UNIVERSITY OF SOUTHAMPTON

FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

SCHOOL OF CHEMISTRY

Quantum mechanical simulation of magnetic resonance imaging

by

Ahmed J. Allami

Thesis for the degree of Doctor of Philosophy

January 2019

Abstract

Many recent magnetic resonance imaging experiments involve spin states other than longitudinal or transverse magnetization. For example, singlet state imaging requires two-spin correlations to be taken into account, hyperpolarised imaging should correctly account for multi-spin dynamics, multiple-quantum imaging must consider the dynamics of the corresponding coherences, *etc*. Simulation of such experiments requires accurate quantum mechanical treatment of spin processes together with an accurate treatment of classical processes, such as diffusion and convection.

In this thesis I report the theoretical and software infrastructure for magnetic resonance simulations using the Fokker-Planck formalism, which simultaneously accounts for spatial dynamics and quantum mechanical spin processes. Fokker-Planck equation is superior to the usual Liouville - von Neumann equation formalism in that spatial dynamics processes (diffusion, hydrodynamics, magic angle spinning, off-resonance pulses, *etc*.) are represented by constant matrices that are more convenient from the programming and numerical efficiency point of view than time-dependent Hamiltonians in the Liouville - von Neumann equation formalism.

It is demonstrated below that NMR and MRI experiments with elaborate spatial encoding and complicated spatial dynamics are no longer hard to simulate, even in the presence of spin-spin couplings and exotic relaxation effects, such as singlet state symmetry lockout. Versions 2.3 and later of *Spinach* library support arbitrary stationary flows and arbitrary distributions of anisotropic diffusion tensors in three dimensions simultaneously with Liouville-space description of spin dynamics, chemical kinetics and relaxation processes. The key simulation design decision that has made this possible is the abandonment of Bloch-Torrey and Liouville - von Neumann formalisms in favour of the Fokker-Planck equation. The primary factors that have facilitated this transition are the dramatic recent improvement in the speed and capacity of digital computers, the emergence of transparent and convenient sparse matrix manipulation methods in numerical linear algebra, and the recent progress in matrix dimension reduction in magnetic resonance simulations.

The principal achievement of this thesis is in programming and software engineering – the reader is encouraged to take a look at the Fokker-Planck module of the *Spinach* library that the work described in this thesis has made possible, and that was programmed as a part of this work: most of the writing performed within this project was writing *code*.

Acknowledgements

I would like first to thank the High Committee for Education Development in Iraq (HCED) to give me the opportunity to study in the UK.

A number of people have influenced this work, particularly my supervisor Prof Ilya Kuprov; without his advice, I would not have had the confidence to undertake this project.

Academics working in the area of nuclear magnetic resonance and magnetic resonance imaging that have had a direct involvement in this work through collaboration and conversations are Ludmilla Guduff, Carine van Heijenoort and Jean-Nicolas Dumez. Singlet state work was an outcome of an interesting conversation with Giuseppe Pileio and Malcolm Levitt.

People who influenced me and strengthened my belief in myself are Liza Suturina and David Goodwin. I am also very grateful to Prof George Attard, who has strongly supported Ilya and I when we fought, and eventually won, a six-month battle with Southampton University, which initially refused to believe that I could speak English, and simply rejected my application.

Finally, a personal acknowledgement is due to my wife Taisir, my dad and my mom who helped me in an uncountable number of ways. These people have become a bedrock for support for me, accepting my eccentricities without question. I am also grateful to my brother and sisters who support me all the time.

This thesis is dedicated to my daughters Reham and Retal, born in UK during my PhD journey, and who I hope will surpass all of my achievements.

Ahmed J. Allami

List of figures

List of tables

Contents

Objectives and chapter outlines

Simulation software is essential for the continued development of magnetic resonance imaging (MRI) instruments and experiments. On the instrumental side (electromagnetic waves, their propagation and detection) classical simulations using Maxwell's equations usually suffice. On the experiment side (spin dynamics), the current way is to use Bloch equations at each voxel and diffusion between voxels [1]. Sophisticated MRI simulation packages, some of them advanced enough to use GPU processing, have existed for some time. A major problem, however, is that all current MRI simulation packages are unable to handle situations when quantum mechanical spin interactions play a role in the dynamics: *J*-couplings, nuclear quadrupolar effects, cross-relaxation and other quantum mechanical processes cannot be described by Bloch equations [2].

The need for quantum mechanical description of MRI is increasingly obvious: a number of research groups are developing singlet state MRI [3, 4], other groups are already performing clinical trials of hyperpolarized MRI [5, 6], and research is being carried out into selective excitation schemes using optimal control theory [7]. Due to lack of institutional experience with quantum mechanics within the MRI community, no simulation packages exist that could handle the spin sub-space quantum mechanically and this is the significance point of this work. Hence, the aim of this thesis is to investigate and implement quantum mechanical MRI simulation techniques. Specifically:

1. To develop a physical theory that combines quantum mechanical description of nuclear spin dynamics, classical description of electrodynamics and classical description of spatial dynamics (diffusion and flow) into a single algebraic framework.

2. To apply the resulting framework to simulation of magnetic resonance imaging experiments that essentially include quantum effects, such as singlet state MRI and metabolic MRI.

3. To solve or mitigate the matrix dimension problem in quantum mechanical simulations of magnetic resonance imaging.

The work is partitioned into the following chapters:

Chapter 1:

An introduction to spin in both classical and quantum approaches, Bloch equations and Liouville - von Neumann equation. A review the literature on MRI concepts such as frequency encoding, phase encoding, *k*-space trajectory, *etc.* NMR spectroscopy is also present in this review *via* ultrafast NMR and pure shift NMR. A brief review of spin Hamiltonians, relaxation process, spatio-temporal building

blocks of MRI, simulation of magnetic resonance imaging experiments using the Fokker-Planck equation and a description of the process of moving magnetic resonance simulations from Liouville - von Neumann equation to the Fokker-Planck equation.

Chapter 2:

Mathematical description of the simulation platform and solving the matrix dimension problem in the equation of motion. A detailed walk through the formal mathematics, including derivations for the various dynamics generators. A discussion of the merits of various numerical methods in the solution of the resulting equations.

Chapter 3:

Simulation of magnetic resonance imaging experiments using the Fokker-Planck equation. The chapter goes through different MRI contrast modes and describes the simulation of common MRI stages like slice selection, spin echo, *etc*. Examples involving complicated spatial dynamics are given: diffusion weighted MRI, localized NMR, and the bright fat effect.

Chapter 5:

In this chapter, a brief description of the whole thesis, the achievements and the limitations are also present. A future look to what I except for further developments.

Chapter 6:

A paper published in collaboration with L. Guduff, C. van Heijenoort, and J.N. Dumez.

1 State of the art

1.1 Classical description of NMR

In 1937, Rabi discovered magnetic resonance in two different ways [8]: firstly, he concluded that the particle in the molecule beam was absorbing energy from an oscillating field and moving from one quantum state to another. Secondly, in the same beam, particles could precess around the external magnetic field. In 1945, Purcell, Torrey and Pound discovered a signal from protons in paraffin [9], while Bloch and his group at the same time observed a signal from protons in liquid water [2].

Resonance happens in the nuclear spins when the electromagnetic radiation frequency matches the energy difference between the nuclear spin levels. However, not all nuclei resonate at the same frequency due to the shielding effect of the electrons surrounding the nucleus. Electrons produce an opposite magnetic field to the applied magnetic field. As a consequence, the magnetic field at the nucleus is reduced. This leads to a reduction in the energy gap and frequency required to achieve the resonance. The electron shielding effect is called 'chemical shift'. If the nucleus is located in the chemical group with high electron density, the NMR frequency will be shifted upfield, while if the nuclei is located in the lower electron density, then the RF frequency will be downfield. This is very useful to chemists and this is why magnetic resonance is so popular.

1.1.1 Spin

In classical physics, angular momentum $\vec{L}$ is defined as linear momentum $\vec{p}$ rotating around the origin at position $\vec{r}$:

$$\vec{L} = \vec{r} \times \vec{p} \tag{1}$$

$\vec{r}$ can be represented as $\vec{r} = x\vec{i} + y\vec{j} + z\vec{k}$ where $\vec{i}, \vec{j}, \vec{k}$ are unit vectors pointing along the three Cartesian axes. In the same way we can define the linear momentum $\vec{p} = p_x\vec{i} + p_y\vec{j} + p_z\vec{k}$. The angular momentum can be written in terms of its components as well $\vec{L} = L_x\vec{i} + L_y\vec{j} + L_z\vec{k}$. As a result, the angular momentum can be written as [10]:

$$\vec{L} = \vec{r} \times \vec{p} = \begin{pmatrix} \vec{i} & \vec{j} & \vec{k} \\ x & y & z \\ p_x & p_y & p_z \end{pmatrix} = \begin{pmatrix} yp_z - zp_y \\ zp_x - xp_z \\ xp_y - yp_x \end{pmatrix} = \begin{pmatrix} L_x \\ L_y \\ L_z \end{pmatrix} \tag{2}$$

In order to understand the interaction between a spin and the external magnetic field, the circular current loop must be introduced [11]. Suppose we have a particle moving in a circular way, a circular loop of current I and area A will be produced. As a magnetic field is applied, the loop will feel *torque* [12]. In order to understand this mathematically, let us start by differentiating equation (1) with respect to time:

$$\frac{d\vec{L}}{dt} = \vec{r} \times \frac{d\vec{p}}{dt} + \frac{d\vec{r}}{dt} \times \vec{p} \tag{3}$$

Now using the definition of force $\vec{F} = d\vec{p}/dt$ (according to the Newton's second law) and the definition of velocity $\vec{v} = d\vec{r}/dt$, we can re-write equation (3) as follows:

$$\frac{d\vec{L}}{dt} = \vec{r} \times \vec{F} + \vec{v} \times \vec{p} \tag{4}$$

The cross product of momentum $\vec{p}$ with its associated velocity $\vec{v}$ is zero because velocity and momentum are parallel, so the second term vanishes.

$$\frac{d\vec{L}}{dt} = \vec{r} \times \vec{F} \tag{5}$$

From literature [12] we know that the torque $\vec{N}$ can be written as:

$$\vec{N} = \vec{r} \times \vec{F} \tag{6}$$

When we put equation (6) into equation (5), the result implies that the angular momentum is evolving according to equation below:

$$\frac{d\vec{L}}{dt} = \vec{N} \tag{7}$$

The formula for the net torque on any current distribution is [13]:

$$\vec{N} = \vec{\mu} \times \vec{B} \tag{8}$$

Where $\vec{\mu}$ is magnetic dipole moment which can be defined for an infinitesimal loop as:

$$\vec{\mu} = I\vec{A} \tag{9}$$

Where I is the current, $\vec{A}$ is area vector which is perpendicular to the plane of the loop.

Proton spin can be thought of as a point magnetic dipole. The relation between spin and its magnetic moment $\vec{\mu}$ can be written as [14]

$$\vec{\mu} = \gamma \vec{L} \tag{10}$$

Where γ is gyromagnetic ratio. By substituting equation (8) and (10) in equation (7) we obtain:

$$\frac{1}{\gamma}\frac{d\vec{\mu}}{dt} = \vec{\mu} \times \vec{B} \tag{11}$$

By re-arranging equation (11) we get Bloch equation [2]

$$\frac{d\vec{\mu}}{dt} = \gamma\left(\vec{\mu} \times \vec{B}\right) \tag{12}$$

For a macroscopic body, the spins can be introduced as their magnetic moment per unit volume V; this is called nuclear magnetization $\vec{M}\left(\vec{r},t\right)$:

$$\vec{M} = \frac{1}{V}\sum_{i \in \text{spins}} \vec{\mu}_i \tag{13}$$

Applying equation (13) in equation (12) we will get:

$$\frac{d\vec{M}}{dt} = \gamma\left(\vec{M} \times \vec{B}_0\right) \tag{14}$$

It is important to describe the magnetization in terms of parallel and perpendicular components to the external magnetic field which relax differently to their equilibrium values [15]. Parallel component, also called longitudinal component is M_z while the perpendicular (transverse) component is:

$$M_{\perp} = \begin{pmatrix} M_{\mathrm{x}} \\ M_{\mathrm{y}} \end{pmatrix} \tag{15}$$

For non-interacting spins, the differential equation related to equation (15) is:

$$\frac{dM_{\mathrm{z}}}{dt} = 0 \tag{16}$$

$$\frac{d\vec{M}_{\perp}}{dt} = \gamma\left(\vec{M}_{\perp} \times \vec{B}_0\right) \tag{17}$$

where the external field $\vec{B}_0$ is always directed along the Z-axis and × represents the vector cross product.

1.1.2 Spin relaxation

The constant interaction with lattice implies that the time derivative of longitudinal magnetization is proportional to the difference between the equilibrium value and the coherent magnetization:

$$\frac{d\vec{M}_z}{dt} \propto \left(\vec{M}_0 - M_z\right) \quad \Rightarrow \quad \frac{d\vec{M}_z}{dt} = \frac{1}{T_1}\left(\vec{M}_0 - M_z\right) \tag{18}$$

Where T_1 is the longitudinal relaxation time, also called 'spin-lattice' relaxation time [16]. If the magnetization is perturbed, the solution of equation (18) shows the evolution from the initial state $M_z(0)$ to the equilibrium state M_0 :

$$M_z(t) = M_z(0)e^{-t/T_1} + M_0\left(1 - e^{-t/T_1}\right) \tag{19}$$

Where T_1 is the longitudinal relaxation time, $M_z(0)$ is the initial magnetization, and M_0 is the equilibrium magnetization.

Moving to the transverse magnetization, equation (17) has an additional evolution term when spin-lattice interaction happens. When the magnetization exists in the *xy* plane, it will to return to zero. Hence, equation (17) will acquire a relaxation term:

$$\frac{d\vec{M}_\perp}{dt} = \gamma\left(\vec{M}_\perp \times \vec{B}_0\right) - \frac{1}{T_2}\vec{M}_\perp \tag{20}$$

Where T_2 is the transverse relaxation time. It is more complicated to follow the magnetization precessing around magnetic field and to simplify it we must rotate the observer together with spin vector. In this situation, equation (20) can be written as:

$$\frac{d\vec{M}_\perp}{dt} = -\frac{1}{T_2}\vec{M}_\perp \tag{21}$$

With following solution:

$$\vec{M}_\perp(t) = \vec{M}_\perp(0)e^{-t/T_2} \tag{22}$$

1.2 Liouville space description of NMR

1.2.1 Quantum mechanical theory of spin

In quantum mechanics, the description of spin is different from the description of angular momentum, although similarities are significant. Uhlenbeck and Goudsmit realized in early 1920 that experimental data cannot be described unless an intrinsic form of angular momentum is postulated in addition to the orbital angular momentum [13].

Replacing quantities with their operators in equation (3) yields:

$$\hat{\vec{L}} = \hat{\vec{r}} \times \hat{\vec{p}} = -i\hbar \begin{pmatrix} \vec{i} & \vec{j} & \vec{k} \\ x & y & z \\ \partial/\partial x & \partial/\partial y & \partial/\partial z \end{pmatrix} = -i\hbar \begin{pmatrix} y\dfrac{\partial}{\partial z} - z\dfrac{\partial}{\partial y} \\ z\dfrac{\partial}{\partial x} - x\dfrac{\partial}{\partial z} \\ x\dfrac{\partial}{\partial y} - y\dfrac{\partial}{\partial x} \end{pmatrix} = \begin{pmatrix} \hat{L}_{\mathrm{x}} \\ \hat{L}_{\mathrm{y}} \\ \hat{L}_{\mathrm{z}} \end{pmatrix} \tag{23}$$

Then, the three components of the angular momentum vector can be written as:

$$\hat{L}_{\mathrm{X}} = -i\hbar\left(y\frac{\partial}{\partial z} - z\frac{\partial}{\partial y} \right); \quad \hat{L}_{\mathrm{Y}} = -i\hbar\left(z\frac{\partial}{\partial x} - x\frac{\partial}{\partial z} \right); \quad \hat{L}_{\mathrm{Z}} = -i\hbar\left(x\frac{\partial}{\partial y} - y\frac{\partial}{\partial x} \right) \tag{24}$$

Components of equation (24), satisfy the following commutation properties:

$$\left[\hat{L}_{\mathrm{x}}, \hat{L}_{\mathrm{y}}\right] = i\hbar\hat{L}_{\mathrm{z}}, \ \left[\hat{L}_{\mathrm{y}}, \hat{L}_{\mathrm{z}}\right] = i\hbar\hat{L}_{\mathrm{x}}, \ \left[\hat{L}_{\mathrm{z}}, \hat{L}_{\mathrm{x}}\right] = i\hbar\hat{L}_{\mathrm{y}} \tag{25}$$

Spin operators can be constructed as finite-dimensional matrices that obey these commutation relations. By analogy with angular momentum, the state space of a spin s is $2s+1$ dimensional. The spin state vector therefore has $2s+1$ complex components. For spin ½, we have:

$$\begin{pmatrix} c_{1/2} \\ c_{-1/2} \end{pmatrix} = c_{1/2}\begin{pmatrix} 1 \\ 0 \end{pmatrix} + c_{-1/2}\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad \left|c_{1/2}\right|^2 + \left|c_{-1/2}\right|^2 = 1 \tag{26}$$

Let us assign the two orthogonal vectors to the two projection states of the spin, that is:

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \left|\alpha\right\rangle, \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \left|\beta\right\rangle \tag{27}$$

This fixes the basis and allows us to build matrix representations of the spin operators. In particular, the two spin projection states should be eigenvectors of the total spin operator $\hat{S}^2$ and the projection

operator $\hat{S}_{\mathrm{Z}}$, which we introduce using the angular momentum analogy. This information yields matrix representations for these operators:

$$\begin{gathered}\hat{S}^2\begin{pmatrix}1\\0\end{pmatrix}=\frac{1}{2}\left(\frac{1}{2}+1\right)\begin{pmatrix}1\\0\end{pmatrix};\quad \hat{S}^2\begin{pmatrix}0\\1\end{pmatrix}=\frac{1}{2}\left(\frac{1}{2}+1\right)\begin{pmatrix}0\\1\end{pmatrix}\quad\Rightarrow\quad \hat{S}^2=\begin{pmatrix}3/4 & 0\\ 0 & 3/4\end{pmatrix}\\ \hat{S}_{\mathrm{Z}}\begin{pmatrix}1\\0\end{pmatrix}=\frac{1}{2}\begin{pmatrix}1\\0\end{pmatrix};\quad \hat{S}_{\mathrm{Z}}\begin{pmatrix}0\\1\end{pmatrix}=-\frac{1}{2}\begin{pmatrix}0\\1\end{pmatrix}\quad\Rightarrow\quad \hat{S}_{\mathrm{Z}}=\begin{pmatrix}1/2 & 0\\ 0 & -1/2\end{pmatrix}\end{gathered} \tag{28}$$

Furthermore, $|\alpha\rangle$ and $|\beta\rangle$ basis vectors should be raised and lowered into each other. This gives matrix representations for the raising and lowering operators:

$$\begin{gathered}\hat{S}_{+}\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix};\quad \hat{S}_{+}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix}\quad\Rightarrow\quad \hat{S}_{+}=\begin{pmatrix}0 & 1\\ 0 & 0\end{pmatrix}\\ \hat{S}_{-}\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}0\\1\end{pmatrix};\quad \hat{S}_{-}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix}\quad\Rightarrow\quad \hat{S}_{-}=\begin{pmatrix}0 & 0\\ 1 & 0\end{pmatrix}\end{gathered} \tag{29}$$

The remaining Cartesian operators can then be constructed from the raising and lowering operators using the relations from the angular momentum theory:

$$\hat{S}_{\mathrm{X}}=\frac{\hat{S}_{+}+\hat{S}_{-}}{2}\qquad \hat{S}_{\mathrm{Y}}=\frac{\hat{S}_{+}-\hat{S}_{-}}{2i} \tag{30}$$

So the operators corresponding to the three Cartesian projections of spin ½ are:

$$\hat{S}_{\mathrm{X}}=\begin{pmatrix}0 & 1/2\\ 1/2 & 0\end{pmatrix}\qquad \hat{S}_{\mathrm{Y}}=\begin{pmatrix}0 & -i/2\\ i/2 & 0\end{pmatrix}\qquad \hat{S}_{\mathrm{Z}}=\begin{pmatrix}1/2 & 0\\ 0 & -1/2\end{pmatrix} \tag{31}$$

These matrices are called Pauli matrices for spin ½. They are a two-dimensional matrix representation of the corresponding spin operators. For spins greater than ½, they have larger dimensions, but may still be constructed using the same method. For example, a spin-1 particle has:

$$\hat{S}_{\mathrm{X}}=\frac{1}{\sqrt{2}}\begin{pmatrix}0 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 0\end{pmatrix},\quad \hat{S}_{\mathrm{Y}}=\frac{1}{\sqrt{2}}\begin{pmatrix}0 & -i & 0\\ i & 0 & -i\\ 0 & i & 0\end{pmatrix},\quad \hat{S}_{\mathrm{Z}}=\begin{pmatrix}1 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & -1\end{pmatrix} \tag{32}$$

It is easy to check by direct inspection that the matrices in Equations (31) and (32) obey the same commutation relations as the angular momentum projection operators:

$$[\hat{S}_{\mathrm{X}},\hat{S}_{\mathrm{Y}}]=i\hat{S}_{\mathrm{Z}};\quad [\hat{S}_{\mathrm{Z}},\hat{S}_{\mathrm{X}}]=i\hat{S}_{\mathrm{Y}};\quad [\hat{S}_{\mathrm{Y}},\hat{S}_{\mathrm{Z}}]=i\hat{S}_{\mathrm{X}};\quad [\hat{S}^2,\hat{S}_{\{\mathrm{X,Y,Z}\}}]=0 \tag{33}$$

Multiplication properties are also different between spin and angular momentum operators.

1.2.2 Spin Hamiltonian

Spin Hamiltonian is the energy operator of the spin system [17]. Hamiltonian eigenvalues are the energy levels of the system. The general spin Hamiltonian can be written as [18]:

$$\begin{gathered}\hat{H}=\hat{H}_{\text{Zeeman}}+\hat{H}_{\text{scalar}}+\hat{H}_{\text{DD+HFC}}+\hat{H}_{\text{Q}}+\hat{H}_{\text{RF}}= \\ =\sum_{k}\hat{\vec{L}}^{(k)}\cdot\mathbf{Z}^{(k)}\cdot\vec{B}+2\pi\sum_{n\neq k}J_{nk}\hat{\vec{L}}^{(n)}\cdot\hat{\vec{S}}^{(k)}+\sum_{n\neq k}\hat{\vec{L}}^{(n)}\cdot\mathbf{D}^{(n,k)}\cdot\hat{\vec{S}}^{(k)}+ \\ +\sum_{k}\hat{\vec{L}}^{(k)}\cdot\mathbf{Q}^{(k)}\cdot\hat{\vec{L}}^{(k)}+2\pi\sum_{k}a_{\text{RF}}^{(k)}\left[\hat{L}_{\text{X}}^{(k)}\cos\left(\omega_{\text{RF}}t\right)+\hat{L}_{\text{Y}}^{(k)}\sin\left(\omega_{\text{RF}}t\right)\right]\end{gathered} \tag{34}$$

The first term is called Zeeman interaction, it is linear in spin operators. It comes from the interaction between spins and the external magnetic field. The full form is:

$$\hat{H}_{\text{z}}=\hat{\vec{L}}\cdot\mathbf{Z}\cdot\vec{B}=\begin{pmatrix}\hat{L}_{\text{x}} & \hat{L}_{\text{y}} & \hat{L}_{\text{z}}\end{pmatrix}\begin{pmatrix}z_{\text{xx}} & z_{\text{xy}} & z_{\text{xz}}\\ z_{\text{yx}} & z_{\text{yy}} & z_{\text{yz}}\\ z_{\text{zx}} & z_{\text{zy}} & z_{\text{zz}}\end{pmatrix}\begin{pmatrix}B_{\text{x}}\\ B_{\text{y}}\\ B_{\text{z}}\end{pmatrix} \tag{35}$$

where the matrix elements are determined by the electronic structure of the system containing the nuclei, and $\hat{\vec{L}}$ is a vector of operators, e.g.

$$\hat{L}_{\text{X}}=\begin{pmatrix}0 & 1/2\\ 1/2 & 0\end{pmatrix}\qquad\hat{L}_{\text{Y}}=\begin{pmatrix}0 & -i/2\\ i/2 & 0\end{pmatrix}\qquad\hat{L}_{\text{Z}}=\begin{pmatrix}1/2 & 0\\ 0 & -1/2\end{pmatrix} \tag{36}$$

The second and third terms in equation (34) are scalar coupling and dipolar (*aka* hyperfine) interaction. They are bilinear in spin operators:

$$\hat{H}_{\text{c}}=\hat{\vec{L}}\cdot\mathbf{A}\cdot\hat{\vec{S}}=\begin{pmatrix}\hat{L}_{\text{x}} & \hat{L}_{\text{y}} & \hat{L}_{\text{z}}\end{pmatrix}\begin{pmatrix}a_{\text{xx}} & a_{\text{xy}} & a_{\text{xz}}\\ a_{\text{yx}} & a_{\text{yy}} & a_{\text{yz}}\\ a_{\text{zx}} & a_{\text{zy}} & a_{\text{zz}}\end{pmatrix}\begin{pmatrix}\hat{S}_{\text{x}}\\ \hat{S}_{\text{y}}\\ \hat{S}_{\text{z}}\end{pmatrix} \tag{37}$$

where $\hat{\vec{L}}$ and $\hat{\vec{S}}$ are vectors of Pauli matrices and $\mathbf{A}$ is the interaction tensor; for a point dipolar coupling the expression is:

$$\hat{H}_{\mathrm{DD}} = -\frac{\mu_0}{4\pi}\frac{\gamma_1\gamma_2\hbar}{r^5}\left(3(\hat{\vec{S}}_1\cdot\vec{r})(\vec{r}\cdot\hat{\vec{S}}_2)-r^2(\hat{\vec{S}}_1\cdot\hat{\vec{S}}_2)\right) = -\frac{\mu_0}{4\pi}\frac{\gamma_1\gamma_2\hbar}{r^5}\times$$
$$\begin{pmatrix}\hat{S}_{\mathrm{X}}^{(1)}\\ \hat{S}_{\mathrm{Y}}^{(1)}\\ \hat{S}_{\mathrm{Z}}^{(1)}\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}3(x_1-x_2)^2-r^2 & 3(x_1-x_2)(y_1-y_2) & 3(x_1-x_2)(z_1-z_2)\\ 3(y_1-y_2)(x_1-x_2) & 3(y_1-y_2)^2-r^2 & 3(y_1-y_2)(z_1-z_2)\\ 3(z_1-z_2)(x_1-x_2) & 3(z_1-z_2)(y_1-y_2) & 3(z_1-z_2)^2-r^2\end{pmatrix}\begin{pmatrix}\hat{S}_{\mathrm{X}}^{(2)}\\ \hat{S}_{\mathrm{Y}}^{(2)}\\ \hat{S}_{\mathrm{Z}}^{(2)}\end{pmatrix} \tag{38}$$

The next term in equation (34) is called electric quadrupole interaction. It is quadratic in spin operators – through a twist of mathematics (nuclear quadrupole moment is strictly collinear with spin) an electrical interaction ends up manifesting as an interaction between the spin and itself:

$$\hat{H}_{\mathrm{Q}} = \hat{\vec{L}}\cdot\mathbf{Q}\cdot\hat{\vec{L}} = \begin{pmatrix}\hat{L}_{\mathrm{x}} & \hat{L}_{\mathrm{y}} & \hat{L}_{\mathrm{z}}\end{pmatrix}\begin{pmatrix}q_{\mathrm{xx}} & q_{\mathrm{xy}} & q_{\mathrm{xz}}\\ q_{\mathrm{yx}} & q_{\mathrm{yy}} & q_{\mathrm{yz}}\\ q_{\mathrm{zx}} & q_{\mathrm{zy}} & q_{\mathrm{zz}}\end{pmatrix}\begin{pmatrix}\hat{L}_{\mathrm{x}}\\ \hat{L}_{\mathrm{y}}\\ \hat{L}_{\mathrm{z}}\end{pmatrix} \tag{39}$$

where $\hat{\vec{L}}$ is a vector of Pauli matrices of dimension greater than two (spin-half Pauli matrices produce zeros or unit matrices when they are multiplied) and $\mathbf{Q}$ is the quadrupole coupling tensor. The explicit form is:

$$\hat{H}_{\mathrm{Q}} = -\frac{\chi}{4I(2I-1)}\left[3\hat{L}_{\mathrm{Z}}^2-\hat{L}^2+\frac{\eta}{2}\left(\hat{L}_{\mathrm{X}}^2-\hat{L}_{\mathrm{Y}}^2\right)\right] \tag{40}$$

$$\chi = \frac{e^2q_{\mathrm{ZZ}}Q}{\hbar} \qquad \eta = \frac{q_{\mathrm{XX}}-q_{\mathrm{YY}}}{q_{\mathrm{ZZ}}} \tag{41}$$

where $\{q_{\mathrm{XX}}, q_{\mathrm{YY}}, q_{\mathrm{ZZ}}\}$ are the eigenvalues of the electric field gradient tensor, e is the electron charge, eQ is the quadrupole moment of the nucleus, S is the spin of the nucleus. Note that with spin-½ Pauli matrices the Hamiltonian in equation (40) is proportional to the unit matrix and does not affect system dynamics – quadrupolar coupling only exists for spin-1 nuclei and higher.

The last term deals with external electromagnetic fields that are characterised by an amplitude and frequency. Common NMR and MRI instrument geometries cannot produce oscillating magnetic fields in the direction of the magnet, only transverse terms therefore appear in Equation (34).

A large amount of research exists on the origins, various expressions and transformations of spin interactions. Because this thesis is computational much more than theoretical, it operates at a level where the physical details of the various interactions are unimportant – they will not be reviewed here; for our purposes, these are simply matrices that chemists know more about.

1.2.3 Density operator formalism and equation of motion

Consider now a simple system that has only spin 1/2, for example a proton. Let us assume that individual protons behave independently. Such a group of independent isolated spins is called an ensemble. Every proton has two polarization states. Some of these protons are the down state and some of them are in the up state, and some in a linear combination. In a calculation of the magnetization for all protons in the ensemble, the magnetization for each proton individually must be computed and added up, however this computation is difficult because of the huge number of protons. On the other hand, there is a formalism that calculates ensemble averages without computing individual spin states and this method called density operator formalism [17, 19].

In order to define the density operator for a quantum mechanical system, we begin with the time-dependent Schrödinger equation [20]:

$$\frac{\partial \psi}{\partial t} = -i\hat{H}\psi \;\Leftrightarrow\; \frac{\partial}{\partial t}\left|\psi\right\rangle = -i\hat{H}\left|\psi\right\rangle \tag{42}$$

Using Taylor series [21] to expand $\psi(t)$ at $t=0$, we get

$$\psi(t) = \psi(0) + \left.\frac{\partial \psi(t)}{\partial t}\right|_{t=0} t + \frac{1}{2}\left.\frac{\partial^2 \psi(t)}{\partial t^2}\right|_{t=0} t^2 + \frac{1}{6}\left.\frac{\partial^3 \psi(t)}{\partial t^3}\right|_{t=0} t^3 + \ldots \tag{43}$$

After substituting equation (42) into equation (43) to get all derivatives of the wavefunction, we get

$$\psi(t) = \psi(0) + \left(-i\hat{H}t\right)\psi(0) + \frac{1}{2}\left(-i\hat{H}t\right)^2\psi(0) + \ldots = \left[\sum_{n=0}^{\infty}\frac{\left(-i\hat{H}t\right)^n}{n!}\right]\psi(0) \tag{44}$$

Summation of the series produces a matrix exponential. Therefore, the general solution of the time dependent Schrödinger equation with a static Hamiltonian can be written as follows:

$$\psi(t) = \left[\sum_{n=0}^{\infty}\frac{\left(-i\hat{H}t\right)^n}{n!}\right]\psi(0) = \exp\left(-i\hat{H}t\right)\psi(0) \tag{45}$$

The problem with this description is the arbitrariness of phase in quantum mechanics – the ensemble average of the wavefunction is zero:

$$\overline{\left|\psi\right\rangle} = \frac{1}{2\pi}\int_0^{2\pi} e^{i\varphi}\left|\psi\right\rangle d\varphi = 0 \tag{46}$$

However, if the following operator is defined:

$$\hat{\rho} = |\psi\rangle\langle\psi| \tag{47}$$

its average does not vanish because the phase cancels:

$$\overline{|\psi\rangle\langle\psi|} = \frac{1}{2\pi}\int_0^{2\pi} e^{i\varphi}|\psi\rangle\langle\psi|e^{-i\varphi}d\varphi = |\psi\rangle\langle\psi| \tag{48}$$

The object in Equation (47) is an operator because when it acts on a wavefunction, it returns another wavefunction:

$$\hat{\rho}|\varphi\rangle = |\psi\rangle\langle\psi|\varphi\rangle = a|\psi\rangle, \qquad a = \langle\psi|\varphi\rangle \tag{49}$$

In addition, the density operator is idempotent:

$$\hat{\rho}^2 = |\psi\rangle\langle\psi|\psi\rangle\langle\psi| = |\psi\rangle\langle\psi| = \hat{\rho} \tag{50}$$

The density operator may therefore be used to describe ensembles of spin systems. Hence, it is important to derive its equation of motion:

$$\begin{gathered}\frac{\partial}{\partial t}\hat{\rho}(t) = \frac{\partial}{\partial t}\left(|\psi(t)\rangle\langle\psi(t)|\right) = \\ = \left(\frac{\partial}{\partial t}|\psi(t)\rangle\right)\langle\psi(t)| + |\psi(t)\rangle\left(\frac{\partial}{\partial t}\langle\psi(t)|\right)\end{gathered} \tag{51}$$

After substituting equation (42) into equation (51):

$$\begin{gathered}\frac{\partial}{\partial t}\hat{\rho} = -i\hat{H}|\psi\rangle\langle\psi| + i|\psi\rangle\langle\psi|\hat{H} = \\ = -i\hat{H}\hat{\rho} + i\hat{\rho}\hat{H}\end{gathered} \tag{52}$$

From the definition of the commutator:

$$\left[\hat{A},\hat{B}\right] = \hat{A}\hat{B} - \hat{B}\hat{A} \tag{53}$$

we can conclude that the equation of motion of the density operator is:

$$\frac{\partial}{\partial t}\hat{\rho}(t) = -i\left[\hat{H}(t),\hat{\rho}(t)\right] \tag{54}$$

This is called Liouville - von Neumann equation. It is the foundation of quantum thermodynamics and of the quantum mechanical formalism of magnetic resonance. It is often formulated in terms of the Hamiltonian communication superoperator:

$$\hat{\hat{H}}\hat{\rho} = \left[\hat{H},\hat{\rho}\right] = \hat{H}\hat{\rho} - \hat{\rho}\hat{H} \tag{55}$$

1.2.4 Rotating frame transformation

The Zeeman interaction is large in magnetic resonance, and it is technically difficult to calculate the spin system trajectory using the full Hamiltonian superoperator. For this reason we need a mathematical transformation to get rid of the large Zeeman terms in Liouville space. The transformation called rotating frame or interaction representation transformation. We will define the new operators:

$$\hat{\sigma}(t) = \exp\left(i\hat{\hat{H}}_0 t\right)\hat{\rho}(t) \qquad \hat{\hat{H}}_1^{\mathrm{R}}(t) = \exp\left(i\hat{\hat{H}}_0 t\right)\hat{\hat{H}}_1(t)\exp\left(-i\hat{\hat{H}}_0 t\right) \tag{56}$$

where $\hat{\sigma}$ is the new density matrix, and $\hat{\hat{H}}_0$ is the time-independent part of the Hamiltonian superoperator. After substituting this into the Liouville – von Neumann equation:

$$\frac{d}{dt}\hat{\rho}(t) = -i\left(\hat{\hat{H}}_0 + \hat{\hat{H}}_1\right)\hat{\rho}(t) \tag{57}$$

and going through a few rounds of simplifications, we get;

$$\frac{d}{dt}\hat{\sigma}(t) = -i\hat{\hat{H}}_1^{\mathrm{R}}(t)\hat{\sigma}(t) \tag{58}$$

The result is that the time-independent Hamiltonian $\hat{H}_0$ formally disappears form equation (58) while the time dependent Hamiltonian $\hat{H}_1$ becomes more complicated, but retains its small norm because the transformation in Equation (56) is unitary. It is therefore easier to solve the equation of motion in the interaction representation rather than in the laboratory frame.

1.2.5 Quantum mechanical treatment of spin relaxation

The relaxation of the spin system to the equilibrium is present in all magnetic resonance experiments. Of all relaxation theories, Bloch-Redfield-Wangsness relaxation theory [22] is the one most commonly used in magnetic resonance. It is a second order time dependent perturbation theory.

We shall start with the Liouville-von Neumann equation in Liouville space:

$$\frac{d}{dt}\hat{\rho}(t) = -i\left(\hat{\hat{H}}_0 + \hat{\hat{H}}_1(t)\right)\hat{\rho}(t) \tag{59}$$

where $\hat{\hat{H}}_0$ is the time-independent part of the Hamiltonian commutation superoperator and $\hat{\hat{H}}_1(t)$ is the time-dependent part. We shall assume a big difference between the norm of $\hat{\hat{H}}_1(t)$ and the norm of $\hat{\hat{H}}_0$:

$$\left\|\hat{\hat{H}}_1(t)\right\| \ll \left\|\hat{\hat{H}}_0\right\| \tag{60}$$

and assume that the ensemble average of $\hat{\hat{H}}_1(t)$ is zero. If it is not zero, the mean can be subtracted and put into $\hat{\hat{H}}_0$. We will then move into the interaction representation with respect to $\hat{\hat{H}}_0$:

$$\hat{\hat{H}}_1^{\mathrm{R}} = \exp\left(i\hat{\hat{H}}_0 t\right)\hat{\hat{H}}_1(t)\exp\left(-i\hat{\hat{H}}_0 t\right) \qquad \hat{\sigma}(t) = \exp\left(i\hat{\hat{H}}_0 t\right)\hat{\rho}(t) \tag{61}$$

As indicated in the previous section we can write the new Liouville - von Neumann equation for $\hat{\sigma}(t)$

$$\frac{d}{dt}\hat{\sigma}(t) = -i\hat{\hat{H}}_1^{\mathrm{R}}(t)\hat{\sigma}(t) \tag{62}$$

After integrating equation (62) we will get:

$$\begin{gathered}\int_0^t d\hat{\sigma}(t) = -i\int_0^t \hat{\hat{H}}_1^{\mathrm{R}}(t)\hat{\sigma}(t)\,dt \\ \Downarrow \\ \hat{\sigma}(t) - \hat{\sigma}(0) = -i\int_0^t \hat{\hat{H}}_1^{\mathrm{R}}(t)\hat{\sigma}(t)\,dt\end{gathered} \tag{63}$$

After taking Equation (63) and substituting back into Equation (62) we will get:

$$\frac{d}{dt}\hat{\sigma}(t) = -i\hat{\hat{H}}_1^{\mathrm{R}}(t)\hat{\sigma}(0) - \int_0^t \hat{\hat{H}}_1^{\mathrm{R}}(t)\hat{\hat{H}}_1^{\mathrm{R}}(t_1)\hat{\sigma}(t)\,dt_1 \tag{64}$$

The first term of right hand side in equation (64) will go to zero after we will take the ensemble average, so equation (64) will become:

$$\frac{\partial}{\partial t}\left\langle\hat{\sigma}(t)\right\rangle = -\int_0^t \left\langle \hat{\hat{H}}_1^{\mathrm{R}}(t)\hat{\hat{H}}_1^{\mathrm{R}}(t_1)\hat{\sigma}(t)\right\rangle dt_1 \tag{65}$$

Now we will represent the time dependent Hamiltonian in terms of time dependent coefficients $f_q(t)$ and time-independent operators $\hat{\hat{A}}_q$ as following:

$$\hat{\hat{H}}_1(t) = \sum_q \hat{\hat{A}}_q f_q(t) = \sum_b \hat{\hat{A}}_b^{\dagger} f_b^{*}(t) \tag{66}$$

The time-independent operators $\hat{\hat{A}}_q$ can also be written in the rotating frame as:

$$\hat{\hat{A}}_q^{\mathrm{R}}(t) = \exp\left(i\hat{\hat{H}}_0 t\right)\hat{\hat{A}}_q \exp\left(-i\hat{\hat{H}}_0 t\right) \tag{67}$$

After putting equation (66) and equation (67) in equation (65) we will get:

$$\frac{\partial}{\partial t}\left\langle\hat{\sigma}(t)\right\rangle = -\sum_q\sum_b\int_0^t\left\langle f_q(t)\, f_b^*(t_1)\,\hat{\hat{A}}_q^{\mathrm{R}}(t)\,\hat{\hat{A}}_b^{\mathrm{R}\dagger}(t_1)\,\hat{\sigma}(t_1)\right\rangle dt_1 \tag{68}$$

We assume lack of correlation between the spin system dynamics and the noise in the lattice, Equation (68) will therefore transform into:

$$\frac{\partial}{\partial t}\left\langle\hat{\sigma}(t)\right\rangle = -\sum_q\sum_b\int_0^t\left\langle f_q(t)\, f_b^*(t_1)\right\rangle\hat{\hat{A}}_q^{\mathrm{R}}(t)\,\hat{\hat{A}}_b^{\mathrm{R}\dagger}(t_1)\left\langle\hat{\sigma}(t)\right\rangle\, dt_1 \tag{69}$$

Second, we will define a correlation function which does not depend on the absolute time *t* because the noise in the system is stationary.

$$\left\langle f_q(t)\, f_b^*(t_1)\right\rangle = g_{qb}(t-t_1) \tag{70}$$

After putting Equations (68), (69) and (70) together we will get:

$$\frac{\partial}{\partial t}\left\langle\hat{\sigma}(t)\right\rangle = -\sum_q\sum_b\int_0^t g_{qb}(t-t_1)\,\hat{\hat{A}}_q^{\mathrm{R}}(t)\,\hat{\hat{A}}_b^{\mathrm{R}\dagger}(t_1)\left\langle\hat{\sigma}(t)\right\rangle\, dt_1 \tag{71}$$

The next assumption is to take density matrix out of integral due to the time scale separation between spin dynamics and the autocorrelation function, and to perform a variable substitution $\tau = t - t_1$:

$$\frac{\partial}{\partial t}\left\langle\hat{\sigma}(t)\right\rangle = -\sum_q\sum_b\int_0^t g_{qb}(\tau)\,\hat{\hat{A}}_q^{\mathrm{R}}(t)\,\hat{\hat{A}}_b^{\mathrm{R}\dagger}(t-\tau)\, d\tau\left\langle\hat{\sigma}(t)\right\rangle \tag{72}$$

The upper limit in the integral can now be extended to infinity due to the (assumed) rapid decay of the correlation functions:

$$\frac{\partial}{\partial t}\left\langle\hat{\sigma}(t)\right\rangle = \left[-\sum_q\sum_b\int_0^\infty g_{qb}(\tau)\,\hat{\hat{A}}_q^{\mathrm{R}}(t)\,\hat{\hat{A}}_b^{\mathrm{R}\dagger}(t-\tau)\, d\tau\right]\left\langle\hat{\sigma}(t)\right\rangle \tag{73}$$

Using equations (59) and (61) to return back to the Schrödinger representation we will get:

$$\frac{\partial}{\partial t}\hat{\rho}(t) = -i\hat{\hat{H}}_0(t)\,\hat{\rho}(t) - \sum_q\sum_b\int_0^\infty g_{qb}(\tau)\,\hat{\hat{A}}_q\exp\left(-i\hat{\hat{H}}_0\tau\right)\hat{\hat{A}}_b^{\dagger}\exp\left(i\hat{\hat{H}}_0\tau\right)\, d\tau\hat{\rho}(t) \tag{74}$$

The final equation contains the relaxation superoperator and is called the master equation:

$$\frac{\partial}{\partial t}\hat{\rho}(t) = -i\hat{\hat{H}}_0(t)\,\hat{\rho}(t) + \hat{\hat{R}}\hat{\rho}(t) \tag{75}$$

where:

$$\hat{\hat{R}} = -\sum_{q}\sum_{b}\int_{0}^{\infty} g_{qb}(\tau)\,\hat{\hat{A}}_{q}\exp\left(-i\hat{\hat{H}}_{0}\tau\right)\hat{\hat{A}}_{b}^{\dagger}\exp\left(i\hat{\hat{H}}_{0}\tau\right)\,d\tau \tag{76}$$

Equation (76) is the commonly used Bloch-Redfield-Wangsness relaxation superoperator [23].

