

A FENICS-HPC FRAMEWORK FOR MULTI-COMPARTMENT BLOCH-TORREY MODELS

D. V. NGUYEN¹, J. JANSSON^{1,2}, J. HOFFMAN¹

¹ Computational Technology Laboratory, Department of Computational Science and Technology, School of Computer Science and Communication, KTH Royal Institute of Technology

² BCAM - Basque Center for Applied Mathematics, Bilbao Basque Country - Spain
e-mail: {vdnguyen, jjan, jhoffman}@kth.se

Keywords: FEniCS-HPC, diffusion NMR, diffusion MRI, simulation.

Abstract. *In diffusion nuclear magnetic resonance (NMR) and diffusion magnetic resonance imaging (MRI), the multi-compartment Bloch-Torrey equation plays an important role in probing the diffusion characteristics from a nanometer scale to a macroscopic scale. The signal attenuation can be computed by solving the equation. If the volume of interest is composed by multiple compartments, interface conditions with permeability are imposed. Depending on applications, different gradient strengths can be used to capture the signal attenuation. In probing porous media, for instance, high gradient strengths are used. In diffusion MRI, since water molecules enter and exit the computational domain in realistic cases, pseudo-periodic boundary conditions are used. These conditions cause difficulties in solving the equation efficiently and many efforts have been made to develop an efficient numerical method. However, large-scale problems for supercomputers with realistic applications have not been considered yet. We propose a framework for the multi-compartment Bloch-Torrey models based on the FEniCS-HPC platform, a part of the FEniCS project that allows for automated discretization, automated error control with mesh adaptivity and high performance computing. The framework runs on supercomputers with near optimal weak and strong scaling. Our work includes two parts. First, we simplify the multi-compartment Bloch-Torrey model used in diffusion MRI by proposing an approximation to the pseudo-periodic boundary conditions to derive a general form for the interface and boundary conditions. The second part includes implementation and numerical validation of our method on the FEniCS-HPC platform. This simplified model is straightforward to implement and to parallelize and shows promise in validation against more realistic models.*

1 Introduction

In diffusion NMR and diffusion MRI, the signal attenuation S represents the diffusion characteristic of molecular motion through the use of magnetic field gradient pulses and it is computed as the average of the transverse magnetization $M(\mathbf{r}, t)$ over a volume of interest Ω . The signal attenuation is measured at the echo time T by integrating the complex transverse magnetization, i.e

$$S = \int_{\mathbf{r} \in \Omega} M(\mathbf{r}, T) d\mathbf{r} \quad (1)$$

The water diffusion is encoded by using time-varying linear spatial magnetic field gradient pulses $f(t)$.

The complex transverse magnetization at a spatial position $\mathbf{r} = (r_1, \dots, r_d)$ (here $d = 2, 3$ is the problem dimensions) can be modeled by the Bloch-Torrey equation [15, 19]

$$\frac{\partial}{\partial t} M(\mathbf{r}, t) = -\mathcal{I} \gamma f(t) \mathbf{g} \cdot \mathbf{r} M(\mathbf{r}, t) + \nabla \cdot \left(\mathbf{D}(\mathbf{r}) \nabla M(\mathbf{r}, t) \right), \quad \mathbf{r} \in \Omega, \quad (2)$$

where \mathcal{I} is the complex unit ($\mathcal{I}^2 = -1$), $\gamma = 2.67513 \times 10^8 \text{ rad s}^{-1} \text{ T}^{-1}$ is the gyromagnetic ratio of the water proton, $\mathbf{D}(\mathbf{r})$ is the diffusion tensor, and $\mathbf{g} = (g_1, \dots, g_d)$ is the diffusion gradient including gradient strength and gradient direction.

Eq. (2) needs an initial condition:

$$M(\mathbf{r}, 0) = \rho, \quad \mathbf{r} \in \Omega \quad (3)$$

where ρ is the initial transverse magnetization in Ω .

The most commonly used temporal profile is called the pulsed-gradient spin echo (PGSE) sequence [18], with two rectangular pulses of duration δ , separated by a time interval $\Delta - \delta$, for which the profile $f(t)$ is

$$f(t) = \begin{cases} 1, & 0 \leq t \leq \delta, \\ -1, & \Delta < t \leq \Delta + \delta, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

The signal attenuation is usually computed and plotted against the gradient strength $q = \|\mathbf{g}\|$ or a quantity called the b -value which is computed as

$$b = \gamma^2 \|\mathbf{g}\|^2 \int_0^T \left(\int_0^u f(s) ds \right) du \quad (5)$$

For the PGSE, it is

$$b = \gamma^2 \|\mathbf{g}\|^2 \delta^2 \left(\Delta - \frac{\delta}{3} \right) \quad (6)$$

If Ω is composed by multiple compartments, i.e $\Omega = \cup^l \Omega^l$, the following interface conditions need to be imposed

$$\mathbf{D}^l(\mathbf{r}) \nabla M^l(\mathbf{r}, t) \cdot \mathbf{n}^l = -\mathbf{D}^n(\mathbf{r}) \nabla M^n(\mathbf{r}, t) \cdot \mathbf{n}^n = \kappa^{ln} \left(M^n(\mathbf{r}, t) - M^l(\mathbf{r}, t) \right), \quad (7)$$

where $\mathbf{r} \in \Gamma^{ln}, \forall l, n$ and \mathbf{n}^k is a normal vector pointing outward Ω^k .

