Journal of Magnetic Resonance 329 (2021) 107011

Contents lists available at ScienceDirect

Journal of Magnetic Resonance

journal homepage: www.elsevier.com/locate/jmr

Accuracy and performance analysis for Bloch and Bloch-McConnell simulation methods

Christina Graf [a], Armin Rund [b], Christoph Stefan Aigner [c], Rudolf Stollberger [a,*]

[a] Institute of Medical Engineering, Graz University of Technology, Graz, Austria
[b] Institute for Mathematics and Scientific Computing, University of Graz, Graz, Austria
[c] Physikalisch-Technische Bundesanstalt (PTB), Braunschweig and Berlin, Germany

ARTICLE INFO

Article history:
Received 6 November 2020
Revised 12 May 2021
Accepted 26 May 2021
Available online 29 May 2021

Keywords:
Bloch equations
Bloch-McConnell equations
Exact solution
Magnetic resonance imaging
Numerical solution
Operator splitting

ABSTRACT

Purpose: To introduce new solution methods for the Bloch and Bloch-McConnell equations and compare them quantitatively to different known approaches.
Theory and Methods: A new exact solution per time step is derived by means of eigenvalues and generalized eigenvectors. Fast numerical solution methods based on asymmetric and symmetric operator splitting, which are already known for the Bloch equations, are extended to the Bloch-McConnell equations. Those methods are compared to other numerical methods including spin domain, one-step and multistep methods, and matrix exponential. Error metrics are introduced based on the exact solution method, which allows to assess the accuracy of each solution method quantitatively for arbitrary example data.
Results: Accuracy and performance properties for nine different solution methods are analyzed and compared in extensive numerical experiments including various examples for non-selective and slice-selective MR imaging applications. The accuracy of the methods heavily varies, in particular for short relaxation times and long pulse durations.
Conclusion: In absence of relaxation effects, the numerical results confirm the rotation matrices approach as accurate and computationally efficient Bloch solution method. Otherwise, as well as for the Bloch-McConnell equations, symmetric operator splitting methods are recommended due to their excellent numerical accuracy paired with efficient run time.

1. Introduction

The Bloch equations [1] are the basis for modeling the spatial-temporal behaviour of the macroscopic magnetization of non-interacting spins in time varying external fields [2,3]. Bloch simulations are frequently exploited for sequence parameter optimization [4,5], MR fingerprinting [6–9], model-based reconstruction [10,11], RF pulse design based on small-tip-angle approximation [12], on perturbation analysis [13] or on optimal control [14–21]. The Bloch-McConnell equations [22] are an extended Bloch model for chemical exchange for various applications including Magnetization Transfer (MT) or Chemical Exchange Saturation Transfer (CEST) [23] imaging. In addition to the general Bloch-McConnell equations, magnetization transfer is described by the model of Graham and Henkelman, [24].

Sometimes, Bloch models can be simplified, typically by neglecting relaxation effects during radio frequency (RF) excitation, e.g. for small pulse durations and large relaxation times. However, simulations with neglected relaxation effects can be too inaccurate for other applications such as arterial spin labeling, [25], which requires a high simulation accuracy. The choice and implementation of a solution method is crucial for reliable simulation of Bloch models. Mathematically, exact/analytical solutions that give the highest accuracy can be derived only for certain classes of ordinary differential equations, which typically results in restrictions to the parameters or discretization of the model. Otherwise, adequate numerical solution methods need to be applied. This paper introduces, analyses and compares different solution method and gives recommendations for their use.

Different exact solution methods for Bloch equations with neglected relaxation effects exist. For a certain discretization (piecewise constant magnetic field) exact solutions can be computed using rotations in magnetization domain [3], in spin domain [26–28], or by quaternions [29]. Beyond, in the special case of a hyperbolic secant function as driving field, a full analytical solution

[*] Corresponding author at: Institute of Medical Engineering, Graz University of Technology, Stremayrgasse 16/III, A-8010 Graz, Austria
E-mail address: rudolf.stollberger@tugraz.at (R. Stollberger).

https://doi.org/10.1016/j.jmr.2021.107011
1090-7807/© 2021 The Authors. Published by Elsevier Inc.

can be given without the restriction of piecewise constantness [30]. In contrast, an analytical solution to the full Bloch equations with relaxation effects and non-constant driving fields does not exist in general, [31]. Even for piecewise constant fields, no general exact solution has been published, [32], however, it will be introduced in this work. An exact solution was only presented for the special case when the longitudinal relaxation time equals the transverse relaxation time, [32]. Other methods that include relaxation effects and an arbitrary, but piecewise constant pulse shape are based on matrix exponentials, whose numerical computation is not trivial and non-exact in the general case [33]. In absence of exact solution methods, numerical solution methods are applied, e.g. operator splitting methods for the Bloch model [34–37], or matrix exponential approaches for the Bloch-McConnell model [38–40]. A rigorous analysis of the accuracy of all these methods for arbitrary example data was not possible so far, since an exact solution method for the general case was missing for numerical comparison. However, an accurate solution method is indispensable for example when designing RF pulses which require a high efficiency such as background suppression pulses for arterial spin labeling [25].

There are three main contributions of this work, all using the widespread assumption of a piecewise constant magnetic field. First, an exact solution method is presented for the first time, that covers the Bloch solution in the general case. The method is based on eigenvalues and generalized eigenvectors. Second, an improved, symmetric operator splitting [41–43] is applied to the Bloch equations and tested on the hypothesis of a minimal numerical error. A sophisticated adaption of this operator splitting was derived and applied for the Bloch-McConnell equations. Third, exact error metrics based on the new exact solver are exploited in extensive numerical experiments, analyzing and comparing the broad set of typical solution methods including matrix exponentials [44,45], asymmetric operator splitting, the Implicit Euler scheme [46,47], the Crank-Nicolson method [48] and a Runge-Kutta method [49]. To accommodate reproducible research, parts of the code used to generate the results for this paper are available at https://github.com/GrafChristina/BlochSim.

2. Theory

In the following, the Bloch equations with and without relaxation effects as well as the Bloch-McConnell equations are introduced. Section 2.1 introduces exact eigenvalue and eigenvector (2.1.2) and approximate asymmetric and symmetric operator splitting schemes (2.1.1) for the Bloch equations. In Section 2.2 we present how to extend the exact eigenvalue and eigenvector and approximate asymmetric and symmetric operator splitting schemes to the Bloch-McConnell equations. Section 2.3 introduces the model of Graham and Henkelman [24] to describe the magnetization transfer.

2.1. Bloch equations

The temporal evolution of the magnetization vector $\mathbf{M}(t,\mathbf{r}) \in \mathbb{R}^3$ in an external magnetic field $\mathbf{B}(t,\mathbf{r}) \in \mathbb{R}^3$ can be described by the Bloch equations. They are given in the rotating frame of reference as

$$\begin{cases} \frac{d\mathbf{M}}{dt}(t,\mathbf{r}) = \gamma\mathbf{B}(t,\mathbf{r}) \times \mathbf{M}(t,\mathbf{r}) + \mathbf{R}(\mathbf{M}(t,\mathbf{r})) = \mathbf{A}(t,\mathbf{r}) \cdot \mathbf{M}(t,\mathbf{r}) + \mathbf{b} \quad \text{for} \quad t \in (0,T), \\ \mathbf{M}(0,\mathbf{r}) = \mathbf{M}^0(\mathbf{r}). \end{cases} \tag{1}$$

Therein, γ is the gyromagnetic ratio, $\mathbf{M}^0$ the initial magnetization, T with $t \in (0,T)$ the terminal time and $\mathbf{r} \in \Omega$ the spatial position with Ω being the spatial domain. The magnetic field $\mathbf{B}(t,\mathbf{r}) = (B_{1,x}(t), B_{1,y}(t), \mathbf{G}_s(t)\cdot\mathbf{r})^\top$ contains the complex radio frequency pulse $B_1(t) = B_{1,x}(t) + iB_{1,y}(t)$ and the slice-selective gradient $\mathbf{G}_s(t)$. $\mathbf{R}(\mathbf{M}(t,\mathbf{r}))$ is the relaxation term defined as

$$\mathbf{R}(\mathbf{M}(t,\mathbf{r})) = (-M_x(t,\mathbf{r})/T_2, -M_y(t,\mathbf{r})/T_2, -(M_z(t,\mathbf{r}) - M_0)/T_1)^\top, \tag{2}$$

where $M_0 = \left|\mathbf{M}^0\right|$ with longitudinal relaxation time T_1 and transverse relaxation time T_2, see [1]. $\mathbf{A}(t,\mathbf{r})$ and $\mathbf{b}$ are given as

$$\mathbf{A}(t,\mathbf{r}) = \begin{pmatrix} -\frac{1}{T_2} & \gamma\mathbf{G}_s(t)\cdot\mathbf{r} & -\gamma B_{1,y}(t) \\ -\gamma\mathbf{G}_s(t)\cdot\mathbf{r} & -\frac{1}{T_2} & \gamma B_{1,x}(t) \\ \gamma B_{1,y}(t) & -\gamma B_{1,x}(t) & -\frac{1}{T_1} \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} 0 \\ 0 \\ \frac{M_0}{T_1} \end{pmatrix}. \tag{3}$$

Treating the Bloch equations as an ordinary differential equation (with arbitrary, but fixed $\mathbf{r}$), the coefficients in the differential equation are non-constant, so a full analytical solution assuming all influences (e.g. relaxation effects, RF and $\mathbf{G}_s$) does not exist in general. The situation changes when a piecewise constant discretization is applied for RF and $\mathbf{G}_s$ on a time grid $0 = t_0 < \ldots < t_N = T$ with constant system matrices $\mathbf{A}_n, n = 1,\ldots,N$ valid on time interval $[t_{n-1}, t_n)$, which is typically used as model for describing the generation of RF waveforms on MR scanners. Assuming neglected relaxation effects, each time step can be solved analytically by rotation matrices [26]. In the general case that relaxation effects, RF and $\mathbf{G}_s$ are present simultaneously, however, rotation matrices alone do not give an exact solution per time step, but will result in a numerical error. This error can be reduced substantially by considering relaxation via operator splitting for the numerical solution, or by an exact solution based on generalized eigenvectors, which both will be discussed next.

2.1.1. Operator splitting methods

We investigate two operator splitting techniques for the Bloch time step, where the system matrices $\mathbf{A}_n = \mathbf{R}_n + \mathbf{C}$ are decomposed additively into a rotation part

$$\mathbf{R}_n = \begin{pmatrix} 0 & \gamma\mathbf{G}_s(t_{n-1})\cdot\mathbf{r} & -\gamma B_{1,y}(t_{n-1}) \\ -\gamma\mathbf{G}_s(t_{n-1})\cdot\mathbf{r} & 0 & \gamma B_{1,x}(t_{n-1}) \\ \gamma B_{1,y}(t_{n-1}) & -\gamma B_{1,x}(t_{n-1}) & 0 \end{pmatrix} \tag{4}$$

and a relaxation part $\mathbf{C} = \mathrm{diag}\left(-\frac{1}{T_2}, -\frac{1}{T_2}, -\frac{1}{T_1}\right)$. Now the asymmetric operator splitting scheme (ASY) that is applied in [34–36] is defined as

$$\frac{d\mathbf{M}_{rot}}{dt} = \mathbf{R}_n \cdot \mathbf{M}_{rot}, \qquad \mathbf{M}_{rot}(0) = \mathbf{M}^{n-1}, \tag{5a}$$

$$\frac{d\mathbf{M}_{rel}}{dt} = \mathbf{C}\cdot\mathbf{M}_{rel} + \mathbf{b}, \qquad \mathbf{M}_{rel}(0) = \mathbf{M}_{rot}(\tau_n), \tag{5b}$$

where $\tau_n = t_n - t_{n-1}$ is the time step size and $\mathbf{M}^n$ is defined as

$$\mathbf{M}^n = \mathbf{M}_{rel}(\tau_n). \tag{6}$$

In contrast, a symmetric operator splitting (Strang splitting, SY) [41,42] scheme is defined as

$$\frac{d\mathbf{M}_{rot,1}}{dt} = \mathbf{R}_n \cdot \mathbf{M}_{rot,1}, \qquad \mathbf{M}_{rot,1}(0) = \mathbf{M}^{n-1}, \tag{7a}$$

$$\frac{d\mathbf{M}_{rel}}{dt} = \mathbf{C}\cdot\mathbf{M}_{rel} + \mathbf{b}, \qquad \mathbf{M}_{rel}(0) = \mathbf{M}_{rot,1}\left(\frac{1}{2}\tau_n\right), \tag{7b}$$

$$\frac{d\mathbf{M}_{rot,2}}{dt} = \mathbf{R}_n \cdot \mathbf{M}_{rot,2}, \qquad \mathbf{M}_{rot,2}(0) = \mathbf{M}_{rel}(\tau_n), \tag{7c}$$

where $\mathbf{M}^n$ is given as

$$\mathbf{M}^n = \mathbf{M}_{rot,2}(\tau_n). \tag{8}$$

Here, all the subsystems are solved exactly using rotation matrices (resulting in an operator G_{rot}) for the rotation part and exponential functions (resulting in an operator G_{rel}) for the relaxation part. Then we can write one time step of the asymmetric operator splitting as

$$\mathbf{M}^n = G_{\text{rel}}(\tau_n) \cdot G_{\text{rot}}(\tau_n) \cdot \mathbf{M}^{n-1} \tag{9}$$

and one time step of the symmetric operator splitting is

$$\mathbf{M}^n = G_{\text{rot}}\left(\frac{\tau_n}{2}\right) \cdot G_{\text{rel}}(\tau_n) \cdot G_{\text{rot}}\left(\frac{\tau_n}{2}\right) \cdot \mathbf{M}^{n-1}. \tag{10}$$

