grebenkov, A fast random walk algorithm for computing the pulsed-gradient spin-echo signal in multiscale porous media, *J. Magnetic Reson.* 208 (2011) 243–255.
- [24] C.A. Waudby, J. Christodoulou, GPU accelerated Monte Carlo simulation of pulsed-field gradient NMR experiments, *J. Magnetic Reson.* 211 (2011) 67–73.
- [25] 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, *Proc. Natl. Acad. Sci.* 113 (2016) 6671–6676.
- [26] E. Panagiotaki, M.G. Hall, H. Zhang, B. Siow, M.F. Lythgoe, D.C. Alexander, High-fidelity meshes from tissue samples for diffusion MRI simulations, in: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*, Berlin, 2010.
- [27] C.-H. Yeh, B. Schmitt, D. Le Bihan, J.-R. Li-Schlittgen, C.-P. Lin, C. Poupon, Diffusion microscopist simulator: a general Monte Carlo simulation system for diffusion magnetic resonance imaging, *PLOS ONE* 8 (10) (2013) 1–12.
- [28] K. Ginsburger, F. Poupon, J. Beaujoin, D. Estournet, F. Matuschke, J.-F. Mangin, M. Axer, C. Poupon, Improving the realism of white matter numerical phantoms: a step toward a better understanding of the influence of structural disorders in diffusion MRI, *Front. Phys.* 6 (2018) 12.
- [29] D.J.R. Meagher, Octree Encoding: a New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer, Rensselaer Polytechnic Institute, Image Processing Laboratory, 1980.
- [30] D. Meagher, Geometric modeling using octree encoding, *Comput. Graphics Image Process.* 19 (1982) 129–147.
- [31] M. Palombo, C. Ligneul, E. Hernandez-Garzon, J. Valette, Can we detect the effect of spines and leaflets on the diffusion of brain intracellular metabolites?, *NeuroImage* (2017)
- [32] L. Xie, M.A. Sparks, W. Li, Y. Qi, C. Liu, T.M. Coffman, G.A. Johnson, Quantitative susceptibility mapping of kidney inflammation and fibrosis in type 1 angiotensin receptor-deficient mice, *NMR Biomed.* 26 (2013) 1853–1863.
- [33] Q. Fang, D.A. Boas, Tetrahedral mesh generation from volumetric binary and gray-scale images, in: *Proceedings of the Sixth IEEE International Conference on Symposium on Biomedical Imaging: From Nano to Macro*, Piscataway, 2009.
- [34] CGAL, Computational Geometry Algorithms Library [Online], Available: <http://www.cgal.org>.
- [35] J.J. Jiménez, R.J. Segura, F.R. Feito, A robust segment/triangle intersection algorithm for interference tests. Efficiency study, *Comput. Geometry* 43 (2010) 474–492.
- [36] D. Madeira, E. Clua, A. Montenegro, T. Lewiner, Gpu octrees and optimized search, in: *VIII Brazilian Symposium on Games and Digital Entertainment*, 2009.
- [37] G.T. Balls, L.R. Frank, A simulation environment for diffusion weighted MR experiments in complex media, *Magnetic Reson. Med.* 62 (2009) 771–778.
- [38] J. Lane, B. Magedson, M. Rarick, An efficient point in polyhedron algorithm, *Computer Vision, Graphics, Image Process.* 26 (1984) 118–125.
- [39] J. Li, W. Wang, Fast and robust GPU-based point-in-polyhedron determination, *Computer-Aided Des.* 87 (2017) 20–28.
- [40] P. Linse, O. Soderman, The validity of the short-gradient-pulse approximation in NMR studies of restricted diffusion. Simulations of molecules diffusing between planes, in cylinders and spheres, *J. Magnetic Reson., Series A* 116 (1995) 77–86.
- [41] T. Möller, B. Trumbore, Fast, minimum storage ray-triangle intersection, *J. Graphics Tools* 2 (1997) 21–28.
- [42] P. Szymczak, A.J.C. Ladd, Boundary conditions for stochastic solutions of the convection-diffusion equation, *Phys. Rev. E* 68 (3) (2003) 036704, 9.