In diffusion MRI, to allow the water molecules to enter and exit the volume of interest $\Omega = \prod_{i=1}^d [a_i, b_i]$, the pseudo-periodic boundary conditions are applied

$$\begin{aligned} M(\mathbf{r}, t)|_{r_i=a_i} &= M(\mathbf{r}, t)|_{r_i=b_i} \exp\left(\mathcal{I} \theta_i(t)\right), \\ \frac{\partial M(\mathbf{r}, t)}{\partial r_i} \Big|_{r_i=a_i} &= \frac{\partial M(\mathbf{r}, t)}{\partial r_i} \Big|_{r_i=b_i} \exp\left(\mathcal{I} \theta_i(t)\right), \end{aligned} \quad (8)$$

where

$$\theta_i(t) := \gamma g_i (b_i - a_i) \int_0^t f(s) ds, i = 1, \dots, d$$

An advantage of using Eq. (8) is significantly reducing the size of computational domains. This is good in terms of saving computer resources but it is challenging to impose and to generate valid meshes.

Solving the full model, i.e Eqs. (2, 7, 8), is challenging and many efforts have been made [1, 10, 13, 14, 16, 20]. However, large-scale problems for supercomputers with realistic applications have not been considered yet.

We propose a framework for solving the multi-compartment Bloch-Torrey equation based on the FEniCS-HPC which is an open source framework for automated solution of PDE on massively parallel architectures, providing automated evaluation of variational forms given a high-level description in mathematical notation, duality-based adaptive error control, implicit parameter-free turbulence modeling by use of stabilized FEM and strong linear scaling up to thousands of cores [4–9]. The FEniCS-HPC is a branch of the FEniCS [2, 11] framework based on components with clearly defined responsibilities:

- FIAT: finite element spaces, basis functions
- FFC+UFL: Automated evaluation of weak forms in mathematical notation.
- DOLFIN-HPC: Automated high performance assembly of weak forms and interface to linear algebra of discrete systems and mesh refinement.

Our work includes two parts. First, we simplify the multi-compartment Bloch-Torrey model by proposing an approximation to the pseudo-periodic boundary conditions (8) to derive a general form for the interface and boundary conditions. In the second part, we implement the approximated model on the FEniCS-HPC platform.

The main contributions of this paper include:

- Proposing a framework on supercomputers for solving the multi-compartment Bloch-Torrey equations applied to diffusion NMR and diffusion MRI.
- Solving the difficulties of mesh generation caused by the pseudo-periodic boundary conditions that enables us to approach more realistic geometries.
- Implementing the method on the FEniCS-HPC platform.
- Validating the correctness and demonstrating the efficiency of the model and the method for some example geometries.

The paper is organized as follows. First, we introduce the multi-compartment Bloch-Torrey models used in diffusion NMR and diffusion MRI with equations and quantities used in this field. In section 2, we propose an approximation of the pseudo-periodic boundary conditions. Then, we introduce the discretizations and implementations of the approximated model on the FEniCS-HPC platform in section 3. In section 4, we first validate the correctness of the model through two examples: a two-layer infinitely-long cylinder and a two-layer spherical cell. Then, we show speedup ratio and parallel efficiency. Finally, in section 5 we conclude our work and propose future directions.

2 An approximation of the pseudo-periodic boundary conditions

To impose the pseudo-periodic boundary conditions (8) for the r_i -direction, vertices belonging to opposite planes $r_i = a_i$ and $r_i = b_i$ should be aligned, i.e $\mathbf{r}_{a_i} = \mathbf{r}_{b_i} - (b_i - a_i) \mathbf{e}_i$, here $\mathbf{r}_{a_i}, \mathbf{r}_{b_i}$ are coordinates of vertices belonging to opposite planes $r_i = a_i$ and $r_i = b_i$, and \mathbf{e}_i is a unit vector in r_i -direction. However, it is usually challenging to generate a mesh with such the enforcement in realistic cases since the interfaces usually cut the exterior boundaries. We propose an approximation of the pseudo-periodic boundary conditions that allows water molecules to enter and exit the boundaries but does not require vertices belonging to the opposite planes to be aligned.

The approximation comes from a combination of the interface conditions between two compartments and the pseudo-periodic conditions for the exterior boundaries. Since the computational domain Ω is extended periodically, in some cases the cell interfaces touch the exterior boundary $\partial\Omega$. To simplify the explanation, we consider a computational domain $\Omega = [a_1, b_1] \times [a_2, b_2]$ in which a cell is placed touching $r_1 = a_1$ (Fig. 1). The boundary needs to be periodic at $r_1 = a_1$ and $r_1 = b_1$. The cell interior is characterized by a diffusion tensor \mathbf{D}^c . The extra-cellular space is the remaining part and is

characterized by a diffusion tensor \mathbf{D}^e . The cell touches the boundary at the interface Σ , which is the intersection between the cell boundary and $\partial\Omega$.

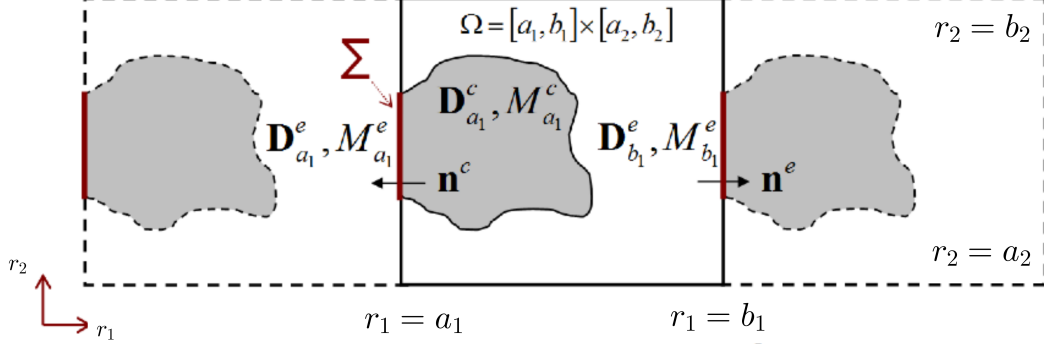


Figure 1: When the cell interface touches $\partial\Omega$, the interface conditions and periodic boundary conditions are combined.