The operator G_{rel} is defined as $G_{\text{rel}}(\tau_n) = diag\left(\exp\left(-\frac{1}{T_2}\tau_n\right), \exp\left(-\frac{1}{T_2}\tau_n\right), \exp\left(-\frac{1}{T_1}\tau_n\right)\right)$. The rotation operator G_{rot} is defined via a rotation axis and rotation angle derived from RF and $\mathbf{G}_s$. The detailed description of G_{rot} is summarized in Appendix 1. The symmetric operator splitting method possesses a larger order of convergence compared to the asymmetric method: The numerical error due to the operator splitting is known to be of first order in terms of the time step size for the asymmetric operator splitting scheme, and of second order for the symmetric operator splitting scheme [50]. Practically this results in a significantly larger drop of the error for the symmetric splitting method when the time grid is refined.

2.1.2. Exact solution

To the best of our knowledge, an exact numerical solution for the Bloch time step in the general case (with simultaneously non-zero RF, $\mathbf{G}_s$, and relaxation terms) has not been presented so far. It is introduced here based on generalized eigenvectors. An implementation of this exact solution method serves as ground truth for the analysis of numerical errors of other solution methods in the numerical experiments.

The exact solution of a time step is obtained using solution theory of ordinary differential equations (ODE) with constant coefficients [31], which is briefly outlined. First, the homogeneous solution is computed by eigenvalue analysis of the Bloch matrix $\mathbf{A}_n$. The eigenvalues and eigenvectors depend on the values of RF, $\mathbf{G}_s$ and relaxation times, they are obtained from MATLAB's `eig` function. However, in order to solve the Bloch model in the general case, the generalized eigenvectors [31, Thm. 6.7.1] need to be taken into account. Based on the detected eigenvalues, seven different cases need to be distinguished. The resulting general homogeneous solution $\mathbf{M}_h(t)$ is summarized in Appendix 2.

With the well-known particular solution of a Bloch time step $\mathbf{M}_p(t) = -\mathbf{A}_n^{-1}\mathbf{b}$ superposition gives the general inhomogeneous solution $\mathbf{M}(t) = \mathbf{M}_h(t) + \mathbf{M}_p(t)$, which still contains three unknown integration constants c_1, c_2, c_3 from the general homogeneous solution. Finally, the exact solution of a time step can be obtained by determining theses constants from the initial condition $\mathbf{M}(0) = \mathbf{M}^{n-1}$ via solving a system of linear equations. Since the method is based on eigenvalues and eigenvectors, it is abbreviated with **EV** throughout this paper.

2.2. Bloch-McConnell equations

For a water proton pool w and two solute proton pools or a solute and a bound pool s_1 and s_2, the Bloch-McConnell equations with magnetization transfer can be described as a system of coupled Bloch equations

$$\begin{cases} \frac{d\mathbf{M}}{dt}(t, \mathbf{r}) = \mathbf{A}(t, \mathbf{r}) \cdot \mathbf{M}(t, \mathbf{r}) + \mathbf{b}, \\ \mathbf{M}(0, \mathbf{r}) = \mathbf{M}^0(\mathbf{r}). \end{cases} \tag{11}$$

The system matrix $\mathbf{A}(t, \mathbf{r})$ is a block matrix

$$\mathbf{A}(t, \mathbf{r}) = \begin{pmatrix} \mathbf{A}_{11}(t, \mathbf{r}) & k_{s_1,w}\mathbf{I} & k_{s_2,w}\mathbf{I} \\ k_{w,s_1}\mathbf{I} & \mathbf{A}_{22}(t, \mathbf{r}) & k_{s_2,s_1}\mathbf{I} \\ k_{w,s_2}\mathbf{I} & k_{s_1,s_2}\mathbf{I} & \mathbf{A}_{33}(t, \mathbf{r}) \end{pmatrix} \tag{12}$$

where $\mathbf{I}$ is the 3×3 identity matrix. See [39] for the full description. The inhomogeneity reads

$$\mathbf{b} = \left(0, \quad 0, \quad \frac{M_{0,w}}{T_{1,w}}, \quad 0, \quad 0, \quad \frac{M_{0,s_1}}{T_{1,s_1}}, \quad 0, \quad 0, \quad \frac{M_{0,s_2}}{T_{1,s_2}}\right)^{\top}. \tag{13}$$

$\mathbf{M} \in \mathbb{R}^9$ describes the magnetization of the water proton pool and both solute proton pools. M_0, T_1 and T_2 are given for each pool individually. The Bloch-McConnell equations can be extended analogously to a higher number of solute proton pools. We use above's Eq. (11) also to describe the MT effect [51].

Again, a piecewise constant discretization of the magnetic fields is assumed. Then, in an arbitrary, but fixed spatial position $\mathbf{r}$, the discrete system matrix $\mathbf{A}_n$ is constant, and the solution of the Bloch-McConnell time step can be given by matrix exponential

$$\mathbf{M}(t) = \left(\mathbf{M}^{n-1} + \mathbf{A}_n^{-1}\mathbf{b}\right) \exp(\mathbf{A}_n t) - \mathbf{A}_n^{-1}\mathbf{b}. \tag{14}$$

However, the matrix exponential $\exp(\mathbf{A}_n t)$ can be calculated analytically only if $\mathbf{A}_n$ is diagonalizable, which is not the case here in general. Therefore, a general applicable numerical solution method based on operator splitting is introduced.

Here, the Bloch-McConnell matrices are splitted into a rotation and a relaxation/chemical exchange part. Rotation matrices are again used to solve the first part. For the second part, a similarity transformation is performed which results in a 3 by 3 block-diagonal matrix where the entries are time-independent. This part is then solved exact using eigenvalues and eigenvectors. Therefore, all subsystems are solved exactly again, hence the error is expected to be governed by the splitting error, which asymptotically decreases linearly when reducing the time step size for the asymmetric splitting and quadratically for the symmetric splitting.

2.3. Pulsed magnetization transfer

In addition to describing the MT effect within the general Bloch-McConnell equations, Section 2.2, we analyze pulsed magnetization transfer using the model of Graham and Henkelman [24]. For a water and a MT pool, the model equations are given as

$$\begin{cases} \begin{pmatrix} \frac{dM_{x,w}}{dt} \\ \frac{dM_{y,w}}{dt} \\ \frac{dM_{z,w}}{dt} \\ \frac{dM_{z,MT}}{dt} \end{pmatrix}(t, \mathbf{r}) = \begin{pmatrix} -\frac{1}{T_{2,w}} & -\Delta\omega_w(\mathbf{r}) & \gamma B_{1,y}(t) & 0 \\ \Delta\omega_w(\mathbf{r}) & -\frac{1}{T_{2,w}} & -\gamma B_{1,x}(t) & 0 \\ -\gamma B_{1,y}(t) & \gamma B_{1,x}(t) & -\frac{1}{T_{1,w}} - k_{w,MT} & k_{MT,w} \\ 0 & 0 & k_{w,MT} & -\frac{1}{T_{1,MT}} - R_{RFB} - k_{MT,w} \end{pmatrix} M(t, \mathbf{r}) + \begin{pmatrix} 0 \\ 0 \\ \frac{M_{0,w}}{T_{1,w}} \\ \frac{M_{0,MT}}{T_{1,MT}} \end{pmatrix}, \\ \mathbf{M}(0, \mathbf{r}) = \mathbf{M}^0(\mathbf{r}). \end{cases} \tag{15}$$

The term R_{RFB} covers saturation of the MT pool and is given as

$$R_{RFB} = B_1^2(t)\gamma^2\pi g_B(\Delta\omega). \tag{16}$$

Therein, g_B describes the absorption line shape of the MT pool, e.g. [52]. All other parameters are defined similarly to Section 2.2. Analogously, Eq. (15) can be extended to incorporate a water pool, a solute pool and the MT pool, whereby no direct exchange between solute and MT is assumed, [39]. The system is again solved either by exact solution based on matrix exponential, or by operator splitting. However, operator splitting now requires splitting into three parts: Rotation, relaxation/chemical exchange, and newly the saturation term R_{RFB} needs to be separated. The solution involves rotation matrices (resulting in an operator G_{rot}) for the rotation parts, eigenvalues and eigenvectors for the relaxation/chemical exchange part (resulting in an operator G_{rc}), and a scalar ODE solve for the saturation part, which can be integrated directly (resulting in an operator G_{sat}). The resulting asymmetric operator splitting scheme reads

$$\mathbf{M}^n = G_{\text{sat}}(\tau_n) \cdot G_{\text{rc}}(\tau_n) \cdot G_{\text{rot}}(\tau_n) \cdot \mathbf{M}^{n-1}. \tag{17}$$

One time step of the symmetric operator splitting is given as

$$\mathbf{M}^n = G_{\text{rot}}\left(\frac{\tau_n}{2}\right) \cdot G_{\text{sat}}\left(\frac{\tau_n}{2}\right) \cdot G_{\text{rc}}(\tau_n) \cdot G_{\text{sat}}\left(\frac{\tau_n}{2}\right) \cdot G_{\text{rot}}\left(\frac{\tau_n}{2}\right) \cdot \mathbf{M}^{n-1}, \tag{18}$$

which involves a half step rotation, half step saturation, then a full step relaxation and chemical exchange, followed by a half step saturation and a half step rotation. This is a generalization of symmetric splitting for more than two effects. The method still features a splitting error of second order, which can be shown by Taylor's expansion.

3. Methods

3.1. Solution methods for Bloch and Bloch-McConnell equations

Nine different solution methods were implemented in MATLAB (The MathWorks, Inc., Natick, USA, Release 2017b) to solve the Bloch and partly the Bloch-McConnell equations (see Sections 2.1, 2.2 and 2.3). The following methods were used in this paper: (i) EV: exact eigenvalue solver, (ii) ASY: asymmetric operator splitting, (iii) SY: symmetric operator splitting, (iv) SD: the well-known spin-domain solution method, (v) `expM`: MATLAB's matrix exponential, (vi) IE: implicit/backward Euler, (vii) CN: Crank-Nicolson's method, and (iix and ix) `ode45`: MATLAB's Runge-Kutta method in two different variants. Table 1 lists and summarizes all solution methods and indicates if they were applied to solve the Bloch and Bloch-McConnell equations and if vectorization was used.