1.2.6 A simple analytical example: spin echo

Spin echo pulse sequence contains two successive pulses to produce an echo signal. It was introduced in 1949 by Erwin Hahn. The pulse sequence contains a 90° pulse to flip the spins into the transverse plane. If an inhomogeneity is present in the magnetic field, it would result in some spins precessing faster (and thus gaining phase) relative to others.

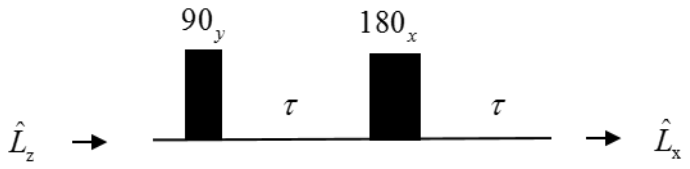


Figure 1. *Schematic diagram of the spin echo pulse sequence.*

Then applying a 180° pulse to flip the spins and puts the faster spins behind, while the slower spins end up in front. As consequence, the faster spins catch up with the slower spins to generate a spin echo signal. The time between the middle of the 90° pulse and the echo signal is called the echo time "TE" in magnetic resonance imaging.

In order to work out how the density operator varies with time we need to know the Hamiltonian (which is also an operator) which is acting during that time. The free precession Hamiltonian (i.e. that for a delay), H_{free} , is

$$\hat{H}_{\text{free}} = \omega_0 \hat{L}_z \tag{77}$$

In the quantum mechanical picture the Hamiltonian involves the z-angular momentum operator, $\hat{L}_z$.

The Hamiltonian for a pulse about the x-axis, $\hat{H}_x$, is

$$\hat{H}_x = \omega_1 \hat{L}_x \tag{78}$$

and for a pulse about the y-axis it is:

$$\hat{H}_y = \omega_1 \hat{L}_x \tag{79}$$

Where ω_0 and ω_1 are the frequencies in Hz.

To make use of the Liouville - von Neumann equation, we will put the initial magnetization on z-axis $\hat{\rho}(0) = \hat{L}_z$ and apply a 90° pulse on y-axis:

$$\hat{H}_y = \omega_1 \hat{L}_y \tag{80}$$

We know that the dynamics of density operator can be written as:

$$\hat{\rho}(t) = \exp\left(-i\hat{H}t\right)\hat{\rho}(0)\exp\left(i\hat{H}t\right) \tag{81}$$

When we substitute Equation (80) in Equation (81), we get:

$$\hat{\rho}(t) = \exp\left(-i\omega_1 t\hat{L}_y\right)\hat{L}_z \exp\left(-i\omega_1 t\hat{L}_y\right) \tag{82}$$

Using an identity from the angular momentum theory:

$$\exp\left(-i\theta\hat{L}_y\right)\hat{L}_z \exp\left(-i\theta\hat{L}_y\right) = \cos(\theta)\hat{L}_z + \sin(\theta)\hat{L}_y \tag{83}$$

Replacing θ by $\omega_1 t_p$:

$$\hat{\rho}(t) = \exp\left(-i\omega_1 t_p\hat{L}_y\right)\hat{L}_z \exp\left(-i\omega_1 t_p\hat{L}_y\right) = \cos\left(\omega_1 t_p\right)\hat{L}_y + \sin\left(\omega_1 t_p\right)\hat{L}_y \tag{84}$$

The first pulse is 90° pulse, so $\omega_1 t_p = \pi/2$, and:

$$\hat{\rho}(t) = \hat{L}_x \tag{85}$$

The magnetization is now in the transverse plane. Equation (85) will be the initial condition of the next stage which is the first delay τ as follows:

$$\hat{H}_{free} = \omega_0 \hat{L}_z \qquad \hat{\rho}(0) = \hat{L}_x \tag{86}$$

Equation (86) with these conditions is:

$$\hat{\rho}(t) = \exp\left(-i\omega_0\tau\hat{L}_z\right)\hat{L}_x \exp\left(i\omega_0\tau\hat{L}_z\right) = \cos\left(\omega_0\tau\right)\hat{L}_x + \sin\left(\omega_0\tau\right)\hat{L}_y \tag{87}$$

The next stage is to apply a 180° pulse on the x-axis. The 180° pulse will not affect the first term, and will flip the sign of the second term:

$$\hat{\rho}(t) = \cos\left(\omega_0\tau\right)\hat{L}_x - \sin\left(\omega_0\tau\right)\hat{L}_y \tag{88}$$

Then, for the second evolution period:

$$\hat{H}_{free} = \omega_0 \hat{L}_z \qquad \hat{\rho}(0) = \cos\left(\omega_0\tau\right)\hat{L}_x - \sin\left(\omega_0\tau\right)\hat{L}_y \tag{89}$$

$$\hat{\rho}(t)=\exp\left(-i\omega_0\tau\hat{L}_z\right)\left(\cos\left(\omega_0\tau\right)\hat{L}_x-\sin\left(\omega_0\tau\right)\hat{L}_y\right)\exp\left(i\omega_0\tau\hat{L}_z\right) \tag{90}$$

$$\begin{aligned}\hat{\rho}(t)=\cos\left(\omega_0\tau\right)\left[\cos\left(\omega_0\tau\right)\hat{L}_x+\sin\left(\omega_0\tau\right)\hat{L}_y\right]-\ldots\\ \sin\left(\omega_0\tau\right)\left[\cos\left(\omega_0\tau\right)\hat{L}_y-\sin\left(\omega_0\tau\right)\hat{L}_x\right]\end{aligned} \tag{91}$$

After simplifying Equation (91) we get:

$$\hat{\rho}(t)=\left[\cos^2\left(\omega_0\tau\right)+\sin^2\left(\omega_0\tau\right)\right]\hat{L}_x \tag{92}$$

We know $\cos^2 x+\sin^2 x=1$, and therefore:

$$\hat{\rho}(t)=\hat{L}_x \tag{93}$$

The physical meaning of Equation (93) is spins rephasing in a way that does not depend on their frequency. The evolution is completely refocused to the original state.

Several spin echo based pulse sequences were implemented into Spinach as a part of the work described in this thesis. **Figure 2** shows the result of a simulation that broadly follows the dynamics described above with the additional complication that the diffusion is present – the treatment of diffusion processes is described in later chapters of this thesis.

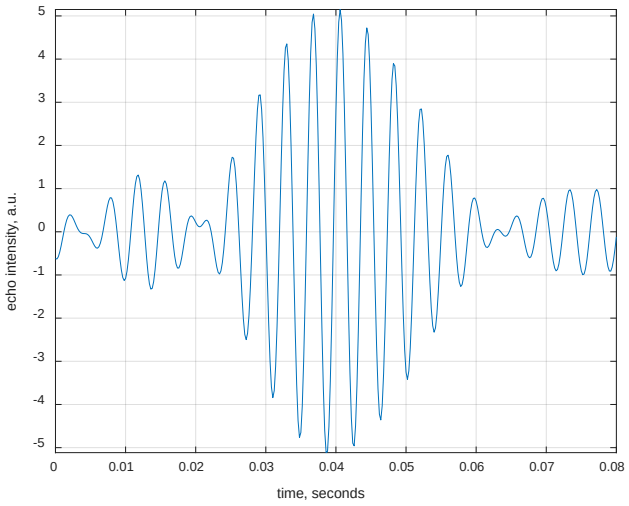


Figure 2. *Spin echo under a magnetic field gradient in the presence of a diffusion process with D = 10^{-6} m^2/s in a one-dimensional sample 30 cm across. The time is counted from the edge of the second pulse. The simulation script that generated this figure is available as a part of the Spinach example set (examples/imaging/spin_echo_1d.m).*

1.3 Singlet state NMR

It is more convenient to introduce a system of two spins -1/2 in term of singlet and triplet states. These states are classified as $|\alpha\rangle$ and $|\beta\rangle$ according to their angular momentum projection along z-axis:

$$\begin{aligned}\hat{L}_z|\alpha\rangle &= \frac{1}{2}|\alpha\rangle \\ \hat{L}_z|\beta\rangle &= -\frac{1}{2}|\beta\rangle\end{aligned} \tag{94}$$

These two spins-1/2 have two possible ways to combine: they may combine for total spin 0, which is called singlet state, or total spin 1, which is called triplet state, defined by [24]:

$$\begin{aligned}|\hat{S}\rangle &= \frac{1}{\sqrt{2}}\left(|\alpha\beta\rangle - \langle\beta\alpha|\right) \\ |\hat{T}_{+1}\rangle &= |\alpha\alpha\rangle \\ |\hat{T}_{0}\rangle &= \frac{1}{\sqrt{2}}\left(|\alpha\beta\rangle + \langle\beta\alpha|\right) \\ |\hat{T}_{-1}\rangle &= |\beta\beta\rangle\end{aligned} \tag{95}$$

Where α represents the projection +1/2 and β represents the projection –1/2 on the Z axis.

A significant interaction inside any molecule that has two spins is the dipole-dipole interaction which is symmetric under exchange between the two nuclei [25]. The singlet state is antisymmetric with respect to spin exchange, whereas the triplet states are symmetric [24]. If the chemical shifts of the two spins are the same, the singlet state is invisible and it is difficult to observe it. On the other hand, if the chemical shifts of the two nuclei are different, the singlet state participates in the dynamics and indirectly becomes visible [24].

In Liouville space, the equation of motion is [26]:

$$\frac{\partial}{\partial t}\hat{\rho}(t) = -i\hat{\hat{L}}(t)\hat{\rho}(t) \qquad \hat{\hat{L}}(t) = \hat{\hat{H}}(t) + i\hat{\hat{R}} \tag{96}$$

Where $\hat{\hat{H}}$ is the Hamiltonian commutation superoperator which contains the coherent interactions between the two spins, while $\hat{\hat{R}}$ is the relaxation superoperator which is responsible for relaxation.

The coherent Hamiltonian contains the chemical shift $\hat{H}_{CS}$ and the spin-spin coupling $\hat{H}_J$:

$$\hat{H}_{\mathrm{CS}} = \omega_{\mathrm{L}}\hat{L}_{\mathrm{z}} + \omega_{\mathrm{S}}\hat{S}_{\mathrm{z}}$$
$$\hat{H}_{\mathrm{J}} = 2\pi J\left(\hat{L}_{\mathrm{x}}\hat{S}_{\mathrm{x}} + \hat{L}_{\mathrm{y}}\hat{S}_{\mathrm{y}} + \hat{L}_{\mathrm{z}}\hat{S}_{\mathrm{z}}\right) \tag{97}$$

Where ω are the Zeeman frequencies of the two spins:

$$\omega = -\gamma\left(1+\delta\right)B_0 \tag{98}$$

and J is the inter-nuclear *J*-coupling in Hz. Equation (97) can be written in terms of two-spin irreducible spherical tensor operators (IST) that are obtained by changing Cartesian coordinates to Cartesian spin operators in spherical harmonics [27]:

Table 1. *Definition of irreducible spherical tensor operators.*

l	m	$Y_{lm}(\theta,\varphi)$	$Y_{lm}(x,y,z)$	$\hat{T}_{lm}(L)$
0	0	$\sqrt{\frac{1}{4\pi}}$	$\sqrt{\frac{1}{4\pi}}$	$\hat{E}$
1	1	$-\sqrt{\frac{3}{8\pi}}e^{i\varphi}\sin\theta$	$-\sqrt{\frac{3}{8\pi}}\frac{x+iy}{r}$	$-\sqrt{\frac{1}{2}}\hat{L}_{+}$
1	0	$\sqrt{\frac{3}{4\pi}}\cos\theta$	$\sqrt{\frac{3}{4\pi}}\frac{z}{r}$	$\hat{L}_{\mathrm{Z}}$
1	–1	$\sqrt{\frac{3}{8\pi}}e^{-i\varphi}\sin\theta$	$\sqrt{\frac{3}{8\pi}}\frac{x-iy}{r}$	$\sqrt{\frac{1}{2}}\hat{L}_{-}$
2	2	$\sqrt{\frac{5}{32\pi}}e^{2i\varphi}\sin^2\theta$	$\sqrt{\frac{15}{32\pi}}\frac{(x+iy)^2}{r^2}$	$\frac{1}{2}\hat{L}_{+}^2$
2	1	$-\sqrt{\frac{5}{8\pi}}e^{i\varphi}\sin\theta\cos\theta$	$-\sqrt{\frac{5}{8\pi}}\frac{(x+iy)z}{r^2}$	$-\frac{1}{2}\left(\hat{L}_{\mathrm{Z}}\hat{L}_{+}+\hat{L}_{+}\hat{L}_{\mathrm{Z}}\right)$
2	0	$\sqrt{\frac{5}{16\pi}}\left(3\cos^2\theta-1\right)$	$\sqrt{\frac{5}{16\pi}}\frac{2z^2-x^2-y^2}{r^2}$	$\sqrt{\frac{2}{3}}\left(\hat{L}_{\mathrm{Z}}^2-\frac{1}{4}\left(\hat{L}_{+}\hat{L}_{-}+\hat{L}_{-}\hat{L}_{+}\right)\right)$
2	–1	$\sqrt{\frac{5}{8\pi}}e^{-i\varphi}\sin\theta\cos\theta$	$\sqrt{\frac{15}{8\pi}}\frac{(x-iy)z}{r^2}$	$\frac{1}{2}\left(\hat{L}_{\mathrm{Z}}\hat{L}_{-}+\hat{L}_{-}\hat{L}_{\mathrm{Z}}\right)$
2	–2	$\sqrt{\frac{5}{32\pi}}e^{-2i\varphi}\sin^2\theta$	$\sqrt{\frac{15}{32\pi}}\frac{(x-iy)^2}{r^2}$	$\frac{1}{2}\hat{L}_{-}^2$

The following table gives the coefficients required for translation between matrix representations and irreducible spherical tensor coefficients for a two-spin system.

Table 2. *Relationship between matrix representation and irreducible spherical tensor coefficients.*

(l,m)	$a_m^{(l)}$	$\hat{T}_m^{(l)}$
$(0,0)$	$-\frac{1}{\sqrt{3}}\left(a_{XX}+a_{YY}+a_{ZZ}\right)$	$-\frac{1}{\sqrt{3}}\hat{\vec{L}}\cdot\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}\cdot\hat{\vec{S}}$
$(1,1)$	$-\frac{1}{2}\left(a_{ZX}-a_{XZ}-i\left(a_{ZY}-a_{YZ}\right)\right)$	$-\frac{1}{2}\hat{\vec{L}}\cdot\begin{pmatrix}0&0&-1\\0&0&-i\\1&i&0\end{pmatrix}\cdot\hat{\vec{S}}$
$(1,0)$	$-\frac{i}{\sqrt{2}}\left(a_{XY}-a_{YX}\right)$	$-\frac{1}{\sqrt{2}}\hat{\vec{L}}\cdot\begin{pmatrix}0&-i&0\\i&0&0\\0&0&0\end{pmatrix}\cdot\hat{\vec{S}}$
$(1,-1)$	$-\frac{1}{2}\left(a_{ZX}-a_{XZ}+i\left(a_{ZY}-a_{YZ}\right)\right)$	$-\frac{1}{2}\hat{\vec{L}}\cdot\begin{pmatrix}0&0&-1\\0&0&i\\1&-i&0\end{pmatrix}\cdot\hat{\vec{S}}$
$(2,2)$	$+\frac{1}{2}\left(a_{XX}-a_{YY}-i\left(a_{XY}+a_{YX}\right)\right)$	$+\frac{1}{2}\hat{\vec{L}}\cdot\begin{pmatrix}1&i&0\\i&-1&0\\0&0&0\end{pmatrix}\cdot\hat{\vec{S}}$
$(2,1)$	$-\frac{1}{2}\left(a_{XZ}+a_{ZX}-i\left(a_{YZ}+a_{ZY}\right)\right)$	$-\frac{1}{2}\hat{\vec{L}}\cdot\begin{pmatrix}0&0&1\\0&0&i\\1&i&0\end{pmatrix}\cdot\hat{\vec{S}}$
$(2,0)$	$+\frac{1}{\sqrt{6}}\left(2a_{ZZ}-\left(a_{XX}+a_{YY}\right)\right)$	$+\frac{1}{\sqrt{6}}\hat{\vec{L}}\cdot\begin{pmatrix}-1&0&0\\0&-1&0\\0&0&2\end{pmatrix}\cdot\hat{\vec{S}}$
$(2,-1)$	$+\frac{1}{2}\left(a_{XZ}+a_{ZX}+i\left(a_{YZ}+a_{ZY}\right)\right)$	$+\frac{1}{2}\hat{\vec{L}}\cdot\begin{pmatrix}0&0&1\\0&0&-i\\1&-i&0\end{pmatrix}\cdot\hat{\vec{S}}$
$(2,-2)$	$+\frac{1}{2}\left(a_{XX}-a_{YY}+i\left(a_{XY}+a_{YX}\right)\right)$	$+\frac{1}{2}\hat{\vec{L}}\cdot\begin{pmatrix}1&-i&0\\-i&-1&0\\0&0&0\end{pmatrix}\cdot\hat{\vec{S}}$

The following commutation properties may be proven by direct inspection:

$$\begin{aligned}
&\left[\hat{H}_{\mathrm{J}}, \hat{T}_0^{(0)}\right] = 0 \\
&\left[\hat{H}_{\mathrm{cs}}^{\Sigma}, \hat{T}_0^{(1)} + \hat{T}_0^{(1)}\right] = 0 \\
&\left[\hat{H}_{\mathrm{cs}}^{\Delta}, \hat{T}_0^{(1)} - \hat{T}_0^{(1)}\right] \neq 0
\end{aligned} \tag{99}$$

where $\hat{H}_{\mathrm{cs}}^{\Sigma}$ is the average chemical shift Hamiltonian of the two spins, $\hat{H}_{\mathrm{cs}}^{\Delta}$ is the difference in the chemical shift Hamiltonians, and indices 1 & 2 denote the first and the second spin respectively. That means there is no effect on the singlet state from *J*-coupling because they commute with each other and the same applies to the sum of the chemical shifts. However the difference in the chemical shifts has an effect on the singlet state because the corresponding operators don't commute with each other. Therefore, the singlet state only evolves under the difference in chemical shift in the coherent Hamiltonian.

The incoherent side of the dynamics is described by the relaxation superoperator:

$$\hat{\hat{R}} = \hat{\hat{R}}_{\mathrm{A}} + \hat{\hat{R}}_{\mathrm{B}} + \hat{\hat{R}}_{\mathrm{C}} + \ldots\ldots\ldots + \hat{\hat{R}}_{\mathrm{A\times B}} + \hat{\hat{R}}_{\mathrm{A\times C}} \tag{100}$$

The total relaxation is the sum of the relaxation superoperator of different interactions $\left(\mathrm{A, B, C, ...}\right)$ and their cross-correlations $\left(\mathrm{A\times B}\right)$ [25]. In nuclear magnetic resonance this happens when a rigid molecule has two or more different anisotropic spin interactions.

We can define the population operator as:

$$\hat{P}_{\Psi} = \left|\Psi\right\rangle\left\langle\Psi\right| \tag{101}$$

For the singlet state the population operator is:

$$\hat{P}_{\mathrm{S}} = \frac{1}{\sqrt{2}}\left(\left|\alpha\beta\right\rangle - \left|\beta\alpha\right\rangle\right)\cdot\frac{1}{\sqrt{2}}\left(\left\langle\alpha\beta\right| - \left\langle\beta\alpha\right|\right) \tag{102}$$

Equation (102) can be expressed *via* a spherical tensor:

$$\hat{P}_{\mathrm{S}} = \frac{\hat{E}}{4} + \sqrt{3}\hat{T}_0^{(0)} \tag{103}$$

Where $\hat{E}$ is a unit matrix, and $\hat{T}_0^{(0)}$ is the irreducible spherical tensor operator of rank zero and projection zero defined in Table 1. ISTs have the following rotation property [28]:

$$\hat{\hat{R}}\left(\alpha,\beta,\gamma\right)\hat{T}_m^{(l)} = \sum_{m'=-l}^{+l}\hat{T}_{m'}^{(l)}D_{m'm}^{(l)}\left(\alpha,\beta,\gamma\right) \tag{104}$$

Where $\hat{\hat{R}}(\alpha,\beta,\gamma)$ is the rotation superoperator, $\hat{T}_{lm}$ is the irreducible spherical tensor operator, $D^{(l)}_{m'm}(\alpha,\beta,\gamma)$ is a Wigner D-matrix element:

$$D^{l}_{m'm}(\Omega) = \left\langle Y_{l,m} \middle| \hat{R}(\Omega) \middle| Y_{l,m'} \right\rangle \tag{105}$$

If we are solving Equation (96) we can use:

$$\hat{\rho}(t+dt) = \hat{\hat{U}}(t,dt)\,\hat{\rho}(t) \tag{106}$$

$$\hat{\hat{U}}(t,dt) = \exp\left(-i\hat{\hat{L}}(t)\,dt\right) = \sum_{m=0}^{\infty} \frac{(-i)^m \hat{\hat{L}}^m(t)\,dt^m}{m!} \tag{107}$$

By substituting Equation (96) in Equation (107) we can find:

$$\hat{\hat{U}}(t,dt) = \exp\left(-i\hat{\hat{L}}(t)\,dt\right) = \sum_{m=0}^{\infty} \frac{(-i)^m dt^m}{m!}\left(\hat{\hat{H}}(t) + i\hat{\hat{R}}\right)^m \tag{108}$$

Redfield relaxation superoperator is fully derived above, and here can be written as:

$$\hat{\hat{R}} = -\int_0^{\infty} \left\langle \hat{\hat{H}}^{R}_{\text{inc}}(t)\,\hat{\hat{H}}^{R}_{\text{inc}}(t+\tau) \right\rangle d\tau \tag{109}$$

where $\hat{H}^{R}_{\text{inc}}(t)$ is the part of the Hamiltonian that contains incoherent noise in the rotating frame. If the noise has a very short autocorrelation time, this expression can be replaced by:

$$\hat{\hat{R}}\hat{\rho} = \left\langle \left[\hat{H}^{R}_{\text{inc}} \left[\hat{H}^{R}_{\text{inc}}, \hat{\rho} \right] \right] \right\rangle \tag{110}$$

For example, for dipole-dipole interaction [29], we can write:

$$\hat{H}_{\text{DD}} = -\frac{\mu_0}{4\pi}\frac{\gamma_1\gamma_2\hbar}{r^5}\left(3(\hat{\vec{S}}_1\cdot\vec{r})(\vec{r}\cdot\hat{\vec{S}}_2) - r^2(\hat{\vec{S}}_1\cdot\hat{\vec{S}}_2)\right) \tag{111}$$

Where μ_0 is the magnetic dipole moment, γ_1 and γ_2 are the gyromagnetic ratio of two spins, r the distance between two spins, $\hat{\vec{S}}_1$ and $\hat{\vec{S}}_2$ are the spin operator vectors of the spin one and spin two respectively. This form is not very convenient for the rotation properties, and therefore the irreducible spherical tensor form is commonly used instead:

$$\hat{H}_{\text{DD}} = \sum_{m=-2}^{2} (-1)^m A^{(2)}_m \hat{T}^{(2)}_{-m} \tag{112}$$

Where A_{2m} are spatial coefficients and $\hat{T}_{2,-m}$ are the irreducible spherical tensor operators. All spin interactions are second-rank, so the following general expression in terms of irreducible spherical tensor operators covers all cases is [30]:

$$\hat{\vec{L}} \cdot \mathbf{A} \cdot \hat{\vec{S}} = \sum_{k,n=\{\mathrm{X,Y,Z}\}} a_{kn} \hat{L}_k \hat{S}_n = a_0^{(0)} \hat{T}_0^{(0)} + \sum_{m=-1}^{1} a_m^{(1)} \hat{T}_m^{(1)} + \sum_{m=-2}^{2} a_m^{(2)} \hat{T}_m^{(2)} \tag{113}$$

$$\begin{aligned} \hat{T}_{\pm 2}^{(2)} &= +\frac{1}{2} \hat{L}_{\pm} \hat{S}_{\pm} \\ \hat{T}_{\pm 1}^{(2)} &= \mp \frac{1}{2} \left(\hat{L}_{\mathrm{Z}} \hat{S}_{\pm} + \hat{L}_{\pm} \hat{S}_{\mathrm{Z}} \right) \\ \hat{T}_0^{(2)} &= +\sqrt{\frac{2}{3}} \left(\hat{L}_{\mathrm{Z}} \hat{S}_{\mathrm{Z}} - \frac{1}{4} \left(\hat{L}_{+} \hat{S}_{-} + \hat{L}_{-} \hat{S}_{+} \right) \right) \end{aligned} \tag{114}$$

The spatial tensor components for homonuclear dipole-dipole interaction are

$$A_{20} = -\sqrt{6} \frac{\mu_0}{4\pi} \frac{\gamma_1 \gamma_2 \hbar}{r^3} \tag{115}$$

Therefore, for dipolar relaxation:

$$\hat{\hat{R}}_{\mathrm{DD}} = -\frac{6}{5} \left(\frac{\mu_0}{4\pi} \frac{\gamma_1 \gamma_2 \hbar}{r^3} \right)^2 \tau_c \sum_m (-1)^m a_m^{(2)} T_{-m}^{(2)} \tag{116}$$

Where γ_1 and γ_2 are magnetogyric ratios, τ_c is the rotational correlation time in the fast motion limit. If we look at the commutators and compare them with Equation (116), we can see the following:

$$\left[\hat{T}_m^{(2)}, \hat{T}_0^{(0)} \right] = 0 \tag{117}$$

That means the singlet state cannot be changed by relaxation superoperator which appears from the dipolar interaction between two nuclei. Therefore, the dipole-dipole interaction which is the biggest cause of relaxation does not affect the singlet state which is therefore long-lived [31].

1.4 Ultrafast NMR

Large range of disciplines such as chemistry, physics, biology and medicine using NMR as a tool for their applications. The reason is NMR considered as a highly multipurpose characterization technique. In biochemistry, for instance, NMR provides an excellent discerption for large molecule geometry by its self or even in complexes with drugs. In 1971, Jean Jeener [32] proposed a 2D NMR spectroscopy

for high resolution NMR in liquid. This technique opens the wide range of applications in NMR spectroscopy. Nevertheless, 2D NMR suffers from numerous time increments required to sample the result. Many strategies are applied to reduce the duration of 2D NMR but they still required 10-100 sec acquisition times to deliver their spectral correlations. Later, 2002 in specific, powerful analytical tool called “Ultrafast NMR” provide a rapid acquisition method to recorded any kind of 2D NMR spectrum in a single scan [33].

Ultrafast NMR is a multidimensional fast single scan technique that parallelized the acquisition process. This mean that ultrafast NMR can be used to obtain 2D,3D, and ND NMR spectra in under a second and within single scan. It does by encoding spin interactions using spatial gradient and then using the gradient again during the acquisition to achieve the encoded information [34].

2D NMR experiments involve the acquisition of multiple data scans regardless of sensitivity considerations. The large number of scan implies that 2D experiments may take many minutes, partially because of number of scan and partially because of delay between scans. While in ultrafast NMR the sample is actually partition, using gradients, into different slices. Each slice evolves with different time after been excited, then the spatial encoded interactions are collected during the acquisition. At the heart of the method lies in extensive use of spatiotemporal encoding (SPEN) which is the key element used to address each slice individually [34].

1.5 Pure shift NMR

Pure shift method is a technique used to improve the resolution of NMR spectra [35]. the term pure shift NMR is also called chemical shift NMR which is designed to yield a pure chemical shift spectrum. Proton spectra are very important to understand many chemical information. However, due to their narrow chemical shift range (10-15 ppm), protons signals overlapped in spectrum which is leads to difficulties for analysis and interpretation. Pure shift NMR has proven to be a tool for the analysis of overpopulated spectra. Pure shift methods rely on the application of radiofrequency pulse sequences which is supress *J*-evolution effect for a short period of time. The idea behind these pulse sequences is to divide spins into active (observed) and passive (manipulated) spins to remove the effect of *J*-coupling. Different methods are implemented to achieve the desired effects. Starting in 1982, Garbow et al. used a bilinear rotation decoupling (BIRD) [36] element to separate between the observed spins that are attached to ^{13}C and the manipulated spins which are attached to ^{12}C to produce a pure shift spectrum that contains only protons bound to ^{13}C. In 1997, Zangger and Sterk [37] proposed a method that relies on application of frequency-selective 180° pulse, the selective pulse chosen should not excite protons that share couplings to avoid the evolution of the mutual coupling, simultaneously with

a weak pulsed field gradient (PFG). The application of PFG produce a signal have different frequencies depending on its positions. As a result, a frequency selective 180° pulse refocuses different signals in different parts of the sample. All signals are refocused simultaneously, but each one arises from a particular narrow region of the sample, a slice [38].

1.6 Magnetic resonance Imaging

MRI is a medical imaging technique. It is particularly useful for neurological conditions, for disorders of muscles and joints, for evaluating tumours, and for showing abnormalities in the heart and blood vessels [12, 39]. The principle of MRI involves the interaction of nuclei in biological tissue with electromagnetic fields. The most abundant atom in biology is hydrogen which contains a proton. The magnetic moments of these protons quantise along the direction of the magnetic field. While each individual proton can only have specific states, an ensemble of protons can appear to behave as though they can have any alignment, and act as macroscopic magnetisation [17].

The picture is often simplified in the clinical literature relative to the rigorous treatment presented below. It is said that a radio frequency pulse is applied, which moves the magnetisation into the transverse plane with respect to the magnet field. The precession of the bulk magnetic moment induces electric current in the coils, which is amplified and detected [40]. The frequency at which a proton or group of protons in a voxel resonates depends on the strength of the local magnetic field – a stronger field corresponds to a larger energy difference and therefore higher frequency of precession [12].

By applying additional coordinate-dependent magnetic fields (gradients) that vary linearly over space, specific slices can be selected, and an image obtained by taking the 2-D Fourier transform of the spatial frequencies of the signal [41]. Different tissues, including malignant ones, can be distinguished because the protons in different tissues return to their equilibrium state at different rates (i.e., they have different relaxation times). By changing the parameters of the experiment this effect may be used to create contrast between different types of tissue [39].

Contrast agents that accelerate spin relaxation may be injected intravenously to enhance the appearance of blood vessels, tumours or inflammation [42]. Contrast agents may also be directly injected into a joint in the case of arthrograms, MRI images of joints. Unlike CT, MRI uses no ionizing radiation and is generally a safe procedure [43].

1.6.1 Magnetic field gradients

A gradient can be defined as a change in the magnetic field along some direction in space. For example, if the gradient is applied in x direction (called Gx), the Z magnetic field will increase and decrease in x

direction depending on the strength of the gradient as shown in **Figure 3**. If a uniform magnetic field is applied, all protons will align in different orientation and start to precess at the same frequency. However, when a gradient is added, the magnetic field of the gradient will be added or subtracted from the main magnetic field. As a result, the spins will precess at different frequencies depending on their positions. This would allow us to distinguish the spin position in space.

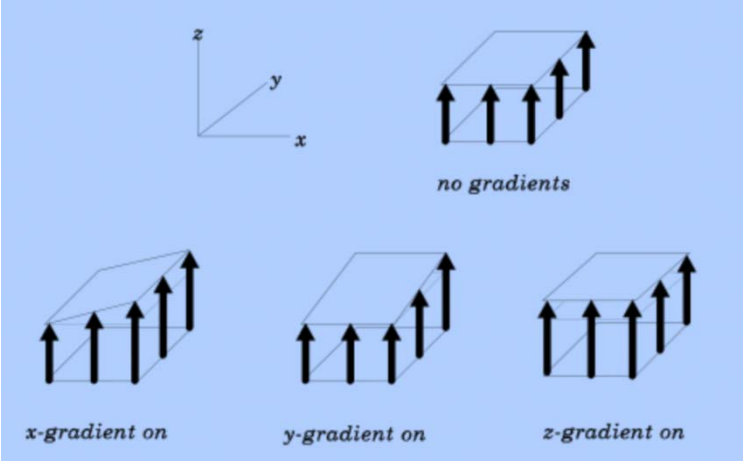


Figure 3. *A schematic of magnetic field gradients in x, y and z directions, reproduced from [27].*

Gradients play several roles in MRI. They may be used to either rephase or dephase the magnetic moment vectors of nuclei. They can also be used to perform slice selection by creating a narrowly localised resonance condition where only certain points of the sample are in resonance with the applied radiofrequency magnetic field.

1.6.2 Slice selection

The signal that comes from an MRI coil contains the information on the magnetisation of the whole body. Without special measures it would be difficult to select the signal from a specific tissue. The patient is positioned in a uniform external field B_0 and that would lead to a uniform excitation of all protons in the body when we apply RF pulse. However, if we make B_0 vary from one point to other, that would mean that each point has a different resonance frequency, and we can therefore apply an RF pulse that would excite a specific point and not excite the others.

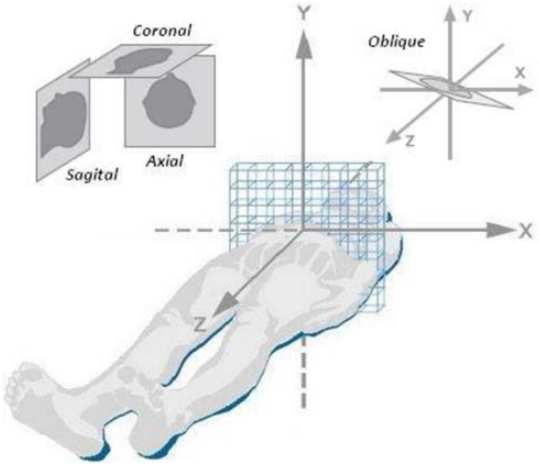


***Figure 4**. A 3-dimensional axial slice through the sample. By varying gradient vector direction, slices can be excited in any orientation. Reproduced from [28].*

A non-uniformity in the magnetic field may be achieved using X, Y, and Z gradient coils, as shown in **Figure 4**. Slice thickness can be controlled by varying the bandwidth of the excitation pulse and the strength of the gradient. The slope of the slice select gradient determines the difference in precession frequency between two points on the gradient. Steep gradient slopes result in a large difference in precession frequency between two points on the gradient, while shallow gradient slopes result in a small difference in precession frequency between the same two points.

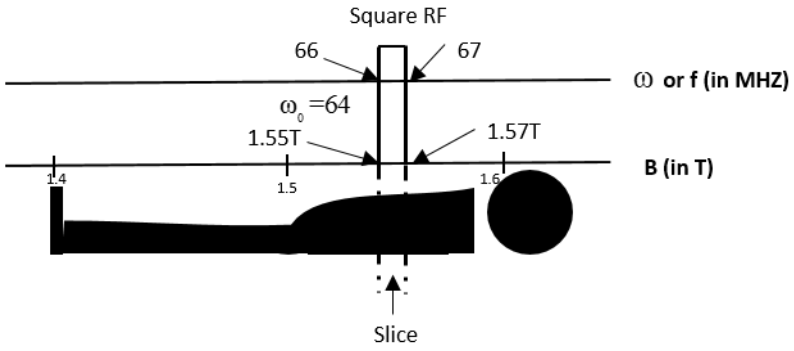


***Figure 5**. An illustration of slice thickness control in MRI. Reproduced from [29].*

Once a certain gradient slope is applied, the RF pulse transmitted to excite the slice must contain a range of frequencies to match the difference in precession frequency between two points. This frequency range is called the bandwidth, and as the RF is being transmitted at this point it is specifically called the transmit bandwidth as shown in **Figure 5**.

1.6.3 K-space

K-space can be defined as the Fourier transform of the real space [44]. The data to fill *k*-space comes directly from the MRI signal. The latter comes from applying the gradients such as slice selection, phase encoding and frequency encoding as shown in **Figure 6**.

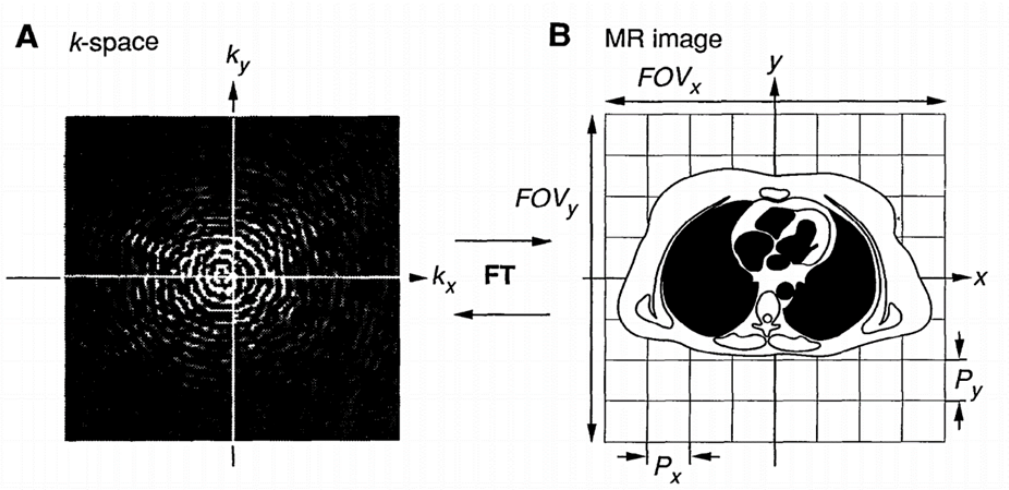


Figure 6. *K-space data (left) and resulting MR image after 2D-Fourier transformation (right). Reproduced from [31].*

The easiest method to implement for most MR imaging sequences is the Cartesian filling of k-space, which means filling line by line in the corresponding *k*-space. This method considered slow, and other methods have been implemented such as spiral and echo planar techniques shown in **Figure 7**.

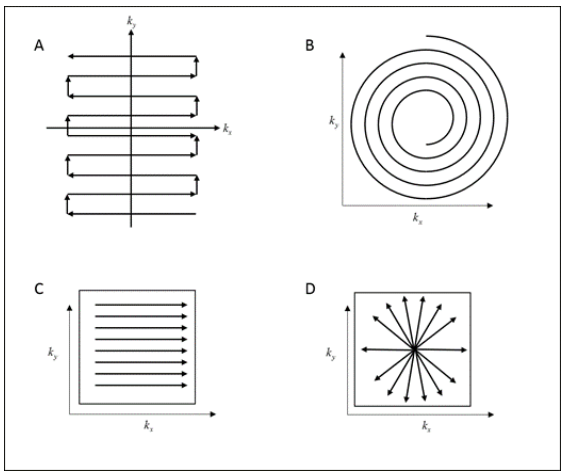


Figure 7. *K-space trajectory examples.(A) Echo planar trajectory, (B) Spiral trajectory, (C) conventional Cartesian trajectory, (D) Radial trajectory*

The echo planar trajectory is the most common technique these days due to the possibility of obtaining individual MR slices in 50-100 ms which is useful to reduce patient motion effects [45]. The square echo planar technique [46] uses a train of echoes after a single RF excitation pulse with a positive and

negative gradient amplitudes to move from left to right and right to left respectively through the k_x direction, while a short phase encoding gradient pulse gives a stepwise increase along the k_y direction as shown in **Figure 8**.

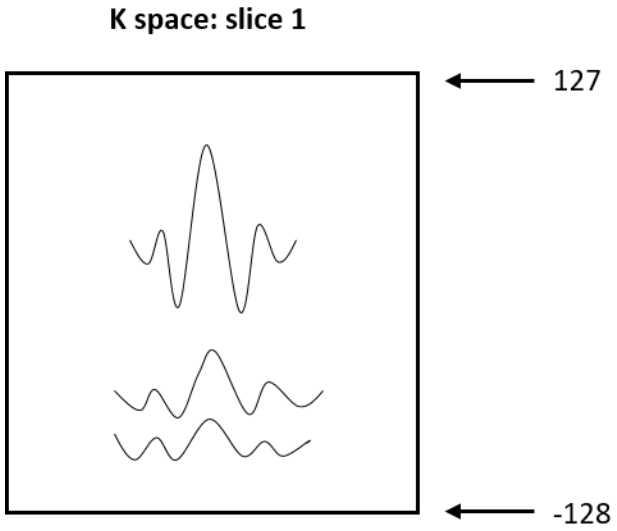


Figure 8. *A schematic illustration of the set of readout echoes recorded for different values of the phase encoding gradient. The resulting data array is a two-dimensional diffractogram that is related to the image by the Fourier transform.*

1.6.4 Frequency encoding

Frequency encoding refers to the fact that, once a gradient is applied, the frequency of the signal is directly linked to the location of the corresponding group of spins. This is illustrated schematically in **Figure 9**. Simply recording the free induction decay and running a Fourier transform is then sufficient to obtain a one-dimensional image. Frequency encoding is therefore the detection step in MRI.

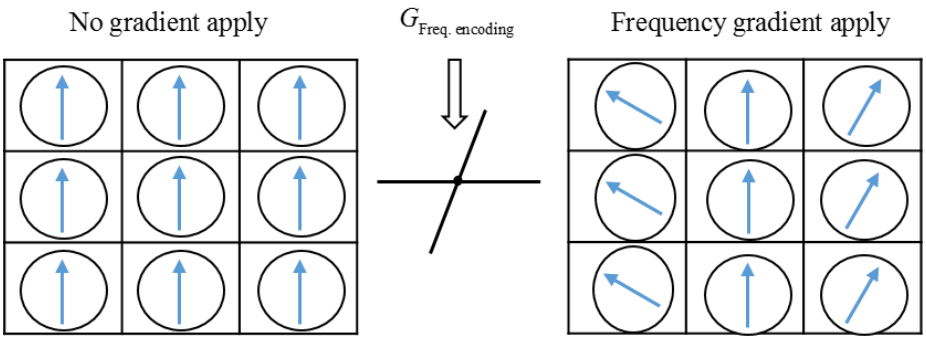


Figure 9. *An illustration of the frequency encoding step in MRI.*

1.6.5 Phase encoding

MRI samples are usually three-dimensional. After slice selection and frequency encoding, this leaves one more dimension to be taken care of. This step is called phase encoding, and it simply amounts to letting the system precess freely under a stepped gradient in the remaining direction. As the gradient

becomes stronger and stronger, greater amount of phase is accumulated, thus effectively collecting a Fourier image in the corresponding dimension (**Figure 10**).

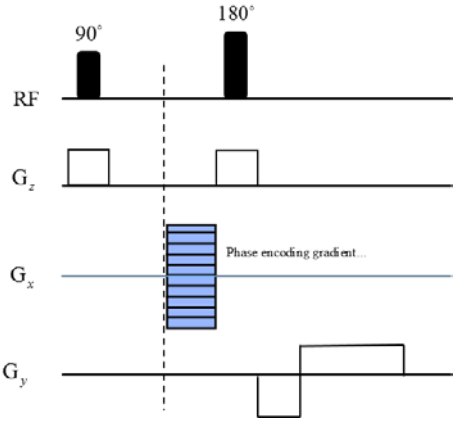


Figure 10. *A schematic illustration of slice selection, phase encoding, and frequency encoding stages of an MRI pulse sequence.*

The result is the simplest possible MRI pulse sequence: a slice selection gradient and pulse are applied to excite a slice in Z, a stepped gradient is used to perform phase encoding in X, and finally a gradient echo detection is performed in Y.