We recall the pseudo-periodic boundary conditions for this specific domain

$$\begin{aligned} M_{a_1}^e &= M_{b_1}^e \exp(\mathcal{I} \theta), \\ \mathbf{D}_{a_1}^e \nabla M_{a_1}^e \cdot \mathbf{n}^c &= -\mathbf{D}_{b_1}^e \nabla M_{b_1}^e \cdot \mathbf{n}^e \exp(\mathcal{I} \theta). \end{aligned} \quad (9)$$

where

$$\theta = \gamma g_1 (b_1 - a_1) \int_0^t f(s) ds,$$

and the interface conditions at Σ are

$$\begin{aligned} \mathbf{D}_{a_1}^c \nabla M_{a_1}^c \cdot \mathbf{n}^c &= \kappa^e (M_{a_1}^e - M_{a_1}^c) \\ \mathbf{D}_{a_1}^e \nabla M_{a_1}^e \cdot \mathbf{n}^e &= \kappa^e (M_{a_1}^c - M_{a_1}^e) \end{aligned} \quad (10)$$

The combination of (9) and (10) with $\mathbf{n}^c = -\mathbf{n}^e$ gives

$$\begin{aligned} \mathbf{D}_{a_1}^c \nabla M_{a_1}^c \cdot \mathbf{n}^c &= \kappa^e (M_{b_1}^e \exp(\mathcal{I} \theta) - M_{a_1}^c), \\ \mathbf{D}_{b_1}^e \nabla M_{b_1}^e \cdot \mathbf{n}^e &= \kappa^e (M_{a_1}^c \exp(-\mathcal{I} \theta) - M_{b_1}^e). \end{aligned} \quad (11)$$

Theoretically, when $\kappa^e \rightarrow +\infty$, Eq. (11) becomes the pseudo-periodic boundary conditions (9). However, we see that practically when $\kappa^e = 1.0 \text{ m/s}$, Eq. (11) is a good approximation of Eq. (9) for free diffusion up to $b = 2000 \text{ s/mm}^2$ corresponding to $S = 2.479\text{e-}3$. If the computed signal is stronger, $\kappa^e = 0.1 \text{ m/s}$ is good enough. Eq. (11) is interesting for the following reasons

- The interface conditions (7) are only a special case of Eq. (11). Their general form is Eq. (12).

$$\begin{aligned} \mathbf{D}^l(\mathbf{r}^l) \nabla M^l(\mathbf{r}^l, t) \cdot \mathbf{n}^l &= \kappa^{ln} \left(M^n(\mathbf{r}^n, t) \exp(\mathcal{I} \theta^{ln}) - M^l(\mathbf{r}^l, t) \right), \\ \mathbf{D}^n(\mathbf{r}^n) \nabla M^n(\mathbf{r}^n, t) \cdot \mathbf{n}^n &= \kappa^{nl} \left(M^l(\mathbf{r}^l, t) \exp(\mathcal{I} \theta^{nl}) - M^n(\mathbf{r}^n, t) \right), \end{aligned} \quad (12)$$

where $\kappa^{ln} = \kappa^{nl}$, $\mathbf{r}^n \in \Gamma^n$, $\mathbf{r}^l \in \Gamma^l$ and

$$\theta^{ln} = -\theta^{nl} = \gamma \mathbf{g} \cdot (\mathbf{r}^n - \mathbf{r}^l) \int_0^t f(s) ds.$$

Further on, instead of solving the complete Bloch-Torrey PDE problem involving the PDE (2), the interface conditions Eqs. (7), the two boundary conditions Eqs. (8), we will solve an approximated model consisting of the Bloch-Torrey equation (2) and the general conditions (12).

- If $M_{a_1}^c$ and $M_{b_1}^e$ are known from the previous time step, the exterior boundaries of the computational domain need not to be treated as periodic. It is also applicable to non confirming interfaces. The constraints of periodicity and confirming interfaces which are usually the most challenging issues in generating meshes can be avoided.
- Since Eq. (12) becomes standard Robin boundary conditions, it is straightforward to implement and parallelize.

3 Discretizations and Implementations on the FEniCS-HPC platform

Let V^l be a function space and $v \in V^l$ be a test function, then the weak form of Eq. (2) on a compartment Ω^l is

$$\frac{\partial}{\partial t} \int_{\Omega^l} v M^l d\mathbf{r} = -\mathcal{I} \gamma f(t) \int_{\Omega^l} v \mathbf{g} \cdot \mathbf{r} M^l d\mathbf{r} - \int_{\Omega^l} \nabla v \cdot \mathbf{D} \nabla M^l d\mathbf{r} + \int_{\partial\Omega^l \cap \Gamma^{ln}} v \mathbf{D} \nabla M^l \cdot \mathbf{n}^l ds \quad (13)$$

Combining with Eq. (12), we get

$$\begin{aligned} \frac{\partial}{\partial t} \int_{\Omega^l} v M^l d\mathbf{r} &= -\mathcal{I} \gamma f(t) \int_{\Omega^l} v \mathbf{g} \cdot \mathbf{r} M^l d\mathbf{r} - \int_{\Omega^l} \nabla v \cdot \mathbf{D} \nabla M^l d\mathbf{r} \\ &+ \int_{\partial\Omega^l \cap \Gamma^{ln}} v \kappa^{ln} \left(M^n(\mathbf{r}^n, t) \exp(\mathcal{I} \theta^{ln}) - M^l(\mathbf{r}^l, t) \right) ds \end{aligned} \quad (14)$$