If possible, the methods were implemented vectorized, i.e. the operations were applied simultaneously for all spatial positions using arrays instead of loops, which results in much faster MATLAB code. The numerical experiments were performed on a MacBook Pro (2 GHz Intel Core i5, 8 GB RAM).

3.2. Numerical evaluation of the Bloch equations

The numerical comparison for the Bloch equation solvers was performed for two distinct RF pulses and corresponding slice selective gradient shapes. First, an RF pulse based on a truncated sinc shape with only slight variations in time (Example 1) was used, see Fig. 1 and Table 2.

Second, a numerically optimized simultaneous multislice RF pulse and time varying slice-selective gradient (Example 2) were used, see Fig. 1 and Table 2, [53]. The RF pulse and the slice-selective gradient are varying heavily over time. Here, only the z-component of the magnetization vector was optimized as the transverse component was spoiled. For the sake of completeness, we perform the simulation study on all components of the magnetization.

For measuring numerical errors of the different solution methods, a relative L^2-norm using the whole magnetization vector at the terminal time T was used

$$\varepsilon_2 = \frac{\|\mathbf{M}_e(\mathbf{r},T) - \mathbf{M}_n(\mathbf{r},T)\|_{L^2(\Omega)}}{\|\mathbf{M}_e(\mathbf{r},T)\|_{L^2(\Omega)}}. \tag{19}$$

Here, $\mathbf{M}_n$ is the magnetization vector which was calculated with one of the numerical solution methods while $\mathbf{M}_e$ is the exact solution from EV. Furthermore, we explored the relative L^2-errors of the z-component of the magnetization vector and of the transverse component individually. For the transverse component $M_{xy} = M_x + i \cdot M_y$, we analyzed the error of $|M_{xy}|$ and of $\text{angle}(M_{xy})$ separately. This approach resulted in relative errors named $\varepsilon_z, \varepsilon_{|M_{xy}|}$ and $\varepsilon_{\text{angle}(M_{xy})}$.

To asses the runtime of different numerical solvers, both examples were simulated 100 times with each solution method and the mean run time was noted. In addition, for the Bloch equations without relaxation effects the conservation of magnetization is checked in Example 2, i.e. it is monitored whether the solvers fulfil

$$\max_{\mathbf{r}\in\Omega}\left|\frac{\max\limits_{t\in[0,T]}|\mathbf{M}(\mathbf{r},t)|}{\min\limits_{t\in[0,T]}|\mathbf{M}(\mathbf{r},t)|} - 1\right| \approx 0. \tag{20}$$

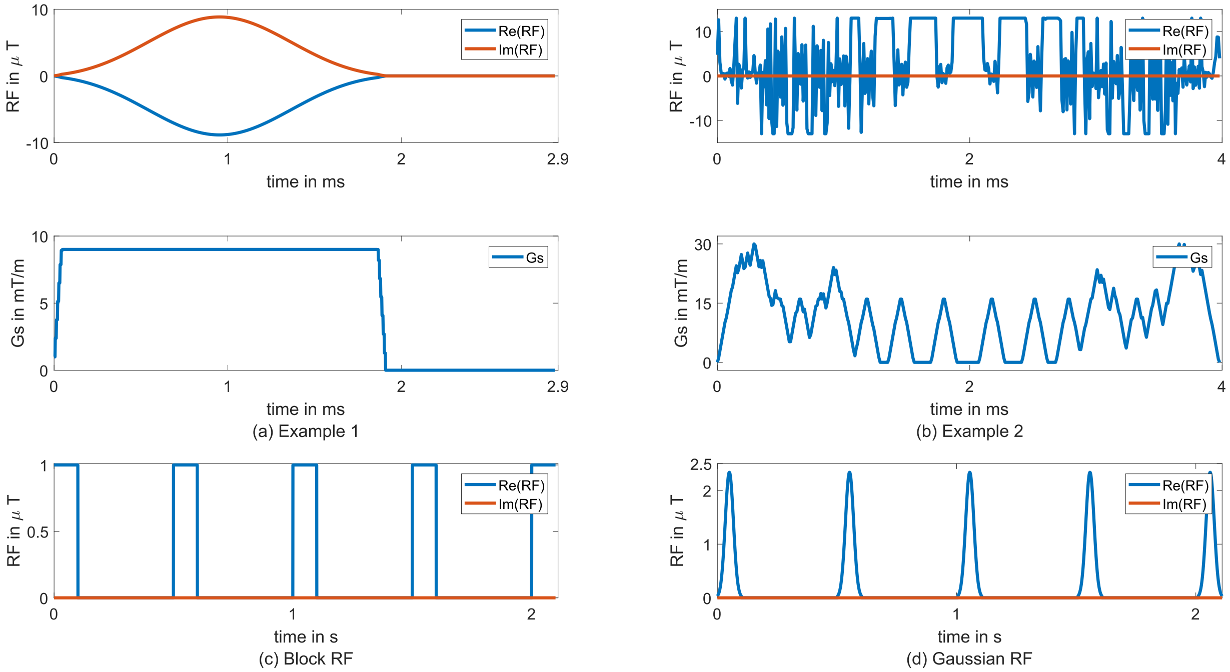
For the Bloch equations with relaxation effects, various sets of relaxation times at 3 T were considered, among them grey matter ($T_1 = 1331$ ms, $T_2 = 110$ ms), white matter ($T_1 = 832$ ms, $T_2 = 79.6$ ms) [54], muscle ($T_1 = 1420$ ms, $T_2 = 31.7$ ms) [55] and tendons ($T_1 = 400$ ms, $T_2 = 5$ ms) [56].

3.3. Numerical evaluation of the Bloch-McConnell equations

The proposed Bloch-McConnell simulation methods ASY, SY and EV were investigated with two distinct RF pulses (block RF pulse (BL) and Gaussian RF pulse (G)), see Fig. 1 and Table 2, for a two pool model with one water proton pool and one solute proton pool in the first place (Example 3), [57]. For the accuracy studies, specific relative L^2- and L^∞-errors were used which are defined analogously to Eq. (19).

Table 1
Abbreviations and description of the analyzed methods (first two columns). The last three columns indicate whether the methods were used in a vectorized manner, for Bloch examples (BE) or Bloch-McConnell examples (BME).

Abbr.	Characteristics	Vectorized	BE	BME
EV	Exact method with eigenvalues and eigenvectors	yes/no	yes	yes
ASY	Asymmetric operator splitting, [42] and (5)	yes	yes	yes
SY	Symmetric operator splitting, [41,42] and (7)	yes	yes	yes
SD	Spin-Domain solution for Bloch equations without relaxation, [26–28]	yes	yes	no
expM	Solution of a time step of Bloch's equation with MATLAB's matrix exponential function `expm` so that $\mathbf{M}^n = \left(\mathbf{M}^{n-1} + \mathbf{A}_n^{-1}\mathbf{b}\right)\texttt{expm}(\mathbf{A}_n\tau_n) - \mathbf{A}_n^{-1}\mathbf{b}$, [44,45]	no	yes	no
IE	Implicit Euler scheme, first-order accurate in time, [46,47]	no	yes	no
CN	Crank-Nicolson method, finite-difference method, second-order accurate in time, [48]	no	yes	no
ode45	MATLAB's `ode45`, differential equation solver based on an explicit Runge-Kutta (4,5) formula [49] with adaptive time grids, with explicit use of the time points from piecewise constant discretization of magnetic field. Tolerances: `abstol=1e−3` and `reltol=5e−2`	no	yes	no
ode45$_2$	MATLAB's `ode45` in standard call with tolerances `abstol = 1e-6`, `reltol = 1e-3`	no	yes	no



(a) Example 1

(b) Example 2

(c) Block RF

(d) Gaussian RF

Fig. 1. (a) Radio frequency pulse (top) and slice-selective gradient (bottom) of Example 1. (b) Radio frequency pulse (top) and slice-selective gradient (bottom) of Example 2. (c) Block RF pulses for Examples 3–7. (d) Gaussian RF pulses for Examples 3–7.

Table 2
Simulation parameters of all examples.

Ex.	RF	G_s	$T(ms)$	$\tau(\mu s)$	steps	$\Omega(mm)$	points
1	180° SINC	trapezoidal	1.9	1.7	1668	±5	100
2	180° optimized	time-varying	3.99	10	399	±60	1000

Examples for Bloch-McConnell Equations

Ex.	RF	$T(ms)$	$\tau(\mu s)$	steps	$\Omega(ppm)$	points	water pool (ms)	solute pool (ms)	MT pool (ms)
3	G & BL	2100	500	4221	±5	100	$T_1 = 1048$ $T_2 = 69$	$T_1 = 1048$ $T_2 = 15$	
4	BL	2100	(4a) 7.8 (4b) 0.16	270.000 1.312.500	±5	100	$T_1 = 1048$ $T_2 = 69$		$T_1 = 1000$ $T_2 = 0.0087$
5	BL	2100	7.8	270.000	±5	100	$T_1 = 1048$ $T_2 = 69$	$T_1 = 1048$ $T_2 = 15$	$T_1 = 1000$ $T_2 = 0.0087$

Examples for Pulsed Magnetization Transfer

Ex.	RF	$T(ms)$	$\tau(\mu s)$	steps	$\Omega(ppm)$	points	water pool (ms)	solute pool (ms)	MT pool (ms)
6	BL	2100	(6a) 500 (6b) 7.8	4221 270.000	±5	100	$T_1 = 1048$ $T_2 = 69$		$T_1 = 1000$ $T_2 = 0.0087$
7	BL	2100	7.8	270.000	±5	100	$T_1 = 1048$ $T_2 = 69$	$T_1 = 1048$ $T_2 = 15$	$T_1 = 1000$ $T_2 = 0.0087$

For a further investigation, time discretizations with different resolutions were analyzed using the experimental order of convergence [58] *EOC*

$$EOC(\tau, \tau') = \frac{\log \frac{\varepsilon_2(\tau)}{\varepsilon_2(\tau')}}{\log \frac{\tau}{\tau'}}. \tag{21}$$

This allows for a statement of the splitting schemes regarding convergence with respect to the time step.

Next, the MT effect was analyzed using a two pool model, one water proton pool and one pool that models the MT (Example 4), Table 2. The specific norm we used was the relative L^2-norm as described in (19). Compared to the former two pool model, the T_2 relaxation time of the MT pool was substantially smaller. Therefore, a smaller time step length of $\tau = 7.8\ \mu s$ was chosen to unravel the relaxation effects accordingly (Example 4a). In Example 4b, the time step size was even more reduced to $\tau = 0.16\ \mu s$.

Last, a three pool model was considered containing a water and a solute proton pool as well as one pool modeling the MT effect [39] (Example 5), Table 2.

3.4. Numerical evaluation of the pulsed magnetization transfer

Similar to Section 3.3 we perform two and three pool experiments with the model described in Section 2.3. Here, we use the input parameters as described in Table 2, Example 6. Additionally, we perform an example using 3 pools, Table 2, Example 7. The line shape is assumed to be Lorentzian, i.e.

$$g_B(\Delta\omega) = \frac{T_{2,MT}}{\pi\left(1 + \left(\Delta\omega \cdot T_{2,MT}\right)^2\right)}. \tag{22}$$

Furthermore, we compare the two different models to include the MT with two pools (using Example 4 and 6) with each other.

4. Results

4.1. Bloch equations without relaxation

Table 3 depicts the simulation results for different Bloch forward solvers in absence of relaxation effects.

The solvers expM, SD, ASY and SY exhibit accuracy in the order of machine precision for both examples. ode45 performs similarly in the first example (slightly varying RF and Gs), but with increased error in the second example (heavily varying RF and Gs). Large errors above $1e-3$ are observed for IE, ode45$_2$, and for CN in the second example.

The table also shows the average run time in seconds for an implementation in MATLAB. We first compare the four vectorized solvers, including EV which was vectorized only for the case without relaxation effects. The smallest run time is observed for SD, followed by EV, ASY and SY, which all need a factor of two to three more time in both examples. We note that non-vectorized implementations expM, IE, CN, ode45 and ode45$_2$ exhibit a factor of 100 to 1000 larger run times. They could be vectorized too, with expectedly comparable run times. However, due to their limited performance in the studies below, we did not invest time in an advanced implementation here.