1.6.6 MRI parameters and contrast types

Different types of images may be collected by varying the basic timing parameters of the MRI sequence [47]. *Repetition Time (TR)* is the amount of time between successive pulse sequence instances applied to the same slice. *Time to Echo (TE)* is the time between the delivery of the RF pulse and the receipt of the echo signal. These parameters have an influence on the resulting image due to the presence of spin relaxation. A parameter that only depends on spin concentration is *proton density* (PD).

The amount of time it takes for protons to return to being aligned with B_0 is determined by the *spin-lattice relaxation* time (T_1) of the tissue. The characteristic time of transverse plane dephasing is called *spin-spin relaxation* time (T_2). The most common MRI sequences are T_1-weighted and T_2-weighted [43]

.

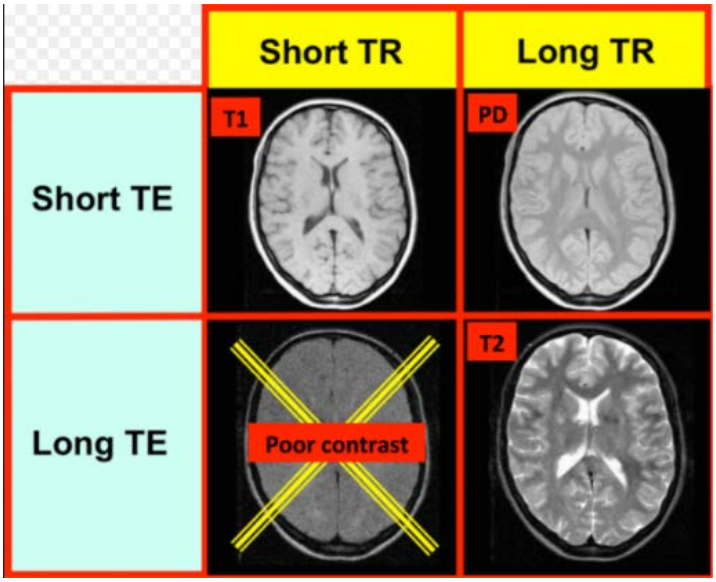


Figure 11. *A schematic [48] showing time parameters for best image contrast where short TE and TR gives T1 contrast while short TE and long TR gives PD contrast. However, long TE and sort TR gives poor contrast while long TE and TR gives T2 contrast.*

T_1-weighted images are produced by using short TE and TR times. The contrast and brightness of the image are predominantly determined by T_1 of tissue. The aim of short timing is to enhance the consequences of T_1 differences between tissues. Fluids which have long T_1 will be dark because the signal from fluids did not have the enough time to relax. Water based tissues have shorter relaxation time than fluids and will appear grey. Fat based tissues have a short T_1, and will therefore appear bright.

T_2-weighted images are produced by using longer TE and TR times. In these images, the contrast and brightness are predominantly determined by the T_2 of tissue. On these images, the T_2 time of fat is shorter than of fluids, so the transverse magnetisation of fat decays faster. The T_2 in fluids is large. Fluids therefore have a high signal and appear bright on T_2 contrast images. However, the T_2 in fat is short, and fat therefore has low signal and appear dark on T_2 contrast images.

PD weighted images are produced by using short TE and long TR times. In PD images, higher signal intensities come from tissues that have higher proton density [43]. To get a PD weighted image we want to minimise the contribution of both T_1 and T_2 contrast. T_1 contrast is minimised with a long TR that leads to complete longitudinal relaxation. T_2 contrast minimised by using a short echo time. Tissues with a high proton density, such as brain tissue, are bright on a proton density contrast image. Tissues with a low proton density, *e.g.* cortical bone, are dark on proton density contrast image. All three cases above are shown in **Figure 11** and **Figure 12**

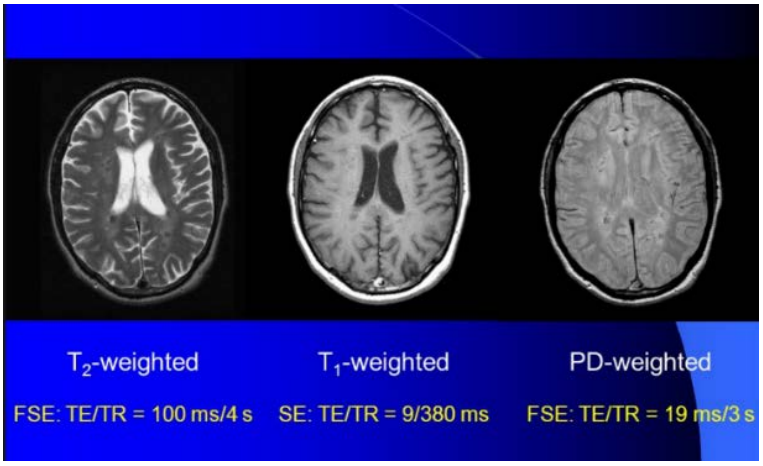


Figure 12. *Difference between T1, T2 and PD weighted images in the brain with time required for each image. Reproduced from [48].*

1.7 Current MRI simulation software

MR imaging involves the nuclear spin system of an object to be imaged, the magnetic fields generated by the imager, the resonance and relaxation phenomena arising from the interaction nuclei / magnetic fields, and the signal detection and reconstruction. Sophisticated packages already exist for MRI simulations. In 1984, Bittoun *et al.* described their simulation program which uses Bloch equation for one point and one frequency for the object which is called "isochromat" (defined as ensemble of spins with same frequency) [49]. Summers *et al.* [50] and Olsson *et al.* [51] applied the same equation for 2D and 3D grid. Afterwards, Shkarin, Pavel, and Spencer [52] proposed a method to increase the accuracy of simulation be increasing the number of the isochromats per voxel. The most important limitation in this method is that they do not include the effect of molecular diffusion on the MRI signal in a general way. Jochimsen *et al.* solved this problem by using the partial derivatives (PDs) of the magnetization vector, with respect to position and frequency but still used Bloch equations [53].

In summary, the subject of single-spin MRI simulation using Bloch equations and Bloch-Torrey equation is very well researched. Even the computer science part of the matter is highly optimised – some of the existing codes already use GPU coprocessors to accelerate calculations [54, 55]. However, the limitation in all the packages is they are insufficient for systems with interacting spins.

2 Fokker-Planck formalism

Nuclear magnetic resonance theory is traditionally based on the Liouville-von Neumann equation. This equation is of central importance in the description of the dynamics of quantum systems and is the dominant formalism in magnetic resonance simulations packages [56-58]. As demonstrated in the previous chapter, from the time-dependent Schrödinger equation we can easily derive the Liouville-von Neumann equation:

$$\frac{\partial}{\partial t}\boldsymbol{\rho}(t) = -i\mathbf{H}(t)\boldsymbol{\rho}(t) \tag{118}$$

Where $\boldsymbol{\rho}(t)$ is the density operator and $\mathbf{H}(t)$ is the spin Hamiltonian commutation superoperator [59]. The latter can be defined as:

$$\mathbf{H}\boldsymbol{\rho} = \left[\mathbf{H}, \boldsymbol{\rho}\right] = \mathbf{H}\boldsymbol{\rho} - \boldsymbol{\rho}\mathbf{H} \tag{119}$$

In general, when relaxation and kinetics are present, equation (118) can be written as:

$$\frac{\partial}{\partial t}\boldsymbol{\rho}(t) = \left[-i\bar{\mathbf{H}}(t) + \mathbf{R} + \mathbf{K}\right]\boldsymbol{\rho}(t) \tag{120}$$

Where $\bar{\mathbf{H}}(t)$ is the ensemble average Hamiltonian, $\mathbf{R}$ is the relaxation superoperator [23] which contains the effects of stochastic terms (for example, rotational diffusion) and $\mathbf{K}$ is the chemical kinetics superoperator which is responsible in the chemical processes in the system. If the thermodynamic equilibrium is non-trivial, we must add the relaxation target:

$$\mathbf{R}\boldsymbol{\rho}(t) \quad \longrightarrow \quad \mathbf{R}\left[\boldsymbol{\rho}(t) - \boldsymbol{\rho}_{\text{eq}}\right] \tag{121}$$

where $\boldsymbol{\rho}_{\text{eq}}$ thermal equilibrium density matrix.

The problem with the current methods of magnetic resonance simulation is that spatial degrees of freedom are only introduced indirectly. For condensed phase spin systems it is commonly assumed that the spin state has no influence on spatial dynamics, but that spatial dynamics has an effect on the spin Hamiltonian. In other words, it is assumed that spatial coordinates are external parameters that make the spin Hamiltonian time-dependent.

2.1.1 Merging spin and spatial degrees of freedom

The Fokker-Planck equation is a partial differential equation describing the time evolution of the probability density of finding system in the state $\mathbf{x}$ at time t :

$$\frac{\partial p(\mathbf{x},t)}{\partial t} = -\nabla_{\mathbf{x}} \cdot \mathbf{j}(\mathbf{x},t) \tag{122}$$

Where $p(\mathbf{x},t)$ is the probability density, $\mathbf{j}(\mathbf{x},t)$ is the probability flux which is determined by the local velocity $\mathbf{v}(\mathbf{x},t)$:

$$\mathbf{j}(\mathbf{x},t) = \mathbf{v}(\mathbf{x},t)\, p(\mathbf{x},t) \tag{123}$$

Equation (122) can be expanded to include the density matrix if we add the spin degrees of freedom:

$$\frac{\partial p(\mathbf{x},\boldsymbol{\rho},t)}{\partial t} = -\nabla_{\mathbf{x}} \cdot \left[\mathbf{v}_{\mathbf{x}}(\mathbf{x},\boldsymbol{\rho},t)\, p(\mathbf{x},\boldsymbol{\rho},t)\right] - \nabla_{\boldsymbol{\rho}} \cdot \left[\mathbf{v}_{\boldsymbol{\rho}}(\mathbf{x},\boldsymbol{\rho},t)\, p(\mathbf{x},\boldsymbol{\rho},t)\right] \tag{124}$$

We must take the integral over the probability density to get the expectation value:

$$\boldsymbol{\rho}(\mathbf{x},t) = \int \boldsymbol{\rho} p(\mathbf{x},\boldsymbol{\rho},t)\, dV_{\boldsymbol{\rho}} \tag{125}$$

Where $dV_{\boldsymbol{\rho}}$ is the volume element of the density matrix space. Taking the derivative of equation (125) with respect to t produces the equation of motion:

$$\frac{\partial}{\partial t}\boldsymbol{\rho}(\mathbf{x},t) = \int \left[\frac{\partial}{\partial t} p(\mathbf{x},\boldsymbol{\rho},t)\right] \boldsymbol{\rho} dV_{\boldsymbol{\rho}} \tag{126}$$

The time derivative of the joint probability density can therefore be written as:

$$\frac{\partial p(\mathbf{x},\boldsymbol{\rho},t)}{\partial t} = -\mathrm{div}_{\boldsymbol{\rho}}\left[\, p(\mathbf{x},\boldsymbol{\rho},t)\, \mathbf{v}_{\boldsymbol{\rho}}\right] - \mathrm{div}_{\mathbf{x}}\left[\, p(\mathbf{x},\boldsymbol{\rho},t)\, \mathbf{v}_{\mathbf{x}}\right] \tag{127}$$

For example, for a rotating NMR sample, the velocity at a point $\boldsymbol{\rho}$ in spin space is given by:

$$\mathbf{v}_{\boldsymbol{\rho}} = -i\mathbf{H}(\mathbf{x},t)\boldsymbol{\rho} \tag{128}$$

The lab space velocity at a specific point may be obtained from the equations of classical mechanics describing motion in three dimensions:

$$\begin{gathered} \mathbf{x}(t) = \exp\left[-i\mathbf{M}t\right]\mathbf{x}(0) \\ \frac{d}{dt}\mathbf{x}(t) = -i\mathbf{M}\exp\left[-i\mathbf{M}t\right]\mathbf{x}(0) = -i\mathbf{M}\mathbf{x}(t) \quad \Rightarrow \quad \mathbf{v}_{\mathbf{x}} = -i\mathbf{M}\mathbf{x} \end{gathered} \tag{129}$$

After placing equations (127), (128) and (129) in equation (126) we will get:

$$\frac{\partial}{\partial t}\boldsymbol{\rho}(\mathbf{x},t) = \int \mathrm{div}_{\boldsymbol{\rho}}\left[\, p(\mathbf{x},\boldsymbol{\rho},t)\left[i\mathbf{H}(\mathbf{x},t)\boldsymbol{\rho}\right]\right]\boldsymbol{\rho} dV_{\boldsymbol{\rho}} + \int \mathrm{div}_{\mathbf{x}}\left[\, p(\mathbf{x},\boldsymbol{\rho},t)\left[i\mathbf{M}\mathbf{x}\right]\right]\boldsymbol{\rho} dV_{\boldsymbol{\rho}} \tag{130}$$

After some calculations [60] we get:

$$\frac{\partial\boldsymbol{\rho}(\mathbf{x},t)}{\partial t}=-i\mathbf{L}(\mathbf{x},t)\boldsymbol{\rho}(\mathbf{x},t)+\mathbf{M}(\mathbf{x},t)\boldsymbol{\rho}(\mathbf{x},t) \tag{131}$$

in which $\mathbf{L}(\mathbf{x},t)=\mathbf{H}(\mathbf{x},t)+i\mathbf{R}+i\mathbf{K}$ is the Liouvillian that is responsible for spin dynamics and $\mathbf{M}(\mathbf{x},t)$ is the spatial dynamics generator. The latter comes from classical mechanics. Some relevant examples are:

1. Flow along a coordinate x and circular motion with a phase φ

$$\frac{\partial p(x,t)}{\partial t}=\left[v(t)\frac{\partial}{\partial x}\right]p(x,t),\qquad \frac{\partial p(\varphi,t)}{\partial t}=\left[\omega(t)\frac{\partial}{\partial\varphi}\right]p(\varphi,t) \tag{132}$$

where v is linear velocity and ω is angular velocity.

2. Isotropic rotational diffusion in three dimensions

$$\frac{\partial p(\Omega,t)}{\partial t}=\left[D_{\mathrm{R}}\left(\mathbf{J}_{\mathrm{X}}^{2}+\mathbf{J}_{\mathrm{Y}}^{2}+\mathbf{J}_{\mathrm{Z}}^{2}\right)\right]p(\Omega,t) \tag{133}$$

where Ω is a shorthand for Euler angles or other orientation parameters, $\{\mathbf{J}_{\mathrm{X}},\mathbf{J}_{\mathrm{Y}},\mathbf{J}_{\mathrm{Z}}\}$ are angular momentum operators and D_{R} is the rotational diffusion coefficient.

2.1.2 Spin dynamics generators

The spin component of the equation of motion is:

$$\begin{gathered}\frac{\partial\boldsymbol{\rho}(\mathbf{r},t)}{\partial t}=-i\mathbf{L}(\mathbf{r},t)\boldsymbol{\rho}(\mathbf{r},t)\\ \mathbf{L}(\mathbf{r},t)=\mathbf{H}(\mathbf{r},t)+i\mathbf{R}(\mathbf{r})\end{gathered} \tag{134}$$

Where $\boldsymbol{\rho}(\mathbf{r},t)$ is the spin density matrix at the spatial point $\mathbf{r}$ at time t, $\mathbf{H}(\mathbf{r},t)$ is the spin Hamiltonian commutation superoperator, $\mathbf{R}(\mathbf{r})$ is the relaxation superoperator. The spin Hamiltonian contains chemical shift terms, pulse field gradients, radiofrequency pulse terms, *J*-couplings, and a great variety of other complications that are discussed in magnetic resonance textbooks [59]:

$$\begin{gathered}\mathbf{H}(\mathbf{r},t)=-\left[B_0+\mathbf{g}^{\mathrm{T}}(t)\cdot\mathbf{r}\right]\sum_n(1+\delta_n)\gamma_n\mathbf{S}_{\mathrm{Z}}^{(n)}-B_{1\mathrm{X}}(t)\sum_n(1+\delta_n)\gamma_n\mathbf{S}_{\mathrm{X}}^{(n)}-\\ -B_{1\mathrm{Y}}(t)\sum_n(1+\delta_n)\gamma_n\mathbf{S}_{\mathrm{Y}}^{(n)}+2\pi\sum_{n<k}J_{nk}\left[\mathbf{S}_{\mathrm{X}}^{(n)}\mathbf{S}_{\mathrm{X}}^{(k)}+\mathbf{S}_{\mathrm{Y}}^{(n)}\mathbf{S}_{\mathrm{Y}}^{(k)}+\mathbf{S}_{\mathrm{Z}}^{(n)}\mathbf{S}_{\mathrm{Z}}^{(k)}\right]+\ldots\end{gathered} \tag{135}$$

where B_0 is the primary magnet field (assumed to be directed along the Z axis), $\mathbf{g}$ is the primary magnet field gradient vector, δ_n are nuclear chemical shifts, γ_n are nuclear magnetogyric ratios, $\left\{\mathbf{S}_{\mathrm{X}}^{(n)},\mathbf{S}_{\mathrm{Y}}^{(n)},\mathbf{S}_{\mathrm{Z}}^{(n)}\right\}$ are nuclear spin operators, J_{nk} are inter-nuclear scalar couplings (traditionally published in Hz, hence the 2π in front), $B_{1\{\mathrm{X,Y}\}}$ are the Cartesian components of the radiofrequency magnetic field, which is assumed to have no Z component, *et cetera* – Equation (135) can be rather long, but for our purposes $\mathbf{H}(\mathbf{r},t)$ is simply a matrix that standard software packages can return on demand. The same applies to the relaxation superoperator, which this communication is not a place to discuss.

An important point from the computational efficiency point of view is that the dimension of the Hilbert space spanned by a realistic spin system trajectory is usually much smaller than the dimension of the full Hilbert space of the spin system [61-64]. A non-constructive proof may be obtained from the fact that the dimension of the space spanned by the simulation trajectory

$$\left\{\boldsymbol{\rho}_0,\mathbf{P}\boldsymbol{\rho}_0,\mathbf{P}^2\boldsymbol{\rho}_0,...,\mathbf{P}^N\boldsymbol{\rho}_0\right\},\qquad \mathbf{P}=\exp\left[-i\mathbf{L}\Delta t\right] \tag{136}$$

is equal the number of linearly independent vectors in it, which is smaller or equal to the number of discrete time points in the trajectory [62]. At a more physically motivated level:

1. Some spin states are not reachable from the initial condition under a given Hamiltonian and a given relaxation superoperator:

$$\left\langle\boldsymbol{\sigma}\middle|e^{-i\mathbf{L}t}\middle|\boldsymbol{\rho}_0\right\rangle=0\qquad\forall t\in[0,t_{\mathrm{sim}}] \tag{137}$$

   All states that *a priori* do not occur in the system trajectory (or whose amplitude does not exceed a user-defined threshold) may be dropped from the basis [62]

2. Some states are never populated to a significant extent because relaxation drains most of the amplitude *en route*. Any states that are reachable, but whose population stays below a set accuracy threshold, may also be dropped from the basis [64].

3. Some states are not reachable from the detection state. Any dynamics involving those states will never impact the observed parameter. For the purposes of simulating the dynamics of that parameter, such states may be dropped from the basis [65].

4. Some states are reachable, generally populated and connected to the detection state, but under specific conditions of a given simulation instance or stage happen to stay unpopulated or have no effect on the observed parameter. Once identified, such states may also be dropped from the basis [62, 65, 66].

These observations are particularly relevant in liquid state magnetic resonance, where the dimension of the space spanned by the system trajectory was recently shown to be orders of magnitude smaller than the dimension of the full Hilbert space [67].

Importantly, matrix representations of spin operators may be built directly in the reduced basis [67]. The procedure makes use of the fact that the Lie algebra of operators acting on a multi-spin system is spanned by direct products of basis operators of the Lie algebras of individual spins:

$$\bigotimes_{k=1}^{N} \mathfrak{su}\left(2s_k+1\right) \tag{138}$$

where N is the number of spins and s_k is the quantum number of each spin. The only consistent way to remove a specific generator $\mathbf{O}_m$ from a Lie algebra while preserving the rest of its structure is to modify the corresponding structure coefficients:

$$[\mathbf{O}_i, \mathbf{O}_j] = \sum_k c_{ijk}\mathbf{O}_k \tag{139}$$

by zeroing the coefficients c_{ijm} that lead into the "ignored" physical state. The resulting object is still a Lie algebra that now generates the time dynamics of the approximate physical model under the exponential map. Importantly, the structure coefficients of the direct product algebra are related to the structure coefficients of the single-spin algebras in a computationally friendly way because:

$$\begin{gathered}\mathrm{Tr}\left(\mathbf{O}_i\mathbf{O}_j\mathbf{O}_k^{\dagger}\right) = \mathrm{Tr}\left[\left(\bigotimes_{n=1}^{N}\mathbf{S}_{in}\right)\left(\bigotimes_{n=1}^{N}\mathbf{S}_{jn}\right)\left(\bigotimes_{n=1}^{N}\mathbf{S}_{kn}\right)^{\dagger}\right] = \\ = \mathrm{Tr}\left[\bigotimes_{n=1}^{N}\left(\mathbf{S}_{in}\mathbf{S}_{jn}\mathbf{S}_{kn}^{\dagger}\right)\right] = \prod_{n=1}^{N}\mathrm{Tr}\left(\mathbf{S}_{in}\mathbf{S}_{jn}\mathbf{S}_{kn}^{\dagger}\right)\end{gathered} \tag{140}$$

where $\mathbf{S}_{kn}$ is the *k*-th generator of the Lie algebra of the *n*-th spin – usually a 2×2 matrix! The cost of computing individual structure coefficients of the full spin system algebra is thus linear with respect to the total number of spins. When the restricted basis scales polynomially in the size of the spin system (true in all liquid state magnetic resonance experiments [64]), this yields an overall polynomially scaling simulation algorithm.

The technical details of how reduced spin operator matrices are generated are outside the scope of this communications. The salient point is that the dimension of the Liouvillian that is realistically necessary to describe an *N*-spin system is much smaller – usually orders of magnitude smaller – than 4^N. For your purposes here we would simply note that all relevant methods [61, 62, 64-68] are implemented in *Spinach* [69] and the program simply provides the necessary matrices on demand.

2.1.3 Spatial dynamics generators

In 1687, a description of viscous fluid motion had been proposed in “Principia” by Newton [70] in which dynamic behaviour of fluids of constant viscosity was investigated. Later, Bernoulli and Euler derived the equations for inviscid flow now known as Euler’s inviscid equations [71]. Even though Navier, Cauchy, Poisson, and St. Venant had carried out studies to explore the mathematical model of fluid flow [72], they had overlooked the viscous (frictional) force. In 1845, Stokes had derived the equation of motion of a viscous flow by adding Newtonian viscous terms[73], thereby the Navier-Stokes equations had been brought to their final form which has been used to generate numerical solutions for fluid flow [74].

The Navier-Stokes equations are a set of nonlinear partial differential equations that describe the flow of fluids. They model weather, the movement of air in the atmosphere, ocean currents, water flow in a pipe, as well as many other fluid flow phenomena [75]. They depend on the combination of Newton’s second law with a fluid stress (due to viscosity) and pressure term. For almost all real situations, they result in a system of nonlinear partial differential equations; however, with certain simplifications (such as 1-dimensional motion) they can sometimes be reduced to linear differential equations [76]. Usually, however, they remain nonlinear, which makes them difficult or impossible to solve; this is what causes turbulence and strong dependence on initial conditions.

The solution of the Navier-Stokes equations can be obtained with either analytical [77] or numerical methods [71] . The analytical method is only valid for simple cases such as Couette flow [78], Poiseuille flow [79], *etc*. On the other hand, almost every case in fluid dynamics comprises non-linear and complex structures in the mathematical model which cannot be ignored without loss of accuracy [80].

Given the complexity and the non-linearity (which is not compatible with either Liouville - von Neumann or with Fokker-Planck equation), a decision was made to only implement *stationary* flow in *Spinach*, because the dynamical generators associated with stationary flow are linear [81]:

$$\mathbf{M}\left(\mathbf{r},t\right)=\nabla^{\mathrm{T}}\cdot\mathbf{v}\left(\mathbf{r},t\right)+\mathbf{v}^{\mathrm{T}}\left(\mathbf{r},t\right)\cdot\nabla \tag{141}$$

where $\nabla=\left[\partial/\partial x \quad \partial/\partial y \quad \partial/\partial z\right]^{\mathrm{T}}$ is the gradient (in the mathematical sense) operator, and $\mathbf{v}\left(\mathbf{r},t\right)$ is the flow velocity. The latter rarely depends on time, meaning that the spatial dynamics generator in Equation (249) is time-independent and may be consigned to the background evolution operator – this is the primary advantage of the Fokker-Planck formalism over other simulation methods for the problem in question [81]. The use of the flow velocity in Equation (249) eliminates the thorny question of how to combine magnetic resonance simulations (where the equation of motion is linear) with hydrodynamics simulations (where it is not). Because spin degrees of freedom may be very accurately

assumed to have no influence on the spatial motion, any hydrodynamics solver may simply be used first to obtain $\mathbf{v}(\mathbf{r},t)$ and put it into Equation (249). In that way, the equation of motion stays linear with respect to the Fokker-Planck state vector $\boldsymbol{\rho}(\mathbf{r},t)$.

Given a concentration profile $c(\mathbf{r},t)$, the diffusion flux is given by Fick's first law [82], and the hydrodynamic flux is the product of concentration and velocity. The total flux therefore is:

$$\mathbf{j}(\mathbf{r},t)=\mathbf{v}(\mathbf{r},t)c(\mathbf{r},t)-\mathbf{D}(\mathbf{r},t)\nabla c(\mathbf{r},t) \tag{142}$$

where $\nabla=\left[\partial/\partial x \quad \partial/\partial y \quad \partial/\partial \mathrm{z}\right]^{\mathrm{T}}$ is the gradient operator, $\mathbf{v}(\mathbf{r},t)$ is the flow velocity and $\mathbf{D}(\mathbf{r},t)$ is the translational diffusion tensor. Conservation of matter requires the local decrease in concentration to be equal to the divergence of its flux, and therefore:

$$\frac{\partial}{\partial t}c(\mathbf{r},t)=-\mathrm{div}\left[\mathbf{j}(\mathbf{r},t)\right]=\left[\nabla^{\mathrm{T}}\cdot\mathbf{v}(\mathbf{r},t)+\mathbf{v}^{\mathrm{T}}(\mathbf{r},t)\cdot\nabla+\nabla^{\mathrm{T}}\cdot\mathbf{D}(\mathbf{r},t)\cdot\nabla\right]c(\mathbf{r},t) \tag{143}$$

At the matrix representation level, finite difference matrices [83] generally suffice for the gradient operator. The important point here is that all differential operators appearing in Equation (143) are direct products. For the gradient operator acting on the vectorised array of concentrations in every voxel:

$$\nabla=\begin{bmatrix}\partial/\partial x\\ \partial/\partial y\\ \partial/\partial z\end{bmatrix} \quad\Rightarrow\quad [\![\nabla]\!]=\begin{bmatrix}[\![\partial/\partial x]\!]\otimes\mathbf{1}_{\mathrm{Y}}\otimes\mathbf{1}_{\mathrm{Z}}\\ \mathbf{1}_{\mathrm{X}}\otimes[\![\partial/\partial y]\!]\otimes\mathbf{1}_{\mathrm{Z}}\\ \mathbf{1}_{\mathrm{X}}\otimes\mathbf{1}_{\mathrm{Y}}\otimes[\![\partial/\partial z]\!]\end{bmatrix} \tag{144}$$

where $[\![\partial/\partial x]\!]$ denotes a matrix representation of $\partial/\partial x$ on a finite grid. If the diffusion tensor is constant and isotropic, the generator only has three terms:

$$D\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2}\right)=D\left(\left[\!\!\left[\frac{\partial^2}{\partial x^2}\right]\!\!\right]\otimes\mathbf{1}_{\mathrm{Y}}\otimes\mathbf{1}_{\mathrm{Z}}+\mathbf{1}_{\mathrm{X}}\otimes\left[\!\!\left[\frac{\partial^2}{\partial y^2}\right]\!\!\right]\otimes\mathbf{1}_{\mathrm{Z}}+\mathbf{1}_{\mathrm{X}}\otimes\mathbf{1}_{\mathrm{Y}}\otimes\left[\!\!\left[\frac{\partial^2}{\partial z^2}\right]\!\!\right]\right) \tag{145}$$

where D is the diffusion coefficient. The dimension of each matrix is equal to the number of grid points along the corresponding dimension, typically of the order of 100. If the diffusion is anisotropic, the number of direct product terms in the sum rises to nine. Even if the entire $\mathbf{D}(\mathbf{r},t)$ array depends so strongly on $\mathbf{r}$ that it must be stored explicitly on the finite grid, the direct product structure of Equation (144) still only acts on one dimension at a time. The same applies to the flow generator.

Assuming hundreds of grid points in each of the three spatial directions, the biggest matrix dimension one would encounter in the polyadic form of Equation (144), and consequently Equation (143), is in the hundreds. The worst-case memory requirements would come from the velocity vector array and the diffusion tensor array: (hundreds)^3×(9 elements)×(64 bits per element) = about a gigabyte. This is well known to numerical hydrodynamics practitioners – the diffusion and flow problem in 3D is easily tractable. Finite difference derivative matrices are also sparse [83].

2.1.4 Matrix representations of spatial dynamics generators

Methods for numerical solution of PDEs [84] can be classified into two types:

1- Deterministic methods [85] where, for a given input, the output is always the same independently of how many times one solves the problem and what computer is used to solve it.

2- Stochastic methods [86] which rely on statistical principles. *Monte Carlo* method [87] is the most widely used stochastic method for solving PDEs.

The finite difference method [88] is a deterministic method. The idea is to substitute a discrete linear combination of function values instead of derivatives. In order to understand how finite difference treat numerically partial differential equations, we must define a function only at a finite number of grid points. At a chosen grid, each derivative in the partial differential equation must be replaced by suitable divided difference of function values, for example:

$$f'(x)=\lim_{\Delta x\to 0}\frac{f(x+\Delta x)-f(x)}{\Delta x}\quad\Leftrightarrow\quad f'(x_k)\approx\frac{f(x_k)-f(x_k-h)}{h} \tag{146}$$

The equation above called a finite difference approximation. It can get an approximate solution at each grid point. The derivative operation may be written in a matrix-vector form, and this provides the necessary connection to matrix representations:

$$\begin{pmatrix} f'(1)\\ f'(2)\\ f'(3)\\ \vdots\\ \\ \end{pmatrix}=\begin{pmatrix} 0 & +1 & & & & -1\\ -1 & 0 & +1 & & & \\ & -1 & 0 & +1 & & \\ & & & \ddots & & \\ & & & & & \\ +1 & & & & -1 & 0 \end{pmatrix}\begin{pmatrix} f_1\\ f_2\\ f_3\\ \vdots\\ \\ \end{pmatrix} \tag{147}$$

where periodic boundary condition is used. A large number of methods exist to generate finite difference matrices [89] of different types and accuracy levels. This is a technical subject that we will not

touch upon – we will simply observe here that matrix representations of controllable accuracy exist for any linear integro-differential operator on any finite grid.

2.1.5 Diffusion generators

Diffusion refers to the random motion of molecules caused by thermal fluctuations [90]. The physical law that describes this phenomenon is called Fick's first law [91]:

$$\vec{J} = -D\vec{\nabla}c \tag{148}$$

where $\vec{J}$ is the diffusion flux vector, D is the diffusion coefficient, $\vec{\nabla}$ is the gradient operator and c is concentration. The diffusion equation can be derived from Fick's law as follows. According to the mass balance, for a short section $\{x, x+\Delta x\}$ of a one-dimensional range:

$$V\frac{\partial c}{\partial t} = \text{input} - \text{output} = J\left(x,t\right)c - J\left(x+\Delta x,t\right)c \tag{149}$$

Since the total amount is $V = c\Delta x$, equation (149) can be rearranged:

$$\frac{\partial c}{\partial t} = \frac{J\left(x,t\right) - J\left(x+\Delta x,t\right)}{\Delta x} \tag{150}$$

In the limit of $\Delta x \to 0$, equation (150) can be written as:

$$\frac{\partial c}{\partial t} = -\frac{\partial}{\partial x}\vec{J} \tag{151}$$

Putting equations (148) and (151) together:

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x}D\left(\frac{\partial c}{\partial x}\right) \tag{152}$$

Finally (for a coordinate-independent diffusion coefficient):

$$\frac{\partial c}{\partial t} = D\nabla^2 c \tag{153}$$

Equation (153) is the diffusion equation. The form implemented in *Spinach* actually allows the diffusion coefficient to be coordinate-dependent.

2.1.6 Diffusion equation: solutions for unit testing

Numerical implementations in *Spinach* were tested against the analytical solutions described in this section. In one dimension, the separation of variables yields

$$c(x,t) = X(x)T(t) \tag{154}$$

One side of the equation (154) only depends on t, and the other only depends on x:

$$\frac{1}{D}\frac{T'(t)}{T(t)} = \frac{X''(x)}{X(x)} \tag{155}$$

Suppose each side equals λ. Hence we can write equation (155) as a system of two ODEs:

$$X''(x) + \lambda X(x) = 0 \tag{156}$$

$$T'(t) + \lambda D T(t) = 0 \tag{157}$$

The problem of finding the solution of equations (156) and (157) can therefore be reduced to solving linear ODEs and consideration of three different cases with respect to sign of λ :

$$\lambda < 0 \quad \Rightarrow \quad X(x) = C_1 e^{\sqrt{-\lambda}x} + C_2 e^{\sqrt{-\lambda}x} \tag{158}$$

Taking the boundary conditions $x(\pm\infty) = 0$ into account, one gets $C_1 = C_2 = 0$, so for $\lambda < 0$, only the trivial solution exists.

$$\lambda = 0 \quad \Rightarrow \quad X(x) = C_1 x + C_2 \tag{159}$$

Due to boundary conditions, this case also yields only trivial solution of the problem.

$$\lambda > 0 \quad \Rightarrow \quad X(x) = C_1 \cos(\lambda x) + C_2 \sin(\lambda x) \tag{160}$$

On the interval $x \in [0, L]$ with some initial condition and (for example) zero boundary conditions:

$$X(0) = C_1 = 0 \qquad X(L) = C_2 \sin(\lambda L) = 0 \tag{161}$$

This yields the quantisation condition for the eigenvalues:

$$\lambda_n = \left(\frac{\pi n}{L}\right)^2, \quad n = 1, 2, 3, \ldots \tag{162}$$

Hence, the solution can be written in the following general form:

$$X(x) = C_n \sin\left(\frac{\pi n}{L} x\right) \tag{163}$$

where C_n are constants. The temporal part in equation (157) is also easily solved:

$$T'(t)+D\left(\frac{\pi n}{L}\right)T(t)=0 \quad \Rightarrow \quad T(t)=B_n\exp\left(-D\left(\frac{\pi n}{L}\right)^2 t\right) \tag{164}$$

Where B_n are constants. The general solution can therefore be written as:

$$c(x,t)=\sum_{n=1}^{\infty}A_n\sin\left(\frac{n\pi x}{L}\right)\exp\left(-D\left(\frac{\pi n}{L}\right)^2 t\right) \tag{165}$$

where A_n are constants. To find A_n using the initial conditions $x\in[0,L]$, we can write the initial condition function $f(x)$ as a Fourier series:

$$f(x)=\sum_{n=1}^{\infty}A_n\sin\left(\frac{n\pi x}{L}\right) \qquad A_n=\frac{2}{L}\int_0^L f(x)\sin\left(\frac{n\pi x}{L}\right)dx \tag{166}$$

The final solution therefore is:

$$c(x,t)=\sum_{n=1}^{\infty}\left(\frac{2}{L}\int_0^L f(x)\sin\left(\frac{n\pi x}{L}\right)dx\right)\sin\left(\frac{n\pi x}{L}\right)\exp\left(-D\left(\frac{\pi n}{L}\right)^2 t\right) \tag{167}$$

2.2 Numerical simulation examples

A variety of initial condition functions can be handled using this formula and its higher dimensional equivalents, and the answers tested against finite difference numerical implementations.

I will now illustrate the performance of the numerical solver that was implemented into *Spinach* during this project. Consider the one-dimensional diffusion equation with a diffusion coefficient D:

$$\frac{\partial c}{\partial t}=D\frac{\partial^2 c}{\partial x^2} \tag{168}$$

At the trivial level, simply starting from a Gaussian function with periodic boundary conditions yields a series of pictures shown in **Figure 13**.

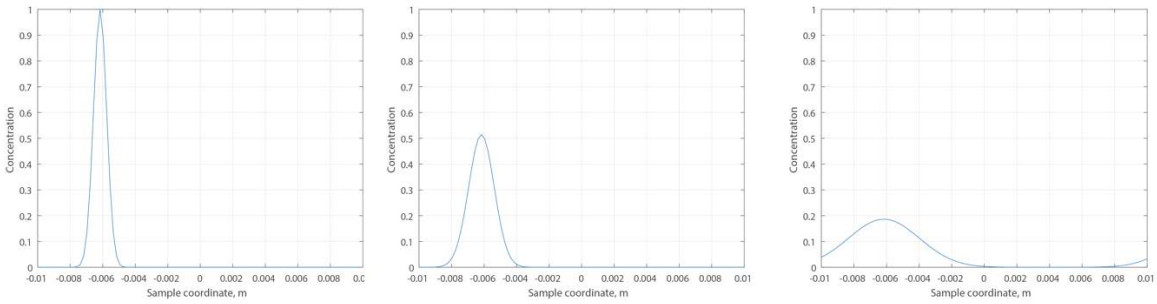


Figure 13. *Evolution of the probability density function for increasing diffusion times in the case of one-dimensional diffusion. D = 5x10^{-5} m^2/s, Gaussian initial condition, periodic boundary conditions. Spinach example set: diffusion_test_1.m*

Going a level up in complexity, for most fluids and some homogeneous solid materials like gels, diffusion is the same in every direction. These systems are called isotropic, and the diffusion process is still characterised by a scalar coefficient:

$$\frac{\partial c}{\partial t} = D\left[\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2}\right] \tag{169}$$

A simple two-dimensional simulation of this type is shown in **Figure 14**.

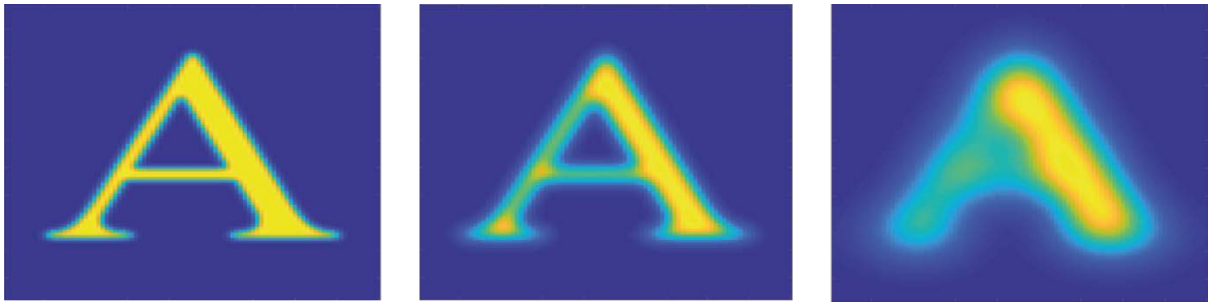


Figure 14. *Evolution of the probability density function for increasing diffusion times in the case of two-dimensional uniform and isotropic diffusion. Initial condition as shown in the left panel, periodic boundary conditions. Spinach example set: diffusion_test_2a.m*

However, even for isotropic diffusion, the diffusion coefficient need not be equal in different parts of the sample. Additional derivatives appear in the diffusion equation in this case because D is no longer-coordinate independent. The implementation in *Spinach* uses the full Equation (191) and therefore has no problem with this case – as illustrated in **Figure 15**.

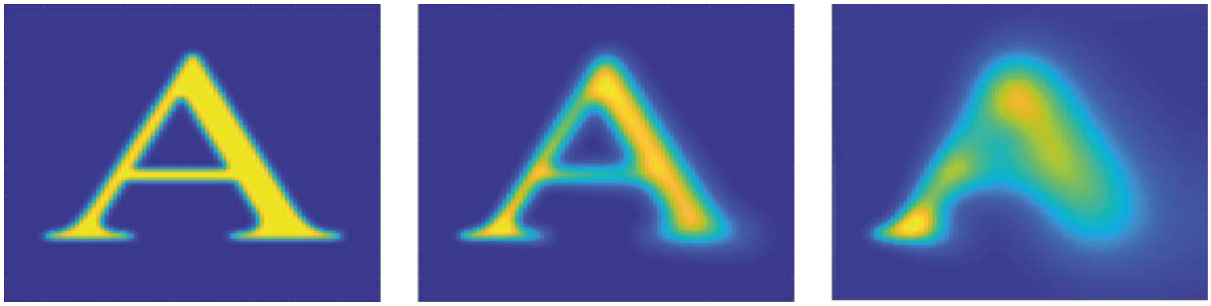


Figure 15. *Evolution of the probability density function for increasing diffusion times in the case of two-dimensional non-uniform and isotropic diffusion. Initial condition as shown in the left panel, periodic boundary conditions. Spinach example set: diffusion_test_2b.m*

Biological tissues are highly structured and typically have different diffusion coefficients along different directions – for example white matter is anisotropic because of the parallel orientation of the nerve fibre tracts. Other tissues with significant anisotropy are muscle and tendons. The diffusion tensor is a matrix in these systems, and can vary in orientation as well as amplitude. An example of such a simulation for a two-dimensional phantom is shown in **Figure 16**.

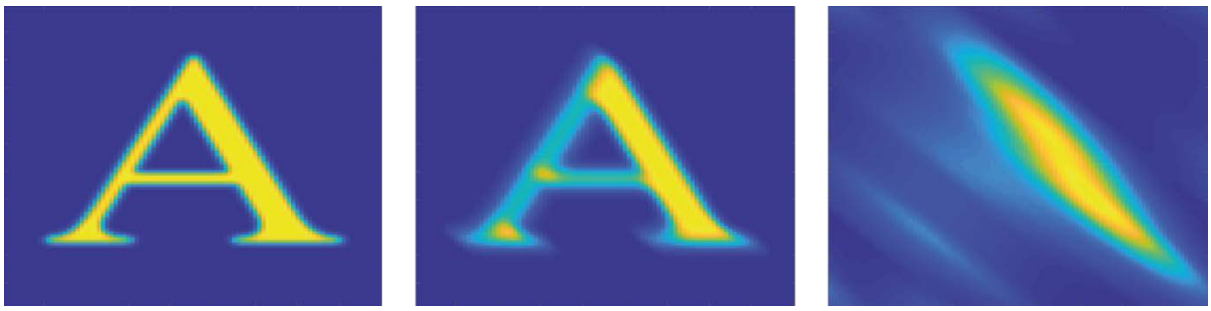


Figure 16. *Evolution of the probability density function for increasing diffusion times in the case of two-dimensional uniform and anisotropic diffusion. Initial condition as shown in the left panel, periodic boundary conditions. Spinach example set: diffusion_test_2c.m*

The general three-dimensional case was implemented in *Spinach* during this project – see v2fplanck.m function for the technical details of the programming involved.