Assuming that V^l is spanned by basis functions φ_α , i.e $V^l = \text{span}\{\varphi_1, \dots, \varphi_N\}$, and $\xi_\alpha^l(t) \approx M^l(\mathbf{r}_\alpha, t)$, $M^l(\mathbf{r}, t)$ can be presented in a linear combination of basis function, i.e $M^l(\mathbf{r}, t) = \sum_{k=1}^N \varphi_k \xi_k^l$, where N is the number of degrees of freedom. Eq. (14) takes $v = \varphi_\beta, \beta = 1, \dots, N$ to give

$$\begin{aligned} \sum_{\alpha=1}^N \int_{\Omega^l} \varphi_\beta \varphi_\alpha d\mathbf{r} \frac{\partial}{\partial t} \xi_\alpha^l(t) = & -\mathcal{I} \gamma f(t) \sum_{\alpha=1}^N \int_{\Omega^l} \varphi_\beta \mathbf{g} \cdot \mathbf{r} \varphi_\alpha d\mathbf{r} \xi_\alpha^l \\ & - \sum_{\alpha=1}^N \int_{\Omega^l} \nabla \varphi_\beta \cdot \mathbf{D} \nabla \varphi_\alpha d\mathbf{r} \xi_\alpha^l + \kappa^{ln} \sum_{\alpha=1}^N \int_{\partial\Omega^l \cap \Gamma^{ln}} \varphi_\beta \varphi_\alpha ds \xi_\alpha^{ln} \end{aligned} \quad (15)$$

Here $\xi_\alpha^{ln} \approx \xi_\alpha^n(t) \exp(I \theta^{ln}) - \xi_\alpha^l(t) = M^n(\mathbf{r}_\alpha^n, t) \exp(I \theta^{ln}) - M^l(\mathbf{r}_\alpha^l, t)$. We note that $M^n(\mathbf{r}_\alpha^n, t)$ comes from $\Gamma^{nl} \subset \Omega^n$ but its values should be projected onto $\Gamma^{ln} \subset \Omega^l$. The spatial position \mathbf{r}_α^n should be computed such that it aligns with \mathbf{r}_α^l . We Assume that $\mathbf{r}_\alpha^l = (r_i^l)_{i=1, \dots, d}$ and $\mathbf{r}_\alpha^n = (r_i^n)_{i=1, \dots, d}$. If \mathbf{r}_α^l belongs to the interfaces between two compartments, then $\mathbf{r}_\alpha^n = \mathbf{r}_\alpha^l$. If \mathbf{r}_α^l belongs to the plane $r_i = a_i$, i.e $r_i^l = a_i$, then we will set $r_i^n = b_i$ and $r_k^n = r_k^l$ for all $k \neq i$. In reverse, if \mathbf{r}^l belongs to the plane $r_i = b_i$, i.e $r_i^l = b_i$, then we will set $r_i^n = a_i$ and $r_k^n = r_k^l$ for all $k \neq i$. Since \mathbf{r}_α^n is computed with respect to \mathbf{r}_α^l , the value of ξ_α^n at \mathbf{r}_α^n may not be available and interpolation will be used.

Let ξ and φ is a column vectors of degrees of freedom and basis functions, i.e $\xi = (\xi_k)$ and $\varphi = (\varphi_k)$, Eq. (15) can be rewritten in a matrix form

$$\mathbf{M}^l \frac{\partial}{\partial t} \xi^l = -\mathcal{I} \gamma f(t) \mathbf{J}^l \xi^l - \mathbf{S} \xi^l + \kappa^{ln} \mathbf{F}^l \xi^{ln} \quad (16)$$

Since ξ^l is complex-valued, we can write $\xi^l = \xi_{\mathcal{R}}^l + \mathcal{I} \xi_{\mathcal{I}}^l$. Eq. (16) is then decoupled to two equations

$$\begin{aligned} \mathbf{M}^l \frac{\partial}{\partial t} \xi_{\mathcal{R}}^l &= \gamma f(t) \mathbf{J}^l \xi_{\mathcal{I}}^l - \mathbf{S} \xi_{\mathcal{R}}^l + \kappa^{ln} \mathbf{F}^l \xi_{\mathcal{R}}^{ln} \\ \mathbf{M}^l \frac{\partial}{\partial t} \xi_{\mathcal{I}}^l &= -\gamma f(t) \mathbf{J}^l \xi_{\mathcal{R}}^l - \mathbf{S} \xi_{\mathcal{I}}^l + \kappa^{ln} \mathbf{F}^l \xi_{\mathcal{I}}^{ln} \end{aligned} \quad (17)$$

where

$$\begin{aligned} \mathbf{M}^l &= \int_{\Omega^l} \varphi \varphi^T d\mathbf{r}, \quad \mathbf{S}^l = \int_{\Omega^l} \nabla \varphi \cdot \nabla \varphi^T d\mathbf{r}, \quad \mathbf{J}^l = \int_{\Omega^l} \mathbf{g} \cdot \mathbf{r} \varphi \varphi^T d\mathbf{r}, \\ \mathbf{F}^l &= \int_{\partial\Omega^l} \chi \varphi \varphi^T ds, \quad \chi(\mathbf{r}) = \begin{cases} 1 & \mathbf{r} \in \partial\Omega \cap \Gamma^{ln} \\ 0 & \text{otherwise} \end{cases}. \end{aligned}$$

Before writing a high-level "solver" in C++ using DOLFIN-HPC, we define the matrices in UFL "form files", closely mapping mathematical notation. The form files are then compiled to low-level C++ source code with FFC.