Furthermore, we analyzed the L^2-norm of the magnetization vectors over time for different points in space. All methods except IE maintain the total magnetization over time. In contrast, IE exhibits a degradation over time depending on the spatial position resulting in a loss of up to 100% of the total magnetization, see Supporting Information Fig. S1.

4.2. Bloch equations with relaxation

Tables 4 and 5 depict the simulation results with relaxation effects including L^2-errors and average run time.

We observe larger errors for expM, IE and SD for both examples and for all sets of relaxation times. CN shows a good performance with errors in the range of $1e-5$ for Example 1, but performs with larger errors for Example 2. ASY and SY outperform the other methods, especially in Example 2. In general, SY exhibits smaller numerical errors than ASY.

While all methods use the same fixed time grid, ode45 and ode45$_2$ are adaptive time-stepping schemes, i.e. they choose its time grid adaptively based on the given tolerances. With the fine tolerances mentioned above ode45 uses in total 18349 respectively 4463 time points in the two examples, and ode45$_2$ uses 1667 respectively 849 time points, each with relaxation times typical for white matter. In general, the accuracy of all methods decreases with a decreasing T_2 relaxation time. Supporting Information Fig. S8 displays the logarithmic error of Example 1 calculated with SY versus EV for various relaxation times over the whole spatial domain. It can be observed that for almost all points in space the performance increases with an increasing T_2 time (tendons, then muscle, white matter, grey matter and last without relaxation). Furthermore, in Supporting Information Fig. S9 we depict the logarithmic errors of the solver SY calculated with five different time step lengths where we observe a uniform convergence through the whole spatial domain.

Figs. 2 and 3 show the magnetization profiles and the numerical accuracy for some solution methods for each magnetization component over the spatial domain. More detailed figures can be found in Supporting Information Figs. S2–S7.

The resulting errors of the numerical solution methods for Example 1 are quite the same across all spatial positions. In contrast, for Example 2, both CN and ode45 show a better performance around the isocenter at 0 mm, see especially Supporting Information Figs. S5–S7. The errors of Example 1 are smoother with respect to the spatial domain while they are oscillating in the case of Example 2. The run times of the simulations with relaxation from white matter are depicted in Tables 4 and 5. All numerical solution

Table 3
Numerical accuracy in absence of relaxation effects with L^2-errors as defined in (19) for all solution methods (line 1 to 3 and 5 to 7) and average run time in seconds (line 4 and 8) out of 100 runs. Solution methods indicated by an asterisk (EV, SD, ASY and SY) are implemented vectorized with respect to the time step. The best values per line are marked green.

		EV*	expM	IE	CN	SD*	ode45	ode45$_2$	ASY*	SY*
Ex. 1	ε_z		$9.30e-16$	$3.39e-03$	$3.26e-06$	$2.68e-14$	$8.38e-14$	$5.54e-04$	$1.36e-12$	$4.09e-15$
	$\varepsilon_{\lvert M_{xy}\rvert}$		$2.23e-15$	$1.70e-02$	$1.41e-05$	$3.91e-14$	$1.87e-13$	$1.33e-02$	$2.62e-12$	$4.08e-14$
	$\varepsilon_{angle(M_{xy})}$		$3.80e-14$	$4.66e-01$	$2.49e-04$	$4.03e-14$	$1.14e-12$	$3.56e-01$	$3.36e-11$	$9.62e-14$
	∅ run time	$4.10e-02$	$1.39e+01$	$2.41e+00$	$1.92e+00$	$1.34e-02$	$1.19e+02$	$6.09e+01$	$3.41e-02$	$4.92e-02$
Ex. 2	ε_z		$3.28e-15$	$7.34e-01$	$7.91e-01$	$3.22e-15$	$7.16e-06$	$2.62e-02$	$2.78e-14$	$2.61e-15$
	$\varepsilon_{\lvert M_{xy}\rvert}$		$8.12e-15$	$9.08e-01$	$1.18e+00$	$4.11e-15$	$1.02e-04$	$1.81e-01$	$7.17e-13$	$3.84e-15$
	$\varepsilon_{angle(M_{xy})}$		$1.58e-14$	$1.11e+00$	$1.25e+00$	$6.52e-15$	$8.08e-05$	$8.28e-01$	$3.45e-12$	$7.97e-15$
	∅ run time	$6.58e-02$	$3.77e+01$	$4.11e+00$	$3.51e+00$	$2.83e-02$	$2.87e+02$	$4.66e+02$	$6.10e-02$	$6.45e-02$

Table 4
L^2-errors as defined in (19) of different solution methods for Example 1. We analyzed different tissue types which all were represented by a typical set of relaxation times, among them grey matter GM ($T_1 = 1331$ ms, $T_2 = 110$ ms), white matter WM ($T_1 = 832$ ms, $T_2 = 79.6$ ms), muscle M ($T_1 = 1420$ ms, $T_2 = 31.7$ ms) and tendons T ($T_1 = 400$ ms, $T_2 = 5$ ms). For each solver we computed the mean of run time in seconds, where 100 simulations were done for the white matter example. Solution methods indicated by an asterisk (SD, ASY and SY) are implemented vectorized with respect to the time step. The best values per line are marked green.

Example 1		EV	expM	IE	CN	SD*	ode45	ode45$_2$	ASY*	SY*
GM	ε_z		$3.13e-04$	$3.38e-03$	$3.24e-06$	$3.22e-03$	$8.24e-14$	$4.85e-04$	$5.33e-07$	$2.44e-09$
	$\varepsilon_{\lvert M_{xy}\rvert}$		$1.07e-03$	$1.69e-02$	$1.42e-05$	$1.97e-02$	$1.86e-13$	$9.02e-03$	$7.85e-06$	$1.68e-09$
	$\varepsilon_{angle(M_{xy})}$		$1.07e-02$	$4.71e-01$	$1.42e-04$	$2.05e-02$	$1.15e-12$	$3.48e-01$	$1.31e-08$	$6.51e-09$
T	ε_z		$1.32e-03$	$3.04e-03$	$3.92e-04$	$6.20e-02$	$3.92e-04$	$7.07e-04$	$3.92e-04$	$3.92e-04$
	$\varepsilon_{\lvert M_{xy}\rvert}$		$7.88e-03$	$1.62e-02$	$3.33e-04$	$5.33e-01$	$3.33e-04$	$9.61e-03$	$3.76e-04$	$3.33e-04$
	$\varepsilon_{angle(M_{xy})}$		$8.47e-02$	$3.71e-01$	$1.52e-02$	$5.78e-01$	$1.52e-02$	$5.12e-01$	$1.52e-02$	$1.52e-02$
WM	ε_z		$4.81e-04$	$3.37e-03$	$3.23e-06$	$4.57e-03$	$8.27e-14$	$4.86e-04$	$8.53e-07$	$3.34e-09$
	$\varepsilon_{\lvert M_{xy}\rvert}$		$1.64e-03$	$1.69e-02$	$1.42e-05$	$2.73e-02$	$1.87e-13$	$9.04e-03$	$1.09e-05$	$2.33e-09$
	$\varepsilon_{angle(M_{xy})}$		$1.51e-02$	$4.73e-01$	$9.46e-05$	$2.87e-02$	$1.21e-12$	$3.45e-01$	$1.84e-08$	$9.11e-09$
M	ε_z		$3.91e-04$	$3.34e-03$	$3.21e-06$	$1.00e-02$	$8.12e-14$	$4.85e-04$	$4.99e-07$	$8.63e-09$
	$\varepsilon_{\lvert M_{xy}\rvert}$		$1.47e-03$	$1.68e-02$	$1.44e-05$	$7.00e-02$	$1.88e-13$	$9.18e-03$	$2.72e-05$	$5.67e-09$
	$\varepsilon_{angle(M_{xy})}$		$3.70e-01$	$4.85e-01$	$1.89e-04$	$7.16e-02$	$1.49e-12$	$3.38e-01$	$4.30e-08$	$2.13e-08$
	⌀ run time	$9.54e+00$	$1.51e+01$	$2.21e+00$	$1.92e+00$	$1.23e-02$	$1.14e+02$	$6.94e+01$	$4.23e-02$	$4.61e-02$

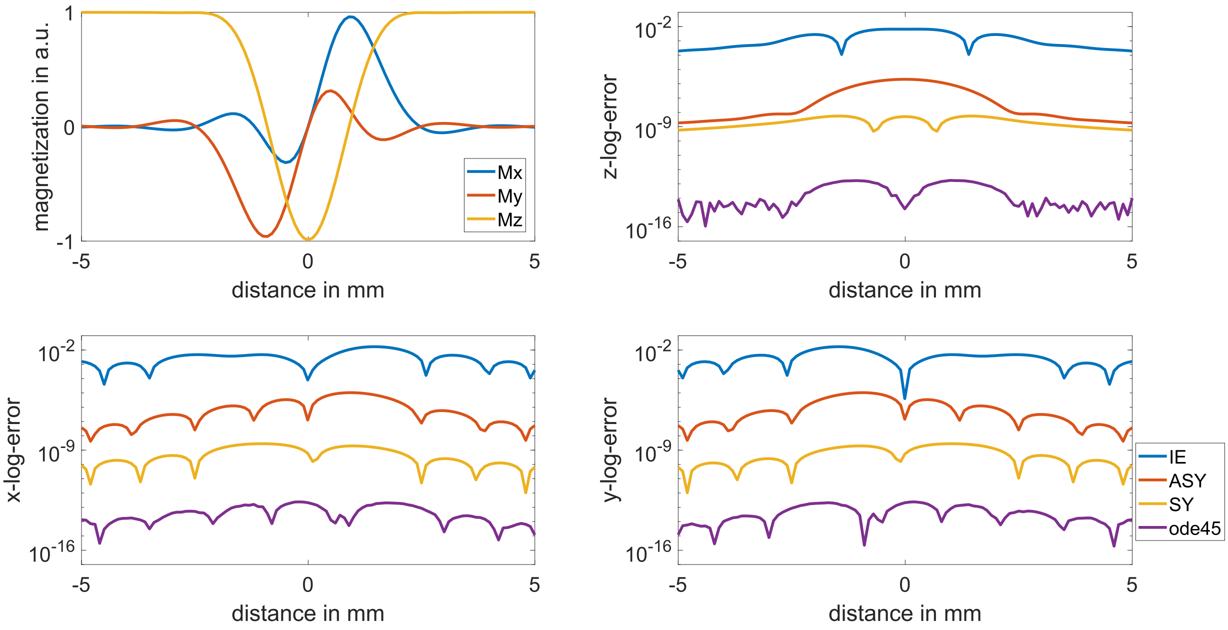


Fig. 2. Magnetization profile in presence of typical relaxation times for white matter for Example 1 at the terminal time T calculated with EV. Depicted are the x- (blue), y- (red) and z-component (yellow) of the magnetization vector (top left). Logarithmic L^2-errors of the individual coordinates for the solvers IE, ASY, SY and ode45. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

methods show the same performance as in the case without relaxation effects. The difference occurs in the exact solution method EV which is now two to three orders of magnitude slower than the splitting schemes and SD.

In Supporting Information Fig. S1 we illustrate the resulting slice profile using EV and the magnetization through time for 3 different points in space. We observe that the loss of magnetization over time is for all solution methods but IE similar to EV. For IE, the total magnetization severely decreases for different points in space.

4.3. Bloch-McConnell equations

First, we analyze the results of Example 3 here. Table 6 shows the accuracy of SY and ASY for Gaussian (G) and block (BL) pulse. For all examples, smaller errors are observed for SY than for ASY.

Table 7 shows the numerical accuracy of ASY and SY for different time discretizations. Both methods show convergence of the error with respect to the time step size, however, with different convergence orders that are approximately depicted using the experimental order of convergence (EOC). In particular, linear convergence is observed for the total error of ASY, and quadratic convergence for SY.

Table 8 shows the numerical accuracy of Example 4 with block-pulse simulation for two different time step lengths.