2.3 Stationary flow and diffusion: numerical examples

The illustrations for the numerical solver that accounts for the diffusion and stationary flow simultaneously (**Figure 17**) are generally similar to the ones already used for the pure diffusion – the software module is actually exactly the same, only the velocity field information is now provided.

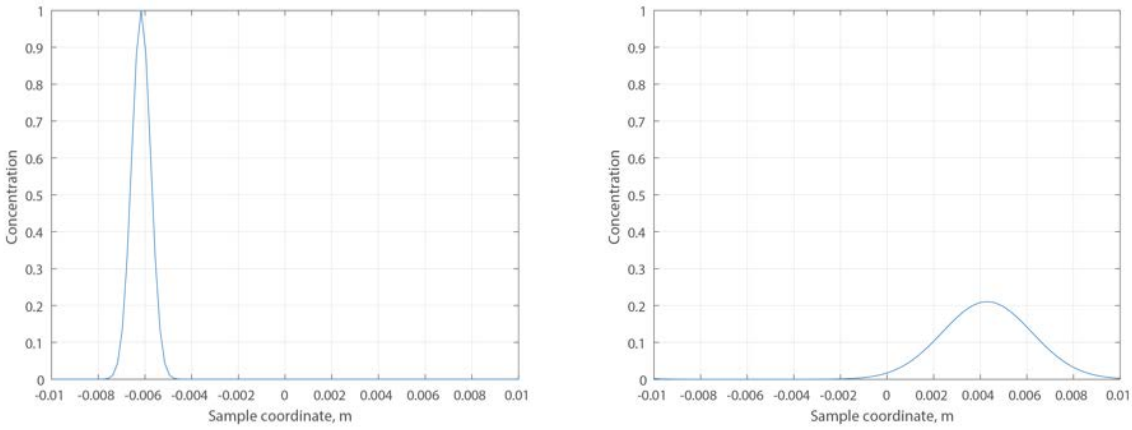


***Figure 17**. A one-dimensional illustration of the combination between diffusion and flow simulated using the Fokker-Planck solver in Spinach. A uniform flow velocity field (0.2 m/s) is used, and the diffusion coefficient is 50 mm²/s. Spinach example set: flow_test_1a.m*

An important matter that must be tackled with flow simulations is that of the boundary conditions. The only type of boundary that is not a veritable minefield in terms of the associated numerical mathematics is the periodic boundary [92]. At this stage in the development I have therefore decided to only implement the periodic boundary condition, which is illustrated in **Figure 18**.

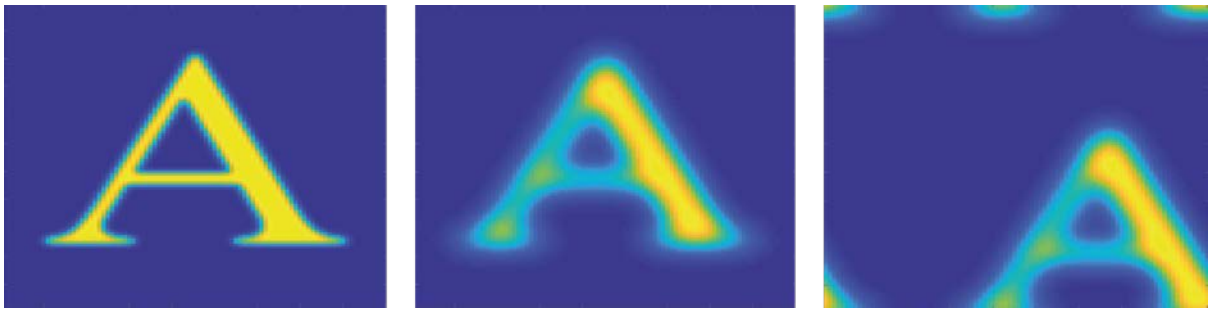


***Figure 18.** Evolution of the probability density function for increasing diffusion times in the case of two-dimensional uniform and isotropic diffusion combined with a linear flow process. D = 5x10⁻⁵ m²/s, initial condition as shown in the left panel, periodic boundary conditions. Spinach example set: flow_test_2.m*

Because Spinach makes claims about being able to simulate three-dimensional MRI processes in reasonable time, an example is required to that effect (**Figure 19**).

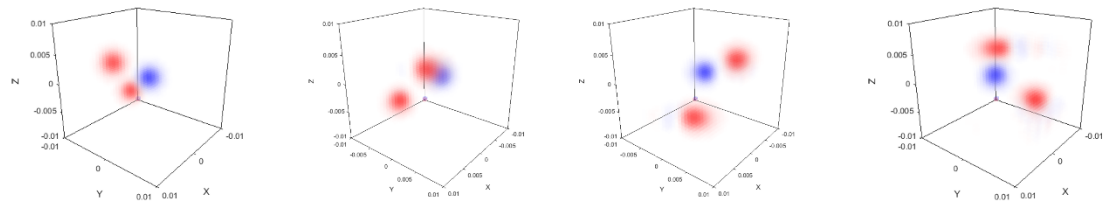


Figure 19. *Frames from a three-dimensional diffusion and flow simulation with concentration spots of three different substances flowing in a circular flow field.The simulation takes a few minutes on a modern computer. Spinach example set: flow_test_3.m*

The summary for this section is that spatial dynamics generators were implemented in code and passed all necessary tests.

As an illustration of Spinach syntax and generality, this section contains a walk through a specific simulation that combines quantum mechanics, diffusion, flow and imaging – a singlet state flow imaging experiment. We first specify the list of isotopes for a specific spin system:

```
% Isotopes
sys.isotopes={'13C','13C'};
```

then the applied magnet field in Tesla:

```
% Magnetic induction
sys.magnet=9.4;
```

Chemical shift must be specified in ppm for nuclei:

```
% Chemical shifts
inter.zeeman.scalar={0.03,-0.03};
```

J-coupling between nuclei

```
% J coupling
inter.coupling.scalar=cell(2);
inter.coupling.scalar{1,2}=55;
```

Bloch-Redfield-Wangsness relaxation theory and its parameters:

```
% Relaxation theory
inter.relaxation={'redfield'};
inter.equilibrium='zero';
inter.rlx_keep='secular';
inter.tau_c={1e-9};
```

Liouville space formalism:

```
% Basis set
bas.formalism='sphten-liouv';
```

Complete basis set, since the spin system is small:

```
bas.approximation='none';
```

Any *Spinach* calculation must begin with a call to create.m and basis.m constructor functions. They process spin system and simulation formalism specifications and create the spin_system object – the primary data structure that is used to store spin system information in *Spinach*:

```
% Spinach housekeeping
spin_system=create(sys,inter);
spin_system=basis(spin_system,bas);
```

Rotating frame assumptions. This setting determines the Hamiltonian:

```
% Assumptions
spin_system=assume(spin_system,'nmr');
```

Relaxation phantom – the same relaxation superoperator everywhere:

```
% Relaxation phantom
parameters.rlx_ph={ones(parameters.npts)};
parameters.rlx_op={relaxation(spin_system)};
```

NMR tube phantom – ones inside the tube, zeros outside:

```
% NMR sample tube phantom
tube=zeros(parameters.npts); tube(:,6:10)=1;
```

The initial condition phantom – Lz everywhere inside the tube:

```
% Initial state phantom
parameters.rho0_ph={tube};
parameters.rho0_st={state(spin_system,'Lz','13C','cheap')};
```

The detection state phantom reflects the fact that different voxels might be detected at different angles and with different sensitivity. We set the detection operator to be uniform L+ across the tube:

```
% Detection state phantom
parameters.coil_ph={tube};
parameters.coil_st={state(spin_system,'L+','13C','cheap')}
```

X and Y components of the velocity vectors for each point in the sample – vertical flow:

```
% Flow
parameters.u=-6e-2*ones(parameters.npts);
parameters.v=zeros(parameters.npts);
```

3x3 diffusion tensor – assuming only longitudinal diffusion:

```
% Diffusion
parameters.diff=[3.6e-6 0 0; 0 0 0; 0 0 0];
```

Parameters of the slice selection pulse:

```
        % Pulse phase
        rf_phi=pi/2;

        % Number of steps in the pulse
        pulse_nsteps=25;

        % Overall pulse duration
        pulse_time=2e-3;

        % Pulse frequency
```

```
pulse_frq=2000;

% Pulse amplitude table
rf_amp_list=2*pi*100*pulse_shape('gaussian',pulse_nsteps);

% Pulse block duration table
rf_dur_list=(pulse_time/pulse_nsteps)*ones(1,pulse_nsteps);

% Gradient amplitude
grad_ampl=6e-3;                                 %                                T/m
```

When the sequence starts, we apply a pulse at the specified frequency under Gx using a shaped pulse in amplitude-frequency coordinates using Fokker-Planck formalism

```
% Apply a pulse at a specific frequency under Gx
rho=shaped_pulse_af(spin_system,L+grad_ampl*G{1},Lx,Ly,parameters.rho0,...
                    pulse_frq*ones(1,pulse_nsteps),rf_amp_list,rf_dur_list,...
                    rf_phi,2,'expv');
```

Parameters of the M2S sequence [93] are then calculated:

```
% M2S parameters
J=55; delta_v=6;
t=1/(4*sqrt(J^2+delta_v^2));
m1=floor(pi*J/(2*delta_v));
if mod(m1,2)~=0, m1=m1+1; end
```

The M2S pulse sequence is run:

```
% M2S sequence
for k=1:m1
    rho=step(spin_system,H,rho,t);
    rho=step(spin_system,Lx,rho,pi);
    rho=step(spin_system,H,rho,t);
end
rho=step(spin_system,Lx,rho,pi/2);
rho=step(spin_system,H,rho,t);
for k=1:m1/2
    rho=step(spin_system,H,rho,t);
    rho=step(spin_system,Lx,rho,pi);
    rho=step(spin_system,H,rho,t);
end
rho_sing=-step(spin_system,H,rho,t);
```

Because only a slice was excited, only a slice gets transformed into singlet. We then run the time evolution and obtain system trajectory:

```
traj{1}=evolution(spin_system,L,[],rho_sing,0.01,100,'trajectory');
```

Another trajectory is also recorded for the situation when the M2S sequence is not run – meaning that simple magnetisation is being detected. We can then inspect the result by projecting the full Fokker-Planck state vector into the phantom space:

```
for n=1:100
    tube_sing=fpl2phan(traj{1}(:,n),singlet(spin_system,1,2),parameters.npts);
    tube_magn=fpl2phan(traj{2}(:,n),state(spin_system,'L+','all'),parameters.npts);
    subplot(1,2,1); imagesc(real(tube_sing),25*[-2e-3 6e-3]/2); colorbar; title('singlet');
    subplot(1,2,2); imagesc(real(tube_magn),25*[-2e-3 6e-3]); colorbar; title('magnetization');
    drawnow; pause(0.01);
end
```

The simulation parameters are shown in **Table 3** and the result is shown in **Figure 53** and published [94].

***Table 3**. Singlet state NMR simulation parameters*

Isotope	13C
Chemical shift	6 ppm
J-coupling	55 Hz
Pulse time	2 ms
Pulse frequency	2 kHz
Gradient amplitude	6 mT/m
Flow rate	60 mm/s
Diffusion coefficient	$3.6*10^{-6}$ m^2/s

A simulation of the singlet state diffusion and flow experiment from Figure 2 of the recent paper by Pileio *et al.* [95] shown in **Figure 53** is particularly good example, because all of the complexities – shaped pulses, gradients, diffusion, flow, quantum mechanical spin evolution and complicated relaxation theory – are present simultaneously.

3 Simulation Platform

3.1 Simulation requirements

There are two well researched limits in the numerical simulation of magnetic resonance imaging: complicated spatial dynamics of simple spin systems, and simple spatial dynamics of complicated spin systems. The former is diffusion and flow MRI of mostly water [96, 97], and the latter is spatially encoded NMR experiments [34, 98], and localised NMR spectroscopy [99]. Both cases are well covered by the existing simulation software. Both are also computationally straightforward because matrix dimensions are manageable: $\mathfrak{aut}\left(\mathbb{R}^3\right)\otimes\mathfrak{su}\left(2\right)$ and $\mathbb{R}^3\otimes\mathfrak{su}\left(2^N\right)$ are both tractable, either directly or with reasonable approximations [61, 63] for the matrix representations of $\mathfrak{su}\left(2^N\right)$, where N is the number of spins.

The simulation problem becomes numerically intractable when complicated spatial dynamics (diffusion, convection, and flow in three dimensions) is combined with complicated spin dynamics (spin-spin coupling, cross-relaxation, chemical kinetics, *etc.*). A well digitised 3D phantom would have at least a hundred points in each of the three directions, meaning a dimension of at least $100^3 = 10^6$ for the spatial dynamics generator matrices. At the same time, a typical metabolite (*e.g.* glucose) contains upwards of ten coupled spins, meaning a Liouville space dimension of at least $4^{10} \approx 10^6$. Direct products of spin and spatial dynamics generators would then have the dimension in excess of 10^{12} even before chemical kinetics is considered – clearly a non-starter, even if sparse matrix arithmetic is used.

At the same time, sophisticated MRI and spatio-temporal NMR techniques are being increasingly called upon to report on neurotransmitters (*e.g.* acetylcholine [100] and γ-aminobutyric acid [101]), bioenergetic metabolites (*e.g.* pyruvate [102] and lactate [103]), osmolytes (*e.g.* sarcosine [104] and taurine [105]), cellular membrane components (*e.g.* triglycerides [106] and cholesterol), and other substances that, unlike water, feature non-trivial quantum mechanical spin processes in liquid phase. Metabolic imaging also implies the presence of chemical kinetics. Further away from the clinic, an increasing number of DNP [107], PHIP [108]and singlet [93] imaging experiments require accurate quantum mechanical treatment of spatially distributed spin dynamics, also including simultaneous contributions from spin-spin coupling, cross-relaxation, convection, diffusion, flow, and chemical kinetics.

Numerical simulation of such systems used to be impossible in particular because the available software [56-58, 109] always insisted on opening the Kronecker products in the structure of the problem:

$$[\text{space dynamics}]\otimes[\text{reaction kinetics}]\otimes[\text{spin dynamics}] \tag{170}$$

before running the simulation. If all three matrices are large, the resulting object has intractable dimensions. However, our long and careful look at the entire structure of the magnetic resonance simulation problem [69] produced an interesting observation: the only algebraic operation *essentially* required in any Liouville space magnetic resonance simulation is matrix-vector product, where the matrix is a sum of objects with the structure shown in Equation (170). Even time propagation, which technically involves a matrix exponential, can be reformulated using only matrix-vector operations [110], for example:

$$\boldsymbol{\rho}(t+\Delta t)=\exp\left[-i\mathbf{L}(t)\Delta t\right]\boldsymbol{\rho}(t)=\sum_{n=0}^{\infty}\frac{(-i\Delta t)^n}{n!}\mathbf{L}\left(\ldots\left(\mathbf{L}\left(\mathbf{L}\boldsymbol{\rho}\right)\right)\right) \tag{171}$$

where $\boldsymbol{\rho}$ is the state vector, $\mathbf{L}$ is Liouvillian superoperator, and t is time. The second important observation then comes from linear algebra – the action by a direct product on a vector may be computed without opening the direct product, for example:

$$\left[\mathbf{A}\otimes\mathbf{B}\right]\mathbf{v}=\text{vec}\left[\mathbf{B}\mathbf{V}\mathbf{A}^{\text{T}}\right] \tag{172}$$

where $\mathbf{A}$ and $\mathbf{B}$ are matrices, $\mathbf{V}$ is obtained by reshaping $\mathbf{v}$ into a matrix of appropriate dimensions for the product on the right hand side, and *vec* stands for column-wise stretch of a matrix into a vector. The right hand side of Equation (172) is much cheaper to compute than the left hand side. The entire class of such relations is extensively researched [111]; they extend to multiple and nested direct products, as well as their sums.

I postulate here that direct products are best left unopened in magnetic resonance simulations, particularly in MRI, where the most problematic product is between spatial and spin degrees of freedom. This hypothesis is explored in this work: I report the design and implementation of a quantum mechanical MRI and spatio-temporal NMR simulation module in *Spinach* [69] that uses polyadics (the formal name for a sum of direct products) without opening them up. It is demonstrated below that previously unthinkable MRI simulations now run in minutes.

3.2 General simulation framework

The general MRI simulation problem contains three principal factors: (a) spatial distributions and spatial dynamics, including field maps of various coils, diffusion, and flow; (b) chemical kinetics; (c) spin

dynamics and relaxation. These factors are in a direct product relationship – each voxel may have different concentrations and transport velocities, each chemical species may have different spin dynamics, and spin dynamics may in turn be different in each voxel due to, for example, a magnetic field gradient or a different viscosity that affects relaxation. Mathematically speaking, the structure of the problem is captured by Equation (170), with the equation of motion consequently having the following algebraic form:

$$\frac{d}{dt}\boldsymbol{\rho}(t)=\left[\sum_{nmk}a_{nmk}(t)\,\mathbf{M}_n\otimes\mathbf{K}_m\otimes\mathbf{S}_k\right]\boldsymbol{\rho}(t) \tag{173}$$

where $\boldsymbol{\rho}(t)$ is the state vector, $a_{nmk}(t)$ are interaction amplitudes, $\mathbf{M}_n$ are spatial operators, $\mathbf{K}_m$ are chemical kinetics operators (possibly themselves dependent on $\boldsymbol{\rho}$ if the kinetics is non-linear), and $\mathbf{S}_k$ are spin dynamics operators. The formal solution is:

$$\boldsymbol{\rho}(t+dt)=\exp\left\{\left[\sum_{nmk}a_{nmk}(t)\,\mathbf{M}_n\otimes\mathbf{K}_m\otimes\mathbf{S}_k\right]dt\right\}\boldsymbol{\rho}(t) \tag{174}$$

The problem of simulating each of the three compartments of Equation (170) *individually* is comprehensively solved and extensively studied, from theoretical foundations [59, 112, 113] all the way to numerical implementations [56-58, 109, 114, 115]. However, the *composite* problem runs into the matrix dimension issue described in the Introduction. The solutions offered here revolve around compressing the dimensions of the three sub-problems to the maximum possible extent, and never opening the Kronecker products in Equation (173). The insight that makes efficient simulation possible is empirical: the polyadic object appearing in Equation (173) is always low-rank. In other words, the sum is short and contains many unit matrices.

3.3 Spinach infrastructure

Consider the general case of a matrix-vector product where the matrix itself is a Kronecker product of smaller matrices, some of which may be unit matrices:

$$\mathbf{y}=\left[\mathbf{A}^{(1)}\otimes\mathbf{A}^{(2)}\otimes\ldots\otimes\mathbf{A}^{(N)}\right]\mathbf{x}$$

The number of elements in $\mathbf{x}$ is the product of column dimensions of $\mathbf{A}^{(k)}$, and the number of elements in $\mathbf{y}$ is the product of their row dimensions. The multiplication procedure, described in detail by Fernandes, Plateau, and Stewart [111] and first implemented for *Matlab* by David Gleich, is as follows.

1. Record column dimensions of $\mathbf{A}^{(N-k+1)}$ into c_k.

2. Reshape $\mathbf{x}$ into an *N*-dimensional array $\mathbf{X}$ with dimensions c_k.

3. Loop index n over the dimensions of $\mathbf{X}$

   Permute the dimensions of $\mathbf{X}$ to make its n^{th} dimension left-most.

   Reshape $\mathbf{X}$ to into a matrix of column dimension c_n and row dimension $\prod_{m\neq n} c_m$.

   Perform reassignment $\mathbf{X}=\mathbf{A}^{(N-n+1)}\cdot\mathbf{X}$ and replace c_n by the row dimension of $\mathbf{A}^{(N-n+1)}$.

   Reshape $\mathbf{X}$ back into an *N*-dimensional array $\mathbf{X}$ with dimensions c_k.

   Put the dimensions of $\mathbf{X}$ back into the original order.

4. Reshape $\mathbf{X}$ back into a vector and return it as $\mathbf{y}$.

This procedure is implemented in the *kronm* function supplied with *Spinach* versions 2.3 and later. Minor logistical optimisations are implemented: multiplication by unit matrices is skipped, and memory-intensive dimension reordering operations are avoided for the first and the last instance of the loop.

A considerable amount of software engineering is required before the method above becomes useful for solving Equation (131). The first significant hurdle is addition: there is no way to add two Kronecker products without opening them up – direct products do not commute with summation. However, MRI Hamiltonians are always sums of direct products, and therefore a workaround is required. This is implemented by simply buffering the terms of the sum as a pointer array:

$$\mathbf{A}\otimes\mathbf{B}+\mathbf{C}\otimes\mathbf{D}\otimes\mathbf{E}+\ldots \quad\Leftrightarrow\quad \left\{\left\{\mathbf{A},\mathbf{B}\right\},\left\{\mathbf{C},\mathbf{D},\mathbf{E}\right\},\ldots\right\} \tag{175}$$

We rely here on the fact that MRI Hamiltonians are always *short* sums of Kronecker products: the number of terms is always much smaller than the dimension of the resulting matrix. A multiplication by the object in Equation (175) is then easy to implement as a loop over its constituent terms.

The second significant hurdle is the norm – numerical implementations of Equation (171) in finite precision arithmetic require the knowledge of the matrix norm. However, the immediate definition of a matrix norm requires either extreme eigenvalues (2-norm), or sums across rows (infinity-norm) and columns (1-norm), or element-wise products (Frobenius norm). These are expensive operations. Thankfully, the Taylor series in Equation (171) has infinite convergence radius, and therefore only an *estimate* of the norm is actually required. The cheapest estimate we could find was published for 1-

norm by Hager [116]. It requires a few matrix-vector products; usually fewer than the subsequent evaluation of Equation (171).

A less problematic, but important observation is that the algorithm described in the previous section, particularly at the dimension reordering stages, has complicated memory access patterns. Non-sequential memory access can be expensive on modern CPU architectures. The algorithm described in the previous section may therefore be viewed as a different trade-off between capacity and latency requirements on the RAM. However, the savings in the arithmetical operation count are always significant.

The last hurdle is that a polyadic object is sometimes pre-multiplied or post-multiplied by diagonal matrices with unknown polyadic structures. Because the only operation the entire object needs to deliver is matrix-vector product, the best strategy is to simply buffer the multipliers, and apply them to the incoming vector before and after the polyadic is applied. The object structure is then extended as follows:

$$\begin{gathered}\mathbf{P}_1\cdot\ldots\cdot\mathbf{P}_N\cdot\left[\mathbf{A}\otimes\mathbf{B}+\mathbf{C}\otimes\mathbf{D}\otimes\mathbf{E}+\ldots\right]\cdot\mathbf{Q}_1\cdot\ldots\cdot\mathbf{Q}_M\\ \Updownarrow\\ \left\{\mathbf{P}_1,\ldots,\mathbf{P}_N\right\}\left\{\left\{\mathbf{A},\mathbf{B}\right\},\left\{\mathbf{C},\mathbf{D},\mathbf{E}\right\},\ldots\right\}\left\{\mathbf{Q}_1,\ldots,\mathbf{Q}_M\right\}\end{gathered}\tag{176}$$

The entire product sequence is simply replayed from right to left every time a product into a vector is needed. Once the elements of this object are themselves allowed to be polyadic, the object can buffer an arbitrary sequence of additions, multiplications and Kronecker products. Importantly, this representation is only computationally efficient when the number of terms is much smaller than the dimension of the matrix it represents, but this is always the case for MRI evolution generators.

It should be noted that more sophisticated representations for tensor structured objects have existed for some time [117, 118]. We do have an MPS/TT object in *Spinach* [119], but it is certainly not a simple machine-precision parameter-free black box of the same kind as polyadics. Because the polyadic decomposition in Equation (143) is known *a priori*, and is nearly always already canonical, the benefits of going to compressed tensor structured formats such as TT [117] are not worth having to deal with their harrowing logistics.

The objects discussed above were implemented in *Spinach.* As a result, it currently contains a very general implementation of the Fokker-Planck formalism that is able to treat 3D diffusion and hydrodynamics simultaneously with Liouville-space spin dynamics, relaxation and chemical kinetics. This is dictated by the direction in which the field is moving: many emerging magnetic resonance methods (ultrafast NMR, singlet state imaging, spatially encoded NMR, metabolite-selective MRI, catalyst MRI,

hyperpolarised imaging, *etc.*) fall in between the established simulation frameworks: on the one hand, they require accurate simulation of three-dimensional diffusion, hydrodynamics and chemical processes; on the other, it is essential that the spin evolution in every molecule is treated quantum mechanically in a way that accurately describes spin relaxation processes. At the same time, sophisticated spatially and temporally modulated radiofrequency pulses must be accounted for. Within the Fokker-Planck formalism, things like three-dimensional diffusion, hydrodynamics and off-resonance radiofrequency appear in a very simple way – each of these is just another constant matrix to add to the background evolution Hamiltonian.

3.4 Synthetic benchmarks

Direct product component matrices for synthetic benchmarks were filled with normally distributed complex random numbers. The dimension of each matrix was chosen randomly between 1 and 64, and a random complex vector was generated to match the Kronecker product dimension. For sparse matrices, an optimistic non-zero density estimate was used: five complex non-zeroes per column. The matrix-vector multiplication operation was timed for 100 instances of this setup; the resulting statistics is presented in Table 1. The source code for this benchmark is available in versions 2.3 and later of Spinach library [69]

***Table 4**. Wall clock time benchmarks for matrix-vector multiplication using a single Xeon E5-2698 CPU core in Matlab on a computer equipped with 256 GB of RAM.*

Matrix-vector multiplication task	Wall clock time, polyadic	Wall clock time, flat
$\left[\mathbf{A}\otimes\mathbf{B}\right]\mathbf{v}$ with $\dim\left(\mathbf{A},\mathbf{B}\right)\leq 64$, full	0.37 ± 0.01 ms	0.88 ± 0.12 ms
$\left[\mathbf{A}\otimes\mathbf{B}\otimes\mathbf{C}\right]\mathbf{v}$ with $\dim\left(\mathbf{A}\text{-}\mathbf{C}\right)\leq 64$, full	1.8 ± 0.3 ms	Out of RAM
$\left[\mathbf{A}\otimes\mathbf{B}\otimes\mathbf{C}\otimes\mathbf{D}\right]\mathbf{v}$ with $\dim\left(\mathbf{A}\text{-}\mathbf{D}\right)\leq 64$, full	97 ± 14 ms	Out of RAM
$\left[\mathbf{A}\otimes\mathbf{B}\right]\mathbf{v}$ with $\dim\left(\mathbf{A},\mathbf{B}\right)\leq 64$, sparse	0.21 ± 0.01 ms	0.05 ± 0.01 ms
$\left[\mathbf{A}\otimes\mathbf{B}\otimes\mathbf{C}\right]\mathbf{v}$ with $\dim\left(\mathbf{A}\text{-}\mathbf{C}\right)\leq 64$, sparse	2.1 ± 0.3 ms	11.4 ± 1.6 ms
$\left[\mathbf{A}\otimes\mathbf{B}\otimes\mathbf{C}\otimes\mathbf{D}\right]\mathbf{v}$ with $\dim\left(\mathbf{A}\text{-}\mathbf{D}\right)\leq 64$, sparse	105 ± 16 ms	Out of RAM

It is clear that 3D MRI experiments with quantum mechanical description of spin cannot be simulated on a system with 256 GB of RAM – this confirms the estimates given in the *Introduction*. However, multiplications using the polyadic object are all in the milliseconds; they would remain realistic even if hundreds of terms are present in the sum in Equation (176). There is no wall clock time advantage,

but the memory problem is solved. From the computational feasibility point of view, the polyadic object puts quantum mechanical 3D MRI simulations within reach.

3.5 Numerical accuracy conditions

Diffusion and flow operators on finite grids have two popular matrix representations: finite difference matrices /ref/ and spectral differentiation matrices /ref/. The accuracy of the former depends on the grid spacing and the choice of the finite difference coefficients; both methods require all spatial frequencies to be below the Nyquist frequency of the grid.

On the Nyquist condition side, for the expected cases of laminar flow and diffusion, the dominant source of high frequencies along spatial coordinates are pulsed field gradients. An accurate simulation is the one that guarantees correct treatment of diffusive and hydrodynamic transport for the tightest phase spiral that the pulse sequence can generate in any element of the spin density matrix. Because this includes unobservable as well as observable coherences, the upper bound on the highest spatial frequency must assume that each gradient, even if intended to refocus some spin states, would continue defocusing some other states. Therefore, in each spatial direction:

$$\Omega_{\max} \leq \frac{L}{2B_0}\left\|\mathbf{H}_{\mathrm{Z}}\right\|_2 \int \left|g\left(t\right)\right| dt \qquad (177)$$

where L is the length of the sample in the chosen dimension, pulsed field gradients are assumed to be zero in the middle, B_0 is the magnet induction, $\left\|\mathbf{H}_{\mathrm{Z}}\right\|_2$ is the 2-norm (largest absolute eigenvalue) of the Zeeman Hamiltonian, $g\left(t\right)$ is the gradient amplitude in the chosen dimension as a function of time, and the integral is taken over the duration of the experiment. The grid point spacing h is limited by the Nyquist condition [120]to have more than two grid points per period of $\Omega_{\max}$:

$$\Omega_{\max} h < \pi \qquad (178)$$

For sequences that make extensive use of gradients, the bound in Equation (177) may be overly cautious. An alternative is to perform spatial Fourier transforms after each gradient event in the pulse sequence, and to inspect the edges of the frequency domain representation. Non-zero amplitude at the edges would indicate that the number of points in the spatial grid must be increased.

Equation (178) is the only accuracy condition when Fourier spectral differentiation matrices are used. However, they are expensive because they are not sparse. For diffusion and flow, it is therefore common to see finite difference matrices used instead; they have further accuracy conditions on the grid

point spacing. These conditions follow from Taylor series expressions for derivatives on finite grids. For the simplest example of the central first derivative:

$$\begin{aligned} f\left(x+h\right) &= f\left(x\right)+f'\left(x\right)h+\frac{f''\left(x\right)}{2}h^2+\int_x^{x+h} f'''\left(t\right)\frac{\left(x+h-t\right)^2}{2}dt \\ f\left(x-h\right) &= f\left(x\right)-f'\left(x\right)h+\frac{f''\left(x\right)}{2}h^2+\int_x^{x-h} f'''\left(t\right)\frac{\left(x-h-t\right)^2}{2}dt \end{aligned} \tag{179}$$

where Lagrange's remainders are used after second order. Solving this for $f'\left(x\right)$ yields:

$$f'\left(x\right)=\frac{f\left(x+h\right)-f\left(x-h\right)}{2h}+R\left(x,h\right) \tag{180}$$

The functions being differentiated are dominated by gradient spirals with the worst-case frequency obtained in Equation (177). The worst case is therefore $f\left(x\right)=\exp\left(-i\Omega_{\max}x\right)$. This permits a more detailed examination of the remainder $R\left(x,h\right)$ in Equation (180). After taking the integrals and simplifying the result, we obtain the following expression for the relative error when $\left|\Omega_{\max}h\right|<1$:

$$\left|\frac{R\left(x,h\right)}{f'\left(x\right)}\right|=1-\frac{\sin\left(\Omega_{\max}h\right)}{\Omega_{\max}h}\approx\frac{\left(\Omega_{\max}h\right)^2}{6}\ll 1 \tag{181}$$

This condition is more stringent than Equation (178) and necessitates a finer grid; this is the price to pay for the sparsity of the differentiation operators.

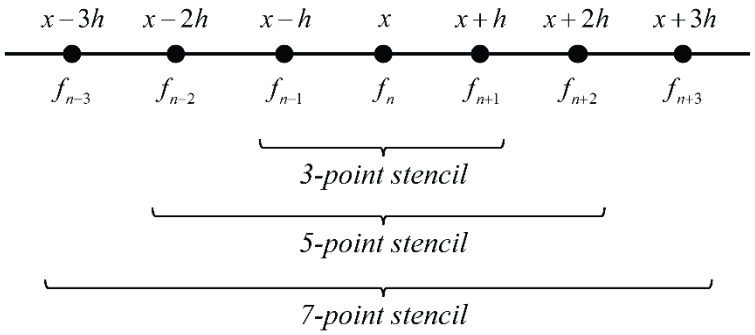


Figure 20. *A schematic illustration of the central finite-difference stencils implemented in the coordinate derivative module of Spinach.*

A variety of other finite difference schemes (**Figure 20**) are available in the literature [83], usually accompanied by the accuracy estimate of the form $O\left[h^n\right]$ where h is the grid spacing and n is a small integer. By following the same procedure as the one in Equations (179)-(181), it is easy to obtain

a more accurate estimate of the worst-case relative error on the derivative. In our context, it is $O\left[\left(\Omega_{\max} h\right)^n\right]$ where $\Omega_{\max}$ is the maximum spatial frequency that occurs in the experiment. In practice, it is reasonable to start from $\left|\Omega_{\max} h\right| \ll 1$, and to make the grid finer until the simulation result stops changing.

A good practical example of this is the ideal Stejskal-Tanner experiment [121], for which the analytical expression for the signal attenuation due to isotropic diffusion is known:

$$S(g) = S(0) e^{-\gamma^2 \delta^2 g^2 D(\Delta - \delta/3)} \tag{182}$$

where $S(g)$ is the signal intensity in an experiment with the gradient amplitude g, γ is the magnetogyric ratio (including any applicable chemical shift corrections) of the working spins, δ is the duration of the gradient pulse, and D is the diffusion coefficient. Running a numerical Stejskal-Tanner simulation on a finite grid and then fitting Equation (182) to extract the diffusion coefficient provides an illustration of how numerical accuracy depends on the grid spacing and the stencil size.

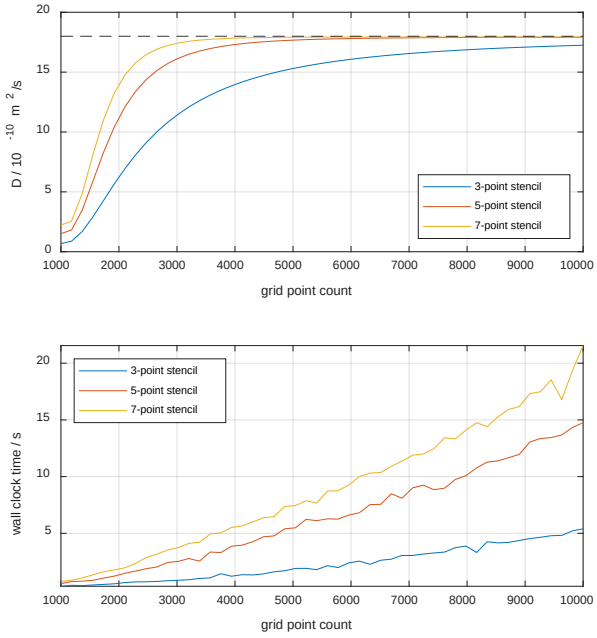


Figure 21. *Accuracy and run time statistics for a Stejskal-Tanner pulse sequence simulation followed by extraction of the known diffusion coefficient (18×10^{-10} m^2/s, dashed line) from the simulated data. The 1.5 cm long sample contains a single type of protons with a chemical shift of 4.6 ppm at 11.74 Tesla. The Stejskal-Tanner pulse sequence simulation uses ideal radiofrequency pulses; perfectly rectangular gradients are assumed with no stabilisation delay. The duration of gradient pulses is δ=2 ms, and the duration of the diffusion delay is Δ=50 ms. Gradient amplitudes are varied from zero to 0.5 Tesla/m.* ***Top:*** *the diffusion coefficient extracted by fitting Equation (182) to the simulated data for three finite difference stencil sizes as a function of the spatial grid size.* ***Bottom:*** *wall clock time (2 x Intel Xeon E5-2698) for the pulse sequence simulation for three finite difference stencil sizes as a function of the spatial grid size. The stencil size refers to the number of points used in the finite difference approximation to the spatial coordinate derivative operator.*

Figure 21 illustrates quantitatively the dilemma mentioned in the *Introduction*: accurate simulations of diffusion NMR experiments require thousands of points in each spatial dimension. Even with sparse matrix arithmetic, three-dimensional samples with complicated spin systems and chemical processes are beyond modern computers unless Kronecker products are left un-opened in equation (173). The minimum grid satisfying the Nyquist condition for spatial frequencies in Figure 2 has 1,280 points; Fourier spectral derivative operators become accurate from that size onwards. However, the wall clock time is actually much longer with a 2,000-point Fourier differentiation operator than with a 10,000-point finite difference operator because Fourier operators are not sparse.

The same accuracy analysis may be viewed from a different perspective: as the maximum accumulated gradient winding that a given spatial grid can support. This is illustrated in Figure 3, which shows a comparison between signal attenuation observed in the simulation and the exact analytical solution as a function of the grid size and the differentiation stencil.

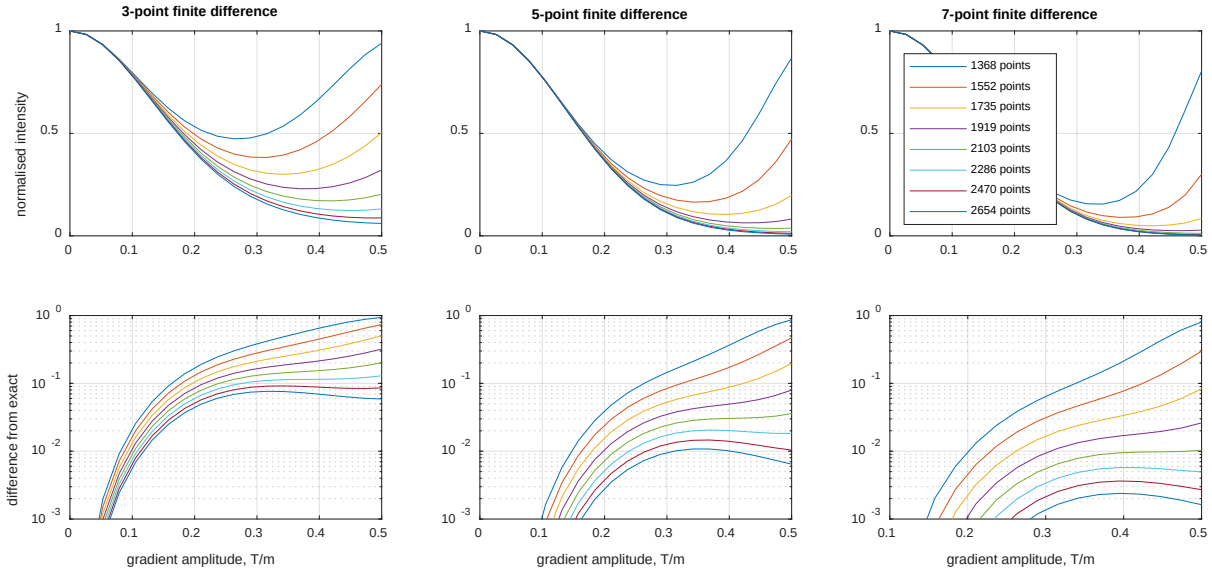


Figure 22. *Simulated Stejskal-Tanner diffusion attenuation profiles as a function of spatial grid size. The 1.5 cm long sample contains a single type of protons with a chemical shift of 4.6 ppm at 11.74 Tesla. The Stejskal-Tanner pulse sequence simulation uses ideal radiofrequency pulses; perfectly rectangular gradients are assumed with no stabilisation delay. The duration of gradient pulses is δ=2 ms, and the duration of the diffusion delay is Δ=50 ms. The diffusion coefficient is 18×10⁻¹⁰ m²/s.* ***Top row:*** *diffusion attenuation profiles for spatial grids and finite difference stencils of different sizes (the minimal grid that satisfies the spatial Nyquist condition in this system has 1280 points). Bottom row: difference between the simulated diffusion attenuation profiles and the exact analytical answer for grids and finite difference stencils of difference sizes.*

The simulation monotonically becomes more accurate as the grid point count increases. The practical conclusion again is that it is reasonable to use a seven-point stencil, to start from about twice the point count dictated by the Nyquist condition, and to increase the point count until the simulation result stops changing to the accuracy required. Similar conclusions were reached for other diffusion NMR pulse sequences: pulsed gradient spin echo [122], pulsed gradient stimulated echo [123], bipolar pulse pair stimulated echo [124] and DOSY ONESHOT[125].

An important consideration is the type of the boundary condition. Fourier differentiation matrices require periodic boundaries, and finite difference operators for diffusion and flow all have a particularly simple form in periodic boundary conditions. The logistics of implementing other types of boundaries in a general simulation package is formidable, and therefore only the periodic boundaries are supported in version 2.3 of Spinach. In practice this means that sufficient margin must be left on either side of the sample concentration phantom to prevent diffusion and flow processes from folding over.

3.6 Numerical efficiency

3.6.1 Computationally efficient time propagation

The methods listed in Section 2.5 can be applied *a priori* in order to decide which spin correlations would in practice contribute to the dynamics. However, when the equations of motion are integrated in the resulting matrix-vector representation, there are further efficiency opportunities – problem dimension may be squeezed further by looking at the actual trajectory. These "trajectory level" methods [62] (implemented in *Spinach* 2.3 and later [69]) include:

1. Krylov subspace analysis: all simulation trajectories have finite duration and finite step length. The space spanned by the trajectory vectors is known as the Krylov subspace of the propagator matrix [126, 127]. The dimension of that subspace (*i.e.* the number of linearly independent vectors in its basis) can in favourable cases be orders of magnitude smaller than the full state space dimension [62]. The minimal basis can only be generated *a posteriori*, after the full simulation is run, but it can facilitate subsequent calculations with similar physical parameters, for example for nearby orientations in a powder average.

2. Zero track elimination: if a particular state has remained unpopulated within a certain initial period in the system evolution, rigorous bounds can be placed on its subsequent contribution to the system dynamics [62]. States that remain unpopulated up to a user-specified tolerance can be dropped, and subsequent steps in the simulation would proceed faster. The dropped states can be re-introduced with their original indices and zero amplitudes if the return to the original basis is required for the next simulation stage.

3. Liouvillian path tracing: spin dynamics generators are always very sparse [128]. The corresponding connectivity graphs between spin states often contain disconnected subgraphs that correspond to non-interacting subspaces [66] induced by unobvious or accidental symmetries and conservation laws. They may be simulated separately. Tarjan's graph partitioning algorithm [129] scales linearly with respect to the number of non-zeroes in the generator, but the resulting block-diagonalization that provides cubic benefit in complexity.

4. Destination state screening: detection states in magnetic resonance are very simple – typically some form of transverse magnetisation. It is therefore advantageous to perform a part of the simulation in reverse – by starting from the detection state, going backwards in time, and meeting the evolving initial condition in the middle. This is possible even with dissipative dynamics [67]. The reduced state space for a simulation is the intersection of the reduced state spaces obtained from forward and backward propagation – that intersection is usually smaller than either of its operands [61].

We will not touch upon the technical matters of implementing all of the above, noting only that this was done [69], are set out in the papers cited above. Fully quantum mechanical simulations of protein-size spin systems became possible as a result [67].

3.6.2 Optimal time stepping

Consider the task of propagating a state vector $\boldsymbol{\rho}(0)$ for a time $T>0$ under a sparse and possibly dissipative time-independent evolution generator $\mathbf{L}$ of dimension N, where the final state vector $\boldsymbol{\rho}(T)$ is required with machine precision accuracy on both phases and amplitudes. The solution:

$$\boldsymbol{\rho}(T)=\exp(-i\mathbf{L}T)\boldsymbol{\rho}(0) \tag{183}$$

has significant efficiency caveats in situations when $\|-i\mathbf{L}T\|\gg 1$. In the treatment below, the following common magnetic resonance settings shall be assumed:

1. The dimension of $\mathbf{L}$ is too large for diagonalisation or inversion. This, and the presence of dissipative terms, rules out Padé, Chebyshev, and Newton exponentiation methods, even in hypothetical cases where they could be superior to the scaled and squared Taylor series.