Matrices \mathbf{M} and \mathbf{J} can be defined in a same form with $f = 1$ for matrix \mathbf{M} and $f = \mathbf{g} \cdot \mathbf{r}$ for matrix \mathbf{J} . Here f is a dolfin "Function" defined in the "solver".

```
V = FiniteElement("CG", "tetrahedron", 1)
u = TrialFunction(V)
v = TestFunction(V)
f = Coefficient(V)
a = f*u*v*dx
```

The stiffness matrix \mathbf{S} with a diffusion coefficient D is defined as

```
V = FiniteElement("CG", "tetrahedron", 1)
u = TrialFunction(V)
v = TestFunction(V)
D = Coefficient(V)
a = D*dot(grad(u), grad(v))*dx
```

Finally, the matrix \mathbf{F} with a boundary marker χ is defined as

```
V = FiniteElement("CG", "tetrahedron", 1)
v = TestFunction(V)
chi = Coefficient(V)
L = chi*v*ds
```

For the time discretization, similar to [10, 14] we apply the second-order adaptive Runge-Kutta Chebyshev (RKC) [17]. This method has second order in time and is especially suitable for parabolic PDEs. Since this method is explicit, the right-hand side of Eq. (17) is known and the two equations can be solved separately. With the same reason, each equation in turn is solved independently on each compartment. To adapt the FEniCS-HPC platform, the serial code of the RKC available at <http://www.netlib.org/ode/rkc.f> was rewritten and parallelized.

4 Results

4.1 Validation of the framework

We validate the framework through two examples: a two-layer infinitely-long cylinder and a two-layer sphere. In the first example, we show that the pseudo-periodic boundary conditions can be well approximated by imposing high permeabilities at the exterior boundaries that solves difficulties of mesh generation and well facilitates for parallelization. In the second example, we show that the framework works efficiently in a wide range of gradient strength.

Signals for a two-layer infinitely-long cylinder

In this sample, we compute signals for a cylinder which is composed of two lay-

ers with $R_1 = 3\mu\text{m}$ and $R_2 = 5\mu\text{m}$ and $L = 5\mu\text{m}$. The diffusion coefficients are $D_1 = D_2 = 3\text{e-}3\text{ mm}^2/\text{s}$. We allow water molecules to enter and exit the computational domain in the z -direction by imposing $\kappa^e = 0.1\text{ m/s}$. So, the signals represent the diffusion in an infinitely-long cylinder. Between two layers, a permeability of $\kappa=5\text{e-}5\text{ m/s}$ is imposed and the diffusion time is $\delta = \Delta = 10\text{ms}$.

The mesh was generated in such a way that it is totally non conforming and non-periodic. The interior layer is twice as fine as the exterior one. The top base is twice as coarse as the bottom (Fig. 2a).

We performed simulations for three gradient directions $\frac{\mathbf{g}}{|\mathbf{g}|} = [1, 0, 0]$, $\frac{[1,0,1]}{\sqrt{2}}$ and $[0, 0, 1]$. The exact signals are computed by a combination between free diffusion and matrix formalism. In all three gradient directions, we obtain good approximations. For high b -values, the approximations are less good since the signals are too weak (Fig. 2b).

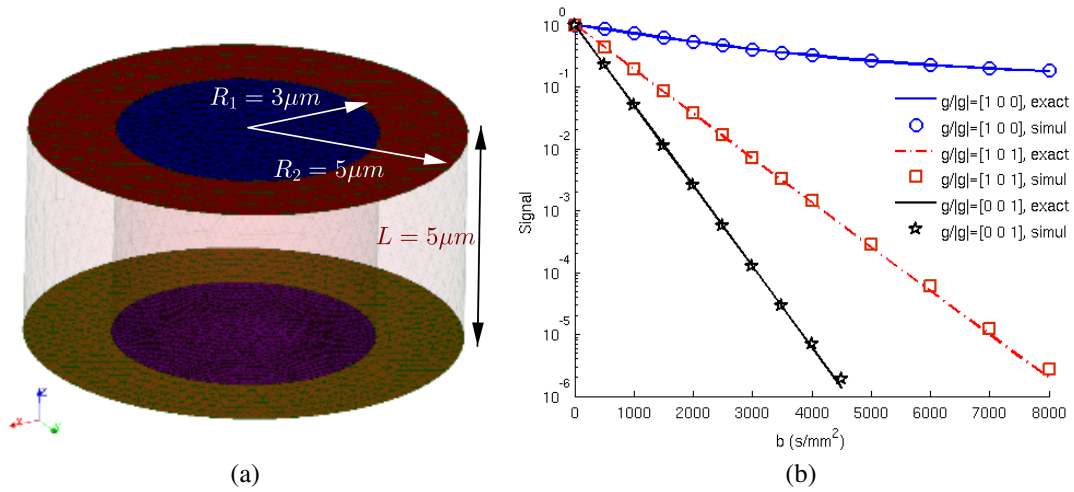


Figure 2: Signal on a double-layer cylinder with $R_1 = 3\mu\text{m}$ and $R_2 = 5\mu\text{m}$, $L = 5\mu\text{m}$. The diffusion coefficients are $D_1 = D_2 = 3\text{e-}3\text{ mm}^2/\text{s}$. The permeability is $\kappa=5\text{e-}5\text{ m/s}$ for the membrane between two layers. The approximation of the pseudo-periodic boundary conditions is applied in z -direction with $\kappa^e = 0.1\text{ m/s}$.