We observe a better accuracy for both solution methods in the case of the smaller time step size (Example 4b) compared to the greater one (Example 4a). Moreover, performance for the water proton pool is better than for the MT pool which yields numerical errors in the range of $1e-4$ compared to $1e-1$ to $1e-3$ for the MT for Example 4a. In the case of Example 4b, the errors are reduced by a factor of 1 to 4. More precisely, as before, a linear con-

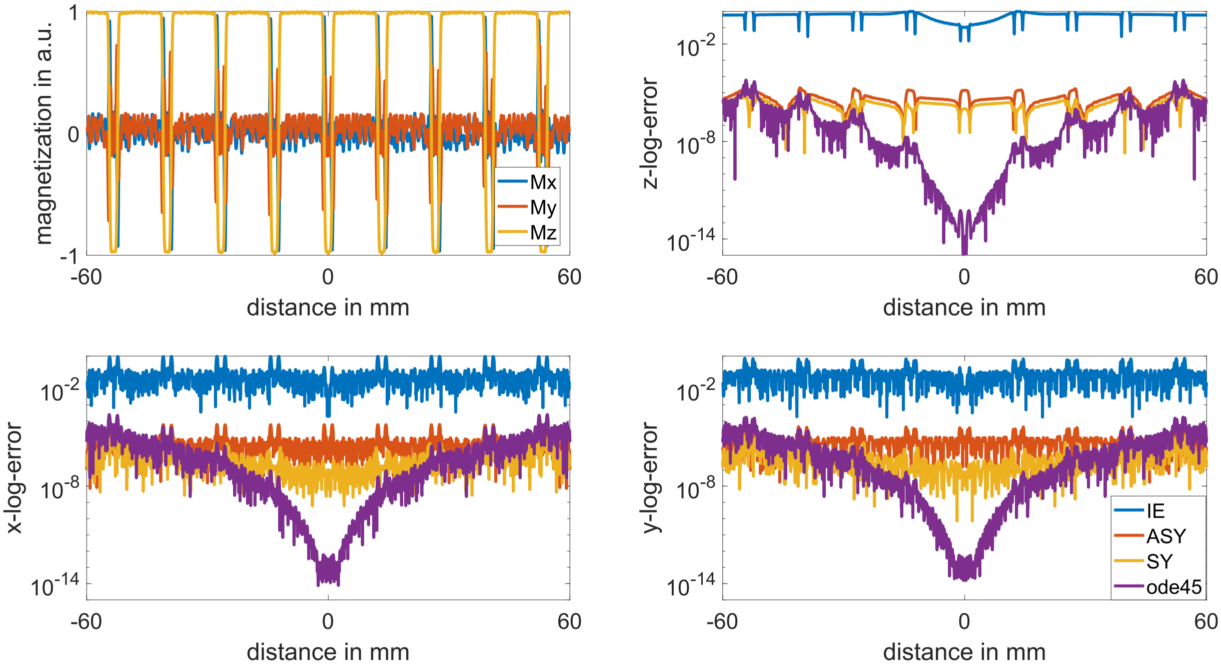


Fig. 3. Magnetization profile in presence of typical relaxation times for white matter at 3T for Example 2 at the terminal time T calculated with EV. Depicted are the x- (blue), y- (red) and z-component (yellow) of the magnetization vector (top left). Logarithmic L^2-errors of the individual coordinates for the solvers IE, ASY, SY and ode45. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
L^2-errors as defined in (19) of different solution methods for Example 2. We analyzed different tissue types which all were represented by a typical set of relaxation times, among them grey matter GM ($T_1 = 1331$ ms, $T_2 = 110$ ms), white matter WM ($T_1 = 832$ ms, $T_2 = 79.6$ ms), muscle M ($T_1 = 1420$ ms, $T_2 = 31.7$ ms) and tendons T ($T_1 = 400$ ms, $T_2 = 5$ ms). For each solver we computed the mean of run time in seconds, where 100 simulations were done for the white matter example. Solution methods indicated by an asterisk (SD, ASY and SY) are implemented vectorized with respect to the time step. The best values per line are marked green.

Example 2		EV	expM	IE	CN	SD*	ode45	ode45$_2$	ASY*	SY*
GM	ε_z		$1.12e-03$	$7.32e-01$	$7.90e-01$	$6.50e-03$	$7.07e-06$	$2.59e-02$	$5.16e-06$	$2.10e-06$
	$\varepsilon_{\lvert M_{xy}\rvert}$		$1.05e-03$	$9.07e-01$	$1.18e+00$	$2.22e-02$	$1.02e-04$	$1.73e-01$	$4.57e-05$	$3.43e-06$
	$\varepsilon_{angle(M_{xy})}$		$7.16e-01$	$1.11e+00$	$1.25e+00$	$7.25e-02$	$7.96e-05$	$7.96e-01$	$3.26e-05$	$1.72e-05$
T	ε_z		$5.28e-03$	$7.06e-01$	$7.82e-01$	$1.32e-01$	$5.36e-06$	$2.31e-02$	$8.91e-05$	$4.53e-05$
	$\varepsilon_{\lvert M_{xy}\rvert}$		$9.42e-03$	$8.78e-01$	$1.25e+00$	$5.88e-01$	$9.97e-05$	$1.84e-01$	$1.02e-03$	$9.01e-05$
	$\varepsilon_{angle(M_{xy})}$		$2.38e-01$	$1.07e+00$	$1.24e+00$	$5.97e-01$	$6.41e-05$	$7.26e-01$	$3.91e-04$	$2.06e-04$
WM	ε_z		$1.76e-03$	$7.31e-01$	$7.90e-01$	$9.06e-03$	$7.03e-06$	$2.69e-02$	$7.50e-06$	$2.87e-06$
	$\varepsilon_{\lvert M_{xy}\rvert}$		$1.61e-03$	$9.06e-01$	$1.18e+00$	$3.09e-02$	$1.02e-04$	$1.73e-01$	$6.32e-05$	$4.73e-06$
	$\varepsilon_{angle(M_{xy})}$		$7.16e-02$	$1.11e+00$	$1.24e+00$	$1.24e-01$	$7.93e-05$	$7.80e-01$	$4.54e-05$	$2.39e-05$
M	ε_z		$1.29e-03$	$7.29e-01$	$7.89e-01$	$2.16e-02$	$6.84e-06$	$2.74e-02$	$1.49e-05$	$7.45e-06$
	$\varepsilon_{\lvert M_{xy}\rvert}$		$1.71e-03$	$9.04e-01$	$1.19e+00$	$7.78e-02$	$1.02e-04$	$1.86e-01$	$1.59e-04$	$1.21e-05$
	$\varepsilon_{angle(M_{xy})}$		$7.17e-02$	$1.10e+00$	$1.24e+00$	$2.08e-01$	$7.79e-05$	$8.33e-01$	$1.08e-04$	$5.71e-05$
	∅ run time	$2.58e+01$	$3.75e+01$	$4.21e+00$	$3.81e+00$	$2.87e-02$	$2.88e+02$	$5.62e+02$	$6.11e-02$	$6.38e-02$

vergence of the total error is observed for ASY, and a quadratic convergence for SY.

Furthermore, Table 8 shows the numerical results of Example 5. We observe significantly larger errors for ASY compared to SY which performs up to 3 orders of magnitude better. Moreover, the simulation of the MT pool shows a much smaller numerical accuracy compared to the water and solute proton pools, in particular for a time step length similar to the T_2 of the bound pool.

4.4. Pulsed magnetization transfer

Table 9 shows the simulation results of Examples 6 and 7. We observe an increased numerical accuracy for SY compared to ASY for throughout all errors and examples, i.e. using SY the errors are reduced by 1 to 3 orders of magnitude compared to ASY. Furthermore, reduction of time step length (Example 6a vs. 6b) reduces the individual errors 1 to 4 orders of magnitude.

When analyzing Tables 8 and 9 we observe that the L^2- and L^∞-errors of the water and the solute pool are comparable. The difference occurs in the MT pool, where Table 9 shows an increased numerical accuracy for the z-coordinate of the MT pool (no transverse component in this model).

In Table 10 we observe the difference between the two models describing the MT. With EV (both simulations exact) we result in a deviation of $4.6e-2$. The same deviation is attained with SY for small time step size.

Table 6
Bloch-McConnell example using a water proton pool and a solute proton pool. L^2- and L^∞-errors as defined by (19) of the numerical solution methods for Gaussian (G) and block (BL) pulse and for both splitting schemes.

	Solver	Error	$\varepsilon_{x,w}$	$\varepsilon_{y,w}$	$\varepsilon_{z,w}$	$\varepsilon_{x,s}$	$\varepsilon_{y,s}$	$\varepsilon_{z,s}$
BL	ASY	L^2	$3.7e-3$	$4.2e-3$	$1.1e-4$	$2.4e-2$	$3.1e-2$	$2.5e-3$
		L^∞	$3.7e-3$	$3.8e-3$	$2.6e-4$	$2.3e-2$	$2.6e-2$	$5.7e-3$
	SY	L^2	$4.4e-5$	$1.1e-3$	$5.1e-6$	$2.2e-4$	$6.0e-3$	$1.9e-5$
		L^∞	$4.5e-5$	$3.7e-4$	$1.0e-5$	$4.4e-4$	$3.4e-3$	$3.2e-5$
G	ASY	L^2	$3.7e-3$	$3.7e-3$	$9.4e-5$	$2.6e-2$	$2.6e-2$	$1.6e-3$
		L^∞	$3.7e-3$	$3.7e-3$	$2.3e-4$	$2.2e-2$	$2.3e-2$	$3.7e-3$
	SY	L^2	$2.3e-5$	$7.1e-5$	$7.5e-6$	$1.3e-3$	$1.9e-3$	$2.1e-5$
		L^∞	$2.7e-5$	$2.2e-5$	$1.2e-5$	$2.1e-3$	$1.4e-3$	$3.3e-5$

Table 7
Bloch-McConnell example using a water proton pool and a solute proton pool. L^2-errors for different time discretizations and corresponding experimental orders of convergence as defined by (19) and (21), where the study was done with BL.

τ	ASY		SY	
in s	L^2	EOC	L^2	EOC
$1e-3$	$1.2e-3$		$2.2e-4$	
$5e-4$	$5.9e-4$	1.02	$4.4e-5$	2.18
$1e-4$	$1.2e-4$	0.99	$1.7e-6$	2.02
$5e-5$	$5.9e-5$	1.02	$4.2e-7$	2.02
$1e-5$	$1.2e-5$	0.99	$1.7e-8$	1.99
$5e-6$	$5.9e-6$	1.02	$4.2e-9$	2.02

5. Discussion

The performance of different Bloch solvers was analyzed under the widespread assumption of a piecewise constant magnetic field using typical examples of RF and slice selective gradient shapes. The numerical accuracy of the individual solvers was assessed using mean error (L^2-norm) and maximum or worst-case error (L^∞-norm) over the whole spatial domain. In absence of relaxation effects, many solvers give solutions in machine precision, including methods based on rotation matrices (ASY, SY) or Cayley-Klein parameters (SD), numerical solution with ode45, or MATLAB's matrix exponential method expM. The latter is the case, since the system matrix is here always diagonalizable. However, other common ODE solvers turn out to be inappropriate for Bloch models already in absence of relaxation effects, including ode45$_2$.

The choice of an adequate Bloch solver is more involved for the Bloch equation with relaxation effects. Neglecting relaxation effects (SD) leads to considerable errors, Tables 4 and 5. Likewise, the matrix exponential implementation expM shows significant

Table 8
L^2- and L^∞-errors of ASY and SY for a two pool example performed with BL, where one pool models the water protons (first three columns) and the other the MT effect (last three columns), whereby Example 4a was done with a time step length of $\tau = 7.8\mu s$ (Lines 1–4) and Example 4b was done with a time step length of $\tau = 0.16\mu s$ (Lines 5–8). Lines 9–12 show the corresponding three pool example, where one pool models the water protons (first three columns), one the solute protons (second three columns) and one the MT effect (last three columns).