2. The cost of a matrix-matrix product is N^{α} multiplications, where $\alpha\leq 3$.

3. The cost of a matrix-vector product is N^{β} multiplications, where $\beta\leq 2$.

4. Taylor series procedure with scaling and squaring [130] is used to compute the matrix exponential; the scaling is done using Hager's estimate [116] of the 1-norm.

With these settings in place, Equation (183) transforms into:

$$\boldsymbol{\rho}(T)=\left[\exp(-i\mathbf{L}\Delta t)\right]^{2^n}\boldsymbol{\rho}(0),\qquad \Delta t=T/2^n,\qquad n=\operatorname{ceil}\left[\log_2\left(T\|\mathbf{L}\|_1\right)\right] \tag{184}$$

where n is the number of propagator squaring operations that is dictated by the numerical accuracy condition for the Taylor series – that all eigenvalues of $-i\mathbf{L}\Delta t$ be scaled into the unit circle to make

sure that there is no hump [130]. Estimates of the largest eigenvalue are expensive, but the more easily estimated 1-norm [116] is an upper bound.

The optimisation problem is then created by the following:

1. A matrix-vector product is cheaper than a matrix-matrix product. Some of the exponentials in Equation (184) may be sequentially multiplied into $\boldsymbol{\rho}(0)$ rather than squared.

2. Propagator squaring increases the time step exponentially, whereas propagator-vector product only takes the system forward at a rate that is linear with respect to the number of matrix-vector products.

An optimal point must therefore exist between the number of propagator squaring operations and the number of propagator-vector products; the cost functional is the total number of multiplications.

The cost of 1-norm estimation and scaling is $5N^{\beta}$ multiplications [116]. The cost of computing the Taylor series indexed by j and involving a matrix with eigenvalues scaled inside the unit circle is determined by the upper bound on the magnitude of the last term, which is $1/j!$, meaning that

$$j_{\max}=\operatorname{ceil}\left\{\Gamma^{-1}\left(1/\varepsilon\right)-1\right\} \tag{185}$$

where $\Gamma(x)$ is Euler's gamma function and $j_{\max}\approx 18$ for double precision arithmetic. The number of matrix-matrix products in the Taylor series procedure is therefore $j_{\max}$ and the total cost of computing the Taylor series is $j_{\max}N^{\alpha}$ multiplications. The cost of squaring the propagator to compensate for the scaling of $-i\mathbf{L}\Delta t$ is nN^{α} multiplications.

If we decide not to do m propagator squaring operations and multiply the propagator into the state vector 2^m times instead, the total cost will be:

$$\Omega(m)=\left[j_{\max}+n-m\right]N^{\alpha}+2^{m}N^{\beta}k+5N^{\beta} \tag{186}$$

multiplications, where k is the number of state vectors in $\boldsymbol{\rho}(0)$ – there may be several. The minimum of this function with respect to m is:

$$\frac{d}{dm}\left(\left[j_{\max}+n-m\right]N^{\alpha}+2^{m}N^{\beta}l\right)=0 \quad\Rightarrow\quad m=\log_2\left(\frac{N^{\alpha-\beta}}{k\ln 2}\right) \tag{187}$$

It does not depend on the norm of the Liouvillian, the cost of its calculation, or machine precision. While α and β strongly depend on the sparsity of the matrix, their difference does not: $\alpha-\beta\approx 1$.

Imposing the non-negative integer constraint produces the optimal number of squaring operations to skip:

$$m_{\text{opt}} = \max\left\{0 \quad \text{ceil}\left[\log_2\left(N^{\alpha-\beta}/k\ln 2\right)\right]\right\} \tag{188}$$

where ceiling is preferred to the floor because sparse matrices fill up when multiplied, and each subsequent propagator squaring will be more expensive than the previous one. Equation (188) is used in *Spinach* [69] whenever the explicit matrix representation of the evolution generator is available.

4 Fokker-Planck models of MRI

4.1 Introduction

This chapter describes the MRI simulation tools and examples that were implemented into *Spinach* as a part of this project. The reader is encouraged to look through the rather large body of source code in the imaging module of *Spinach* – most of it was written within this project.

All previous MRI simulation packages appear to be based on Bloch equations [2] or Bloch-Torrey equations [1]. Many magnetic resonance imaging simulators are very well developed [49, 52, 131, 132], but still insufficient because they cannot handle coupled spin systems. On the NMR and EPR side, there is also a great variety of packages that, however, cannot handle the necessary spatial dynamics. The first quantum mechanics software package was called NMR-SCOPE [133]. It implemented product operator description coupled spin systems. It was followed by GAMMA [58] written in C++ programming language. Many packages appeared afterwards, such as SIMPOSN [134], BlochLib [134], SPINEVOLUTION [56], SIMPLTN [135], NMRSIM[136], SpinDynamica [137], and Spinach [69].

I would argue that the implementation presented here is currently unique in combining sophisticated quantum mechanical description of the entirety of magnetic resonance with a sophisticated description of three-dimensional coherent and dissipative dynamics.

4.2 MRI pulse sequence implementation

This section goes through the algebraic side of Spinach MRI module code implemented during this project. The corresponding source code is extensively annotated and should be read alongside. We begin with Fokker-Planck equation:

$$\begin{gathered}\frac{\partial\boldsymbol{\rho}(\mathbf{r},t)}{\partial t}=-i\mathbf{L}(\mathbf{r},t)\boldsymbol{\rho}(\mathbf{r},t)+\mathbf{M}(\mathbf{r},t)\boldsymbol{\rho}(\mathbf{r},t)=-i\mathbf{F}(\mathbf{r},t)\boldsymbol{\rho}(\mathbf{r},t)\\ \mathbf{L}(\mathbf{r},t)\boldsymbol{\rho}(\mathbf{r},t)=\left[\mathbf{H}(\mathbf{r},t),\boldsymbol{\rho}(\mathbf{r},t)\right]+i\mathbf{R}\boldsymbol{\rho}(\mathbf{r},t)+i\mathbf{K}\boldsymbol{\rho}(\mathbf{r},t)\end{gathered}\tag{189}$$

where $\boldsymbol{\rho}(\mathbf{r},t)$ is the spin density matrix at the spatial point $\mathbf{r}$ at time t, $\mathbf{H}(\mathbf{r},t)$ is the spin Hamiltonian, square brackets denote a commutator, $\mathbf{R}$ is the thermalized relaxation superoperator, $\mathbf{K}$ is the chemical kinetics superoperator, and $\mathbf{M}(\mathbf{r},t)$ is the spatial dynamics generator dealing with diffusion and flow [81].

In liquid state spatially encoded NMR spectroscopy and imaging simulations, the laboratory frame spin Hamiltonian in Equation (247) contains the terms describing chemical shifts, pulsed field gradients, radiofrequency pulses and *J*-couplings [26]:

$$\begin{aligned}\mathbf{H}(\mathbf{r},t)=&-\left[B_0+\mathbf{g}^{\mathrm{T}}(t)\cdot\mathbf{r}\right]\sum_n(1+\delta_n)\gamma_n\mathbf{S}_{\mathrm{Z}}^{(n)}-B_{1\mathrm{X}}(t)\sum_n(1+\delta_n)\gamma_n\mathbf{S}_{\mathrm{X}}^{(n)}-\\&-B_{1\mathrm{Y}}(t)\sum_n(1+\delta_n)\gamma_n\mathbf{S}_{\mathrm{Y}}^{(n)}+2\pi\sum_{n<k}J_{nk}\left[\mathbf{S}_{\mathrm{X}}^{(n)}\mathbf{S}_{\mathrm{X}}^{(k)}+\mathbf{S}_{\mathrm{Y}}^{(n)}\mathbf{S}_{\mathrm{Y}}^{(k)}+\mathbf{S}_{\mathrm{Z}}^{(n)}\mathbf{S}_{\mathrm{Z}}^{(k)}\right]\end{aligned}\tag{190}$$

where B_0 is the primary magnet field (assumed to be directed along the Z axis), $\mathbf{g}$ is the primary magnet field gradient vector, δ_n are nuclear chemical shifts, γ_n are nuclear magnetogyric ratios, $\left\{\mathbf{S}_{\mathrm{X}}^{(n)},\mathbf{S}_{\mathrm{Y}}^{(n)},\mathbf{S}_{\mathrm{Z}}^{(n)}\right\}$ are nuclear spin operators, J_{nk} are inter-nuclear scalar couplings (traditionally published in Hz, hence the 2π in front) and $B_{1\{\mathrm{X,Y}\}}$ are the Cartesian components of the radiofrequency magnetic field, which is assumed to have no Z component.

The spatial dynamics generator in Equation (247) has contributions from diffusion and flow [81]:

$$\mathbf{M}(\mathbf{r},t)=\nabla^{\mathrm{T}}\cdot\mathbf{v}(\mathbf{r},t)+\mathbf{v}^{\mathrm{T}}(\mathbf{r},t)\cdot\nabla+\nabla^{\mathrm{T}}\cdot\mathbf{D}(\mathbf{r},t)\cdot\nabla\tag{191}$$

where $\nabla=\left[\partial/\partial x\quad\partial/\partial y\quad\partial/\partial z\right]^{\mathrm{T}}$ is the gradient (in the mathematical sense) operator, $\mathbf{v}(\mathbf{r},t)$ is the flow velocity and $\mathbf{D}(\mathbf{r},t)$ is the translational diffusion tensor.

On a finite grid, the Fokker-Planck spin Hamiltonian has a block structure, with a different spin Hamiltonian in each voxel block:

$$\begin{pmatrix}\mathbf{H}_1 & & \\ & \ddots & \\ & & \mathbf{H}_N\end{pmatrix}\tag{192}$$

where N is the total number of voxels. Spin Hamiltonian generation is handled by the existing modules of *Spinach*, and the spatial dynamics generators are obtained as described above, and the initial condition is concentration-weighted longitudinal magnetisation:

$$\boldsymbol{\rho}(0)=\begin{pmatrix}c_1\\ \vdots\\ c_N\end{pmatrix}\otimes\mathbf{L}_{\mathrm{z}}\tag{193}$$

To make a functioning imaging simulation, we must add spatially variable relaxation to the dynamics generator in equation (189). This is accomplished using a polyadic decomposition:

$$\mathbf{F} = \ldots + \sum_k \mathbf{Ph}_k \otimes \mathbf{R}_k + \ldots \tag{194}$$

Where $\mathbf{F}$ is the total Fokker-Planck evolution generator from Equation (189), $\mathbf{R}_k$ are the relaxation superoperators responsible for the individual relaxation processes (longitudinal, transverse, *etc.*), and $\mathbf{Ph}_k$ are the corresponding "phantoms" – three-dimensional cubes of data describing the amplitude of each particular relaxation mechanism at each voxel in the sample.

From this point on, the generality of the Fokker-Planck formalism shows its power and the general solution to equation (189) is very simple:

$$\boldsymbol{\rho}(t+dt) = \exp\left[-i\mathbf{F}(t)\,dt\right]\boldsymbol{\rho}(t) \tag{195}$$

4.2.1 Slice selection and echo detection

The sequence intended to diagnose slice selection is shown in **Figure 23** and the simulation parameters shown in **Table 5**. It involves a shaped pulse under the slice selection gradient and then a 1D spin echo readout using frequency encoding.

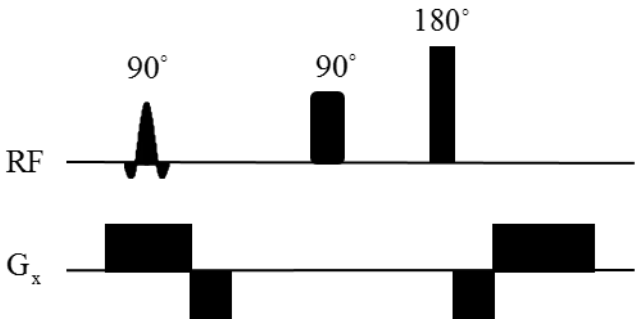


Figure 23. *One dimensional slice selection MRI sequence followed by conventional spin echo.*

Table 5. Slice selection simulation parameters

Isotope	1H
Pulse steps	50
Pulse time	0.05 ms
Pulse frequency	100 kHz
Gradient amplitude	6 mT/m
Flow rate	10 mm/s
Diffusion coefficients	$5*10^{-6}$ m^2/s

It is a good illustration of the amount of programming and software engineering done within this project to go though the corresponding sequence implementation line by line.

We start by specifying the isotope:

```
% Isotopes
sys.isotopes={'1H'};
```

Then the applied magnet field in Tesla:

```
% Magnetic induction
sys.magnet=5.9;
```

Liouville space formalism:

```
% Formalism
bas.formalism='sphten-liouv';
```

Complete basis set, since the system only contains one spin:

```
% Basis set
bas.approximation='none';
```

Spinach housekeeping functions:

```
% Spinach housekeeping
spin_system=create(sys,inter);
spin_system=basis(spin_system,bas);
```

Relaxation theory phantom – no relaxation:

```
% Relaxation phantom
parameters.rlx_ph={zeros(parameters.npts,1)};
parameters.rlx_op={relaxation(spin_system)};
```

Initial condition phantom – Lz everywhere:

```
% Initial state phantom
parameters.rho0_ph={ones(parameters.npts,1)};
parameters.rho0_st={state(spin_system,'Lz','1H','cheap')};
```

Detection state phantom – L+ everywhere:

```
% Detection state phantoms
```

```
parameters.coil_ph={ones(parameters.npts,1)};
parameters.coil_st={state(spin_system,'L+','1H','cheap')};
```

X component of the velocity vector for each point in the sample:

```
% Flow
parameters.u=1e-2*ones(parameters.npts,1);
```

Diffusion constant for each point in the sample:

```
% Diffusion
parameters.diff=5e-6;
```

The previous parts are exactly the same for the Gaussian pulse and the square pulse. However, the 'Pulse parameters' block is different between these two examples. Gaussian pulse parameters:

```
% Pulse parameters
pulse_nsteps=50; pulse_time=0.5e-4;
pulse_frq=+100e3; pulse_pwr=2*pi*20000;
parameters.rf_phi=pi/2;
parameters.rf_frq_list=pulse_frq*ones(1,pulse_nsteps);
parameters.rf_dur_list=(pulse_time/pulse_nsteps)*ones(1,pulse_nsteps);
parameters.max_rank=3;
parameters.rf_amp_list=pulse_pwr*pulse_shape('gaussian',pulse_nsteps);
time_grid=linspace(-2,2,npoints);
waveform=normpdf(time_grid)/sqrt(2);
```

Square pulse parameters:

```
% Pulse parameters
pulse_nsteps=50; pulse_time=0.5e-4;
pulse_frq=+100e3; pulse_pwr=2*pi*20000;
parameters.rf_phi=pi/2;
parameters.rf_frq_list=pulse_frq*ones(1,pulse_nsteps);
parameters.rf_dur_list=(pulse_time/pulse_nsteps)*ones(1,pulse_nsteps);
parameters.max_rank=3;
parameters.rf_amp_list=pulse_pwr*pulse_shape('rectangular',pulse_nsteps);
time_grid=linspace(-2,2,npoints);
waveform=ones(1,npoints);
```

Slice selection pulse syntax is the same in both cases:

```
% Slice selection pulse
parameters.rho0=shaped_pulse_af(spin_system,L+parameters.ss_grad_ampl*G{1},Lx,Ly,...
               parameters.rho0,parameters.rf_frq_list,parameters.rf_amp_list,...
               parameters.rf_dur_list,parameters.rf_phi,parameters.max_rank);
```

We then move to the edge of the *k*-space:

```
% Rephasing gradient
parameters.rho0=evolution(spin_system,L-parameters.ss_grad_ampl*G{1},[],...
                         parameters.rho0,sum(parameters.rf_dur_list)/2,1,'final');
```

and call the one dimensional basic hard imaging sequence for acquisition:

```
% Call to hard imaging
fid=basic_1d_hard(spin_system,parameters,H,R,K,G,F);
```

The resulting fid is Fourier transformed and plotted:

```
mri=real(fftshift(fft(ifftshift(fid))));
figure(); plot_1d(spin_system,mri,parameters);
```

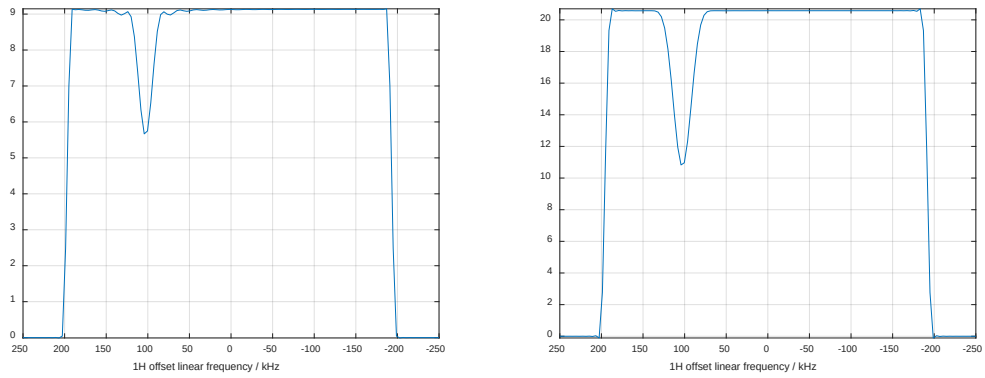


Figure 24. *One-dimensional slice selection illustration using slice selection diagnostics function. The left figure was obtained using a rectangular shaped pulse while the right figure used a Gaussian shaped pulse. The simulation includes soft radiofrequency pulse, a field gradient, as well as identical amounts of flow and diffusion. The flow rate was is 0.01 m/s and the diffusion coefficient equal to 5 mm²/s. The pulse sequence is illustrated in* ***Figure 23****.*

The resulting slice selection diagnostics outputs are shown in **Figure 24**. It is clear that the accuracy of the calculation is rather high on the chosen spatial grid – even the sinc wobbles from the square pulse are visible. Note that a great amount of mathematics is hidden in the functions being called, for example shaped_pulse_af.m function. Those are the subject of other chapters of this thesis.

4.2.2 Phase and frequency encoded imaging

The simplest possible MRI pulse sequence for 3D imaging is shown in **Figure 10**. It involves repeated slice selection, followed by echo detection with frequency encoding. The code below walks through a simple MRI simulation for a three-dimensional brain phantom. Furthermore, the simulation parameters are listed in **Table 6.**

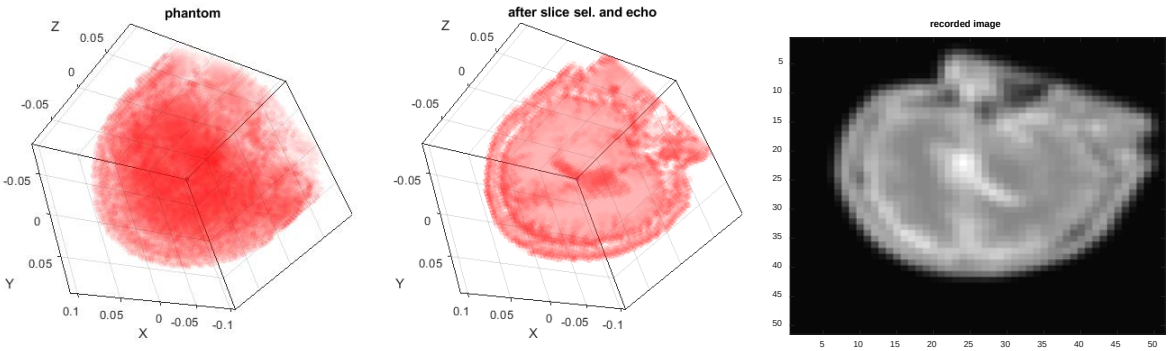


Figure 25. *The left panel shows a volumetric plot of a 3D brain phantom with a distribution in spin density and relaxation rates. Applying three spatial gradients and an explicit soft pulse to excite an arbitrary slice through the 3D volume results in the figure presented in the middle panel. The resulting image (right panel) is then acquired in a standard way.*

Specify the isotope:

```
% Isotopes
sys.isotopes={'1H'};
```

Specify the applied magnet field:

```
% Magnetic induction
sys.magnet=5.9;
```

Specify longitudinal and transverse relaxation rates:

```
% Relaxation model

inter.relaxation={'t1_t2'};
inter.rlx_keep='diagonal';
inter.equilibrium='zero';
inter.r1_rates=30;
inter.r2_rates=70;
```

Liouville space formalism:

```
% Basis set
bas.formalism='sphten-liouv';
```

Single spin system, hence no approximations:

```
bas.approximation='none';
```

Construct the spin system object:

```
% Spinach housekeeping
spin_system=create(sys,inter);
spin_system=basis(spin_system,bas);
```

Load the brain phantom

```
% Phantoms
[R1,R2]=rlx_t1_t2(spin_system);
[R1_Ph,R2_Ph,PD_Ph,dims,npts]=phantoms('brain-lowres');
```

Specify the relaxation phantoms as per Equation (194):

```
% Relaxation phantom
parameters.rlx_op={R1,R2};
parameters.rlx_ph={R1_Ph,R2_Ph};
```

The initial condition phantom reflects the proton density distribution:

```
% Initial state phantom
parameters.rho0_ph={PD_Ph};
parameters.rho0_st={state(spin_system,'Lz','1H','cheap')};
```

Uniform L+ for the detection state:

```
% Detection state phantoms
parameters.coil_ph={ones(prod(parameters.npts,1))};
parameters.coil_st={state(spin_system,'L+','1H','cheap')};
```

X, Y and Z components of the velocity vectors for each point in the sample:

```
% Flow
parameters.u=zeros(parameters.npts);
parameters.v=zeros(parameters.npts);
parameters.w=zeros(parameters.npts);
```

Diffusion coefficient for situations when this parameter is the same in every voxel:

```
% Diffusion
parameters.diff=0;
```

Slice selection pulse parameters:

```
% Pulse phase
parameters.rf_phi=pi/2;

% Number of steps in the pulse
pulse_nsteps=50;

% Overall pulse duration
pulse_time=2.0e-4;

% Slice selection pulse frequency table
parameters.rf_frq_list=-5e3*ones(1,pulse_nsteps);

% Slice selection pulse amplitude table
parameters.rf_amp_list=2*pi*7500*pulse_shape('gaussian',pulse_nsteps);

% Slice selection pulse duration table
parameters.rf_dur_list=(pulse_time/pulse_nsteps)*ones(1,pulse_nsteps);
```

Imaging sequence parameters, including slice selection, phase encoding and readout gradients amplitude and time required for echo:

```
 % Imaging parameters
parameters.image_size=[51 51];
parameters.ss_grad_amp=32.0e-3;  % T/m
parameters.ss_grad_dur=1.0e-4;
parameters.pe_grad_amp=25.0e-3;  % T/m
parameters.pe_grad_dur=2.0e-4;
parameters.ro_grad_amp=25.0e-3;  % T/m
parameters.ro_grad_dur=3.0e-4;
parameters.t_echo=20e-3;
```

Slice selection, phase encoding and readout gradient vectors. The three gradients must be orthogonal:

```
% Slice selection gradient vector
parameters.gs_ampl=[1 1 1]/sqrt(3);

% Phase encoding gradient vector
parameters.gp_ampl=[1 -2 1]/sqrt(6);

% Readout gradient vector
parameters.gr_ampl=[1 0 -1]/sqrt(2);
```

Draw the 3 dimensional phantom

```
% Draw the phantom
figure(); dims=zeros(1,6);
dims([1 3 5])=-parameters.dims;
dims([2,4,6])=+parameters.dims;
volplot(R1_Ph.^(1/16),dims);
title('phantom'); drawnow();
```

Call the 3D imaging pulse sequence:

```
% Select a slice
mri=imaging(spin_system,@slice_phase_enc,parameters);
```

We will now go through the code of that sequence. We start with the slice selection gradient operator and the corresponding pulse:

```
% Gradient operator for slice selection
Gs=(parameters.gs_ampl(1)*G{1}+parameters.gs_ampl(2)*G{2}+parameters.gs_ampl(3)*G{3});

% Apply the slice selection pulse
parameters.rho0=shaped_pulse_af(spin_system,L,Hx,Hy,parameters.rho0,...
                                parameters.rf_frq_list,parameters.rf_amp_list,...
                                parameters.rf_dur_list,parameters.rf_phi,2,'expv');
```

The phase must be restored after the slice selection gradient:

```
 % Rollback operator
L=B-parameters.ss_grad_amp*Gs;

% Run the rollback gradient
parameters.rho0=evolution(spin_system,L,[],parameters.rho0,...
                         sum(parameters.rf_dur_list)/2,1,'final');
```

We proceed to run the evolution during the echo time:

```
% Evolve for the echo time
parameters.rho0=evolution(spin_system,B,[],parameters.rho0,parameters.t_echo,1,'final');
```

Apply a 180° hard pulse and then evolve under echo time:

```
% Apply 180-degree pulse
parameters.rho0=step(spin_system,Hy,parameters.rho0,pi);

% Evolve for the echo time
parameters.rho0=evolution(spin_system,B,[],parameters.rho0,parameters.t_echo,1,'final');
```

Now that slice selection is complete, we draw the slice for diagnostic purposes:

```
 % Draw the slice
mri_slice=fpl2phan(parameters.rho0,state(spin_system,'L+','1H','cheap'),parameters.npts);
dims=zeros(1,6); dims([1 3 5])=-parameters.dims; dims([2,4,6])=+parameters.dims;
figure(); volplot(abs(mri_slice),dims); title('after slice sel. and echo'); drawnow();
```

Specify the phase encoding and readout gradients operators:

```
% Gradient operator for phase encoding
Ge=parameters.gp_ampl(1)*G{1}+parameters.gp_ampl(2)*G{2}+parameters.gp_ampl(3)*G{3};

% Gradient operator for readout
Gr=parameters.gr_ampl(1)*G{1}+parameters.gr_ampl(2)*G{2}+parameters.gr_ampl(3)*G{3};
```

Now go forward to spatial encoding which is defined as the signals are returned with a particular frequency and phase and these are slotted into their respective locations

```
% Preallocate the image
fid=zeros(parameters.image_size);

% Loop over frequencies
parfor n=1:parameters.image_size(2) %#ok<*PFBNS>
```

```
    % Phase encoding gradient
    rho=evolution(spin_system,B+pe_grad_amps(n)*Ge,[],parameters.rho0,...
                  parameters.pe_grad_dur,1,'final');

    % Get the timing parameters
    nsteps=parameters.image_size(1)-1; step_length=parameters.ro_grad_dur/nsteps;

    % Run the pre-roll gradient
    rho=evolution(spin_system,B-parameters.ro_grad_amp*Gr,[],rho,step_length/2,...
                  nsteps,'final');

    % Detect under the readout gradient
    fid(:,n)=evolution(spin_system,B+parameters.ro_grad_amp*Gr,...
                       parameters.coil,rho,step_length,nsteps,'observable');

end
```

Applying an apodization function:

```
% Apodization
fid=apodization(fid,'sqsinbell-2d');
```

Applying the Fourier transform:

```
% Fourier transform
mri=real(fftshift(fft2(ifftshift(fid))));
```

Plotting the result:

```
% Plotting
figure(); mri_plot_2d(mri); title('recorded image');
```

Table 6. *Phase and frequency encoding simulation parameters*

Isotope	1H
Echo time	20 ms
Slice selection gradient	32 mT/m
Phase encoding gradient	25 mT/m
Readout Gradient	30 mT/m

This is a simulation of a 100 x 100 x 100 voxel phantom with a three-spin system treated at the Liouville space level. As a result of the work presented in this thesis, the calculation takes only a few minutes.

4.2.3 Fast spin echo imaging

One of the important disadvantages in the imaging is the time required to collect the data. This time is at least in part due to the longitudinal relaxation that determines the time before the spin system can be excited again. Many researches tried to reduce the time of imaging [46], and the current record is about ten milliseconds [138].

The "fast spin echo" is a pulse sequence that has a shorter scan time in comparison with the conventional pulse sequence, which uses a 90° pulse followed by a 180° rephasing pulse and provides only one echo per transient – filling in only one line in *k*-space.

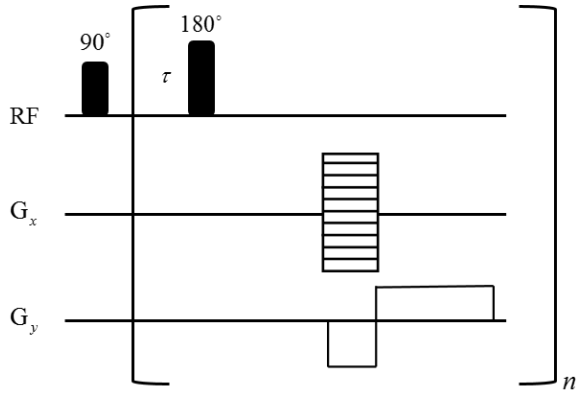


Figure 26. *Fast spin echo pulse sequence with τ delay and number of echoes n. image sampling representing by phase encoding gradient in x axis and frequency encoded in y axis. The picture assumes that slice selection is already done.*

In the fast spin echo pulse sequence, the excitation 90° pulse is followed by a 180° echo pulse train that provides multiple lines in the *k*-space – one for each echo. For as long as the T_2 relaxation has not completely destroyed the signal, it is possible to apply additional 180° pulses and generate additional echoes. The amplitude of each subsequent echo is smaller due to the T_2 decay. The role of the Gx gradient in **Figure 26** is to move in the *k*-space in the direction perpendicular to the readout direction.

Fast spin echo is nearly always preferred to conventional imaging in T_2 weighted MRI because more information may be collected per unit time, leading to better sensitivity and / or resolution. Another feature is that the chain of echoes partially refocuses the *J*-couplings with the result that fat remains bright. Muscles do however appear darker because of the transverse relaxation acceleration by the paramagnetic iron in myoglobin [41].

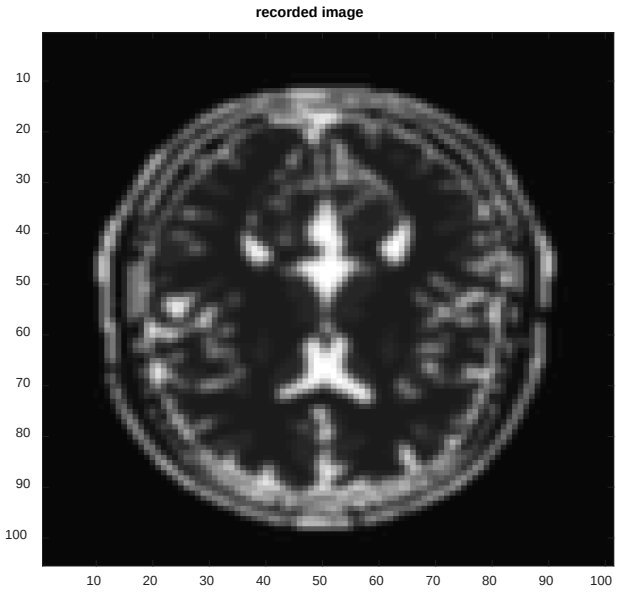


Figure 27. *MRI image of a brain phantom simulated using 2D fast spin echo pulse sequence. The simulation contains phase encoding gradient with 4.8 mT/m amplitude and 1 ms duration for the outer edge of k-space. The readout gradient was of 5.3 mT/m amplitude and 2 ms duration.*

As a part of this project, the fast spin echo sequence was implemented into *Spinach*. An example image is shown in **Figure 27** and the simulation parameters shown in **Table 7.**

Table 7. *Fast spin echo simulation parameters*

Isotope	1H
Phase encoding gradient	4.8 mT/m
Readout Gradient	5.3 mT/m
Phase encoding gradient duration	1 ms
Readout Gradient duration	2 ms

A walk through the implementation is given below:

We start with assembling the background Liouvillian that contains spin Hamiltonian, diffusion and flow, relaxation, and chemical kinetics:

```
% Assemble the background
B=H+F+1i*R+1i*K;
```

Pulse operators are made next and kroneckered up into the Fokker-Planck space:

```
% Make pulse operators
Lp=operator(spin_system,'L+',parameters.spins{1});
Ly=kron(speye(prod(parameters.npts)),(Lp-Lp')/2i);
```

The sequence starts with a 90-degree pulse

```
% Apply 90-degree pulse
rho=step(spin_system,Ly,parameters.rho0,pi/2);
```

A burst of readout gradient moves us to the left edge of the k-space:

```
% Move to left edge of k-space
rho=step(spin_system,B+parameters.ro_grad_amp*G{2},...
         rho,parameters.ro_grad_dur/2);
```

Some housekeeping calculations are done:

```
% Preallocate k-space image
fid=zeros(parameters.image_size);

% Phase encoding gradient amplitudes
pe_grad_amps=linspace(-parameters.pe_grad_amp,...
                       parameters.pe_grad_amp,...
                       parameters.image_size(2));
```

The outer loop of the sequence begins:

```
% Phase encoding loop
for n=1:parameters.image_size(2)
```

We apply a 180-degree pulse:

```
    % 180-degree pulse
    rho=step(spin_system,Ly,rho,pi);
```

We move to the current line of the k-space

```
    % Kick up the k-space
    rho=step(spin_system,B+pe_grad_amps(n)*G{1},...
             rho,parameters.pe_grad_dur);
```

And read out that line using Krylov propagation (matrix dimensions are expected to be large):

```
    % Get trajectory using Krylov propagation
    timestep=parameters.ro_grad_dur/(parameters.image_size(1)-1);
    rho_stack=krylov(spin_system,B+parameters.ro_grad_amp*G{2},...
                     parameters.coil,rho,timestep,...
                     parameters.image_size(1)-1,'trajectory');
    rho=rho_stack(:,end); fid(:,n)=(parameters.coil'*rho_stack)';
```

We then move to the original location in the k-space:

```
    % Kick down the k-space
    rho=step(spin_system,B-pe_grad_amps(n)*G{1},...
             rho,parameters.pe_grad_dur);

end
```

Finally, apodization and Fourier transform are applied:

```
% Apodization
fid=apodization(fid,'sqsinbell-2d');

% Fourier transform
mri=-real(fftshift(fft2(ifftshift(fid)),1));
```

4.2.4 Echo planar imaging

One of the earliest attempts at rapid acquisition was by Mansfield in 1977 [45]. He proposed the echo planar imaging technique (EPI), which can provide a 2D image in 30-100 ms. EPI pulse sequence contains a 90° pulse followed by spatial encoding gradients that fill the k-space as shown in **Figure 28** [139].

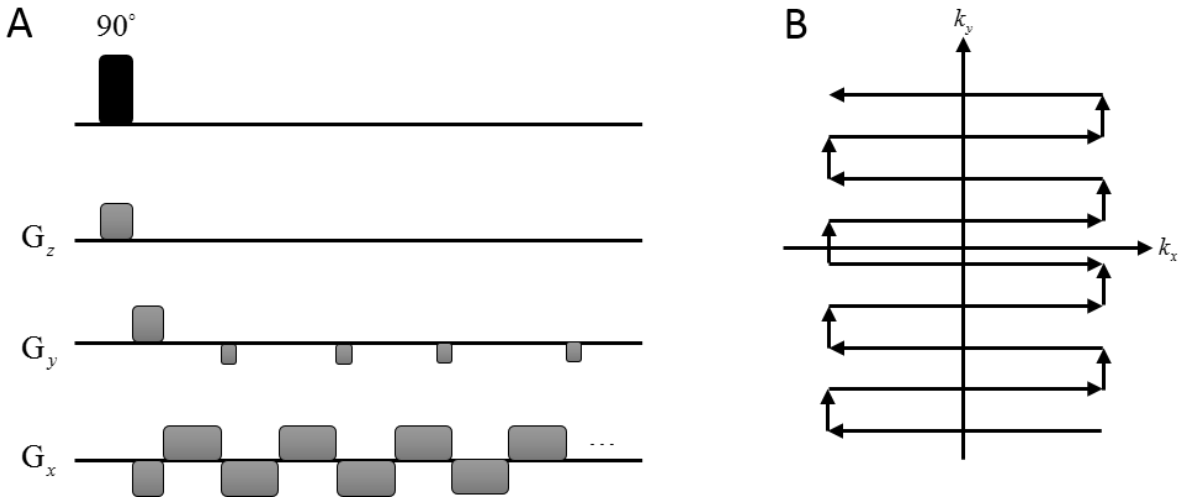


Figure 28. *(A) the echo planar pulse sequence which contains a 90° pulse with slice selection gradient followed by phase encoding and readout gradients to move in k-space upward to downward and left to right respectively. (B) the k-space path traced by frequency and phase encoding gradients.*

This pulse sequence was implemented in *Spinach*. The code of the corresponding example file is highly indicative of the decision making that went into the user-end syntax and the internal architecture of the MRI module, and therefore merits detailed discussion. Noting that the simulation parameters are shown in **Table 8.**

Table 8*. Echo planar simulation parameters*

Isotope	1H
Phase encoding gradient	4.8 mT/m
Readout Gradient	5.3 mT/m
Phase encoding gradient duration	1 ms
Readout Gradient duration	2 ms

The user-end syntax begins with isotope specification:

```
% Isotopes
sys.isotopes={'1H'};
```

We then specify the applied magnet field in Tesla:

```
% Magnetic induction
sys.magnet=5.9;
```

Relaxation superoperator is specified with unit relaxation rates (the actual rates are in the phantom):

```
% Relaxation model
inter.relaxation={'t1_t2'};
inter.rlx_keep='diagonal';
inter.equilibrium='zero';
inter.r1_rates=1;
inter.r2_rates=1;
```

This is a single-spin simulation; no approximations are necessary:

```
% Basis set
bas.formalism='sphten-liouv';
bas.approximation='none';
```

Spinach housekeeping functions use the above information to create the spin system object:

```
% Spinach housekeeping
spin_system=create(sys,inter);
spin_system=basis(spin_system,bas);
```

The user then specifies pulse sequence parameters:

```
% Sequence parameters
parameters.spins={'1H'};
parameters.decouple={};
parameters.offset=0.0;
parameters.image_size=[101 105];
parameters.ro_grad_amp=5.3e-3; % T/m
parameters.pe_grad_amp=4.8e-3; % T/m
parameters.grad_dur=2e-3;
```

The relaxation superoperator is requested from Spinach:

```
% Relaxation superoperators
[R1Op,R2Op]=rlx_t1_t2(spin_system);
```

The relaxation theory phantom (a standard 3D brain) is loaded from disk and a 2D slice is taken:

```
% Phantom library call
[R1Ph,R2Ph,PDPh,dims,npts]=phantoms('brain-medres');
R1Ph=R1Ph(:,:,50); R2Ph=R2Ph(:,:,50); PDPh=PDPh(:,:,50);
```

Sample geometry is set to match the phantom:

```
% Sample geometry
parameters.dims=dims([1 2]);
parameters.npts=npts([1 2]);
parameters.deriv={'period',3};
```

Relaxation superoperator is built out of phantoms:

```
% Relaxation phantoms
```

```
parameters.rlx_ph={R1Ph,R2Ph};
parameters.rlx_op={R1Op,R2Op};
```

The initial state is modulated with proton density; the detection state is assumed to be uniform:

```
% Initial and detection state phantoms
parameters.rho0_ph={PDPh};
parameters.rho0_st={state(spin_system,'Lz','1H','cheap')};
parameters.coil_ph={ones(parameters.npts)};
parameters.coil_st={state(spin_system,'L+','1H','cheap')};
```

Diffusion and flow rates are set:

```
% Diffusion and flow
parameters.u=zeros(parameters.npts);
parameters.v=zeros(parameters.npts);
parameters.diff=0;
```

And finally the pulse sequence is called within the imaging context:

```
% Run the simulation
mri=imaging(spin_system,@epi,parameters);
```

The output is shown in **Figure 29**.

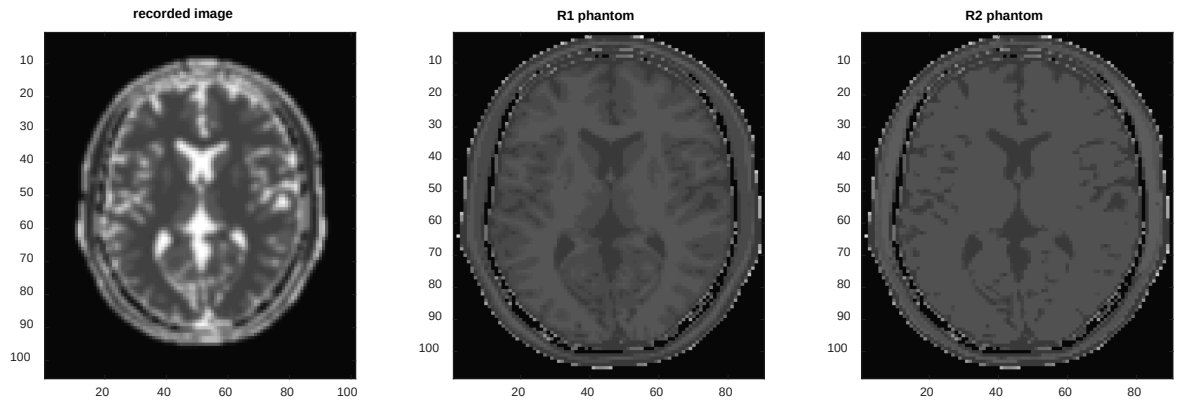


Figure 29. *An illustration of the performance of the pulse sequence shown in* ***Figure 28****. The simulation contains a relaxation phantom (R1 and R2) and proton density phantom, and takes only a few seconds.*

The pulse sequence itself starts with a hard 90° pulse:

```
% Apply 90-degree pulse
parameters.rho0=step(spin_system,Ly,parameters.rho0,pi/2);
```

A preliminary gradient moves the system to the bottom left corner of the k-space:

```
% Preroll the gradients
rho=evolution(spin_system,B-parameters.pe_grad_amp*G{1}...
                          -parameters.ro_grad_amp*G{2},[],rho,...
                           parameters.grad_dur/2,1,'final');
```

A hard 180° hard pulse

```
% Apply 180-degree pulse
parameters.rho0=step(spin_system,Hy,parameters.rho0,pi);
```

Propagators are precomputed for efficiency

```
% Precompute propagators
P_ro_p=propagator(spin_system,B+parameters.ro_grad_amp*G{2},...
                  parameters.grad_dur/(parameters.image_size(1)-1));
P_ro_m=propagator(spin_system,B-parameters.ro_grad_amp*G{2},...
                  parameters.grad_dur/(parameters.image_size(1)-1));
P_pe_p=propagator(spin_system,B+parameters.pe_grad_amp*G{1},...
                  parameters.grad_dur/(parameters.image_size(2)-1));
```

Finally, the echo planar readout loop is executed exactly as shown in the right panel of **Figure 28**:

```
% Phase encoding loop
for n=1:parameters.image_size(2)

    % Determine readout gradient sign
    ro_grad_sign=2*mod(n,2)-1;

    % Readout loop
    for k=1:parameters.image_size(1)

        % Detect under readout gradient
        if ro_grad_sign>0
            fid(k,n)=parameters.coil'*rho;
            rho=P_ro_p*rho;
        else
            rho=P_ro_m*rho;
            fid(parameters.image_size(1)-k+1,n)=parameters.coil'*rho;
        end

    end

    % Propagate under encoding gradient
    rho=P_pe_p*rho;

end
```

The result is apodised, Fourier transformed and plotted (**Figure 29**).