Signals for a two-layer sphere

In this example, we compute signals on a double-layer spherical cell with $R_1 = 20\mu\text{m}$ and $R_2 = 40\mu\text{m}$ (Fig. 3a). A wide range of gradient strengths [3, 13] is used. The signals are computed and plotted against b -values and q -values. In Fig. 3b, the gradient strength q is varied between 0 and 1.5 T/m for $\delta=1\text{e-}3\text{s}$, $\Delta=1\text{s}$. Two mesh sizes of 1922 and 8276 vertices are considered. For high gradient strength, finer mesh is needed to give good approximation. When computed against b -values, the gradient strength is

large for short-time diffusion and smaller for long-time diffusion. Three time sequences, $\delta = \Delta = 0.1, 0.5, 1.0$ s with b varied between 0 and $1e5$ s/mm² are considered (Fig. 3c). The largest gradient strength is 0.045 T/m for $\delta = \Delta = 0.1$ s and $b=1e5$ s/mm². Numerical signals are compared to the reference ones computed by matrix formalism [3]. The diffusion coefficients are $D_1 = D_2 = 3e-3$ mm²/s, the permeability for the membrane between two layers is $\kappa=5e-5$ m/s, and $\kappa^e = 0$ is set for exterior boundaries.

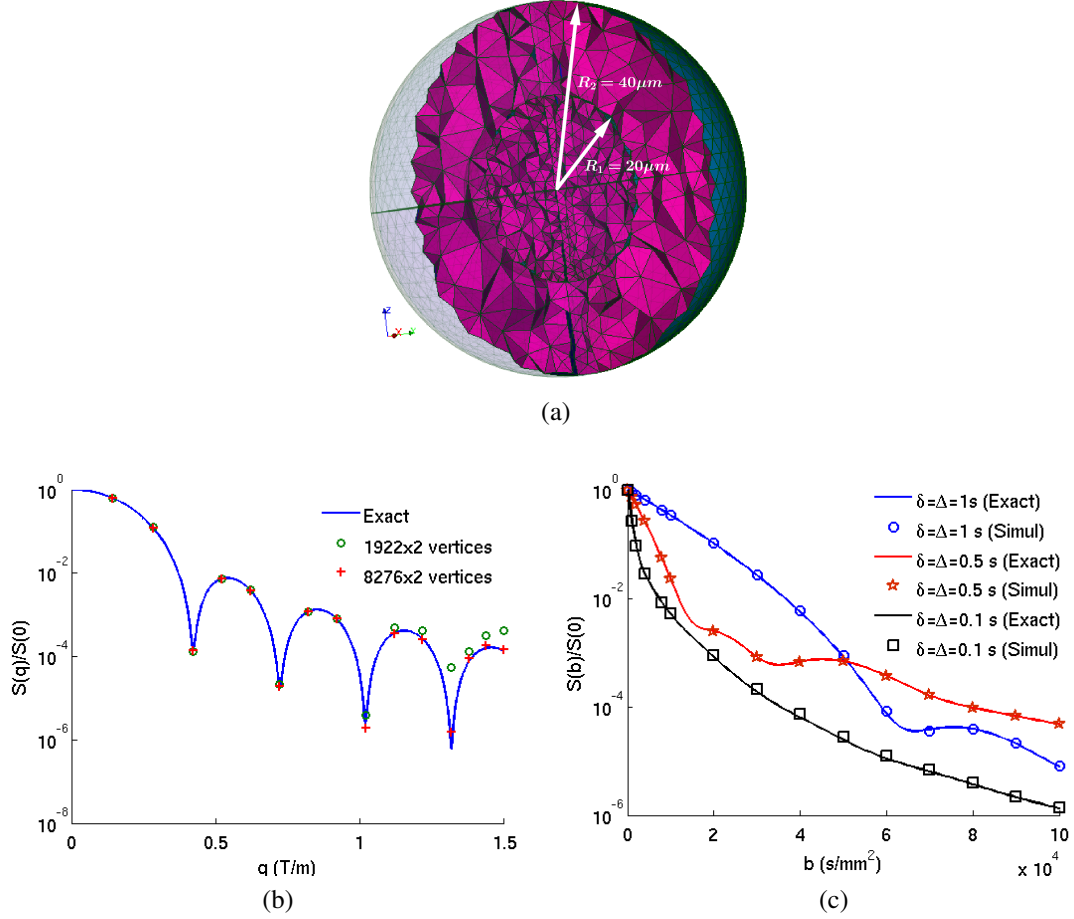


Figure 3: Signal on a double-layer spherical cell with $R_1 = 20\mu\text{m}$ and $R_2 = 40\mu\text{m}$. The diffusion coefficients are $D_0=D_1=3e-3$ mm²/s, the permeability for the membrane between two layers is $\kappa=5e-5$ m/s, and $\kappa^e = 0$ is set for exterior boundaries. The gradient strength is varied between 0 and 1.5 T/m for $\delta=1e-3$ s, $\Delta=1$ s and two mesh sizes of 1922 and 8276 vertices are considered (a). When computed against b -values, the gradient strength is large for short-time diffusion and smaller for long-time diffusion (b). Three time sequences, $\delta = \Delta = 0.1, 0.5, 1.0$ s with b varied between 0 and $1e5$ s/mm² are considered.

Each b-value is computed in between 5 and 10 minutes using 4 MPI processes for the mesh size of 8276×2 vertices. Timing for each q is less than 2 minutes for the mesh size of 1922×2 and 15 minutes for the mesh size of 8276×2 vertices.

4.2 Speedup ratio and parallel efficiency

We clarify the timing, speedup ratio and parallel efficiency on the Hydra and Beskow computers for a sphere of radius $R = 80\mu\text{m}$ with $b = 4000 \text{ s/mm}^2$ and the diffusion time $\delta = \Delta = 100\text{ms}$. On the Hydra a moderate-scale problem with a tetrahedral mesh of 124886 vertices and on the Beskow a large-scale problem with 4307475 vertices are considered. The number of MPI processes vary between 1 and 42 on the Hydra and between 4 and 232 on the Beskow.