	Solver	Error	$\varepsilon_{x,w}$	$\varepsilon_{y,w}$	$\varepsilon_{z,w}$	$\varepsilon_{x,s}$	$\varepsilon_{y,s}$	$\varepsilon_{z,s}$	$\varepsilon_{x,MT}$	$\varepsilon_{y,MT}$	$\varepsilon_{z,MT}$
Ex. 4a	ASY	L^2	$6.6e-4$	$3.3e-4$	$6.3e-4$				$1.5e-1$	$3.8e-1$	$1.2e-3$
		L^∞	$6.3e-4$	$1.7e-4$	$6.6e-4$				$2.3e-1$	$3.8e-1$	$1.2e-3$
	SY	L^2	$5.7e-4$	$2.6e-4$	$6.3e-4$				$1.5e-2$	$6.5e-2$	$1.2e-3$
		L^∞	$5.4e-4$	$1.4e-4$	$6.7e-4$				$2.3e-2$	$6.5e-2$	$1.2e-3$
Ex. 4b	ASY	L^2	$2.3e-6$	$2.1e-6$	$3.1e-7$				$4.1e-3$	$9.3e-3$	$5.5e-7$
		L^∞	$2.3e-6$	$2.0e-6$	$3.2e-7$				$6.3e-3$	$9.3e-3$	$5.3e-7$
	SY	L^2	$2.4e-7$	$9.5e-8$	$2.7e-7$				$6.1e-6$	$2.9e-5$	$5.2e-7$
		L^∞	$2.2e-7$	$4.9e-8$	$2.9e-7$				$9.4e-6$	$2.9e-5$	$5.4e-7$
Ex. 5	ASY	L^2	$6.3e-3$	$6.6e-3$	$9.2e-5$	$6.5e-4$	$6.1e-4$	$1.0e-4$	$3.4e-1$	$3.9e-1$	$9.0e-3$
		L^∞	$6.2e-3$	$6.7e-3$	$1.6e-4$	$7.1e-4$	$5.9e-4$	$3.4e-4$	$3.5e-1$	$3.9e-1$	$9.0e-3$
	SY	L^2	$7.4e-6$	$2.0e-5$	$1.1e-7$	$9.5e-7$	$8.4e-7$	$2.5e-7$	$2.7e-3$	$5.7e-2$	$9.0e-3$
		L^∞	$8.4e-6$	$1.5e-5$	$2.0e-7$	$1.8e-6$	$5.5e-7$	$4.6e-7$	$2.7e-2$	$5.7e-2$	$9.0e-3$

Table 9
L^2- and L^∞-errors of ASY and SY for a two pool pulsed MT example performed with BL, where one pool models the water protons (first three columns) and the other the MT effect (last column), whereby Example 6a was done with a time step length of $\tau = 500\mu s$ (Lines 1–4) and Example 6b was done with a time step length of $\tau = 7.8\mu s$ (Lines 5–8). Lines 9–12 show the corresponding three pool example, where one pool models the water protons (first three columns), one the solute protons (second three columns) and one the MT effect (last column) with a time step length of $\tau = 7.8\mu s$.

	Solver	Error	$\varepsilon_{x,w}$	$\varepsilon_{y,w}$	$\varepsilon_{z,w}$	$\varepsilon_{x,s}$	$\varepsilon_{y,s}$	$\varepsilon_{z,s}$	$\varepsilon_{z,MT}$
Ex. 6a	ASY	L^2	$5.0e-3$	$7.3e-3$	$1.2e-4$				$3.2e-4$
		L^∞	$5.0e-3$	$5.1e-3$	$2.7e-4$				$9.2e-4$
	SY	L^2	$4.2e-5$	$2.7e-3$	$1.3e-5$				$1.3e-5$
		L^∞	$4.8e-5$	$8.8e-4$	$1.5e-5$				$1.7e-5$
Ex. 6b	ASY	L^2	$7.8e-5$	$7.8e-5$	$1.8e-6$				$5.0e-6$
		L^∞	$7.8e-5$	$7.8e-5$	$4.4e-6$				$1.5e-5$
	SY	L^2	$2.2e-7$	$6.4e-7$	$2.1e-7$				$2.1e-7$
		L^∞	$2.1e-7$	$1.9e-7$	$2.0e-7$				$2.0e-7$
Ex. 7	ASY	L^2	$5.8e-3$	$6.5e-3$	$9.2e-5$	$4.1e-4$	$4.0e-4$	$5.3e-5$	$8.0e-5$
		L^∞	$5.9e-3$	$6.7e-3$	$1.6e-4$	$4.4e-4$	$3.5e-4$	$1.6e-4$	$1.5e-4$
	SY	L^2	$6.9e-6$	$1.9e-5$	$1.2e-7$	$7.6e-7$	$9.7e-7$	$2.4e-7$	$7.7e-5$
		L^∞	$8.1e-6$	$1.5e-5$	$2.2e-7$	$1.4e-6$	$5.9e-7$	$4.8e-7$	$1.4e-4$

Table 10
L^2- errors Eq. (19) between the two magnetization vectors calculated with the Bloch-McConnell equations and the pulsed MT model. The top rows show the errors for different time discretizations using SY, the bottom row shows the error between the two models using the method EV. Additionally, the second column shows the time discretization step normalized to the transverse relaxation time of the MT. The RF pulse used was BL and two pools were considered (water and MT).

Solver	τ in μs	$\frac{\tau}{T_{2,MT}}$	L^2–Error
SY	500	57.47	$3.0e-1$
	125	14.37	$1.2e-1$
	31.3	3.60	$5.7e-2$
	7.8	0.90	$4.7e-2$
	2.0	0.23	$4.6e-2$
	1.0	0.11	$4.6e-2$
EV	500	57.47	$4.6e-2$

errors even for the simple Example 1. Its results give just one more correct digit than SD. However, there are many other implementations of matrix exponentials that might yield different results, confer [33]. In contrast, operator splitting methods can be easily implemented and give a substantially higher precision, see Tables 4 and 5, and Figs. 2 and 3. The error is evenly distributed over space and components. Compared to expM another two to six correct digits are obtained from SY. In general, SY outperforms ASY in accuracy. The inclusion of relaxation effects via ASY and SY is possible with a factor of two to three increased run time compared to the fast spin-domain solution SD in absence of relaxation effects. The performance of CN heavily depends on the problem, with a good accuracy for the smooth RF pulse in Example 1, and bad accuracy for the heavily varying RF pulse of Example 2. Besides, the error of CN shows an undesired uneven distribution over space with huge errors in isolated points. This behaviour together with the reduced robustness and increased run time leaves CN not competitive to e.g. SY. The results of ode45 show that adaptive numerical ODE solvers can reach high accuracy in the case of piecewise constant magnetic fields, however, at cost of run time, if the tolerances are chosen very fine and if the events (jumping points in piecewise constant RF and Gs) are included in the adaptive time grid. In contrast, the second version of `ode45`, ode45$_2$, where the time grid of the magnetic fields was not explicitly enforced, suffers from an increased numerical error comparable to IE and CN. Concluding, SY clearly outperforms the other Bloch solvers in the test field in presence of relaxation effects. The solution methods can be applied to longer pulse sequences such as those used in Magnetic Resonance Fingerprinting. In experiments the error behaviour was found to be the same (not shown here). For speed-up one could simulate RF and Gs periods with one of the simulation methods, and model the pure relaxation periods (i.e. where no RF or Gs are present) by a single relaxation term, see [59].

Asymmetric splitting methods [34–36] have been applied to Bloch simulations for a long time. Recently, the symmetric splitting method [41,42] has shown its advantage in the sense of accuracy for Bloch simulations with relaxation [43]. With the above introduced exact solution it was possible to analyze the errors of these methods and other numerical approximations for piecewise constant RF and gradient pulses as provided by typical modern scanners. In the presented work, the splitting approaches were extended to the Bloch-McConnell equations and the model of Graham and Henkelman [24]. Both, the splitting of the system matrix and the development of adequate solvers for the resulting subsystems, was explained in detail. For the Bloch-McConnell equations, splitting into rotation and relaxation/chemical exchange allowed us to implement exact solvers for both subsystems. In case of the pulsed MT model, a consistent extension of symmetric splitting for more than two effects was shown, which retains the favorable fast second-order convergence of the numerical error, which allows for substantially increased numerical accuracy of the simulations compared to first-order methods.

A comprehensive investigation of numerical accuracy with respect to time discretizations using different resolutions was done for ASY and SY in Table 7. Although the study was done for the

Bloch-McConnell equations, the results can be directly transfered to the Bloch equations, which are included as special case. The results show in general an increased numerical accuracy for SY compared to ASY, together with an increased order of convergence: SY exhibits a quadratic order of convergence and ASY a linear order, which is in accordance with general theory of operator splitting methods [50]. Necessary for reaching this maximal order was to solve the subsystems with adequate precision. In our case all subsystems were solved exactly. Especially for involved examples with small relaxation times as for example in Table 8, a quadratic convergent time stepping method is vital for fast and accurate solution. Furthermore, using SY less time steps are necessary for a certain accuracy compared to ASY which results in a reduced run time.

The detailed comparison of accuracy of Bloch solvers was made possible by application of an exact solution method EV, which allows to reliably compute the exact solution in machine precision. EV for Bloch with relaxation effects was based on a comprehensive investigation of the eigenvalues and eigenvectors of the Bloch system matrix. Therefore, the Bloch equations per time step are solved exactly, however, at the cost of a significantly increased run time. This prohibits the use of EV itself in time-demanding applications. In both Bloch-McConnell cases, an exact matrix exponential based on eigenvalues and eigenvectors was only derived for diagonalizable system matrices, which is not always the case. However, we note that for the Bloch-McConnell examples considered in this paper it was ensured that the system matrix is diagonalizable to provide a fair comparison between exact and numerical solution methods. Moreover, the operator splitting solvers work well also for non-diagonalizable system matrices (not shown here).

Besides numerical accuracy, run time of the simulations is a decision criterion for the quality of the method. Since numerical accuracy of stable ODE solvers increases with refining the time grid at the cost of run time, the time grid was fixed throughout all solvers excluding ode45 and ode45$_2$, which are adaptive time stepping schemes. Then a comparison regarding run time is adequate, however, only within vectorized or not-vectorized implementations. Within the vectorized solvers (SD, SY, ASY) SD is the fastest solution method, with a factor of two to four increased run time for SY and ASY, see Tables 4 and 5. This is mainly due to the fact that SD is profiting from complex arithmetic in the spin domain, while the other two need to work in the magnetization domain. An efficient implementation of ASY and SY in the spin domain was prohibited since a way of computing the relaxation effects in the spin domain could not be found. Therefore, results need to be transferred between magnetization and spin domain in each time step which results in a substantially higher computational effort. ASY is slightly faster than SY due to the additional step involved in the symmetric scheme. All non-vectorized solvers (IE, CN, expM, EV) exhibit larger run times with CN being the fastest solver in this group. The adaptive time stepping schemes ode45 and ode45$_2$ cannot be compared directly, since they chooses their own time grids based on given tolerances. For the chosen tolerances, ode45 was using 18349 respectively 4463 time points in the two Bloch examples with relaxation effects from white matter compared to the 1668 respectively 399 that were used by all other solvers. This factor of 11 increase in numerical effort brings an increased accuracy compared to SY only in Example 1. Especially in Example 2, ode45 can give comparable accurate results than SY only with a drastically increased computational effort, which renders ode45 not competitive to operator splitting methods for Bloch and Bloch-McConnell models. In contrast, ode45$_2$ uses 1667 and 849 points in time, respectively, but its accuracy results are not competitive compared to other methods. For all solution methods we observed a greater run time for Example 2 (399 time discretization points, 1000 spatial discretization points) compared with Example 1 (1668 time discretization points, 100 spatial discretization points) which shows the significant influence of the number of spatial points on the performance of the methods. It is noted, that each spatial point is treated independently and can be assigned to have it's own T_1 and T_2. Therefore, the extension to use spatially dependent relaxation times is straight forward and does not impact run time. While the work rigorously analyses accuracy of different methods, some topics regarding the implementation (distributed-memory systems, GPU programming) are beyond the scope.