4.2.5 Diffusion weighted imaging

In the one-dimensional diffusion equation for a concentration profile $c(x,t)$, the scalar diffusion coefficient D can in principle depend on the spatial location:

$$\frac{\partial c(x,t)}{\partial t}=\frac{\partial}{\partial x}D(x)\frac{\partial}{\partial x}c(x,t) \tag{196}$$

The three-dimensional version of equation (196) features a symmetric 3x3 diffusion tensor, all six independent components of which can have their own coordinate dependence:

$$\frac{\partial c(\mathbf{r},t)}{\partial t}=\left[\vec{\nabla}^{\mathrm{T}}\cdot\mathbf{D}(\mathbf{r})\cdot\vec{\nabla}\right]c(\mathbf{r},t) \tag{197}$$

In real biological tissue this is often the case: even if microscopic diffusion is isotropic, the tissue itself is highly structured at the macroscopic level. If there is any barrier along molecular random walk (like a cell membrane), the observed effective value of D will be different from the free motion case, and molecular mobility may not be the same in all directions.

The subject of diffusion was introduced into magnetic resonance by Hann 1950. They demonstrated that NMR is made sensitive to diffusion by adding magnetic field gradients [121]. The line of the argument is approximately as follows.

To understand equation above, we start with a particular normalised gradient shape $s(\varepsilon)$, where $0 \le \varepsilon \le 1$. The total effect of that gradient can be expressed by its shape factor σ :

$$\sigma = \int_0^1 s(\varepsilon)\, d\varepsilon \tag{198}$$

Two further shape gradient parameters are relevant:

$$\lambda = \frac{1}{\sigma}\int_0^1 S(\varepsilon)\, d\varepsilon \qquad \kappa = \frac{1}{\sigma^2}\int_0^1 S^2(\varepsilon)\, d\varepsilon \tag{199}$$

where $S(\varepsilon)$ is a cumulative integral of $s(\varepsilon)$:

$$S(\varepsilon) = \int_0^{\varepsilon} s(\varepsilon)\, d\varepsilon \tag{200}$$

The Gradient Shape Functions and Calculated Values of Shape Parameters could be found in [140]

The coherence pathway order can be written according to the gradients applied in the spin echo pulse sequence:

$$P(t)G(t) = \begin{Bmatrix} \gamma g a(t) & 0 \le t \le \delta \\ 0 & \delta \le t \le \Delta \\ -\gamma g a(t-\Delta) & \Delta \le t \le \Delta + \delta \end{Bmatrix} \tag{201}$$

Where $P(t)$ is the coherence order as a function of time and effectively representing the coherence transfer pathway, $G(t)$ is the linear gradient varying strength over time, g is the gradient field strength and $a(t)$ is the shape of the pulse covering the pulse length δ :

$$a(t) = s\left(t/\delta\right) \tag{202}$$

Substitute equation (202) in equation (201)

$$P(t)G(t)=\begin{cases}\gamma g s\left(t/\delta\right) & 0\le t\le\delta\\ 0 & \delta\le t\le\Delta\\ -\gamma g s\left([t-\Delta]/\delta\right) & \Delta\le t\le\Delta+\delta\end{cases} \tag{203}$$

Return to equation (200), we can find:

$$\int_{t_0}^{t} s\left([t-t_0]/\delta\right)\,dt=\delta\int_{0}^{\left([t-t_0]/\delta\right)} s\left([t-t_0]/\delta\right)\,d\left([t-t_0]/\delta\right)=\delta\left([t-t_0]/\delta\right) \tag{204}$$

Now we can introduce the function that can describe the amount of dephasing per unit length:

$$q(t)=\int_{0}^{t}P(t)G(t)dt \tag{205}$$

Put equation (203) in equation (205):

$$q(t)=\begin{cases}\gamma\delta g S\left(t/\delta\right) & 0\le t\le\delta\\ \gamma\delta g\sigma & \delta\le t\le\Delta\\ -\gamma g\left[\sigma-S\left([t-\Delta]/\delta\right)\right] & \Delta\le t\le\Delta+\delta\end{cases} \tag{206}$$

The square of $q(t)$ is important because the Einstein diffusion relation contains a square root:

$$q^2(t)=\begin{cases}\gamma^2\delta^2 g^2 S^2\left(t/\delta\right) & 0\le t\le\delta\\ \gamma^2\delta^2 g^2\sigma^2 & \delta\le t\le\Delta\\ \gamma^2\delta^2 g^2\sigma^2+S^2\left([t-\Delta]/\delta\right)-2\sigma S\left([t-\Delta]/\delta\right) & \Delta\le t\le\Delta+\delta\end{cases} \tag{207}$$

At this point we should introduce the previously defined shape parameter λ and κ to evaluate the integral of equations (206) and (207) over all pulse sequence

$$\begin{aligned}\int_{t_0}^{t_0+\delta} S\left([t-t_0]/\delta\right)\,dt&=\delta\int_{0}^{\delta}S\left(t/\delta\right)\,d\left(t/\delta\right)=\delta\sigma\lambda\\ \int_{t_0}^{t_0+\delta} S^2\left([t-t_0]/\delta\right)\,dt&=\delta\int_{0}^{\delta}S^2\left(t/\delta\right)\,d\left(t/\delta\right)=\delta\sigma^2\kappa\end{aligned} \tag{208}$$

The integral over $q(t)$ required to assess the effects of unidirectional translation E_{transl}

$$\ln\left(E_{\text{transl}}\right)=\int_{0}^{\Delta+\delta} q(t)\,dt=\gamma\delta g\left[\underbrace{\left(\delta\sigma\lambda\right)}_{0\le t\le\delta}+\underbrace{\left(\Delta-\delta\right)\sigma}_{\delta\le t\le\Delta}+\underbrace{\left(\delta\sigma-\delta\sigma\lambda\right)}_{\Delta\le t\le\Delta+\delta}\right]=\gamma\delta\sigma g\Delta \quad (209)$$

where:

$$E_{\text{transl}}=\exp\left(iv_z\int_{o}^{\Delta+\delta} q(t)\,dt\right) \quad (210)$$

where v_z is the velocity of the tr anslation along the z-axis.

Similarly, the integral of $q^2(t)$, to assess the effects of diffusion, is evaluated as:

$$\begin{aligned}\ln\left(E_{\text{diff}}\right)&=\int_{0}^{\Delta+\delta} q^2(t)\,dt\\&=\gamma^2\delta^2\mathrm{g}^2\times\left[\underbrace{\left(\delta\sigma^2\kappa\right)}_{0\le t\le\delta}+\underbrace{\left(\Delta-\delta\right)\sigma^2}_{\delta\le t\le\Delta}+\underbrace{\left(\delta\sigma^2+\delta\sigma^2\kappa-2\delta\sigma^2\lambda\right)}_{\Delta\le t\le\Delta+\delta}\right] \quad (211)\\&=\gamma^2\delta^2\sigma^2\mathrm{g}^2\left[\Delta+2\left(\kappa-\lambda\right)\delta\right]\end{aligned}$$

where:

$$E_{\text{transl}}=\exp\left(-D\int_{o}^{\Delta+\delta} q^2(t)\,dt\right) \quad (212)$$

For rectangular shape gradient, the shape parameters are equal to:

$$\sigma=1,\ \lambda=\frac{1}{2},\ \kappa=\frac{1}{3} \quad (213)$$

By substituting equation (213) in equation (211), the complete ST equation become:

$$E_{diff}=e^{-D\gamma^2\delta^2\mathrm{g}^2\left(\Delta-\frac{\delta}{3}\right)} \quad (214)$$

In which E_{diff} is the signal intensity. The resulting attenuation is:

$$c=c_0\exp\left(-b\cdot D\right)\qquad b=\gamma^2G^2\delta^2\left(\Delta-\delta/3\right) \quad (215)$$

Where c and c_0 are signal intensities in the presence and absence the diffusion respectively, D is the apparent diffusion coefficient where γ is the magnetogyric ratio, G is gradient amplitude, δ is

gradient duration and Δ is the diffusion interval. The b factor characterises the diffusion sensitivity of the sequence.

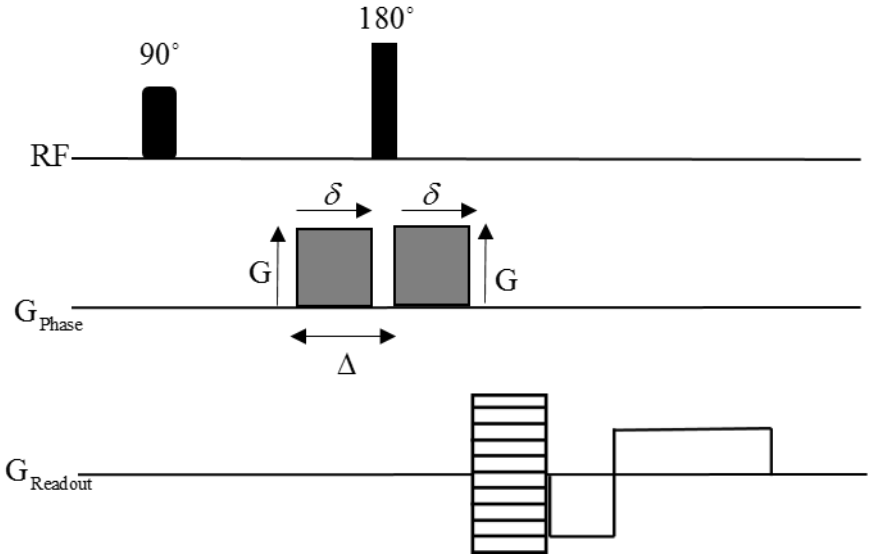


Figure 30. *Diffusion weighted imaging sequence based on the Stejskal-Tanner pulse sequence.G is the strength of the gradient, δ is the duration of the gradient field and Δ is time interval between the first and the second gradients.*

This technique forms the basis for most modern DWI pulse sequences. Its core consists of two strong gradient pulses of magnitude G and duration δ, separated by time interval Δ. Pure imaging sequences must have low b values, ideally around 1 s/mm^2, so that the diffusion effects are negligible. However, when the diffusion-sensitive gradient pair is inserted, b value must be higher.

Programming a diffusion-sensitive pulse sequence into *Spinach* essentially amounts to inserting diffusion-sensitive gradient pairs into standard imaging sequences. An example appears below. The sequence starts with a hard 90° pulse:

```
% Apply 90-degree pulse
parameters.rho0=step(spin_system,Ly,parameters.rho0,pi/2);
```

after which the diffusion gradient is applied

```
% Apply diffusion gradient
parameters.rho0=step(spin_system,B+parameters.diff_g_amp(1)*G{1}+...
                    parameters.diff_g_amp(2)*G{2},parameters.rho0,...
                    parameters.diff_g_dur);
```

A hard 180° pulse is intended to refocus the spin evolution

```
% Apply 180-degree pulse
parameters.rho0=step(spin_system,Hy,parameters.rho0,pi);
```

The second diffusion gradient is then applied

```
% Apply diffusion gradient
parameters.rho0=step(spin_system,B+parameters.diff_g_amp(1)*G{1}+...
                     parameters.diff_g_amp(2)*G{2},parameters.rho0,...
                     parameters.diff_g_dur);
```

The rest is a standard phase encoding + frequency encoding imaging sequence:

```
% Get phase encoding gradient range
pe_grad_amps=linspace(-parameters.pe_grad_amp,...
                       parameters.pe_grad_amp,...
                       parameters.image_size(2));

% Get phase encoding stack
rho_stack=zeros(numel(parameters.rho0),parameters.image_size(2));
parfor n=1:parameters.image_size(2) %#ok<*PFBNS>
    rho_stack(:,n)=step(spin_system,B+pe_grad_amps(n)*G{1},...
                        parameters.rho0,parameters.pe_grad_dur);
end

% Run the pre-roll gradient
rho_stack=step(spin_system,B-parameters.ro_grad_amp*G{2},...
               rho_stack,parameters.ro_grad_dur/2);

% Get the timing parameters
nsteps=parameters.image_size(1)-1;
step_length=parameters.ro_grad_dur/nsteps;

% Detect under the readout gradient
fid=evolution(spin_system,B+parameters.ro_grad_amp*G{2},parameters.coil,...
              rho_stack,step_length,nsteps,'observable');

% Apodization
fid=apodization(fid,'sqsinbell-2d');

% Fourier transform
mri=real(fftshift(fft2(ifftshift(fid))));
```

In the special case of isotropic diffusion, the diffusion coefficient is a scalar – an example of such an imaging simulation appears in **Figure 31** and the simulation parameters are shown in **Table 9.**

***Table 9**. Diffusion weighted simulation parameters*

Isotope	1H
Diffusion gradient	1 mT/m
Diffusion gradient duration	0.1 ms
Phase encoding gradient	3.8 mT/m
Readout Gradient	4.3 mT/m
Phase encoding gradient duration	1 ms
Readout Gradient duration	2 ms

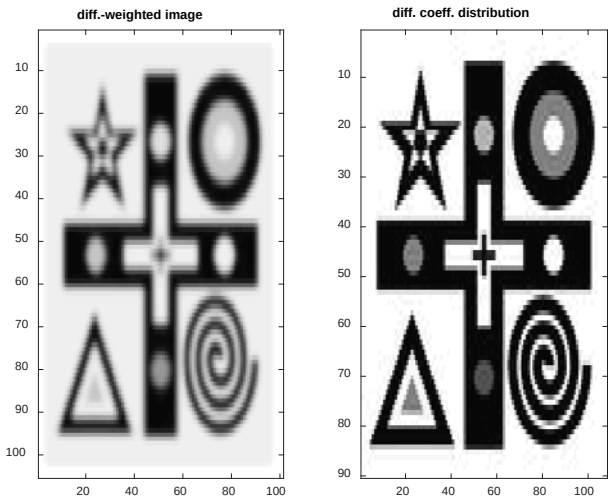


Figure 31. *A two-dimensional diffusion weighted image.The isotropic diffusion coefficient is set to $1{\cdot}10^{-3}$ m^2/s in the dark areas and to zero in the bright areas. The initial condition is a uniform spin density phantom.*

The non-uniform and anisotropic diffusion case is more complicated because a diffusion image would be a distribution of a tensor quantity. The anisotropy must therefore be somehow reduced to a scalar. The most common way of doing this is to use the "fractional anisotropy" (FA) measure, which is defined in the following way:

$$FA=\sqrt{\frac{3}{2}}\frac{1}{\sqrt{\lambda_1^2+\lambda_2^2+\lambda_3^2}}\sqrt{\frac{\left(\lambda_1-\left\langle\lambda\right\rangle\right)^2+\left(\lambda_2-\left\langle\lambda\right\rangle\right)^2+\left(\lambda_3-\left\langle\lambda\right\rangle\right)^2}{3}} \tag{216}$$

where λ_k are the eigenvalues of the diffusion tensor and $\left\langle\lambda\right\rangle$ is their average. The resulting measure is a scalar between zero and one. The zero value means the diffusion is isotropic and one means maximum possible second rank anisotropy.

The measure shown in Equation (216), along with the corresponding pulse sequence, were implemented into Spinach as a part of this project, but the corresponding code is too long to realistically appear in this thesis. It is simply a variation on the themes already explored above; the reader is encouraged to inspect the source code of the fractional_anisotropy_2d.m example supplied with Spinach, and the code of the pulse sequence that it calls.

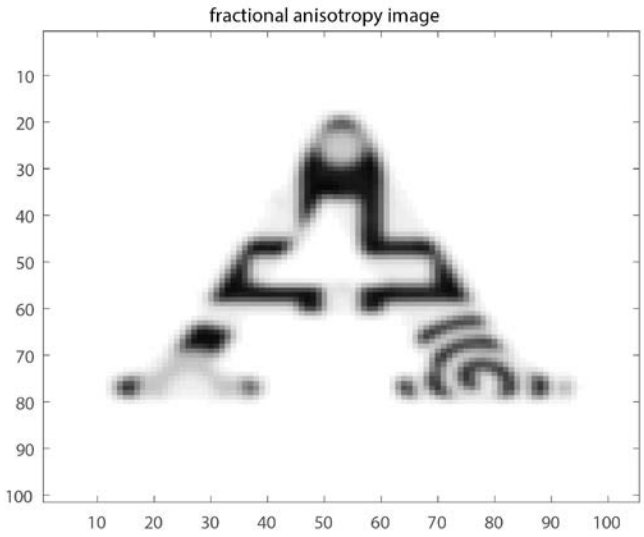


Figure 32. *A two dimensional fractional anisotropy diffusion imaging simulation. The calculation includes the full Liouville space of a two-spin system; the full dipolar relaxation superoperator is obtained using Redfield theory; diffusion is handled using the explicit finite difference diffusion operator within the Fokker-Plank equation for simultaneous spin and spatial dynamics.*

The example of a sophisticated calculation combining non-trivial spin dynamics with a complicated diffusion process is shown in **Figure 32**: the calculation involves a *J*-coupled two-spin system with the full dipolar relaxation superoperator.

4.3 Quantum effects in MRI

4.3.1 Localised NMR of complex molecules

It is very important for many medical and industrial applications to be able to record an NMR spectrum *at a particular location*. For example, in central nervous system diagnostics, neurological disorders such as dementia, schizophrenia, and ischaemic processes can be detected early by the metabolite concentration abnormalities [141]. Localised NMR can report on neurotransmitters (glutamate, aspartate and gamma-aminobutyric acid), bioenergetic metabolites (creatine, lactic acid), osmolytes, molecules involved with membrane synthesis, sugars, and other CNS metabolites present at millimolar concentrations [142]. This gives valuable insight into the nature of disease metabolism [143].

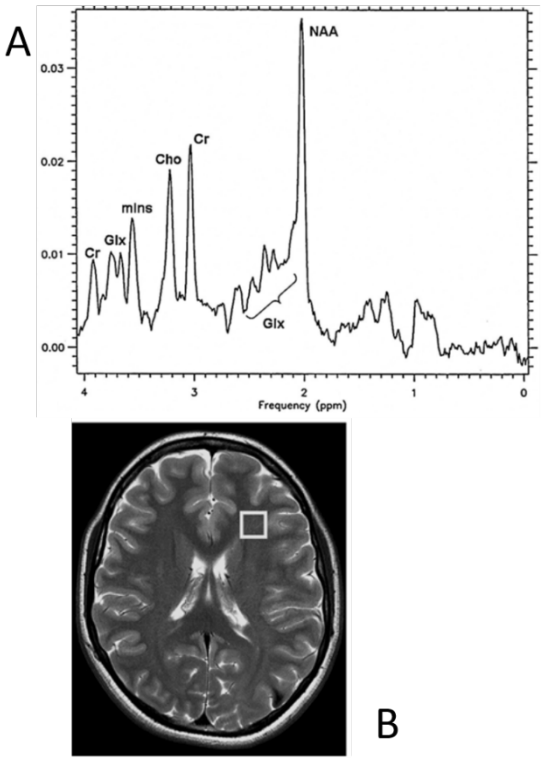


Figure 33. *A T2-weighted FSE normal brain image. A, localized proton spectrum use the PRESS sequence. B, Location of voxel used for localized proton spectra. Reproduced from [144].*

From the simulation point of view, this is exactly the nightmare scenario discussed in the Introduction – a large spin state space kroneckered into three-dimensional spatial dynamics. Even conservative estimates for the resulting matrix dimensions are in the millions. However, the prize is tempting: modern localised NMR is severely limited by the lack of sophistication in the localised excitation techniques, and any improvements must necessarily rely on simulations.

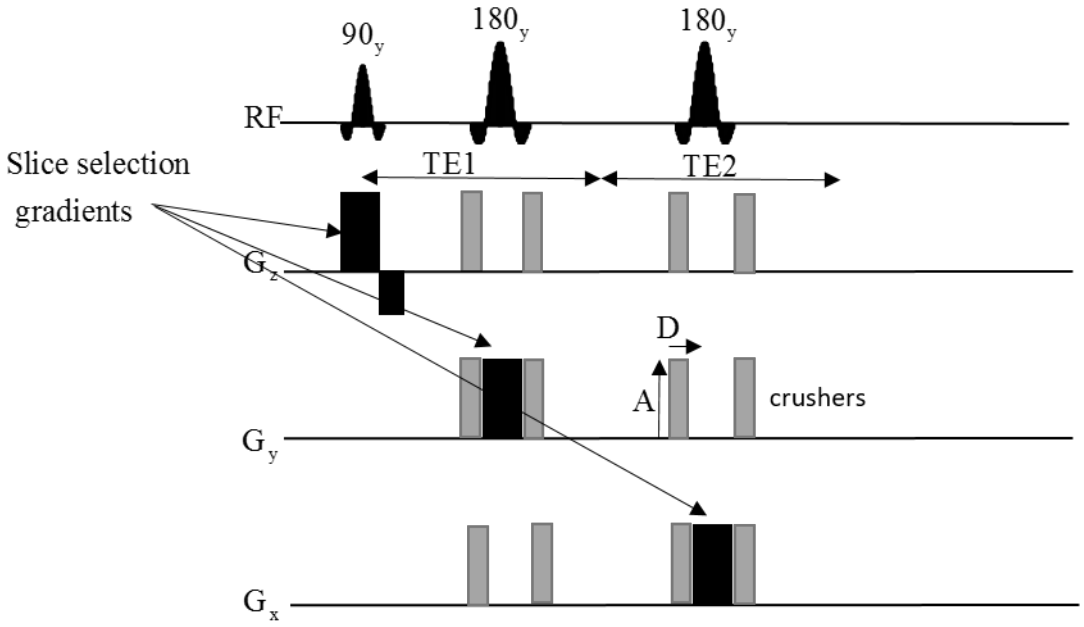


Figure 34. *PRESS pulse sequence.The sequence consists of 90°-180°-180° pulses with three gradients in x,y and z directions. The crusher gradients are required to dephase any unwanted signals. Three rounds of gradients define three orthogonal planes and excite a cubic voxel.*

Better results can be obtained from point resolved spectroscopy sequence (PRESS) sequence which does not excite the whole body and results in a high resolution MR spectrum. PRESS pulse sequence [145] is one of the methods that detect signal from a localized voxel. The sequence consists of three slice selection pulses (90°, 180°, 180°) with three gradients in x, y, z directions. Each RF pulse applied with different gradients (Gx, Gy, Gz). The first 90° pulse is applied to excite protons in z direction to xy-plane. Following 180° pulses applied in x and y directions depending on their gradient in that direction. The PRESS echo can be generated by the sample located in three plane crossing point. It is formed at time TE=TE1+TE2. Additionally, crusher gradients may be applied to dephase any unnecessary signals produced from 180° refocusing pulses.

PRESS sequence was implemented in *Spinach*. In this section we show some examples of a PRESS simulation for an AB spin system. The simulation was performed on a phantom containing three substances A,B and C with different chemical shifts and *J*-couplings. Simulated free induction decays were generated for 100 points over an acquisition time of 25 ms and NMR spectra obtained by Fourier transform of the free induction decays. Again, *Spinach* syntax is highly indicative of the software engineering that was done. It is presented below and the simulation parameters are shown **Table 10**.

Table 10. 1D PRESS simulation parameters

Isotope	1H
Pulse time	0.05 ms
Pulse frequency	[-1 0 1] kHz
Slice selection gradient	30 mT/m
Readout gradient	30 mT/m

We specify a non-trivial spin system:

```
% Isotopes
sys.isotopes={'1H','1H','1H','1H','1H','1H'};
```

Specify the magnet in Tesla

```
% Magnetic induction
sys.magnet= 3.0;
```

The chemical shift for each spin is specified in ppm

```
% Chemical shifts
inter.zeeman.scalar={-3,3,-2,2,-1,1};
```

Specify the J-coupling between protons

```
% J coupling
inter.coupling.scalar=cell(6);
inter.coupling.scalar{1,2}=10;
inter.coupling.scalar{3,4}=20;
inter.coupling.scalar{5,6}=30;
```

The system is composed of three independent spin pairs that mimic three different substances. A connectivity-adaptive basis set is chosen to make use of this fact automatically:

```
% Basis set
bas.formalism='sphten-liouv';
bas.approximation='IK-2';
bas.space_level=1;
bas.connectivity='scalar_couplings';
```

Spinach housekeeping is called:

```
% Spinach housekeeping
spin_system=create(sys,inter);
spin_system=basis(spin_system,bas);
```

Imaging experiment parameters are specified:

```
 % Sequence parameters
parameters.spins={'1H'};
parameters.decouple={};
parameters.offset=0;
parameters.sweep=500000;
parameters.npoints=128;
parameters.axis_units='kHz';
parameters.invert_axis=1;
parameters.ss_grad_ampl=30e-3;
```

```
parameters.ro_grad_ampl=30e-3;
```

Pulse parameters are specified:

```
% Pulse parameters
parameters.rf_phi=pi/2;
parameters.rf_frq_list=-100e3;       % change this to scan through the sample
parameters.rf_amp_list=2*pi*5000;
parameters.rf_dur_list=0.5e-4;
parameters.max_rank=3;
```

Relaxation is set to zero everywhere:

```
% Relaxation phantom
parameters.rlx_ph={zeros(parameters.npts,1)};
parameters.rlx_op={relaxation(spin_system)};
```

The initial condition phantom has substance A on the left, substance B on the right, and substance C spread across the entire sample:

```
% Initial state phantom
parameters.rho0_ph={[zeros(10,1); ones(30,1); zeros(60,1)];
                    [zeros(60,1); ones(30,1); zeros(10,1)];
                    [ones(60,1);  ones(30,1); ones(10,1)]};
parameters.rho0_st={state(spin_system,{'Lz'},{1})+state(spin_system,{'Lz'},{2});
                    state(spin_system,{'Lz'},{3})+state(spin_system,{'Lz'},{4});
                    state(spin_system,{'Lz'},{5})+state(spin_system,{'Lz'},{6})};
```

The detection state is uniform across the sample:

```
% Detection state phantoms
parameters.coil_ph={ones(prod(parameters.npts,1))};
parameters.coil_st={state(spin_system,'L+','1H','cheap')};
```

No diffusion and no flow:

```
% Flow
parameters.u=zeros(parameters.npts);

% Diffusion
parameters.diff=0;
```

Finally, the PRESS sequence is called via the imaging context function:

```
% Run PRESS 1D
parameters.sweep=1000;
parameters.axis_units='ppm';
fid=imaging(spin_system,@press_1d,parameters);
```

The pulse sequence itself is quite simple in one dimension. It applies a slice selection pulse:

```
% Slice selection pulse
parameters.rho0=shaped_pulse_af(spin_system,L+parameters.ss_grad_ampl*G{1},Lx,Ly,...
                parameters.rho0,parameters.rf_frq_list,parameters.rf_amp_list,...
                parameters.rf_dur_list,parameters.rf_phi,parameters.max_rank);
```

then rephases the spins:

```
% Rephasing gradient
parameters.rho0=evolution(spin_system,L-parameters.ss_grad_ampl*G{1},[],...
                          parameters.rho0,sum(parameters.rf_dur_list)/2,1,'final');
```

Then runs the standard acquisition function:

```
% Call to acquisition
fid=acquire(spin_system,parameters,H+F,R,K);
```

The output is shown in Figure 35 for three different slice selection frequencies.

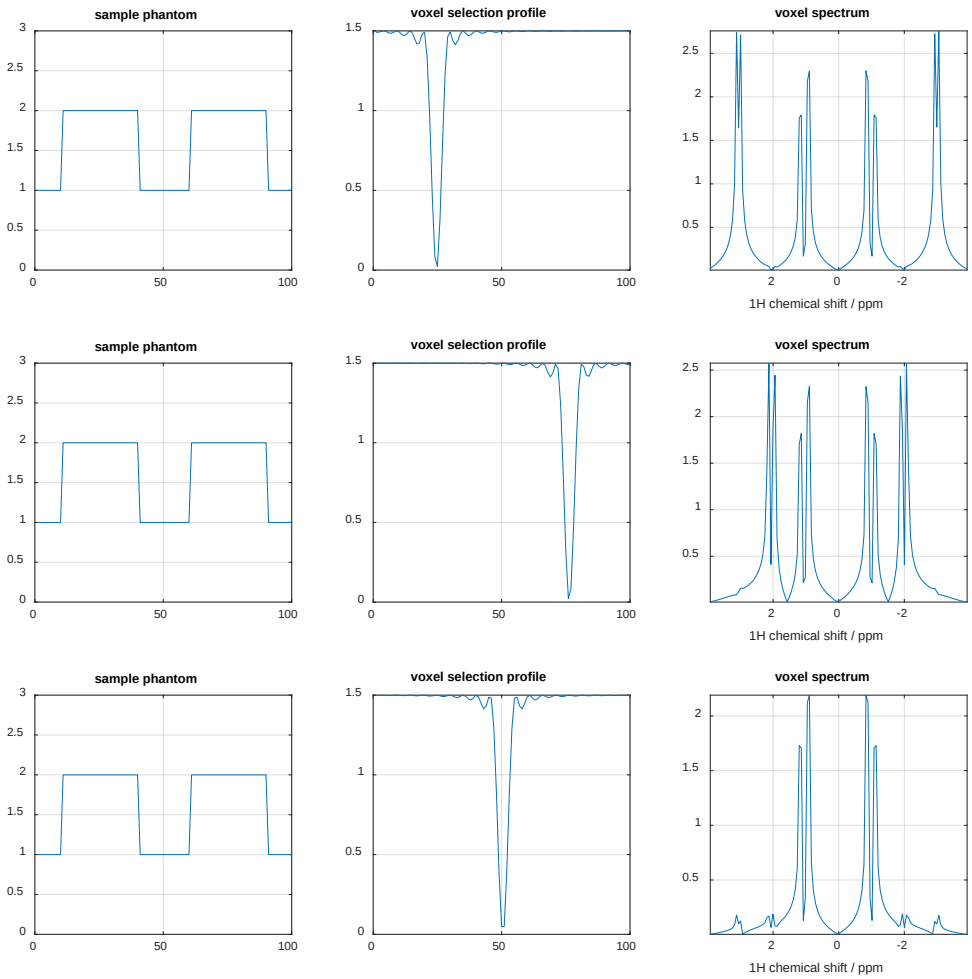


Figure 35. *Simulation of one-dimensional PRESS. Three chemical substances were used containing pairs of protons with J=10 Hz, J=20 Hz, and J=30 Hz. The first row uses the spatial location that excites substances A and C. The second row shows the excitation of B and C. Finally the third row shows excitation of only C.*

The two-dimensional version of the PRESS pulse sequence and simulation parameters are very similar. The same spin system specification block is run, but the sequence and pulse parameters are richer because we now have two spatial dimensions:

```
% Sequence parameters
parameters.spins={'1H'};
parameters.decouple={};
parameters.offset=0;
parameters.sweep=1000;
parameters.axis_units='ppm';
parameters.npoints=128;
parameters.invert_axis=1;
parameters.ss_grad_ampl=[25e-3 25e-3];
```

```
parameters.image_size=[101 105];
parameters.sp_grad_amp=5e-3; % T/m
parameters.sp_grad_dur=5e-4;

% Pulse parameters
parameters.rf_phi={pi/2 pi/2};
parameters.rf_frq_list={-120e3 -100e3};       % change this to scan through the sample
parameters.rf_amp_list={2*pi*5000 2*pi*5000};
parameters.rf_dur_list={0.5e-4 1.0e-4};
parameters.max_rank={3 3};

% Sample geometry
parameters.dims=[0.30 0.25];
parameters.npts=[108 90];
parameters.deriv={'period',3};
```

The initial condition phantom is now three spots for the tree chemicals at random locations:

```
% Initial state phantom
[X,Y]=ndgrid(1:parameters.npts(1),1:parameters.npts(2));
parameters.rho0_ph={(X-15).^2+(Y-15).^2<10^2;
                    (X-30).^2+(Y-40).^2<10^2;
                    (X-70).^2+(Y-80).^2<10^2};
parameters.rho0_st={state(spin_system,{'Lz'},{1})+state(spin_system,{'Lz'},{2});
                    state(spin_system,{'Lz'},{3})+state(spin_system,{'Lz'},{4});
                    state(spin_system,{'Lz'},{5})+state(spin_system,{'Lz'},{6})};
```

The pulse sequence runs the first slice selection with a 90-degrees pulse:

```
% Slice selection pulse
rho=shaped_pulse_af(spin_system,L+parameters.ss_grad_ampl(1)*G{1},Lx,Ly,...
                    rho,parameters.rf_frq_list{1},parameters.rf_amp_list{1},...
                    parameters.rf_dur_list{1},parameters.rf_phi{1},...
                    parameters.max_rank{1});
```

This necessitates a rephasing gradient:

```
% Rephasing gradient
rho=evolution(spin_system,L-parameters.ss_grad_ampl(1)*G{1},[],...
              rho,sum(parameters.rf_dur_list{1})/2,1,'final');
```

A crusher gradient is then applied as per **Figure 34**:

```
% Apply a crusher gradient
parameters.rho0=evolution(spin_system,L+parameters.sp_grad_amp*(G{1}+G{2}),[],...
                         parameters.rho0,parameters.sp_grad_dur,1,'final');
```

The slice in the orthogonal direction is selected with a 180-degree pulse:

```
% Slice selection 180 degree pulse
parameters.rho0=shaped_pulse_af(spin_system,L+parameters.ss_grad_ampl(2)*G{2},Lx,Ly,...
                parameters.rho0,parameters.rf_frq_list{2},2*parameters.rf_amp_list{2},...
                parameters.rf_dur_list{2},parameters.rf_phi{2},parameters.max_rank{2});
```

with another crusher:

```
% Apply a crusher gradient
parameters.rho0=evolution(spin_system,L+parameters.sp_grad_amp*(G{1}+G{2}),[],...
                         parameters.rho0,parameters.sp_grad_dur,1,'final');
```

The resulting excited and un-dephased pixel is submitted for acquisition:

```
% Acquisition
fid=acquire(spin_system,parameters,H+F,R,K);
```

The result is shown in **Figure 36**.

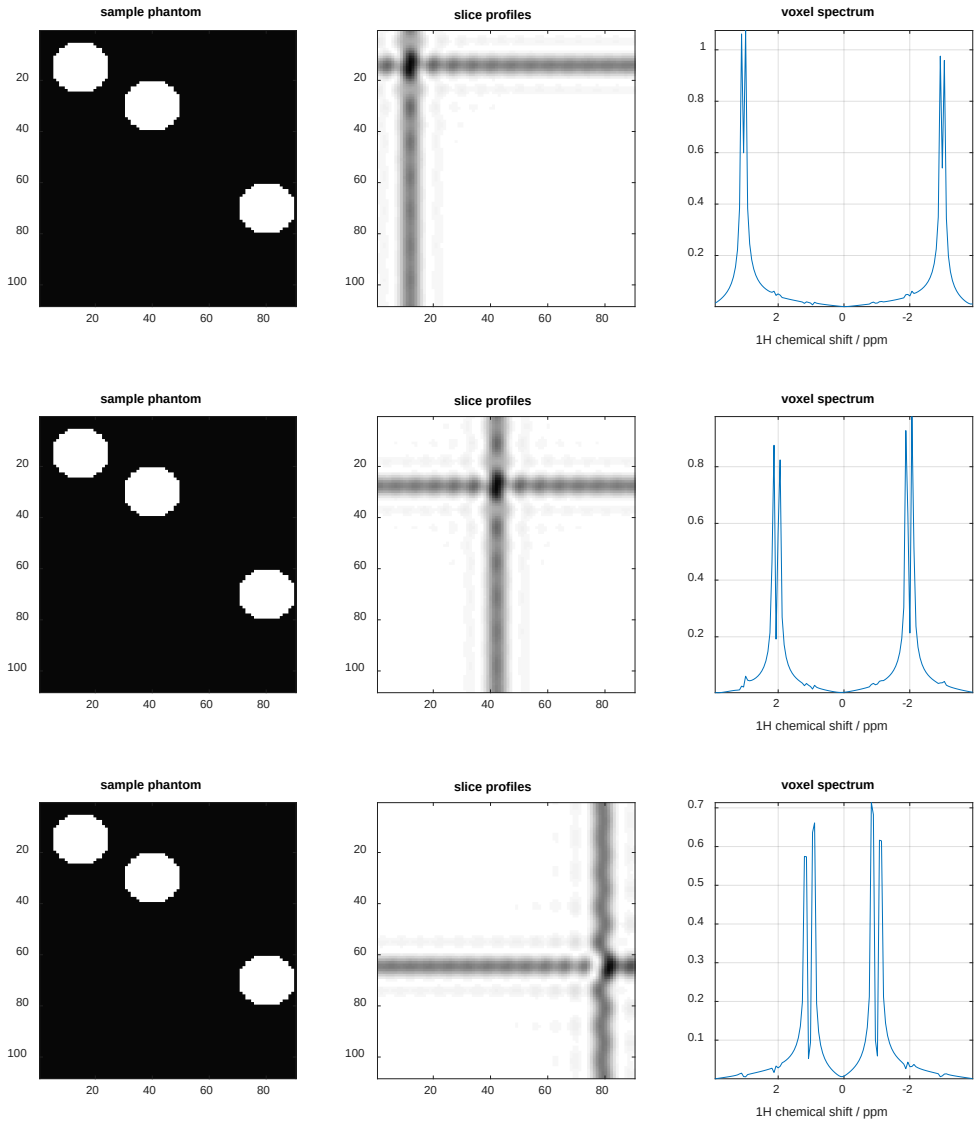


Figure 36. *Simulation of two-dimensional PRESS. The three spots on the phantom correspond to three substances, each containing a pair of protons with J=10 Hz, J=20 Hz, and J=30Hz. Three sets of vertical and horizontal excitation profile intersections are used to excite the three substances individually.*

It is instructive to consider the Fokker-Planck state space dimensions for the examples covered within this section. It is clear from **Table 11** that such calculations were previously completely impossible.

Table 11. *Matrix dimension statistics for the imaging example files supplied with Spinach.*

Example file	FP space dimension	Wall clock time
diff_weighted_2d.m	38880	Minutes
fractional_anisotropy_2d.m	155520	Minutes
singlet_imaging_1.m	180000	Minutes
singlet_imaging_2.m	224000	days

4.3.2 Gradient-based selective suppression and excitation methods

Local chemical information about human tissues is valuable in clinical diagnostics [146], but the corresponding NMR spectra are crowded and dominated by very intense peaks (mostly water and fat [147]). This makes localised NMR relatively insensitive to metabolites [148]. They have concentrations in the millimolar range that are hard to detect alongside much stronger signals. Water and fat suppression is therefore necessary to obtain accurate quantification. The same applies to biological NMR spectroscopy where the residual water signal is also a problem [149].

Many techniques are used to supress strong signals [150]. Most of them, like chemical shift selective suppression (CHESS), use excitation pulses followed by dephasing gradients to reduce the water magnetization [151]. A particularly successful technique is Double Pulse Field Gradient Spin Echo (DPFGSE) pulse sequence [152]. The DPFGSE scheme consists of two selective 180° pulses, each one flanked by two gradients (G1-S-G1-G2-S-G2). By using different values for G1 and G2, only the selected resonances will be refocused at the end of the echo whereas all other signals are defocused. The phase properties of the DPFGE scheme depend only on the inversion profile of the refocusing pulse, while the amplitude is scaled both by the inversion profile of the refocusing pulse and by the unavoidable losses due to relaxation during the spin echo[152].

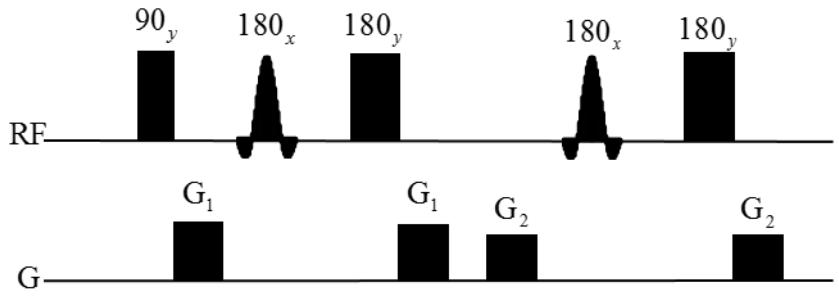


Figure 37. *DPFGSE pulse sequence. The gradients G_1 and G_2 are different in amplitude or duration. The 180° shaped pulses are signal selection pulses. The rectangular shapes represent the hard pulses.*

As a spatio-temporal pulse sequence involving complicated spin systems, DPFGSE is an excellent example of the functionality that was implemented in *Spinach* within this project. In this section we show DPFGSE-based selective excitation and selective suppression of signals of γ-aminobutyric acid, which is an important inhibitory neurotransmitter in the brain. Furthermore, **Table 12** shows the simulation parameters for this example. Many brain disorders, such as depression, epilepsy, Alzheimer's disease,

and schizophrenia could be early diagnosed by knowing the concentration of GABA [153]. Therefore, a method for in *vivo* assessment of GABA is useful for understanding the pathophysiology, diagnosis, and treatment of related disorders [154].