On the Hydra, the serial computation is about 1.7 hours and the fastest computation with 42 MPI processes is about 4 minutes. On the Beskow, the timing is about 14.7 hours for 4 MPI processes and 20 minutes for 232 MPI processes. See Fig. 4.

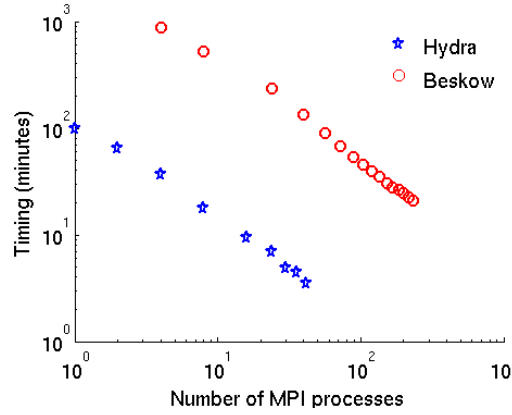


Figure 4: Timing tested on the Hydra and on the Beskow for a sphere of radius $R = 80\mu\text{m}$ with $b = 4000 \text{ s/mm}^2$ and the diffusion time $\delta = \Delta = 100\text{ms}$. The tetrahedral mesh has 124886 vertices on the Hydra and 4307475 vertices on the Beskow.

The speedup ratio is the ratio between timing for serial execution T_{serial} and timing for parallel execution T_{parallel} , i.e

$$\mathcal{S}(p) = \frac{T_{\text{serial}}}{T_{\text{parallel}}(p)} \quad (18)$$

where p is the number of MPI processes used in the parallel execution.

The ideal speedup is $\mathcal{S}_{\text{ideal}}(p) = p$, i.e when p MPI processes are used, the parallel execution will be p times faster than the serial execution.

The parallel efficiency is computed by

$$\mathcal{E} = \frac{\mathcal{S}}{p} \quad (19)$$

In Fig. 5, we show the speedup ratio and parallel efficiency corresponding to the timing in Fig. 4. We obtain a strong scaling on both computers and the parallel efficiency is always higher than 60%.

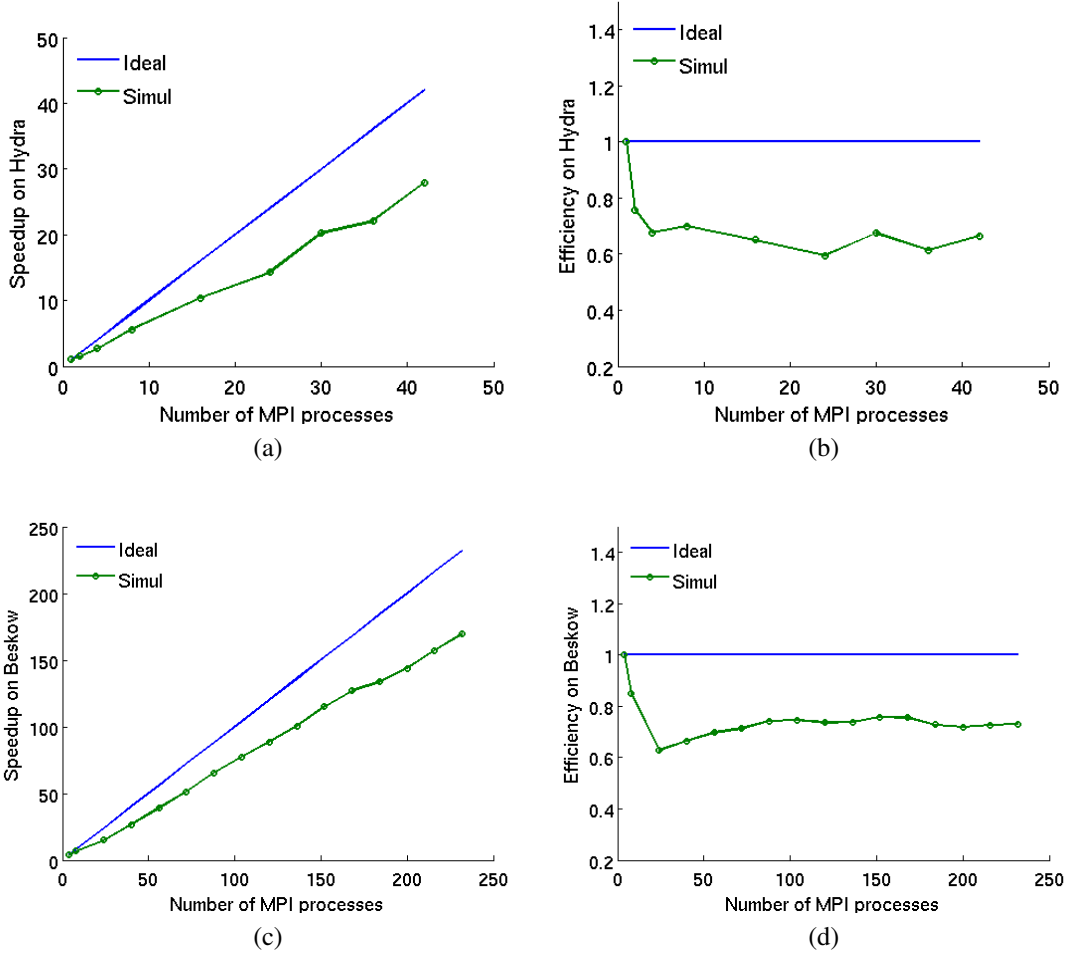


Figure 5: Speedup ratio and parallel efficiency on the Hydra and on the Beskow for a sphere of radius $R = 80\mu\text{m}$ with $b = 4000\text{ s/mm}^2$ and the diffusion time $\delta = \Delta = 100\text{ms}$. The tetrahedral mesh has 124886 vertices on the Hydra and 4307475 vertices on the Beskow. We obtain a strong scaling on both computers and the parallel efficiency is always higher than 60%.