An important aspect of a Bloch solver is the preservation of the total magnetization in absence of relaxation effects. A study showed that all solvers in the test set but IE precisely preserve total magnetization. In contrast, IE shows a severe loss of magnetization throughout space. This renders standard IE infeasible for Bloch equations, also in presence of relaxation effects, where IE looses much faster total magnetization than the relaxation shown by all the other solvers in that case.

One aspect regarding usability of a numerical solution method was the MT effect and therein especially the very short relaxation times which need to be dissolved accordingly. We presented here two options of including the MT. Within the Bloch-McConnell equations, the choice of the time step length was crucial. This can be seen in Table 7 where a finer time discretization leads to an increase in numerical accuracy. This is also the case in Table 8, where an increased time step length yields a better numerical accuracy for the MT pool. In particular, a linear convergence for ASY and a quadratic convergence for SY are observed again. Within the model of Graham and Henkelman [24] the splitting methods yielded already a better accuracy with a larger time step length. However, the choice of the line shape (we decided for a Lorentzian here) is an issue, Table 9. The short time step length together with the small T_2 relaxation time of the MT pool indicates that the error is governed by the z-component.

There exist many software toolboxes, among them MRISIMUL [60], MRiLab [61], SIMRI [62] and JEMRIS [63], that can simulate whole MR experiments including k-space acquisition and image reconstruction, which are not covered here. The Bloch equations are solved using an asymmetric operator splitting as described in [34] by means of GPU or multi-threading CPU techniques, thus making them computationally affordable methods. Moreover, MRiLab [61] is a software tool that includes MT and CEST as well. In contrast to the aforementioned software tools, this paper investigates MR simulations on a low level, namely just solving the Bloch- and Bloch-McConnell equations. We note, that GPU implementations of asymmetric operator splitting methods can be extended comfortably to symmetric operator splitting methods. In particular, a GPU implementation of the symmetric operator splitting for Bloch- and Bloch-McConnell and a further expansion to simulate whole MR experiments are logical improvements of this work.

For the Bloch equations without relaxation effects, the spin domain solution algorithm SD is the method of choice. It provides an exact solution in this case with a smaller run time compared to the other solution methods. For the Bloch equations with relaxation effects, EV provides the only general exact solution. However, it suffers from a severe increase in run time. Therefore, it is the method of choice for a single computation where the computational effort is not a disadvantage or for the sake of comparison to other numerical solution methods. For numerous calculations, for example during iterative RF pulse optimization, the symmetric operator splitting is the method of choice. It does not only provide a solution method with a good numerical accuracy, but is also one of the fastest numerical solution methods. Furthermore, it is the method of choice for the Bloch-McConnell models since the run time is only slightly increased compared to ASY and it can dissolve relaxation effects accordingly.

6. Conclusion

In this work exact and numerical solution methods for the Bloch and Bloch-McConnell equations were introduced, accuracy and performance properties were analyzed, and a recommendation was drawn. In absence of relaxation effects, the numerical results confirm the rotation matrices approach as accurate and computationally efficient Bloch solution method. Otherwise, as well as for the Bloch-McConnell equations symmetric operator splitting methods are recommended due to it's excellent numerical accuracy paired with efficient run time.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

Definition 1 (*Rotation Operator*). The rotation operator G_{rot} is given as

$$G_{\text{rot}}(\tau)=\begin{pmatrix} n_1^2(1-\cos(\varphi))+\cos(\varphi) & n_1n_2(1-\cos(\varphi))-n_3\sin(\varphi) & n_1n_3(1-\cos(\varphi))+n_2\sin(\varphi) \\ n_1n_2(1-\cos(\varphi))+n_3\sin(\varphi) & n_2^2(1-\cos(\varphi))+\cos(\varphi) & n_2n_3(1-\cos(\varphi))-n_1\sin(\varphi) \\ n_1n_3(1-\cos(\varphi))-n_2\sin(\varphi) & n_2n_3(1-\cos(\varphi))+n_1\sin(\varphi) & n_3^2(1-\cos(\varphi))+\cos(\varphi) \end{pmatrix}.$$

Therein, $\overrightarrow{n}(\tau)=(n_1(\tau),n_2(\tau),n_3(\tau))^T$ is the rotation axis and $\varphi(\tau)$ is the rotation angle defined as

$$\varphi(\tau)=-\gamma\tau\sqrt{\left|B_{1,x}(\tau)-iB_{1,y}(\tau)\right|^2+(z\cdot G_s(\tau))^2},$$
$$n_1(\tau)=\gamma\tau\frac{B_{1,x}(\tau)}{|\varphi(\tau)|},\qquad n_2(\tau)=\gamma\tau\frac{B_{1,y}(\tau)}{|\varphi(\tau)|},\qquad n_3(\tau)=\gamma\tau\frac{z\cdot G_s(\tau)}{|\varphi(\tau)|}.$$

Definition 2 (*Exact solution based on eigenvalues and generalized eigenvectors*). Below, the homogeneous part of the exact solution for the piecewise constant Bloch equation is derived based on eigenvalues and generalized eigenvectors, confer [31, Thm. 6.7.1]. Calculating the characteristic polynomial of the system matrix $\mathbf{A}_n$ leads to a standard eigenvalue problem

$$\chi(\lambda)=\lambda^3+p\lambda^2+q\lambda+r\overset{!}{=}0 \tag{23}$$

for some coefficients $p,q,r\in\mathbb{R}$, with three roots/eigenvalues λ_1, λ_2 and λ_3. Based on the properties of geometric and algebraic multiplicity of these eigenvalues, the solution can be calculated. However, for a complete set of solution cases, the definition of generalized eigenvectors is recalled.

$\mathbf{v}\in\mathbb{C}^n\setminus\{0\}$ is called *generalized eigenvector* of order $l\in\mathbb{N}$ corresponding to the eigenvalue λ of the matrix $\mathbf{A}\in\mathbb{C}^{n\times n}$, if

$$(\mathbf{A}-\lambda\mathbf{I})^l\mathbf{v}=\mathbf{0}\quad\text{and}\quad(\mathbf{A}-\lambda\mathbf{I})^{l-1}\mathbf{v}\neq\mathbf{0}, \tag{24}$$

where $\mathbf{I}$ is the identity matrix in $\mathbb{C}^{n\times n}$.

With this definition, the general solution of the homogeneous equation

$$\frac{d\mathbf{M}_h}{dt}=\mathbf{A}_n\mathbf{M}_h \tag{25}$$

is now given in seven cases.

Case 1: λ_1 has algebraic multiplicity 3. Therefore $\lambda_1=\lambda_2=\lambda_3\in\mathbb{R}$.

- Case 1a: λ_1 has geometric multiplicity 1. Let $\mathbf{v}_1$ be the eigenvector corresponding to λ_1. The corresponding generalized eigenvectors of order 2 and 3 are $\mathbf{u}_2$ and $\mathbf{u}_3$. Now the general solution of the homogeneous Eq. (25) is given as

$$\begin{aligned}\mathbf{M}_h(t)=&\,c_1e^{\lambda_1t}\mathbf{v}_1+c_2e^{\lambda_1t}(\mathbf{u}_2+t(\mathbf{A}_n-\lambda_1\mathbf{I})\mathbf{u}_2)\\&+c_3e^{\lambda_1t}\left(\mathbf{u}_3+t(\mathbf{A}_n-\lambda_1\mathbf{I})\mathbf{u}_3+\tfrac{t^2}{2}(\mathbf{A}_n-\lambda_1\mathbf{I})^2\mathbf{u}_3\right).\end{aligned} \tag{26}$$

  with constants c_1,c_2,c_3 in $\mathbb{R}$.
- Case 1b: λ_1 has geometric multiplicity 2. Let $\mathbf{v}_1$ and $\mathbf{v}_2$ the corresponding linearly independent eigenvectors. Let $\mathbf{u}_2$ be the generalized eigenvector of order 2. Then the homogeneous solution is

$$\mathbf{M}_h(t)=c_1e^{\lambda_1t}\mathbf{v}_1+c_2e^{\lambda_1t}\mathbf{v}_2+c_3e^{\lambda_1t}(\mathbf{u}_2+t(\mathbf{A}_n-\lambda_1\mathbf{I})\mathbf{u}_2). \tag{27}$$

- Case 1c: λ_1 has geometric multiplicity 3. Therefore, we can choose three linearly independent eigenvectors $\mathbf{v}_1,\mathbf{v}_2$ and $\mathbf{v}_3$ and the homogeneous solution is given as

$$\mathbf{M}_h(t)=c_1e^{\lambda_1t}\mathbf{v}_1+c_2e^{\lambda_1t}\mathbf{v}_2+c_3e^{\lambda_1t}\mathbf{v}_3. \tag{28}$$

Case 2: λ_1 has algebraic multiplicity 1 and $\lambda_2\neq\lambda_1$ has algebraic multiplicity 2, i.e. $\lambda_2=\lambda_3\in\mathbb{R}$.

- Case 2a: λ_2 has geometric multiplicity 1. Let $\mathbf{v}_1$ the be eigenvector corresponding to λ_1 and $\mathbf{v}_2$ the one corresponding to λ_2. Let the generalized eigenvector of order 2 associated with λ_2 be $\mathbf{u}_2$. Now the homogeneous solution is

$$\mathbf{M}_h(t)=c_1e^{\lambda_1t}\mathbf{v}_1+c_2e^{\lambda_2t}\mathbf{v}_2+c_3e^{\lambda_2t}(\mathbf{u}_2+t(\mathbf{A}_n-\lambda_2\mathbf{I})\mathbf{u}_2). \tag{29}$$

- Case 2b: λ_1 has geometric multiplicity 1 with eigenvector $\mathbf{v}_1$ and λ_2 has geometric multiplicity 2 with linearly independent eigenvectors $\mathbf{v}_2$ and $\mathbf{v}_3$ which leads to

$$\mathbf{M}_h(t)=c_1e^{\lambda_1t}\mathbf{v}_1+c_2e^{\lambda_2t}\mathbf{v}_2+c_3e^{\lambda_2t}\mathbf{v}_3. \tag{30}$$

Case 3: Three different eigenvalues λ_1,λ_2 and λ_3. Therefore, the geometric multiplicity of each eigenvalue is 1.

- Case 3a: The eigenvalues are real, which means that the corresponding eigenvectors $\mathbf{v}_1,\mathbf{v}_2$ and $\mathbf{v}_3$ are real as well and the homogeneous solution is

$$\mathbf{M}_h(t)=c_1e^{\lambda_1t}\mathbf{v}_1+c_2e^{\lambda_2t}\mathbf{v}_2+c_3e^{\lambda_3t}\mathbf{v}_3. \tag{31}$$

- Case 3b: λ_1 is real, but λ_2 is not. It is well-known that non-real eigenvalues can arise only in pairs of conjugated eigenvalues, i.e. $\lambda_{2,3}=\alpha\pm i\beta$ with corresponding eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_{2,3}=\mathbf{c}\pm i\mathbf{d}$. Here the homogeneous solution is

$$\begin{aligned}\mathbf{M}_h(t)=&\,c_1\mathbf{v}_1e^{\lambda_1t}+c_2e^{\alpha t}(\cos(\beta t)\cdot\mathbf{c}-\sin(\beta t)\cdot\mathbf{d})\\&+c_3e^{\alpha t}(\sin(\beta t)\cdot\mathbf{c}+\cos(\beta t)\cdot\mathbf{d}).\end{aligned} \tag{32}$$

Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.jmr.2021.107011.

References

[1] F. Bloch, Nuclear induction, Phys. Rev. 70 (1946) 460–474.
[2] D. Hoult, The solution of the Bloch equations in the presence of a varying B1 field - an approach to selective pulse analysis, J. Magn. Reson. 35 (1979) 69–86.