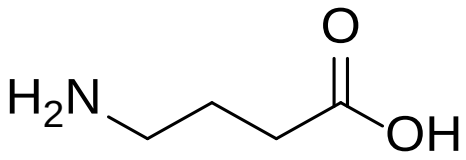


Figure 38 *Chemical structure of gamma-aminobutyric acid (GABA)*

Table 12*. DPFGSE simulation parameters*

Isotope	1H
Pulse time	10 ms
Pulse frequency	750 Hz
gradient	[1 1.5] mT/m
Readout gradient	30 mT/m

The specification for the GABA spin system is straightforward:

```
% Isotopes
sys.isotopes={'1H','1H','1H','1H','1H','1H'};

% Magnet (Tesla)
sys.magnet=5.9;

% Chemical shifts (ppm)
inter.zeeman.scalar={3.00 3.00 1.88 1.88 2.28 2.28 4.80};

% J-couplings (Hz)
inter.coupling.scalar{1,3}=7.36;
inter.coupling.scalar{1,4}=7.36;
inter.coupling.scalar{2,3}=7.36;
inter.coupling.scalar{2,4}=7.36;
inter.coupling.scalar{3,5}=7.58;
inter.coupling.scalar{3,6}=7.58;
inter.coupling.scalar{4,5}=7.58;
inter.coupling.scalar{4,6}=7.58;
inter.coupling.scalar{7,7}=0;
```

A connectivity-adaptive basis set ensures that unimportant spin states are excluded:

```
% Basis set
bas.formalism='sphten-liouv';
bas.approximation='IK-2';
bas.space_level=1;
bas.connectivity='scalar_couplings';
```

All of the above information is supplied to Spinach:

```
% Spinach housekeeping
spin_system=create(sys,inter);
spin_system=basis(spin_system,bas);
```

Imaging parameters including slice selection and readout gradients amplitudes are specified next:

```
% Sequence parameters
parameters.spins={'1H'};
parameters.decouple={};
parameters.offset=800;
parameters.sweep=1200;
parameters.npoints=512;
parameters.zerofill=2048;
parameters.axis_units='Hz';
parameters.invert_axis=1;
parameters.g_amp=[1e-3 1.5e-3];
parameters.g_dur=1e-3;
```

The shaped pulse in **Figure 37** can be, for example, a Gaussian:

```
% Signal selection
pulse_nsteps=10;
parameters.max_rank=2;
parameters.rf_phi=0;
parameters.rf_frq_list=750*ones(1,pulse_nsteps);
parameters.rf_amp_list=2*pi*340*pulse_shape('gaussian',pulse_nsteps);
parameters.rf_dur_list=(10e-3/pulse_nsteps)*ones(1,pulse_nsteps);
```

Relaxation is set to zero, and all phantoms are uniform:

```
% Relaxation phantom
parameters.rlx_ph={zeros(parameters.npts,1)};
parameters.rlx_op={relaxation(spin_system)};

% Initial state phantom
parameters.rho0_ph={ones(parameters.npts,1)};
parameters.rho0_st={state(spin_system,'Lz','1H')};

% Detection state phantoms
parameters.coil_ph={ones(prod(parameters.npts,1))};
parameters.coil_st={state(spin_system,'L+','1H','cheap')};
```

There is no diffusion and no flow:

```
% Flow
parameters.u=zeros(parameters.npts);

% Diffusion
parameters.diff=0;
```

Finally, the pulse sequence itself is called using the imaging context:

```
% Run simulation
fid=imaging(spin_system,@dpfgse_select,parameters);
```

The pulse sequence itself is by now a fairly ordinary example of Spinach syntax. The walkthrough below is given for DPFGSE signal selection. The suppression sequence only differs in the sign of one gradient. We start with a hard 90° pulse:

```
% Hard 90 on everything
rho=step(spin_system,Ly,parameters.rho0,pi/2);
```

We proceed to dephase spins using positive gradient:

```
% Gradient
rho=step(spin_system,L+parameters.g_amp(1)*G{1},rho,parameters.g_dur);
```

We then apply a shaped 180° pulse to rephase a specific signal:

```
% Soft 180 on user-specified frequency
rho=shaped_pulse_af(spin_system,L,Lx,Ly,rho,parameters.rf_frq_list-parameters.offset,...
                    parameters.rf_amp_list,parameters.rf_dur_list,...
                    parameters.rf_phi,parameters.max_rank,'expv');
```

Two gradients are then applied as per **Figure 37**:

```
% Gradients
rho=step(spin_system,L+parameters.g_amp(1)*G{1},rho,parameters.g_dur);
rho=step(spin_system,L+parameters.g_amp(2)*G{1},rho,parameters.g_dur);
```

A shaped 180° pulse is again used to rephase a specific signal:

```
% Soft 180 on user-specified frequency
rho=shaped_pulse_af(spin_system,L,Lx,Ly,rho,parameters.rf_frq_list-parameters.offset,...
                    parameters.rf_amp_list,parameters.rf_dur_list,...
                    parameters.rf_phi,parameters.max_rank,'expv');
```

The final gradient is applied:

```
% Gradient
rho=step(spin_system,L+parameters.g_amp(2)*G{1},rho,parameters.g_dur);
```

and the signal is detected:

```
% Run the evolution and watch the coil state
fid=evolution(spin_system,L,parameters.coil,rho,...
    1/parameters.sweep,parameters.npoints-1,'observable');
```

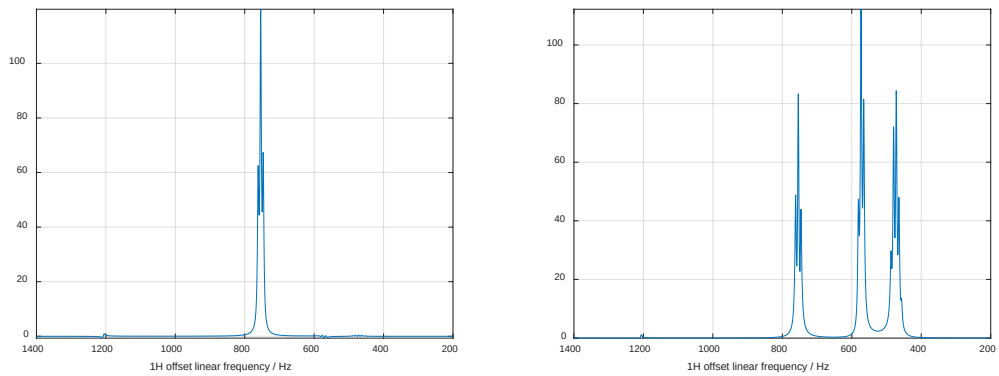


Figure 39. *Water-GABA simulation spectrum.(Right): DPFGSE water suppression for GABA solution in water. The suppressed water signal was located at 1.2 kHz. (Left): DPFGSE signal selection, where only a specific frequency window is excited using the DPFGSE method. It has been shown in this figure that the use of DPFGSE in one-dimensional water-GABA simulation result in spectra of much high quality. Such high-quality spectra make it possible to measure very small GABA enhancements and also make it straightforward to measure.*

4.3.3 Slice-selective M2S pulse sequence with diffusion and flow

An emerging class of MRI experiments that essentially requires quantum mechanical treatment of spin processes is singlet state imaging [93, 155, 156]. One reason is the delicate interplay of symmetry, chemical shift anisotropy and dipolar coupling in the relaxation superoperator that makes the two-spin singlet state long-lived. The other is the M2S and S2M pulse sequences that rely on the *J*-coupling to get the spin system in and out of the singlet state [157]. Accurate Liouville space description of the spin dynamics [158] is therefore essential. At the same time, recent proposals for using singlet states as a delivery vehicle for hyperpolarisation in MRI [93, 159] make it necessary to include diffusion and flow. The slice-selective singlet experiment is designed to extend the time scale in MRI and monitor slow motional process using singlet tagging [54].

M2S pulse sequence is the "magnetization to singlet" pulse sequence which is contains two blocks of 180° pulses separated by time delay τ and 90° pulse at the beginning of the pulse sequence [155, 160]. It is built from spin echo blocks.

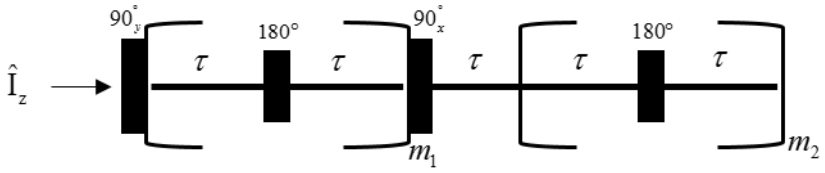


Figure 40. *M2S pulse sequence, where τ is the duration, m_1 and m_2 are the number of repetitions.*

M2S converts the magnetization into singlet when $\Delta\omega \ll 2\pi J$. For a pair of spins in a magnetic field, the basis is products of the Zeeman spin functions:

$$\mathrm{B}_z = \left\{ \left|\alpha\alpha\right\rangle, \left|\beta\alpha\right\rangle, \left|\alpha\beta\right\rangle, \left|\beta\beta\right\rangle \right\} \tag{217}$$

In the case of two magnetically equivalent spins with the same Larmor frequency, the operator for the strong coupling can be diagonalised by choosing a different basis called singlet-triplet basis. It consists of the three triplet states:

$$\begin{aligned} \left|T_{+1}\right\rangle &= \left|\alpha\alpha\right\rangle \\ \left|T_{0}\right\rangle &= \frac{1}{\sqrt{2}}\left(\left|\alpha\beta\right\rangle + \left|\beta\alpha\right\rangle\right) \\ \left|T_{-1}\right\rangle &= \left|\beta\beta\right\rangle \end{aligned} \tag{218}$$

And one singlet state:

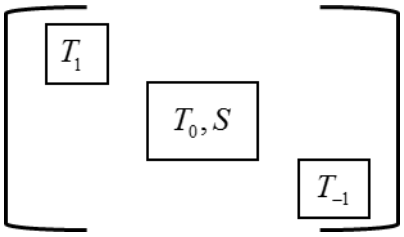
$$|S_0\rangle = \frac{1}{\sqrt{2}}\left(|\alpha\beta\rangle - |\beta\alpha\rangle\right) \tag{219}$$

Let us now understand what will happen to the singlet-triplet basis in that sequence. First, the Hamiltonian in the singlet-triplet basis is diagonal as shown below:

$$\hat{H} = \begin{pmatrix} -\frac{3}{2}\pi J & 0 & 0 & 0 \\ 0 & \omega + \frac{1}{2}\pi J & 0 & 0 \\ 0 & 0 & \frac{1}{2}\pi J & 0 \\ 0 & 0 & 0 & -\omega + \frac{1}{2}\pi J \end{pmatrix} \tag{220}$$

Where J is the spin-spin coupling between the two coupled spin

This may be represented using the following block structure:



J-coupling only acts non-trivially on the central block. By isolating the singlet-triplet subspace $\{|T_0\rangle, |S\rangle\}$ and working only on this block, we obtain an object that behaves like a spin-1/2. For any pair of energy levels $\{|1\rangle, |2\rangle\}$ we can define the corresponding Pauli matrices as:

$$\begin{aligned} \hat{L}_x^{12} &= \frac{1}{2}\left(|1\rangle\langle 2| + |2\rangle\langle 1|\right) \\ \hat{L}_y^{12} &= \frac{1}{2i}\left(|1\rangle\langle 2| - |2\rangle\langle 1|\right) \\ \hat{L}_z^{12} &= \frac{1}{2}\left(|1\rangle\langle 1| + |2\rangle\langle 2|\right) \\ \hat{E}^{12} &= \left(|1\rangle\langle 1| + |2\rangle\langle 2|\right) \end{aligned} \tag{221}$$

By using these formulas and considering the separated singlet-triplet block, we can define the fictitious operators for the pseudospin:

$$\begin{aligned}
\hat{L}_{\mathrm{x}}^{12} &= \frac{1}{2}\left(\left|\hat{T}_0\right\rangle\left\langle\hat{S}\right| + \left|\hat{S}\right\rangle\left\langle\hat{T}_0\right|\right) \\
\hat{L}_{\mathrm{y}}^{12} &= \frac{1}{2i}\left(\left|\hat{T}_0\right\rangle\left\langle\hat{S}\right| - \left|\hat{S}\right\rangle\left\langle T_0\right|\right) \\
\hat{L}_{\mathrm{z}}^{12} &= \frac{1}{2}\left(\left|\hat{T}_0\right\rangle\left\langle\hat{T}_0\right| + \left|\hat{S}\right\rangle\left\langle\hat{S}\right|\right) \\
\hat{E}^{12} &= \left(\left|\hat{T}_0\right\rangle\left\langle\hat{T}_0\right| + \left|\hat{S}\right\rangle\left\langle\hat{S}\right|\right)
\end{aligned} \tag{222}$$

The Hamiltonian can then be separated into:

$$\hat{H} = \hat{H}^{\mathrm{ST_0}} + \hat{H}^{\mathrm{T_1T_{-1}}} \tag{223}$$

Where $\hat{H}^{\mathrm{ST_0}}$ is the singlet-triplet Hamiltonian while $\hat{H}^{\mathrm{T_1T_{-1}}}$ is the rest of the Hamiltonian. The first term of Equation (223) can be written as:

$$\hat{H}^{ST_0} = \omega_\Delta \hat{L}_x^{ST_0} + \omega_J \hat{L}_z^{ST_0} - \frac{\omega_J}{4}\hat{E}^{ST_0} \tag{224}$$

Where:

$$\begin{aligned}
\omega_J &= 2\pi J \\
\omega_\Delta &= \omega_1 - \omega_2
\end{aligned} \tag{225}$$

If we translate the singlet-triplet subspace $\left\{\left|T_0\right\rangle, \left|S\right\rangle\right\}$ into a Bloch sphere:

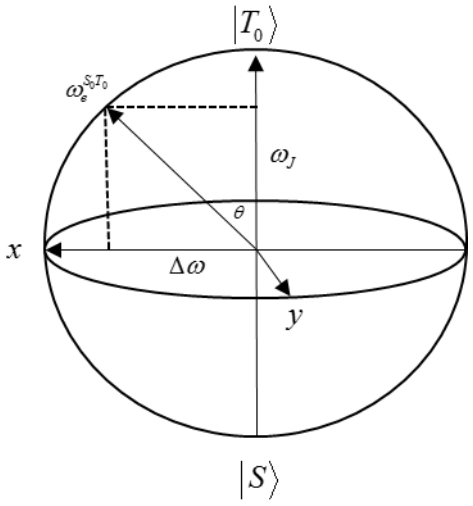


Figure 41. *Bloch sphere representation of the singlet-triplet subspace. Pure and mixed states of the two-level system correspond to points on the sphere. Incoherent mixtures of the basis states correspond to points along the Z axis.*

The constituent operators of Equation (224) are infinitesimal rotations. The combined effect of the Hamiltonian in Equation (224) may be viewed as a rotation of $\hat{L}_z^{ST_0}$ around y by angle $\theta = \arctan\left(\Delta\omega / \omega_J\right)$ that we will denote $\hat{\hat{R}}_y\left(\theta\right)$. The role of $\hat{L}_z^{ST_0}$ is to increase the strength of the effective field $\omega_e = \sqrt{\omega_J^2 + \Delta\omega^2}$. By putting that information into the $\hat{H}^{ST_0}$, we can rewrite it as:

$$\hat{H}^{ST_0} = \omega_e \hat{\hat{R}}_y\left(\theta\right)\hat{L}_z^{ST_0} - \frac{\omega_J}{4}\hat{E}^{ST_0} \tag{226}$$

We can now calculate the propagator for the free evolution:

$$\hat{U}^{ST_0}\left(\tau\right) = \exp\left(-i\tau\hat{H}^{ST_0}\right) = \exp\left(-i\tau\omega_e\hat{\hat{R}}_y\left(\theta\right)\hat{L}_z^{ST_0}\right) \tag{227}$$

This is a rotation parameterised in the angle-axis convention. When the two spins are nearly equivalent $\left(\theta \ll 1\right)$, this propagator has no effect on singlet order.

A more effective way to implement the rotations on the singlet-triplet Bloch sphere is J-synchronised spin echoes that are used in M2S and S2M pulse sequences. If the echo delay is set to

$$\tau = 1/2\omega_e \tag{228}$$

then for $\omega_J \gg \omega_\Delta$:

$$\omega_e = \sqrt{\omega_J^2 + \omega_\Delta{}^2} = \omega_J \tag{229}$$

This connects the echo delay to the *J*-coupling:

$$\tau = \frac{1}{2\omega_J} = \frac{1}{4\pi J} \tag{230}$$

It is convenient to pre-compute the π pulse propagator:

$$\hat{R}_x\left(\pi\right) = \exp\left(-i\pi\hat{L}_x\right) = \begin{array}{c} \begin{array}{cccc} T_1 & T_0 & S_0 & T_{-1} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{231}$$

As expected, the π pulse has no effect on the singlet state $\left|S_0\right\rangle$. It changes the sign of $\left|T_0\right\rangle$ and also changes the sign and position of $\left|T_1\right\rangle$ and $\left|T_{-1}\right\rangle$. After looking at the Bloch sphere introduced above, we can rewrite the rotation operator as follows:

$$\hat{R}_{\mathrm{x}}\left(\pi\right)=-i\hat{R}_{\mathrm{z}}^{ST_0}\left(\pi\right)\hat{R}_{\mathrm{x}}^{T_1T_{-1}}\left(\pi\right) \tag{232}$$

Where $\hat{R}_{\mathrm{z}}^{ST_0}$ is the rotation operator for singlet-triplet state along z-axis and $\hat{R}_{\mathrm{x}}^{T_1T_{-1}}$ is the rotation operator of the triplet states along x-axis. Therefore, the full echo propagator is:

$$\hat{U}\left(2\tau\right)=\hat{U}_{\mathrm{echo}}^{ST_0}\left(\tau\right)\hat{U}_{\mathrm{echo}}^{T_1T_{-1}}\left(\tau\right) \tag{233}$$

where:

$$\hat{U}_{\mathrm{echo}}^{ST_0}=-i\hat{R}_{\mathrm{y}}^{ST_0}\left(\theta\right)\hat{R}_{\mathrm{z}}^{ST_0}\left(\omega_{\mathrm{e}}^{ST_0}\tau\right)\hat{R}_{\mathrm{y}}^{ST_0}\left(-\theta\right)\hat{R}_{\mathrm{z}}^{ST_0}\left(\pi\right)\hat{R}_{\mathrm{y}}^{ST_0}\left(\theta\right)\hat{R}_{\mathrm{z}}^{ST_0}\left(\omega_{\mathrm{e}}^{ST_0}\tau\right)\hat{R}_{\mathrm{y}}^{ST_0}\left(-\theta\right) \tag{234}$$

Now let us analyse the full M2S sequence. We start with the thermal equilibrium population of $\hat{L}_{\mathrm{z}}$:

$$\hat{L}_{\mathrm{z}}=\left(\left|\alpha\alpha\right\rangle\left\langle\alpha\alpha\right|-\left|\beta\beta\right\rangle\left\langle\beta\beta\right|\right)=\left(\left|\hat{T}_1\right\rangle\left\langle\hat{T}_1\right|-\left|\hat{T}_{-1}\right\rangle\left\langle\hat{T}_{-1}\right|\right) \tag{235}$$

We start by applying a 90_{y}° pulse that converts longitudinal polarization into triplet-triplet coherence. Up to a constant multiplier, the system state becomes:

$$\hat{L}_{\mathrm{x}}=\left(\left|\hat{T}_1\right\rangle+\left|\hat{T}_{-1}\right\rangle\right)\left\langle\hat{T}_0\right|+\left|\hat{T}_0\right\rangle\left(\left\langle\hat{T}_1\right|+\left\langle\hat{T}_{-1}\right|\right) \tag{236}$$

After the first echo train, we have two options:

1- If the number of echoes m_1 is even, the state will be:

$$\frac{i}{2}\left[\left|\hat{T}_1\right\rangle+\left|\hat{T}_{-1}\right\rangle\right]\left\langle\hat{S}_0\right|-\left|\hat{S}_0\right\rangle\left[\left\langle\hat{T}_1\right|+\left\langle\hat{T}_{-1}\right|\right] \tag{237}$$

2- If the number of echoes m_1 is odd, the state will be:

$$\frac{i}{2}\left[\left|\hat{T}_1\right\rangle+\left|\hat{T}_{-1}\right\rangle\right]\left\langle\hat{S}_0\right|+\left|\hat{S}_0\right\rangle\left[\left\langle\hat{T}_1\right|+\left\langle\hat{T}_{-1}\right|\right] \tag{238}$$

By a historical chance, m_1 is always chosen to be even. We can now apply the 90_x° pulse to convert $\hat{T}_{\pm}$ into $\hat{T}_0$ and vice versa without affecting the singlet order:

$$\begin{aligned}\left|\hat{T}_1\right\rangle+\left|\hat{T}_{-1}\right\rangle &\rightarrow -i\sqrt{2}\left|\hat{T}_0\right\rangle\\ \left|\hat{T}_0\right\rangle &\rightarrow \frac{-i}{\sqrt{2}}\left(\left|\hat{T}_1\right\rangle+\left|\hat{T}_{-1}\right\rangle\right)\\ \left|\hat{S}_0\right\rangle &\rightarrow \left|\hat{S}_0\right\rangle\end{aligned} \tag{239}$$

Up to a constant, our state will become:

$$\left[\left|\hat{T}_1\right\rangle+\left|\hat{T}_{-1}\right\rangle\right]\left\langle\hat{S}_0\right|-\left|\hat{S}_0\right\rangle\left[\left\langle\hat{T}_1\right|+\left\langle\hat{T}_{-1}\right|\right] \quad\rightarrow\quad \left[\left|\hat{T}_0\right\rangle\left\langle\hat{S}\right|+\left|\hat{S}\right\rangle\left\langle\hat{T}_0\right|\right]=2\hat{L}_{\mathrm{x}}^{ST_0} \quad \text{for even } m \tag{240}$$

$$\left[\left|\hat{T}_1\right\rangle+\left|\hat{T}_{-1}\right\rangle\right]\left\langle\hat{S}_0\right|-\left|\hat{S}_0\right\rangle\left[\left\langle\hat{T}_1\right|+\left\langle\hat{T}_{-1}\right|\right] \quad\rightarrow\quad \left[\left|\hat{T}_0\right\rangle\left\langle\hat{S}\right|-\left|\hat{S}\right\rangle\left\langle\hat{T}_0\right|\right]=2i\hat{L}_{\mathrm{x}}^{ST_0} \quad \text{for odd } m \tag{241}$$

The result is a coherence between singlet and triplet state. After evolving it under the echo train propagator $\hat{U}(\tau)$ we would get:

$$\hat{L}_{\mathrm{x}}^{ST_0} \quad\rightarrow\quad \hat{L}_{\mathrm{y}}^{ST_0} \tag{242}$$

Finally the second echo train of the M2S pulse sequence will accomplish a 90° rotation about the x-axis of the $\left\{\left|S\right\rangle,\left|T_0\right\rangle\right\}$ Bloch sphere:

$$\hat{L}_{\mathrm{y}}^{ST_0} \quad\rightarrow\quad \hat{L}_{\mathrm{z}}^{ST_0}=\frac{1}{\sqrt{2}}\left(\left|\hat{T}_0\right\rangle\left\langle\hat{T}_0\right|-\left|\hat{S}\right\rangle\left\langle\hat{S}\right|\right) \tag{243}$$

This state contains $\left|\hat{T}_0\right\rangle$, which has a lifetime of the order of T_1, and $\left|\hat{S}_0\right\rangle$ which has a (hopefully) longer lifetime T_s. The numerical equivalent of Equations (217)-(243) is fairly trivial, the challenge from the simulation point of view is to compute the outcome of an imaging experiment where the entire process is slice-selective. This means spatial degrees of freedom, gradients, and shaped pulses.

Spinach implementation of the M2S sequence is literal and straightforward – just a sequence of calls to the step function that executes all the necessary pulses and other evolution events. The only difference that exists in the imaging context is that the initial excitation pulse should be slice-selective.

```
% M2S parameters
J=55; delta_v=6;
t=1/(4*sqrt(J^2+delta_v^2));
m1=floor(pi*J/(2*delta_v));
if mod(m1,2)~=0, m1=m1+1; end

% M2S sequence
for k=1:m1
    rho=step(spin_system,H,rho,t);
    rho=step(spin_system,Lx,rho,pi);
    rho=step(spin_system,H,rho,t);
end
rho=step(spin_system,Lx,rho,pi/2);
rho=step(spin_system,H,rho,t);
for k=1:m1/2
```

```
    rho=step(spin_system,H,rho,t);
    rho=step(spin_system,Lx,rho,pi);
    rho=step(spin_system,H,rho,t);
end
rho_sing=-step(spin_system,H,rho,t);
```

Spatial dynamics generators operate in the background – the entire mathematical infrastructure dealing with them is transparent to the user. The outcome of a typical simulation is shown in **Figure 43** – singlet state decays much slower when it is flow through an NMR tube.

In this situation, the Fokker-Planck equation is clearly superior to the more traditional Liouville - von Neumann equation formalism because all spatial dynamics processes (diffusion, hydrodynamics, magic angle spinning, spatially selective pulses, *etc.*) are represented by *constant* matrices that are more convenient from the programming and numerical efficiency point of view than the time-dependent Hamiltonians in the Liouville - von Neumann formalism [81].

This simulation is a particularly good example, because all of the complexities – shaped pulses, gradients, diffusion, flow, quantum mechanical spin evolution and complicated relaxation theory – are present simultaneously:

$$\frac{\partial}{\partial t}\boldsymbol{\rho}(t)=\left[-i\mathbf{L}(t)+\omega(t)\mathbf{D}_{\varphi}+D_{T}\mathbf{D}_{z}^{2}+v\mathbf{D}_{z}\right]\boldsymbol{\rho}(t) \tag{244}$$

$$\mathbf{D}_{\varphi}=\left[\frac{\partial}{\partial\varphi}\right]_{M}\otimes\mathbf{1}_{N}\otimes\mathbf{1}_{K}\qquad\mathbf{D}_{z}=\mathbf{1}_{M}\otimes\left[\frac{\partial}{\partial z}\right]_{N}\otimes\mathbf{1}_{K} \tag{245}$$

Where $\mathbf{L}$ the matrix representation of the spin Liouvillian, $\mathbf{D}_{\varphi}$ is the matrix representation of the RF phase derivative operator, $\mathbf{D}_{z}^{2}$ is matrix representation of the diffusion operator, $\mathbf{D}_{z}$ is the matrix representation of the flow operator, $\omega(t)$ is the RF pulse frequency, D_{T} is the translational diffusion coefficient and v is the flow velocity.

An average over 100,000 Monte-Carlo instances had to be performed by the authors of the original paper, with only statistical guarantees of convergence [93]. The Fokker-Planck formalism performs the same simulation deterministically in a single run. This simulation is included with the standard example set of *Spinach* library and the simulation parameters are shown in **Table 13**

***Table 13**. Slice selective singlet state simulation parameters*

Isotope	1H
Chemical shift	0.6 ppm
J coupling	12 Hz

Pulse time	2 ms
Pulse frequency	[-20 20 0 10 -10] kHz
gradient	[0.075 0.075] T/m

Another interesting possibility is ultrafast singlet state relaxation measurement – an experiment where different slices of the sample have different amount of relaxation delay. One possible sequence in this class is shown below.

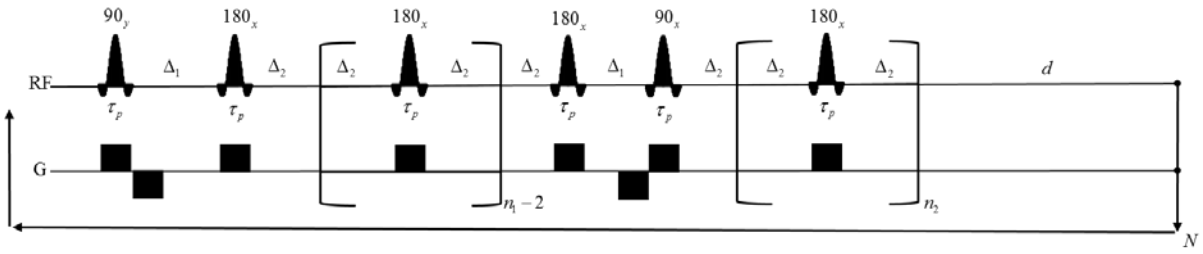


Figure 42 *A slice-selective equivalent of the M2S pulse sequence cast as a relaxation experiment*

We start with two protons at a distance with a *J* coupling:

```
% Spin system and interactions
sys.magnet=14.1;
sys.isotopes={'1H','1H'};
inter.zeeman.scalar={-5e-4,5e-4};
inter.coupling.scalar=cell(2);
inter.coupling.scalar{1,2}=12;
inter.coordinates={[0.00 0.00 0.00];
                   [2.43 0.00 0.00]};
```

Singlet state effects require accurate relaxation theory treatment:

```
% Relaxation theory
inter.relaxation={'redfield'};
inter.equilibrium='zero';
inter.rlx_keep='secular';
inter.tau_c={50e-12};
```

At two-spin system is too small to worry about restricted state spaces:

```
% Basis set
bas.formalism='sphten-liouv';
bas.approximation='none';
```

After specifying the spin system and setting the basis options, we create the spin system object and generate the basis set:

```
% Spinach housekeeping
spin_system=create(sys,inter);
spin_system=basis(spin_system,bas);
```

The next stage is to specify simulation parameters:

```
% Sample geometry
parameters.spins={'1H'};
parameters.dims=0.02;
parameters.npts=2000;
parameters.deriv={'period',5};
```

Relaxation theory phantom contains relaxation superoperators in each voxel:

```
% Relaxation phantom
R=relaxation(spin_system);
R=thermalize(spin_system,R,state(spin_system,'Lz','all'));
parameters.rlx_ph={ones(parameters.npts,1)};
parameters.rlx_op={R};
```

The initial condition phantom reflects the fact that different voxels might start off in a different spin state. The detection state phantom reflects the fact that different voxels might be detected at different angles and with different sensitivity. There is no flow, but diffusion is present:

```
% Diffusion and flow
parameters.u=zeros(parameters.npts,1);
parameters.diff=1e-9;
```

Pulse sequence parameters contain all information that the sequence needs:

```
% Pulse sequence parameters
parameters.J=12.0;
parameters.dv=0.6;
parameters.pulse_nsteps=150;
parameters.pulse_dur=2e-3;
parameters.pulse_frq=[-2e4 -1e4 0e4 +1e4 +2e4];
parameters.rlx_time=10;
parameters.fp_rank=3;
parameters.offset=0;

% Gradient amplitude
parameters.grad_ampl=[0.075 0.075] ; % T/m
```

The next thing is to call the imaging context function that generates the Hamiltonian, the relaxation superoperator, the kinetics superoperator, the Fokker-Planck spatial dynamics generator (including diffusion and flow), gradient operators, and passes all of that to the pulse sequence, which is supplied as a handle.

```
% Simulation
singlet_image=imaging(spin_system,@spen_singlet,parameters);
```

The sequence itself treats all of those operators as given:

```
function singlet_image=spen_singlet(spin_system,parameters,H,R,K,G,F)

% Compute the M2S echo train time
t=1/(4*sqrt(parameters.J^2+parameters.dv^2));
report(spin_system,['M2S echo train time: ' num2str(t) ' seconds']);

% Get M2S echo train dimensions
n_echoes=floor(pi*parameters.J/(2*parameters.dv));
if mod(n_echoes,2)~=0, n_echoes=n_echoes+1; end
report(spin_system,['M2S 1st echo train: ' num2str(n_echoes) ' pulses']);
report(spin_system,['M2S 2nd echo train: ' num2str(n_echoes/2) ' pulses']);

% Make sure the timing is sensible
if t-3*parameters.pulse_dur/2<0
    error('pulses are longer than M2S sequence permits.');
end

% Assemble the background evolution generator
```

```
B=H+F+1i*R+1i*K;

% Make pulse operators
Lp=operator(spin_system,'L+',parameters.spins{1});
Ly=polyadic({{speye(prod(parameters.npts)),(Lp-Lp')/2i}});
Lx=polyadic({{speye(prod(parameters.npts)),(Lp+Lp')/2}});

% Build and normalize a 90-degree shaped sinc/hamming pulse
rf_amp_list=pulse_shape('sinc3',parameters.pulse_nsteps).*hamming(parameters.pulse_nsteps)';
rf_amp_list=(pi/2)*rf_amp_list/(sum(rf_amp_list)*(parameters.pulse_dur/parame-
ters.pulse_nsteps));
rf_dur_list=(parameters.pulse_dur/parameters.pulse_nsteps)*ones(1,parameters.pulse_nsteps);

% Get the initial condition
rho=parameters.rho0;
```

After all of these preliminaries, we run slice-selective M2S sequence in a loop with shifting slice frequency and a fixed delay after each round:

```
% Run the slice-selective M2S for each slice
for n=1:numel(parameters.pulse_frq)

    % Apply a 90-degree shaped Y pulse at a specific frequency under Gx
    rho=shaped_pulse_af(spin_system,B+parameters.grad_ampl(1)*G{1},Lx,Ly,rho,...
                        parameters.pulse_frq(n)*ones(1,parameters.pulse_nsteps),...
                        rf_amp_list,rf_dur_list,pi/2,parameters.fp_rank,'expv');

    % Roll back the phase
    rho=step(spin_system,B-parameters.grad_ampl(1)*G{1},rho,parameters.pulse_dur/2);

    % Evolve the for an appropriate time
    rho=step(spin_system,B,rho,t-3*parameters.pulse_dur/2);

    % Apply a 180-degree shaped X pulse at a specific frequency under Gx
    rho=shaped_pulse_af(spin_system,B+parameters.grad_ampl(2)*G{1},Lx,Ly,rho,...
                        parameters.pulse_frq(n)*ones(1,parameters.pulse_nsteps),...
                      2*rf_amp_list,rf_dur_list,0,parameters.fp_rank,'expv');

    % Evolve the rest of the time
    rho=step(spin_system,B,rho,t-parameters.pulse_dur/2);


    % Run the first echo train
    for k=1:(n_echoes-2)

        % Let the system evolve
        rho=step(spin_system,B,rho,t-parameters.pulse_dur/2);

        % Apply a 180-degree shaped X pulse at a specific frequency under Gx
        rho=shaped_pulse_af(spin_system,B+parameters.grad_ampl(2)*G{1},Lx,Ly,rho,...
                            parameters.pulse_frq(n)*ones(1,parameters.pulse_nsteps),...
                          2*rf_amp_list,rf_dur_list,0,parameters.fp_rank,'expv');

        % Let the system evolve
        rho=step(spin_system,B,rho,t-parameters.pulse_dur/2);

    end

    % Let the system evolve
    rho=step(spin_system,B,rho,t-parameters.pulse_dur/2);

    % Apply a 180-degree shaped X pulse at a specific frequency under Gx
    rho=shaped_pulse_af(spin_system,B+parameters.grad_ampl(2)*G{1},Lx,Ly,rho,...
                        parameters.pulse_frq(n)*ones(1,parameters.pulse_nsteps),...
                      2*rf_amp_list,rf_dur_list,0,parameters.fp_rank,'expv');

    % Evolve the for an appropriate time
    rho=step(spin_system,B,rho,t-3*parameters.pulse_dur/2);

    % Pre-roll the phase
    rho=step(spin_system,B-parameters.grad_ampl(2)*G{1},rho,parameters.pulse_dur/2);

    % Apply a 90-degree shaped X pulse at a specific frequency under Gx
    rho=shaped_pulse_af(spin_system,B+parameters.grad_ampl(2)*G{1},Lx,Ly,rho,...
```

```
                        parameters.pulse_frq(n)*ones(1,parameters.pulse_nsteps),...
                        rf_amp_list,rf_dur_list,0,parameters.fp_rank,'expv');

    % Evolve the for an appropriate time
    rho=step(spin_system,B,rho,t-parameters.pulse_dur/2);

    % Run the second echo train
    for k=1:(n_echoes/2)

        % Let the system evolve
        rho=step(spin_system,B,rho,t-parameters.pulse_dur/2);

        % Apply a 180-degree shaped X pulse at a specific frequency under Gx
        rho=shaped_pulse_af(spin_system,B+parameters.grad_ampl(2)*G{1},Lx,Ly,rho,...
                            parameters.pulse_frq(n)*ones(1,parameters.pulse_nsteps),...
                          2*rf_amp_list,rf_dur_list,0,parameters.fp_rank,'expv');

        % Let the system evolve
        rho=step(spin_system,B,rho,t-parameters.pulse_dur/2);

    end

    % Let the system evolve (relaxation period, assuming no coherent evolution)
    rho=evolution(spin_system,F+1i*R,[],rho,parameters.rlx_time,1,'final');

end

% Return the answer
coil=singlet(spin_system,1,2); coil(1)=0;
singlet_image=fpl2phan(rho,coil,[parameters.npts 1]);

end
```

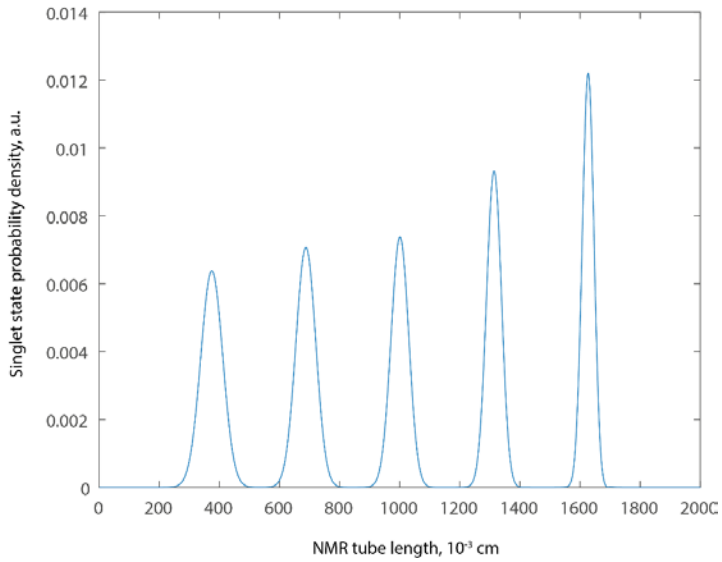


***Figure 43**. A simulation of the outcome of five slice-selective M2S singlet state excitation sequences performed at equal time interval at different points along the length of a standard NMR sample tube. The sharp initial distribution of the singlet state amplitude (rightmost signal) is gradually broadened by diffusion (from right to left). The calculation was performed in Spinach 1.9 and includes the full Liouville space of a two-spin system; the full dipolar relaxation superoperator is obtained using Redfield theory; diffusion is handled using the explicit finite difference diffusion operator within the Fokker-Plank equation for simultaneous spin and spatial dynamics.*

The result is a pulse sequence that populates several slices with a singlet state. As the next slice is being converted into a singlet, the previous ones undergo relaxation, meaning that the pulse sequence is wasting no time – it is a discrete equivalent of an ultrafast relaxation measurement sequence. The continuous version is in the works at the Pileio lab.

4.3.4 Bright fat effect in magnetic resonance imaging

Many researchers looked at the best way to improve fat imaging [161]. The reason for bright fat in MRI is that the protons in lipids have slightly different chemical environments. The slightly different chemical shifts make the protons in lipid precess at slightly different frequencies. It should be noted that TE's of 60-80 ms are typically used to produce T2-weighed images in conventional spin-echo imaging. By choosing TE's in this range, the signal recorded from fat is actually lower due to J-coupling effects than it might have been should a shorter or longer TE have been chosen.

In a sense, therefore, it might be said that fat on T2-weighted CSE imaging is abnormally dark, rather than the fat on FSE being abnormally bright. The point is bright fat effect is an essentially quantum mechanical effect that requires a multi-spin system to be present. It is therefore of interest as an application of the methods developed in this work.

One way of preventing magnetisation loss from happening is to use a train of 180° pulses with a spacing of 10 ms or less – which is enough to keep the spins collinear and thus to prevent the *J*-coupling evolution. In the early SE experiments by Hahn (1950) and Carr and Purcell (1954) [162] RF pulses were all applied along the same axis (usually x-direction). In practice, this method resulted in measured T2 values that were too short because of (1) cumulative phase errors from repetitive imperfect 180° pulses, and (2) B1 inhomogeneity effects that spread the magnetization out in a plane containing B1 and Bo. In 1958 Meiboom and Gill [163] proposed that such pulse-related errors could be reduced if the 180°-pulses in a SE train were phase shifted 90° with respect to the initial 90° pulse. In other words, if the 90°-pulse were applied along the x-axis, the 180°-pulses would be applied alternately along the ±y-axes. This technique, subsequently known by the acronym CPMG (Carr-Purcell-Meiboom-Gill), was extremely robust and is still employed on modern MR imagers when the SE technique is selected. It is the one of the techniques that is used to measure the relaxation times in liquid in the presence of diffusion. However, it was found later that CPMG pulse sequence is also useful to remove the effect of scalar coupling [164]. CPMG provided as 90° pulse followed by train of 180° pulses separated by uniform time interval τ .

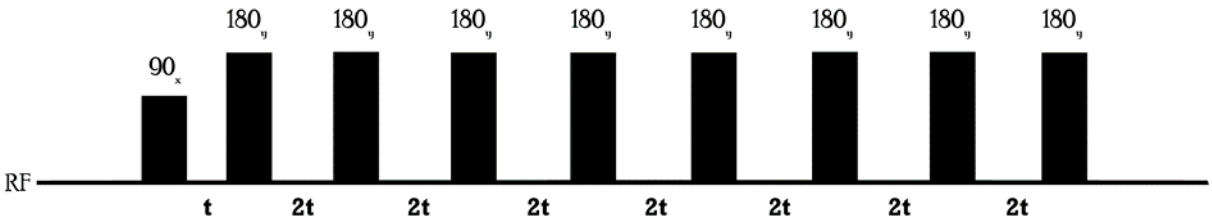


Figure 44. *CPMG pulse sequence. The idea is to use RF pulses in the same direction and uniform delay between the π pulses spin flips to stop the build-up of phase accumulation for a given spin. The performance of the sequence is improved somewhat by changing the phase of the π pulses relative to the initial π/2 excitation pulse.*

Figure 44 shows the details of CPMG pulse sequence. Recently, Uhrig [165] demonstrated that an optimized sequence of π pulses can increase the lifetime of coherence. After that, Warren *et al*. [166] showed the non-uniform time interval can improve fat signal. This technique is called Uhrig Dynamic decoupling (UDD). It is the same as CPMG in term of pulses but it's different in that the intervals between pulses are not uniform.

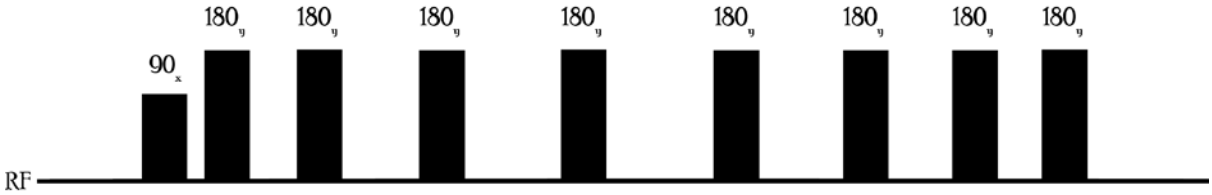


Figure 45. *UDD pulse sequence. The key point in this pulse sequence is using non-uniform delay between π pulses and the timing can be calculated by Equation* (246)*. The reason for that is recently found that unequal spacing can provide better refocusing than CPMG.*

It was observed that UDD technique can preserve electron spin coherence better than CPMG [167] followed that by a paper demonstrated that using Uhrig dynamic decoupling to single $^{43}Ca^{+}$ trapped-ion qubit can extend the coherence time of a single memory qubit [168]. However, more recent paper observed that CPMG can be better than UDD [169] in relaxometry results using tissue-mimicking gel phantoms. This is an interesting contradiction that we can try to resolve by high-fidelity simulations.

More recently, Warren *et al*. observed that the uniform time is not ideal for this kind of refocusing and the best timing for $n\pi$ pulses that the UDD spacing performs better when the $j\text{th}$ pulse should be located at time[166]:

$$\delta_j = \tau\left\{\sin^2\left(\pi j/\left(2n+2\right)\right)-0.5\right\} \tag{246}$$

Warren demonstrated this fact experimentally by getting different T2-weighted contrast between UDD and CPMG in a live mouse tumour model.

I have therefore performed within this project a fully quantum mechanical numerical simulation using the methods described above.

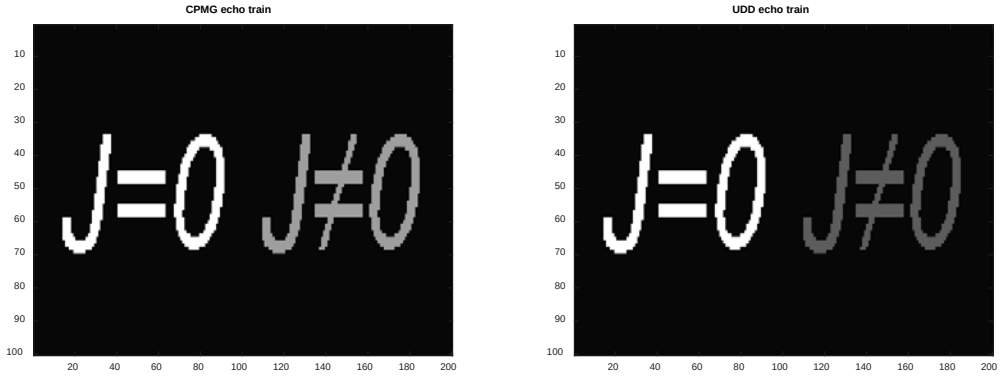


Figure 46. *Illustration of magnetisation loss into multiple-quantum coherences as a result of J-couplings being present. A spin echo imaging simulation was performed for a three-spin system with J = 11, 13, 17 Hz and chemical shifts of 1.0, 2.0, 3.0 ppm at 3.0 Tesla. Echo train length was 70 milliseconds, there were 32 inversion pulses in the train. Simulation results contradict literature claims that the UDD train is superior to CPMG.*

Our numerical results do not support this experimental observation – **Figure 46** illustrates that CPMG echo train yields a stronger signal for a *J*-coupled system. This supports the counter-claims that exist in the literature about the relative merits of CPMG and UDD [169] in relaxometry.

5 Research paper: spatio-temporal NMR

Published as *L. Guduff, A.J. Allami, C. van Heijenoort, J.N. Dumez, I. Kuprov, "Efficient simulation of ultrafast magnetic resonance experiments", Physical Chemistry Chemical Physics 19, 17577-17586* and reproduced here without changes.