5 Conclusion and future works

We proposed a FEniCS-HPC framework for the multi-compartment Bloch-Torrey models applied for diffusion NMR and diffusion MRI. This framework works efficiently for a wide range of gradient strengths and shows promise in validation against more realistic models. A comparison with other methods will be performed and more realistic applications will be considered in the future.

Acknowledgments

The first author would like to thank Jing-Rebecca Li and Denis Grebenkov for their teachings about the multi-compartment Bloch-Torrey models applied to diffusion NMR and diffusion MRI.

The authors would like to thank Niyazi Cem Degirmenci and Jeannette Spühler for their enthusiasm helps with the FEniCS-HPC platform.

This work has been funded by the Swedish Foundation for Strategic Research, the European Research Council, and the Swedish Research Council. The simulations were performed on resources provided by the SNIC.

References

- [1] L. BELTRACHINI, Z. A. TAYLOR, AND A. F. FRANGI, *A parametric finite element solution of the generalised Bloch-Torrey equation for arbitrary domains*, Journal of Magnetic Resonance, 259 (2015), pp. 126 – 134.
- [2] FENICS, *Fenics project*, <http://www.fenicsproject.org>, (2003).
- [3] D. S. GREBENKOV, *Pulsed-gradient spin-echo monitoring of restricted diffusion in multilayered structures*, Journal of Magnetic Resonance, 205 (2010), pp. 181–195.
- [4] J. HOFFMAN, J. JANSSON, R. V. DE ABREU, N. C. DEGIRMENCI, N. JANSSON, K. MÜLLER, M. NAZAROV, AND J. H. SPÜHLER, *Unicorn: Parallel adaptive finite element simulation of turbulent flow and fluid-structure interaction for deforming domains and complex geometry*, Computers and Fluids, (2012).
- [5] J. HOFFMAN, J. JANSSON, C. DEGIRMENCI, N. JANSSON, AND M. NAZAROV, *Unicorn: a Unified Continuum Mechanics Solver*, Springer, 2012, ch. 18.
- [6] J. HOFFMAN, J. JANSSON, N. JANSSON, C. JOHNSON, AND R. V. DE ABREU, *Turbulent flow and fluid-structure interaction*, in Automated Solutions of Differential Equations by the Finite Element Method, Springer, 2011.

- [7] J. HOFFMAN, J. JANSSON, N. JANSSON, AND M. NAZAROV, *Unicorn: A unified continuum mechanics solver*, in *Automated Solutions of Differential Equations by the Finite Element Method*, Springer, 2011.
- [8] N. JANSSON, J. HOFFMAN, AND J. JANSSON, *Framework for Massively Parallel Adaptive Finite Element Computational Fluid Dynamics on Tetrahedral Meshes*, SIAM J. Sci. Comput., 34 (2012), pp. C24–C41.
- [9] R. C. KIRBY, *FIAT: Numerical Construction of Finite Element Basis Functions*, Springer, 2012, ch. 13.
- [10] J.-R. LI, D. CALHOUN, C. POUPON, AND D. L. BIHAN, *Numerical simulation of diffusion mri signals using an adaptive time-stepping method*, Physics in Medicine and Biology, 59 (2014), p. 441.
- [11] A. LOGG, K.-A. MARDAL, G. N. WELLS, ET AL., *Automated Solution of Differential Equations by the Finite Element Method*, Springer, 2012.
- [12] ———, *Automated Solution of Differential Equations by the Finite Element Method*, Springer, 2012.
- [13] B. F. MORONEY, T. STAIT-GARDNER, B. GHADIRIAN, N. N. YADAV, AND W. S. PRICE, *Numerical analysis of NMR diffusion measurements in the short gradient pulse limit*, Journal of Magnetic Resonance, 234 (2013), pp. 165–175.
- [14] D. V. NGUYEN, J.-R. LI, D. GREBENKOV, AND D. L. BIHAN, *A finite elements method to solve the Bloch - Torrey equation applied to diffusion magnetic resonance imaging*, Journal of Computational Physics, 263 (2014), pp. 283 – 302.
- [15] W. S. PRICE, *NMR studies of translational motion: principles and applications*, Cambridge University Press, 2009.
- [16] G. RUSSELL, K. D. HARKINS, T. W. SECOMB, J.-P. GALONS, AND 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 (2012), p. N35.
- [17] B. P. SOMMEIJER, L. F. SHAMPINE, AND J. G. VERWER, *RKC: An explicit solver for parabolic PDEs*, Journal of Computational and Applied Mathematics, 88 (1998), pp. 315–326.
- [18] E. O. STEJSKAL AND J. E. TANNER, *Spin diffusion measurements: Spin echoes in the presence of a time-dependent field gradient*, The Journal of Chemical Physics, 42 (1965), pp. 288–292.

- [19] H. TORREY, *Bloch equations with diffusion terms*, Physical Review Online Archive (Prola), 104 (1956), pp. 563–565.
- [20] J. XU, M. DOES, AND J. GORE, *Numerical study of water diffusion in biological tissues using an improved finite difference method*, Physics in medicine and biology, 52 (2007).