[3] P. Mansfield, P. Morris, NMR Imaging in Biomedicine, Elsevier Academic Press, Oxford, 1982.
[4] A. Sbrizzi, H. Hoogduin, J.V. Hajnal, C.A. van den Berg, P.R. Luijten, S.J. Malik, Optimal control design of turbo spin-echo sequences with applications to parallel-transmit systems, Magn. Reson. Med. 77 (2017) 361–373.
[5] E. Van Reeth, H. Ratiney, M. Tesch, D. Grenier, O. Beuf, S.J. Glaser, D. Sugny, Optimal control design of preparation pulses for contrast optimization in MRI, J. Magn. Reson. 279 (2017) 39–50.
[6] D. Ma, V. Gulani, N. Seiberlich, K. Liu, J. Sunshine, J. Duerk, M. Griswold, Magnetic resonance fingerprinting, Nature 495 (2013) 187–192.
[7] A. Sbrizzi, T. Bruijnen, O. van der Heide, P.R. Luijten, C.A. van den Berg, Dictionary-free MR fingerprinting reconstruction of balanced-GRE sequences. Working paper, 2017.
[8] J. Assländer, S.J. Glaser, J. Hennig, Pseudo steady-state free precession for MR-fingerprinting, Magn. Reson. Med. 77 (2017) 1151–1161.
[9] P. Lee, L. Watkins, T. Anderson, G. Buonincontri, B. Hargreaves, Flexible and efficient optimization of quantitative sequences using automatic differentiation of Bloch simulations, Magn. Reson. Med. 82 (2019) 1438–1451.
[10] M. Doneva, P. Börnert, H. Eggers, C. Stehning, J. Sénégas, A. Mertins, Compressed sensing reconstruction for magnetic resonance parameter mapping, Magn. Reson. Med. 64 (2010) 1114–1120.
[11] O. Maier, J. Schoormans, M. Schloegl, G. Strijkers, A. Lesch, T. Benkert, T. Block, B. Coolen, K. Bredies, R. Stollberger, Rapid T1 quantification from high resolution 3D data with model-based reconstruction, Magn. Reson. Med. 81 (2019) 2072–2089.
[12] J. Pauly, D. Nishimura, A. Macovski, A k-Space Analysis of Small-Tip-Angle Excitation, J. Mang. Reson. 81 (1989) 43–56.
[13] H. Zheng, T. Zhao, Y. Qian, T. Ibrahim, F. Boada, Improved large tip angle parallel transmission pulse design through a perturbation analysis of the bloch equations, Magn. Reson. Med. 66 (3) (2011) 687–696.
[14] S. Conolly, D. Nishimura, A. Macovski, Optimal control solutions to the magnetic resonance selective excitation problem, IEEE Trans. Med. Imaging 5 (1986) 106–115.
[15] D. Xu, K.F. King, Y. Zhu, G.C. McKinnon, Z. Liang, Designing multichannel, multidimensional, arbitrary flip angle RF pulses using an optimal control approach, Magn. Reson. Med. 59 (2008) 547–560.
[16] W.A. Grissom, D. Xu, A.B. Kerr, J.A. Fessler, D.C. Noll, Fast large-tip-angle multidimensional and parallel RF pulse design in MRI, IEEE Trans. Med. Imaging 28 (2009) 1548–1559.
[17] M.S. Vinding, I.I. Maximov, Z. Tošner, N.C. Nielsen, Fast numerical design of spatial-selective RF pulses in MRI using Krotov and quasi-Newton based optimal control methods, J. Chem. Phys. 137 (2012) 054203.
[18] C.S. Aigner, C. Clason, A. Rund, R. Stollberger, Efficient high-resolution RF pulse design applied to simultaneous-multislice excitation, J. Magn. Reson. 263 (2016) 33–44.
[19] A. Rund, C.S. Aigner, K. Kunisch, R. Stollberger, Magnetic resonance RF pulse design by optimal control with physical constraints, IEEE Trans. Med. Imaging 37 (2018) 461–472.
[20] A. Rund, C.S. Aigner, K. Kunisch, R. Stollberger, Simultaneous multislice refocusing via time optimal control, Magn. Reson. Med. 80 (2018) 1416–1428.
[21] C.S. Aigner, A. Rund, S. Abo Seada, A.N. Price, J.V. Hajnal, S.J. Malik, K. Kunisch, R. Stollberger, Time optimal control-based RF pulse design under gradient imperfections, Magn. Reson. Med. 83 (2019) 561–574.
[22] H.M. McConnell, Reaction rates by nuclear magnetic resonance, J. Chem. Phys. 28 (1958) 430–431.
[23] P.C. van Zijl, N.N. Yadav, Chemical exchange saturation transfer (CEST): What is in a name and what isn't?, Magn Reson. Med. 65 (2011) 927–948.
[24] S.J. Graham, R.M. Henkelman, Understanding Pulsed Magnetization Transfer, J. Magn. Reson. Imaging 7 (5) (1997) 903–912.
[25] D.M. Garcia, G. Duhamel, D.C. Alsop, Efficiency of Inversion Pulses for Background Suppressed Arterial Spin Labeling, Magn. Reson. Med. 54 (2005) 366–372.
[26] H. Goldstein, Classical Mechanics, First ed., Addison-Wesley, United States of America, 1951.
[27] J. Pauly, P. Le Roux, D. Nishimura, A. Macovski, Parameter relations for the Shinnar-Le Roux selective excitation pulse design algorithm, IEEE Trans. Med. Imaging 10 (1991) 53–65.
[28] M.A. Bernstein, K.F. King, X.J. Zhou, Handbook of MRI Pulse Sequences, Elsevier Academic Press, Oxford, 2004.
[29] D.J. Siminovitch, Rotations in NMR: Part I. Euler–Rodrigues parameters and quaternions, Concepts Magn. Reson. 9 (1997) 149–171.
[30] M.S. Silver, R.I. Joseph, D.I. Hoult, Selective spin inversion in nuclear magnetic resonance and coherent optics through an exact solution of the Bloch-Riccati equation, Phys. Rev. A 31 (1985) 4.
[31] J.R. Brannan, W.E. Boyce, Differential Equations: An Introduction to Modern Methods and Applications, third ed., Wiley, 2015.
[32] D.E. Rourke, A.A. Karabanov, G.H. Booth, I. Frantsuzov, The Bloch equations when T1=T2, Inverse Prob. 23 (2007) 609–623.
[33] C. Moler, C. Van Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later, SIAM Rev. 49 (2003) 3–49.
[34] J. Bittoun, J. Taquin, M. Sauzade, A Computer Algorithm for the Simulation of any Nuclear Magnetic Resonance (NMR) Imaging Method, J. Magn. Reson. 2 (1984) 113–120.
[35] B. Hargreaves, Bloch equation simulator, http://mrsrl.stanford.edu/brian/blochsim/ (accessed: 2019-07-01).
[36] A. Hazra, Numerical Simulation of Bloch Equations for Dynamic Magnetic Resonance Imaging (Ph.D. thesis), Georg-August-Universität Göttingen, 2016.
[37] L. Vidarsson, T2-Selective Magnetization Preparation Pulses, IEEE Trans. Med. Imaging 26 (2007) 981–989.
[38] M. Rezaeian, G. Hossein-Zadeh, H. Soltanian-Zadeh, Numerical solutions to the Bloch-McConnell equations with radio frequency irradiation scheme for CEST MRI, in: ICEE 2012–20th Iranian Conference on Electrical Engineering, 2012.
[39] M. Zaiss, Z. Zhongliang, X. Junzhong, P. Schuenke, D.F. Gochberg, J.C. Gore, M. Ladd, P. Bachert, A combined analytical solution for chemical exchange saturation transfer and semi-solid magnetization transfer, NMR Biomed. 28 (2015) 217–230.
[40] G. Xiao, R. Wu, Sun P. Zhe, Fast simulation and optimization of pulse-train chemical exchange saturation transfer (CEST) imaging, Phys. Med. Biol. 60 (2015) 4719–4730.
[41] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506–517.
[42] R. McLachlan, G. Quispel, Splitting methods, Acta Numer. 11 (2002) 341–434.
[43] K. Majewski, Rotation relaxation splitting for optimizing parallel RF excitation pulses with T1- and T2-relaxations in MRI, J. Magn. Reson. 288 (2018) 43–57.
[44] N. Higham, The scaling and squaring method for the matrix exponential revisited, SIAM J. Matrix Anal. Appl. 26 (2005) 1179–1193.
[45] A. Al-Mohy, N. Higham, A new scaling and squaring algorithm for the matrix exponential, SIAM J. Matrix Anal. Appl. 31 (2009) 970–989.
[46] M. Ascher, S. Ruuth, R. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, Appl. Numer. Math. 25 (1997) 151–167.
[47] E. Hairer, S.P. Norsett, G. Wanner, Solving Ordinary Differential Equations I: Nonstiff Problems, Springer Series in COmputional Mathematics (2009).
[48] J. Crank, P. Nicolson, A practical method for numerical evaluation of solutions of partial differential equations of the heat-conductive type, Adv. Comput. Math. 43 (1996) 50–67.
[49] J. Dormand, P. Prince, A family of embedded Runge-Kutta formulae, J. Comput. Appl. Math. 6 (1980) 19–26.
[50] S. MacNamara, G. Strang, Operator Splitting, In: R. Glowinski, S. Osher, W. YinM Splitting Methods in Communication, Imaging, Science, and Engineering. Scientific Computation, Springer, Cham, 2016.
[51] M. Gloor, Magnetization Transfer Imaging Using Steady-State Free Precession MR Sequences (Ph.D. thesis), Universität Basel, 2010.
[52] R.M. Henkelman, X. Huang, Q.S. Xiang, G.J. Stanisz, S.D. Swanson, M.J. Bronskill, Quantitative interpretation of magnetization transfer, Magn. Reson. Med. 29 (6) (1993) 759–766.
[53] C.S. Aigner, A. Rund, C. Graf, K. Kunisch, R. Stollberger, RF pulse design via time optimal control for combined excitation, refocusing and inversion, in: Proceedings of the 27th Annual Meeting of ISMRM, Abstract 1445, 2019.
[54] J. Wansapura, S. Holland, R. Scott Dunn, W. Ball, NMR relaxation times in the human brain at 3.0 Tesla, J. Magn. Reson. Imaging 9 (1999) 531–538.
[55] G. Stanisz, E. Odrobina, J. Pun, M. Escaravage, S. Graham, M. Bronskill, Henkelmann R. Mark, T1, T2 relaxation and magnetization transfer in tissue at 3T, Magn. Reson. Med. 54 (2005) 507–512.
[56] G. Filho, J. Du, B. Pak, S. Statum, R. Znamorowski, P. Haghighi, G. Bydder, C. Chung, Quantitative characterization of the Achilles tendon in cadaveric specimens: T1 and T2$*$ measurements using ultrashort-TE MRI at 3T, AJR Am. J. Roentgenol. 192 (2009) 117–124.
[57] M. Zaiss, P. Bachert, Exchange-dependent relaxation in the rotating frame for slow and intermediate exchange - modeling off-resonant spin-lock and chemical exchange saturation transfer, NMR Biomed. 26 (2013) 507–518.
[58] E. Süli, D. Mayers, An Introduction to numerical analysis, Cambridge University Press, Cambridge, 2003.
[59] R.M. Summers, L. Axel, S. Israel, A computer simulation of nuclear magnetic resonance imaging, Magn. Reson. Med. 3 (1986) 363–376.
[60] C.G. Xanthis, I.E. Venetis, A.V. Chalkias, A.H. Aletras, MRISIMUL: A GPU-Based Parallel Approach to MRI Simulations, IEEE Trans. Med. Imaging 33 (2014) 607–617.
[61] F. Liu, J.V. Velikkina, W.F. Block, R. Kijowski, A.A. Samsonov, Fast Realistic MRI Simulations Based on Generalized Multi-Pool Exchange Tissue Model, IEEE Trans. Med. Imaging 36 (2) (2017) 527–537.
[62] H. Benoit-Cattin, G. Collewet, B. Belaroussi, H. Saint-Jalmes, C. Odet, The SIMRI project: a versatile and interactive MRI simulator, J. Magn. Reson. 173 (2005) 97–115.
[63] T. Stöcker, K. Vahedipour, D. Pflugfelder, Sh.ah.N. Jon, High-performance computing MRI simulations, Magn. Reson. Med. 64 (2010) 186–193.