5.1 Introduction

Spatial coordinates are more convenient in magnetic resonance than temporal ones – unlike the indelible past, any voxel in the $\mathbb{R}^3$ can be accessed repeatedly. The use of spatial coordinates as replacements for temporal ones was the key idea behind the "ultrafast" NMR methods proposed in 2002 by the Frydman group [34]. Those methods enabled the study of chemical [170] and biological [171, 172] processes on the time scale that was not previously accessible to multi-dimensional NMR [173]. Applied to MRI, the same approach enabled fast localised acquisition of 2D spectra [174]. Another useful application of the same idea is the family of "pure-shift" NMR experiments [175], fine-tuned at Manchester University around 2010 [176] – they finally achieved the dream of eliminating homonuclear *J*-couplings.

Spatial dimensions do have their problems though – the biggest one is the presence of uncontrolled dynamics (*e.g.* diffusion and convection) that is coupled to quantum mechanical evolution [177, 178]. Pulse sequence design must therefore rely on detailed modelling of simultaneous spatial, chemical and quantum dynamics, which would be wonderful *if only the required theory and software existed*: the usual Liouville - von Neumann equation is not good enough because there is no easy way to combine diffusion and hydrodynamics [179] with quantum mechanical spin evolution. The primary nuisance is the *J*-coupling [161, 180] – it dephases transverse magnetisation and generates undesirable modulations in echo trains [161, 181].

Very complicated systems of precisely this kind are emerging all over magnetic resonance: hyperpolarised pyruvate imaging [182] requires simultaneous modelling of quantum spin dynamics, diffusion, hydrodynamics, chemical kinetics and spin relaxation theory; the same applies to hyperpolarised singlet state imaging [159]. Pure-shift NMR [176], ultrafast NMR [183], metabolite-selective imaging [184], PARASHIFT contrast agents [185] and other similar recent developments are all united by their

theory and simulation infrastructure requirements that are a level above anything that is currently available.

The methods currently used in MRI (Bloch-Torrey equations [53, 186], distributed Bloch equations [187, 188], *k*-space Bloch equations [189], *etc.*) are very well developed [190], but insufficient for ultrafast NMR or singlet state MRI because all of those methods are built around Bloch equations for the spin degrees of freedom; this makes accurate treatment of coupled multi-spin systems impossible. Some methods (split propagation [191], random-walk averaging [95], *etc.*) do treat spin dynamics at the density matrix level, but introduce diffusion externally in an approximate way [192].

To our knowledge, the only formalism that simultaneously supports diffusion, hydrodynamics, chemical kinetics and Liouville-space spin dynamics in a general and computationally affordable way is the Fokker-Planck equation [193-195]. In this communication, we present the corresponding algebraic and numerical framework, and illustrate its performance by simulating (on the MRI side) a singlet state diffusion and flow imaging experiment, and (on the ultrafast NMR side) single-scan diffusion-ordered spectroscopy (DOSY) experiments that rely on spatial encoding of the diffusion dimension.

5.2 Simulation formalism

The key feature of ultrafast magnetic resonance experiments is the delicate interplay of spin and spatial dynamics. In designing the simulation methods for spatially encoded pulse sequences we will therefore be using the Fokker-Planck formalism [193, 194] discussed in detail by Kuprov [81, 196] and recently implemented in *Spinach* [197]. Its fundamental equation of motion is:

$$\begin{gathered}\frac{\partial\boldsymbol{\rho}(\mathbf{r},t)}{\partial t}=-i\mathbf{L}(\mathbf{r},t)\boldsymbol{\rho}(\mathbf{r},t)+\mathbf{M}(\mathbf{r},t)\boldsymbol{\rho}(\mathbf{r},t)\\ \mathbf{L}(\mathbf{r},t)\boldsymbol{\rho}(\mathbf{r},t)=\left[\mathbf{H}(\mathbf{r},t),\boldsymbol{\rho}(\mathbf{r},t)\right]+i\mathbf{R}\boldsymbol{\rho}(\mathbf{r},t)+i\mathbf{K}\boldsymbol{\rho}(\mathbf{r},t)\end{gathered}\tag{247}$$

where $\boldsymbol{\rho}(\mathbf{r},t)$ is the spin density matrix at the spatial point $\mathbf{r}$ at time t, $\mathbf{H}(\mathbf{r},t)$ is the spin Hamiltonian, square brackets denote a commutator, $\mathbf{R}$ is the relaxation superoperator, $\mathbf{K}$ is the chemical kinetics superoperator, and $\mathbf{M}(\mathbf{r},t)$ is the spatial dynamics generator dealing with diffusion and flow [81].

5.2.1 Fokker-Planck formalism for MRI and spatially encoded NMR

In liquid state spatially encoded NMR spectroscopy and imaging simulations, the laboratory frame spin Hamiltonian in Equation (247) contains the terms describing chemical shifts, pulsed field gradients, radiofrequency pulses and *J*-couplings [26]:

$$\begin{aligned}\mathbf{H}(\mathbf{r},t)=-\left[B_0+\mathbf{g}^{\mathrm{T}}(t)\cdot\mathbf{r}\right]\sum_n(1+\delta_n)\gamma_n\mathbf{S}_{\mathrm{Z}}^{(n)}-B_{1\mathrm{X}}\sum_n(1+\delta_n)\gamma_n\mathbf{S}_{\mathrm{X}}^{(n)}-\\-B_{1\mathrm{Y}}\sum_n(1+\delta_n)\gamma_n\mathbf{S}_{\mathrm{Y}}^{(n)}+2\pi\sum_{n<k}J_{nk}\left[\mathbf{S}_{\mathrm{X}}^{(n)}\mathbf{S}_{\mathrm{X}}^{(k)}+\mathbf{S}_{\mathrm{Y}}^{(n)}\mathbf{S}_{\mathrm{Y}}^{(k)}+\mathbf{S}_{\mathrm{Z}}^{(n)}\mathbf{S}_{\mathrm{Z}}^{(k)}\right]\end{aligned}\tag{248}$$

where B_0 is the primary magnet field (assumed to be directed along the Z axis), $\mathbf{g}$ is the primary magnet field gradient vector, δ_n are nuclear chemical shifts, γ_n are nuclear magnetogyric ratios, $\left\{\mathbf{S}_{\mathrm{X}}^{(n)},\mathbf{S}_{\mathrm{Y}}^{(n)},\mathbf{S}_{\mathrm{Z}}^{(n)}\right\}$ are nuclear spin operators, J_{nk} are inter-nuclear scalar couplings (traditionally published in Hz, hence the 2π in front) and $B_{1\{\mathrm{X,Y}\}}$ are the Cartesian components of the radiofrequency magnetic field, which is assumed to have no Z component.

The spatial dynamics generator in Equation (247) has contributions from diffusion and flow [81]:

$$\mathbf{M}(\mathbf{r},t)=\nabla^{\mathrm{T}}\cdot\mathbf{v}(\mathbf{r},t)+\mathbf{v}^{\mathrm{T}}(\mathbf{r},t)\cdot\nabla+\nabla^{\mathrm{T}}\cdot\mathbf{D}(\mathbf{r},t)\cdot\nabla\tag{249}$$

where $\nabla=\left[\partial/\partial x\quad\partial/\partial y\quad\partial/\partial z\right]^{\mathrm{T}}$ is the gradient (in the mathematical sense) operator, $\mathbf{v}(\mathbf{r},t)$ is the flow velocity and $\mathbf{D}(\mathbf{r},t)$ is the translational diffusion tensor. The latter rarely depends on time; the flow is also often stationary, meaning that the spatial dynamics generator in Equation (249) is time-independent and may be consigned to the background evolution operator – this is the primary advantage of the Fokker-Planck formalism over other simulation methods for the problem in question [81].

The use of the flow velocity in Equation (249) eliminates the thorny question of how to combine magnetic resonance simulations (where the equation of motion is linear) with hydrodynamics simulations (where it is not). Because spin degrees of freedom may be very accurately assumed to have no influence on the spatial motion, any hydrodynamics solver may simply be used first to obtain $\mathbf{v}(\mathbf{r},t)$ and put it into Equation (249). In that way, the equation of motion that we actually need to solve at the spin dynamics simulation stage – Equation (247) – stays linear with respect to the state vector $\boldsymbol{\rho}(\mathbf{r},t)$

.

Another advantage of the FP method the context of spatially encoded NMR and MRI is that RF pulses at any frequency have time-independent generators. The matter is discussed in detail in Kuprov's recent review of the subject [81]; here we will only note that the Fokker-Planck RF pulse generator is the derivative with respect to the RF phase, which is simply added to the spatial dynamics operator in Equation (249):

$$\mathbf{M} = \ldots + \omega_{\mathrm{RF}} \frac{\partial}{\partial \varphi} \tag{250}$$

where ω_{RF} is the angular frequency of the RF field and the phase φ occurs in the transverse magnetic field terms of Equation (248):

$$B_{1\mathrm{X}} = B_1 \cos\varphi, \qquad B_{1\mathrm{Y}} = B_1 \sin\varphi \tag{251}$$

The derivative term in Equation (250) has a simple physical meaning – its exponential is moving the radiofrequency phase forward at the rate ω_{RF}:

$$\exp\left[\omega_{\mathrm{RF}} t \frac{\partial}{\partial \varphi}\right] f(\varphi) = f(\varphi + \omega_{\mathrm{RF}} t) \tag{252}$$

and so the RF term looks like a static "interaction" within the Fokker-Planck dynamics generator.

Spin relaxation may be introduced in two different ways. The more general and computationally expensive method is to include the molecular rotation and the internal motion into the spatial dynamics generator $\mathbf{M}(\mathbf{r},t)$, and to treat the corresponding spatial coordinates explicitly – this is known as the Stochastic Liouville Equation (SLE) formalism [198, 199]. A somewhat less general, but also less computationally demanding formulation is to treat the molecular dynamics that drives the relaxation perturbatively, as an external stochastic process that rattles operator coefficients in the spin Hamiltonian – this is called Bloch-Redfield-Wangsness (BRW) theory [22, 23, 200]. BRW relaxation superoperator is added algebraically to the spin Hamiltonian commutation superoperator. The use of SLE becomes necessary in EPR systems where the BRW approach breaks down [201]. That is not the case here, and we are therefore using BRW relaxation theory in this work – its complete derivation and the technical details of its numerical implementation are published elsewhere [68, 202].

With all of this in place, Equation (247) has a simple general solution:

$$\boldsymbol{\rho}(\mathbf{r}, t+dt) = \exp\left\{\left[-i\mathbf{L}(\mathbf{r},t) + \mathbf{M}(\mathbf{r},t)\right]dt\right\}\boldsymbol{\rho}(\mathbf{r},t) \tag{253}$$

for which a great variety of numerical evaluation methods exist [203]. We are using Krylov propagation [204] for short events and shaped pulses, and scaling and squaring matrix exponential propagation

[130] for extended evolution periods. The primary technical problem is the generation of sufficiently small and accurate matrix representations for all operators involved [81, 205] – that is the subject of the next section.

5.2.2 Numerical implementation

The choice of the procedure for the generation of spin operators for Equation (248) depends on the size of the spin system. With fewer than ten spins, the standard Pauli matrix product method [26] is appropriate

$$\mathbf{S}_{\{\mathrm{X,Y,Z}\}}^{(k)}=\mathbf{1}\otimes\mathbf{1}\otimes\ldots\otimes\boldsymbol{\sigma}_{\{\mathrm{X,Y,Z}\}}\otimes\ldots\otimes\mathbf{1} \tag{254}$$

where $\mathbf{1}$ is a unit matrix of the dimension equal to the multiplicity of the corresponding spin and $\boldsymbol{\sigma}_{\{\mathrm{X,Y,Z}\}}$ are Pauli matrices [206] that occur in the *k*-th position in the direct product. Larger spin systems are best handled with the restricted state space approximation [64]; technical details of the spin operator generation process in restricted state spaces are given in our recent paper [205].

The most straightforward practical way of generating matrix representations for spatial derivative operators occurring in Equations (249) and (250) is to discretise the spatial coordinates on finite grids. For periodic boundary conditions, Fourier differentiation matrices on uniform grids [92] are convenient:

$$\left[\frac{\partial}{\partial x}\right]_{nk}=\begin{cases}\dfrac{(-1)^{n+k}}{2}\cot\left[\dfrac{(n-k)\pi}{N}\right] & n\neq k\\ 0 & n=k\end{cases} \tag{255}$$

Their disadvantage is high memory requirements – Fourier differentiation matrices have very few zeroes. A less memory-hungry alternative that is also compatible with non-periodic boundaries and non-uniform grids is finite-difference matrices, for which general formulas to arbitrary accuracy order and arbitrary point spacing have been derived by Fornberg [83]. For complicated diffusion and flow domain topologies, more sophisticated differential operator representations exist [207, 208] – they are not discussed here, but the salient point is that all of these approaches are well developed, highly automated and simply produce a matrix in the end. Here and below, we will assume rectangular domains with periodic boundaries.

Standard accuracy considerations apply to discrete representations of spatial dynamics operators: the Fourier differentiation method must have a minimum of two points per period of the fastest oscillation (this means gradient spirals in spatial dimensions), and the finite difference method must satisfy the

accuracy conditions formulated by Fornberg [83]. In practical calculations the number of points is increased from the minimum specified by the Larmor condition until convergence is achieved in the output.

Assuming *N*, *M* and *K* discretisation points in X, Y and Z respectively, the differentiation matrices are multiplied up into the three-dimensional space in the standard way, for example:

$$\mathfrak{D}_X = \left[\partial/\partial x\right]_N \otimes \mathbf{1}_M \otimes \mathbf{1}_K \tag{256}$$

where $\left[\partial/\partial x\right]_N$ is a matrix representation of the derivative operator on an *N*-point grid, and $\mathbf{1}$ are unit matrices of the indicated dimension. Similar expressions apply for derivatives along other dimensions.

The density matrix and the spin Hamiltonian are mapped into the Liouville space in the standard way [26]:

$$\begin{array}{rcl} \mathbf{H} & \rightarrow & \mathbf{1}\otimes\mathbf{H} - \mathbf{H}^{\mathrm{T}}\otimes\mathbf{1} \\ \boldsymbol{\rho} & \rightarrow & \mathrm{vec}\left(\boldsymbol{\rho}\right) \end{array} \tag{257}$$

where 1 is the unit matrix of the same dimension as the Hamiltonian, vec denotes the vectorisation operation – the columns of the matrix are placed sequentially one under the other to make a vector.

After the spin operators and the space operators are obtained individually, they are projected into the Fokker-Planck space by the final round of direct products, in which it is convenient to place spatial coordinates first. Therefore, the general form of a purely space operator is $\mathbf{A}\otimes\mathbf{1}$, the general form of a purely spin operator is $\mathbf{1}\otimes\mathbf{B}$, and for operators that connect spin and space degrees of freedom (such as pulsed field gradients) we have $\mathbf{A}\otimes\mathbf{B}$.

The numerical solution process for Equation (253) either involves a sparse matrix exponential (the exponentiation method [26]) or repeated multiplication of a full vector by a sparse matrix (the Krylov propagation method [204]). Both operations map well on GPU cards manufactured by *Intel* and *NVidia* using standard functionality implemented in *Matlab R2017a* and later. It bears notice that the complexity of the software implementation process is formidable and falls far beyond the scope of this paper; we would advise the reader to make use of the extensively annotated and documented open-source *Matlab* code that is freely available as a part of *Spinach* [197].

5.3 Spatially encoded diffusion spectroscopy applications

5.3.1 Ultrafast DOSY pulse sequence simulation

A good example of classical spatial dynamics coexisting with quantum spin dynamics in the same equation of motion is given by the ultrafast DOSY [209] pulse sequence (**Figure 47**). The sequence is derived from the conventional stimulated-echo DOSY, in which diffusion-encoding gradients (separated by a delay Δ so that spatial motion can have is effect on the spin precession phase) are replaced by frequency-swept chirp pulses applied under a gradient. Such chirp-and-gradient pairs are the defining feature of ultrafast NMR in general [183]. The other key ingredient in UF NMR is the use of a spectroscopic imaging acquisition block to acquire, in a single scan, both the direct dimension and the spatially encoded information.

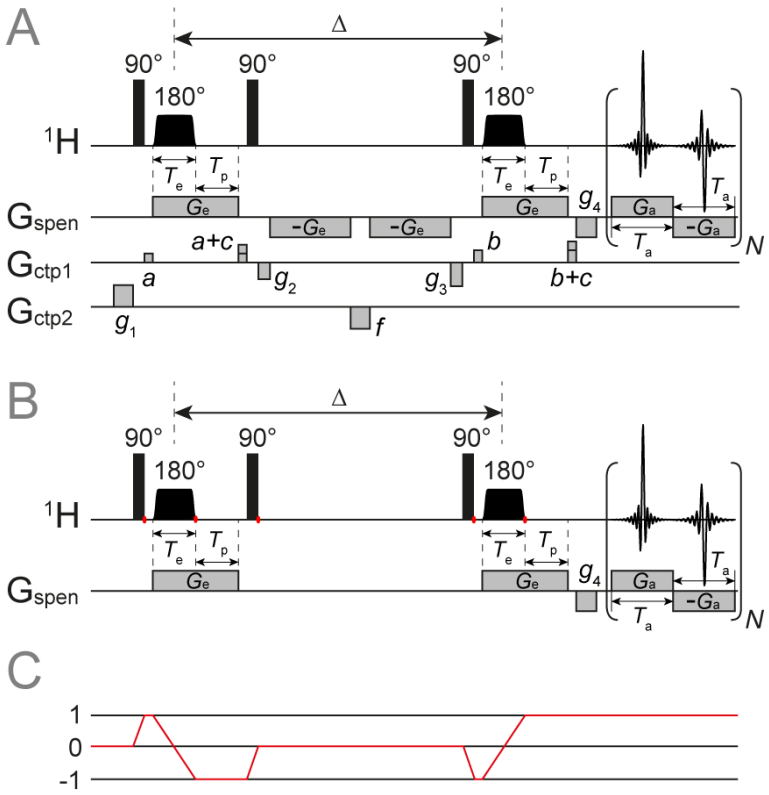


***Figure 47**: SPEN DOSY pulse sequencefor (A) the real experiment executed on the NMR instrument. Gradients a to c are used for coherence selection around the chirp pulses, f is a spoiler gradient, g_1, g_2 and g_3 are balancing pulses for lock signal retention, and g_4 is the prephasing gradient. T_e, T_p and T_a are the duration of the encoding gradient, the post-chirp gradient and the acquisition gradient respectively. G_a is the amplitude of the acquisition gradient, G_e is the amplitude of the encoding gradient. Chirp pulses combined with gradients pulsed are framing the diffusion delay of the stimulated echo so that the signal attenuation due to spin displacement during the diffusion delay Δ is position-dependent. (B) Simulated experiment, in which straightforward coherence selection steps had been replaced by analytical coherence selection commands (indicated by red dots) to improve the numerical performance. (C) coherence selection diagram.*

The principal advantage of the Fokker-Planck formalism here is that the diffusion operator is a part of the static background Hamiltonian. Another convenient feature is that gradient operators are presented to the user in the same algebraic style as spin operators – in Fokker-Planck space, they are matrices [81] that may either be added to the background Hamiltonian, or used with shaped pulse functions in the same way as spin operators. At the pulse sequence level, this makes simulation code neat and easy to write. Multi-spin systems with internal interactions are also straightforward – *Spinach* modifies the Hamiltonian and changes all matrix dimensions as appropriate [197]; pulse program code remains the same.

Figure 48a shows the simulated free induction decay obtained for the SPEN DOSY pulse sequence. It is an echo train created by the train of gradient pulses during acquisition. In order to extract the diffusion information, the data is rearranged in the *k*-*t* space (**Figure 48b**) and Fourier transformed along both dimensions (**Figure 48c**). The diffusion decay curve is then obtained as a slice of the 2D matrix. When the diffusion constant is set to zero, the spatial profile (dotted black curve in **Figure 48d**) reflects the smoothed chirp excitation profile. Otherwise (blue curve in **Figure 48d**), the decay induced by the diffusion is observable in the spatial profile. The plateau of the chirp pulse can then be used for diffusion coefficient determination.

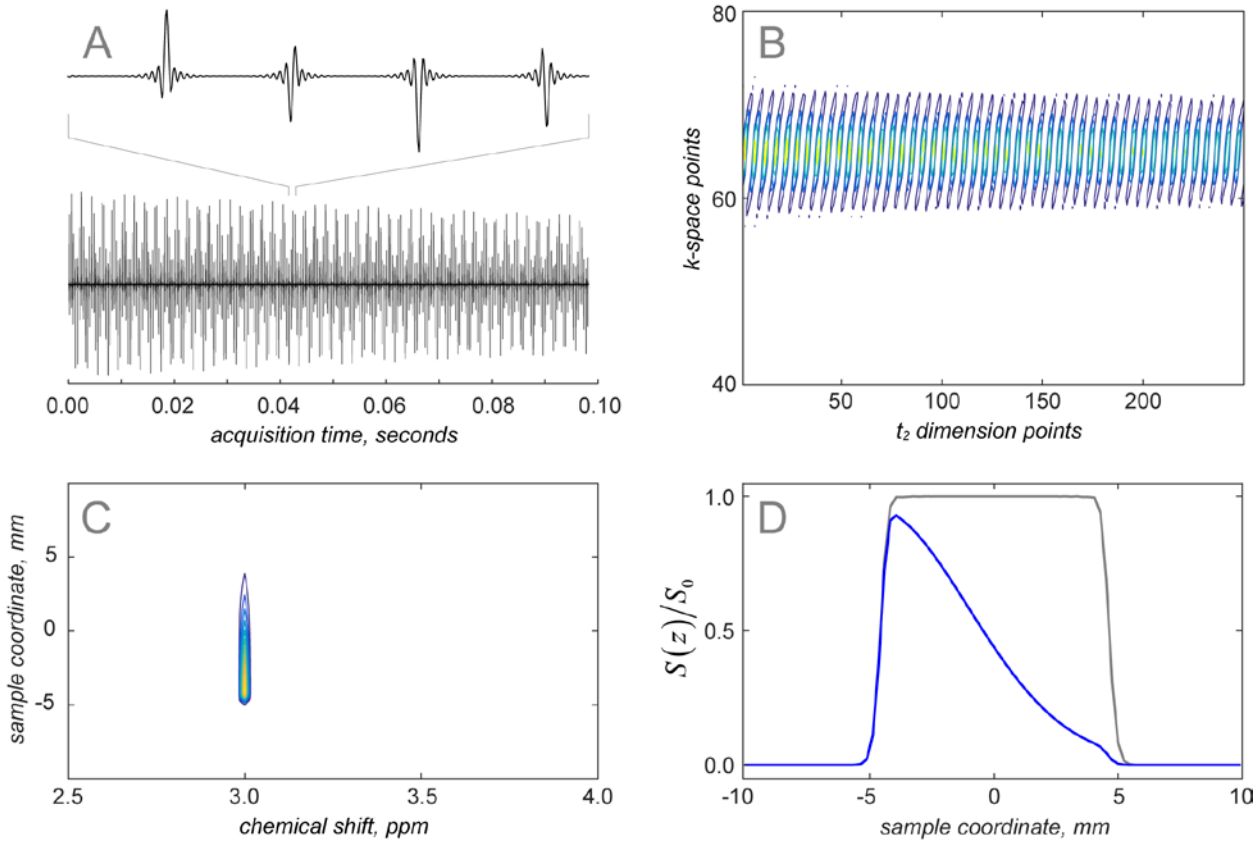


***Figure 48**: SPEN DOSY simulation using the Fokker-Planck formalism. The free induction decay (a) is a series of gradient echoes induced by the bipolar gradients. At the processing stage, it is rearranged into a matrix (b) with horizontal dimension corresponding to the t_2 evolution period, and vertical dimension to the k-space. (b). The spectrum (c) is obtained after Fourier transforms in both dimensions. The extracted diffusion profile (blue curve) and the reference signal recorded in the absence of diffusion (black dotted curve) are shown in (d). Physical dimensions of the sample are shown in the figure. Encoding gradients of 0.2535 T/m were applied with a chirp pulse of 110 kHz bandwidth and a duration of 1.5 ms. The post-chirp gradient had a duration of 1.6 ms. The acquisition was performed train of 256 bipolar gradients, where each gradient had an amplitude of 0.52 T/m, and a duration of 192 μs and 256 points. Diffusion constant was set to $8 \cdot 10^{-10}$ m^2/s (blue curve) or 0 (grey curve).*

5.3.2 Numerical accuracy and convergence

The formalism outlined in Section 4.2 is exact with respect to spin dynamics: the only accuracy parameter is the time step in Equation (253), which must be sufficiently small. In the spatial dynamics part, the essential factors that influence simulation accuracy are the spatial Nyquist condition (all gradient spirals must be adequately digitised) and the accuracy of the finite difference approximation to the spatial dynamics operators, in this case diffusion. The Nyquist condition [120] is that the spatial discretisation grid must have at least two points per period of the fastest gradient spiral.

As illustrated in the left panel of **Figure 49** and in Table 14, once the Nyquist condition is satisfied, distortions disappear abruptly and excellent convergence is achieved shortly thereafter. The accuracy of the finite difference approximation is controlled by the grid point spacing and the formal accuracy order of the finite difference stencil [83]. The effect of the accuracy order is illustrated, and compared to the formally exact (when Nyquist condition is satisfied) Fourier differentiation operator in the right

panel of **Figure 49**. Fourier differentiation operators are very accurate, but computationally expensive because they are not sparse.

Practical experience indicates that periodic boundary conditions (PBC) on spatial degrees of freedom are easier to implement and simulate than absorptive or reflective boundaries. Technically, the formalism above is not restricted to periodic boundaries (finite difference operators may be built to account for other boundary types), but the implementation process is time-consuming. The use of PBC adds another accuracy consideration: for non-periodic systems, sufficient white space should be allowed on all sides of the simulation box to avoid self-interaction across the periodic boundary.

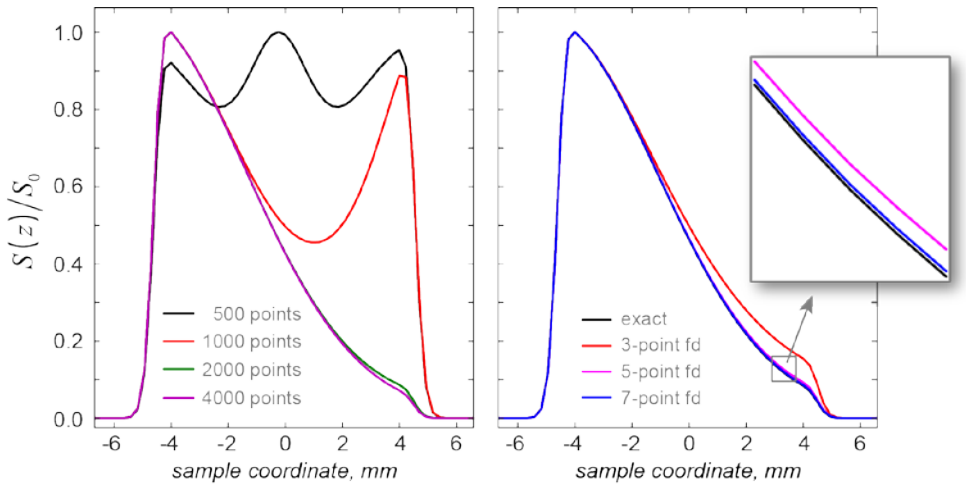


***Figure 49**: Left panel: the effect of the spatial Nyquist condition on the numerical accuracy of a SPEN DOSY simulation with the same physical parameters as in **Figure 48**: the grids with 500 and 1000 points are sub-Nyquist and result in catastrophic loss of accuracy. Grids with 2000 points and more satisfy the Nyquist condition and produce accurate results. Right panel: the effect of the accuracy order of the finite difference approximation [83] to the diffusion superoperator. The Fourier differentiation matrix (marked "exact") is formally exact on a given grid, but has a significantly longer run time (36 minutes) compared to the much sparser finite difference matrix (less than 2 minutes).*

Table 14 *Convergence of the diffusion coefficient retro-fitted (using Stejskal-Tanner equation) to the simulated SPEN DOSY profile in the left panel of Figure 3 as a function of the number of points in the spatial grid. 500-point and 1000-point grids are below the Nyquist condition for the gradient spirals. Note the linear scaling of the simulation time.*

Grid points	Fitted value of the diffusion coefficient, $\times 10^{-10}$ $m^2 s^{-1}$	Wall clock simulation time, seconds
500	–0.07	11.6
1000	1.10	17.6
2000	7.83	33.2
3000	7.97	52.2
4000	7.98	70.0
5000	7.98	90.3

It is impossible to give a general analytical formula for the minimum number of points in the spatial grid – each pulse sequence would have its own minimal grid size. However, a convenient practical diagnostic criterion may still be formulated: the spatial Fourier transform of the Fokker-Planck state vector should not have any probability density at the high-frequency edge points at any time in the simulation.

5.3.3 Experimental parameter optimisation

High-fidelity numerical simulations of the SPEN DOSY pulse sequence provide useful insight into the effects of pulse sequence parameters and instrumental imperfections. An example is shown in **Figure 50**– the exact shape of the chirp pulses in **Figure 47** determines the spatial profile of the excitation region. Normalisation by the spatial profile obtained in the absence of diffusion makes it possible to retrieve a larger range of attenuations for the diffusion decay, as illustrated in **Figure 50**. This also accounts for deviations from an ideal plateau due to factors like the B_1 field inhomogeneity, which is easy to model by assigning different amplitudes and directions of the control operators in Equation (248) to different voxels.

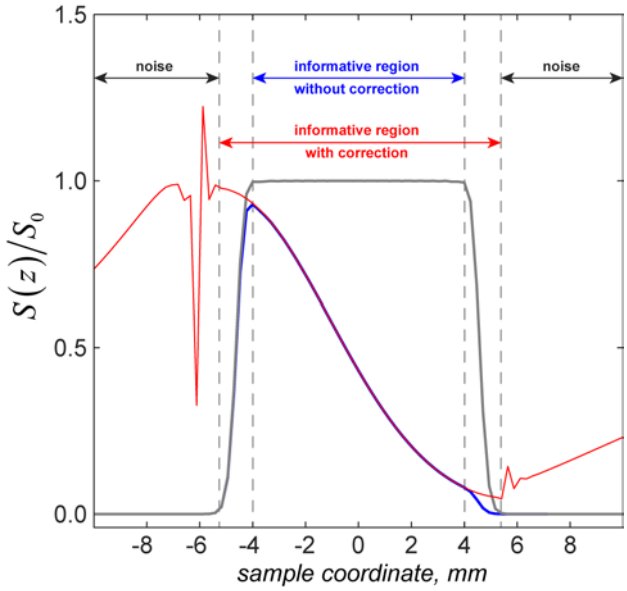


Figure 50. *Diffusion-weighted DOSY lineshape obtained under a linear gradient with (red line) and without (blue line) reference profile (dotted black line) correction. A significant increase in the usable signal area is evident in the background-corrected signal.*

Figure 5 shows the effect of the amplitude of the acquisition gradient G_a. It is clear that a sub-optimal choice of G_a can generate a bias (in this case, an over-estimation is produced) in the calculated diffusion constant because it yields an oscillating lineshape when decreased too much. The cause of the oscillation is signal truncation in *k*-space. This effect can be corrected when the reference profile is taken into account (Table 15) by dividing the diffusion profile by the reference profile computed with the diffusion coefficient set to zero. In our real experiments, G_a of 0.20 T/m is used; this confirms that it is necessary to correct the diffusion weighted image by a reference image with no diffusion present. Since it is impossible to switch the diffusion off in an actual experiment, the reference profile is obtained by cutting the corresponding pulses and evolution periods out of the pulse sequence in **Figure 47**.

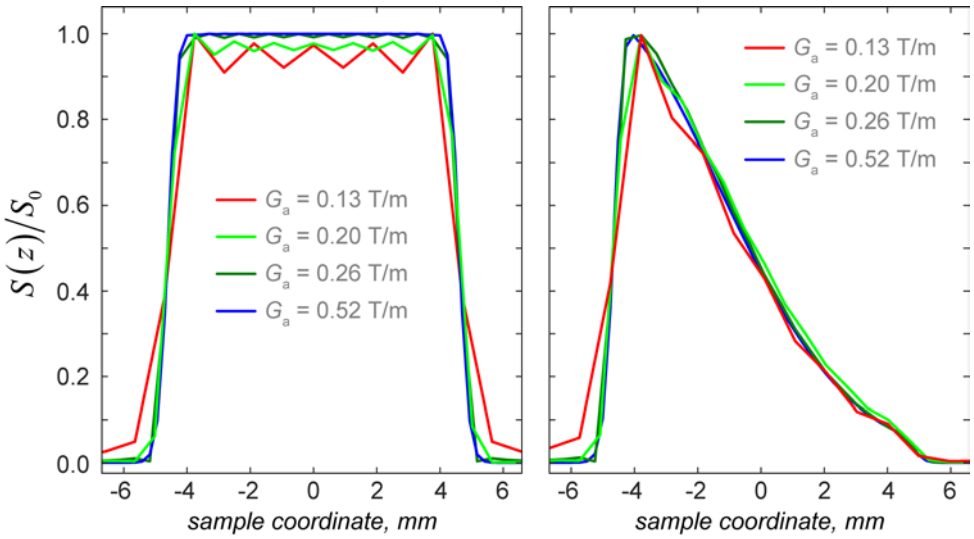


Figure 51. *The effect of gradient amplitude on the shape of the sample image in the absence of spatial diffusion (left) and with a typical diffusion constant of 8·10⁻¹⁰ m²/s (right). The oscillatory distortions, called Gibbs ringing, are due the truncation of the k-space signal.*

Table 15 *The effect of the acquisition gradient amplitude on the extracted diffusion coefficient (true value of 8.0 ·10–10 m2/s) with and without k-space truncation artefact correction.*

Acquisition Gradient (T/m)	D (raw) × 10^{-10} m^2/s	D (corrected) × 10^{-10} m^2/s
0.52	7.97	7.98
0.26	7.97	7.98
0.20	8.06	7.98
0.13	8.18	7.97

In order to extract diffusion coefficients from the experimental SPEN DOSY data, the gradient-induced attenuation in the signal $S(z)$ is usually fitted using the modified Stejskal-Tanner equation [209]:

$$S(z) = S_0 \exp\left[-DK(z)^2\left(\Delta - \frac{T_e + T_p}{2}\right)\right] \tag{258}$$

where D is the diffusion coefficient, Δ is the diffusion delay, T_e is the duration of the encoding gradient, T_p is the duration of the post-chirp gradient, and $K(z)$ is the effective gradient area at position z :

$$K(z) = \frac{d\varphi(z)}{dz} + \gamma G_e T_p \tag{259}$$

where G_e is encoding gradient amplitude, γ is the magnetogyric ratio for the nuclei in question, and the phase $\varphi(z)$ imparted by the chirp pulse is:

$$\varphi(z) = \int_0^{T_e} \sqrt{\left[\Omega_0 - \gamma G_e z - \Omega_{RF}(t)\right]^2 + \omega_{RF}(t)^2}\, dt \tag{260}$$

where Ω_0 is the frequency offset of the NMR signal, $\Omega_{RF}(t)$ is the time-dependent frequency of the RF sweep, $\omega_{RF}(t)$ is the time-dependent amplitude of the RF field.

Numerical simulations are also useful in assessing the effect of chemical shift offsets on the errors in the fitting of Equation (258) to the experimental SPEN DOSY data. Chemical shifts cause two types of signal displacements. The displacement upon encoding is accounted for in Equation (260),but the chemical shift displacement during acquisition also has to be taken into account, as it may otherwise results in significant errors. **Figure 52** shows the spatial diffusion profiles obtained for several values of the chemical shift offset, and Table 16 gives the corresponding fitted values of the diffusion coefficients, for corrected and uncorrected z values in Equation (258), with

$$z_{corr} = z - \frac{\Omega_0}{\gamma G_a} \tag{261}$$

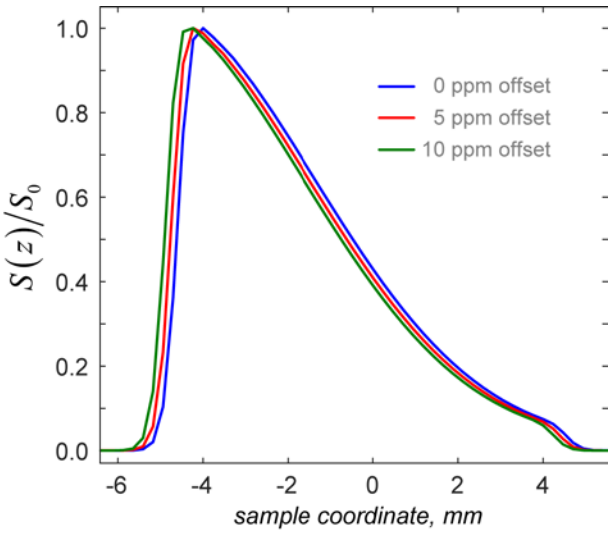


***Figure 52**. Simulated diffusion profile displacement for three different chemical shift offsets of the working signal from the resonance frequency. All other parameters as in Figures 3-5.*

For typical proton chemical shifts and gradient amplitudes of the same order as in Table 2, the chemical shift offset error in the diffusion coefficient is of the order of 5%; the decision on whether this is significant rests with the user, but Equation (261) is in any case straightforward to apply.

Table 16. *Diffusion coefficients extracted from raw and corrected DOSY profiles for different values of NMR signal offset from the resonance.*

Chemical shift (ppm)	D (raw) $\times 10^{-10}$ m^2/s	D (corrected) $\times 10^{-10}$ m^2/s
0	7.97	7.97
5	7.73	7.97
10	7.52	7.97

5.4 Quantum mechanical processes

An emerging class of MRI experiments that essentially requires quantum mechanical treatment of spin processes is singlet state imaging [95, 210, 211]. One reason is the delicate interplay of symmetry, chemical shift anisotropy and dipolar coupling in the relaxation superoperator that makes the two-spin singlet state long-lived [157]. The other is the M2S and S2M pulse sequences that rely on the *J*-coupling to get the spin system in and out of the singlet state [160]. Accurate Liouville space description of the spin dynamics [212] is therefore essential. At the same time, recent proposals for using singlet states as a delivery vehicle for hyperpolarisation in MRI [95, 210, 213] make it necessary to include diffusion and flow.

In this situation, the Fokker-Planck equation is clearly superior to the more traditional Liouville - von Neumann equation formalism because all spatial dynamics processes (diffusion, hydrodynamics, magic angle spinning, spatially selective pulses, *etc.*) are represented by *constant* matrices that are more convenient from the programming and numerical efficiency point of view than the time-dependent Hamiltonians in the Liouville - von Neumann formalism [81].

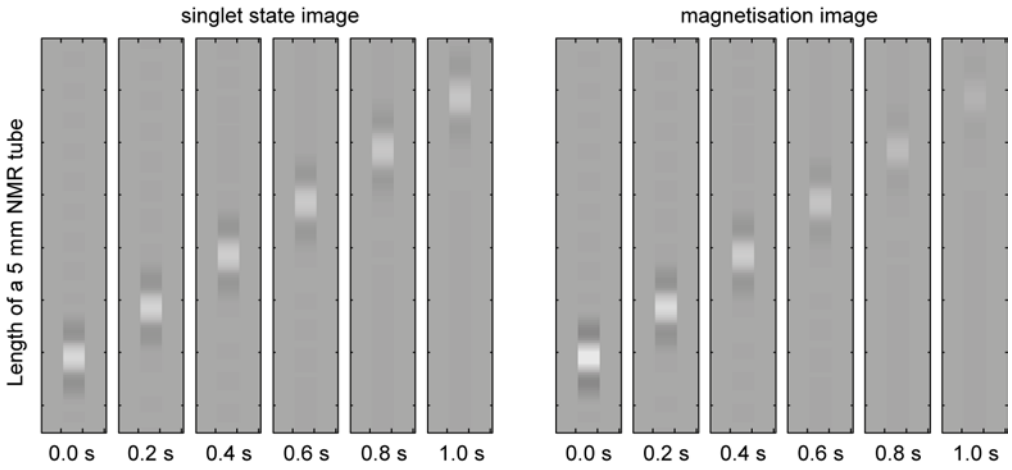


Figure 53. *Fokker-Planck theory simulation of a singlet state NMR imaging experiment in the presence of diffusion and flow. The simulation includes soft radiofrequency pulses with simultaneous explicit simulation of flow, diffusion, magnetic field gradients and spin-spin coupling dynamics as prescribed by the Liouville - von Neumann equation, as well as full Redfield relaxation superoperator treatment.*

The flow rate is set to 5 cm/s and the diffusion coefficient to $3.6 \cdot 10^{-6}$ m^2/s. The singlet imaging sequence used is described in the recent paper from the Pileio group [95]. It is clear that the very slowly relaxing singlet state flow (left panels) can be tracked at longer distances than magnetisation flow (right panels).

A simulation of the singlet state diffusion and flow experiment from Figure 2 of the recent paper by Pileio *et al.* [95] shown in **Figure 53** is particularly good example, because all of the complexities – shaped pulses, gradients, diffusion, flow, quantum mechanical spin evolution and complicated relaxation theory – are present simultaneously. An average over 100,000 Monte-Carlo instances had to be performed by the authors of the original paper, with only statistical guarantees of convergence [95]; here we demonstrate that the Fokker-Planck formalism performs the same simulation deterministically in a single run.

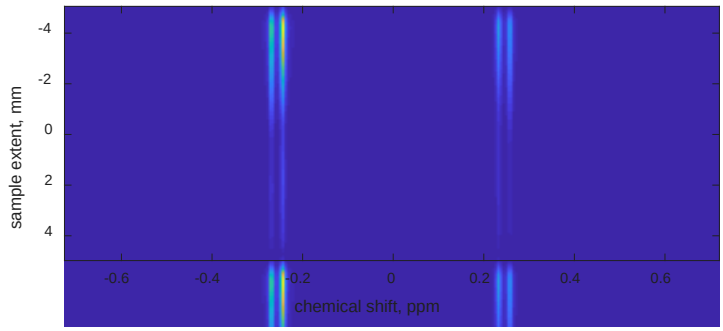


***Figure 54**: A simulated ultrafast DOSY spectrum for a two-proton spin system at 600 MHz with diffusion ($8 \cdot 10^{-10}$ m^2/s), flow (0.1 mm/s), chemical shift difference (0.5 ppm), strong J-coupling (15 Hz), full Redfield DD-CSA relaxation superoperator (2.5 Angstrom, CSA eigenvalues [-10 -10 20] ppm on both spins, with the tensor on spin 1 being collinear to the distance vector and the tensor on spin 2 being perpendicular to it, rotational correlation time 1.0 ns), and a non-symmetric chemical exchange process between the two spins (k_+=5 Hz, k_-=10 Hz). The calculation was set up in ten minutes and took less than five minutes to run with 3000 discretisation points in the spatial grid.*

Figure 54 illustrates the same point – within the Fokker-Planck formalism, diffusion and imaging simulations for systems with multiple coupled spins can be carried out in exactly the same way as for uncoupled spins – Equation (247) supports arbitrary spin Hamiltonian commutation superoperators, arbitrary relaxation theories and a host of other processes, such as chemical kinetics.

5.5 Conclusions and outlook

It is demonstrated above that NMR and MRI experiments with elaborate spatial encoding and complicated spatial dynamics are no longer hard to simulate, even in the presence of spin-spin couplings and exotic relaxation effects, such as singlet state symmetry lockout. Versions 1.10 and later of *Spinach*

library [197] support arbitrary stationary flows and arbitrary distributions of anisotropic diffusion tensors in three dimensions simultaneously with Liouville-space description of spin dynamics, chemical kinetics and relaxation processes. The key simulation design decision that has made this possible is the abandonment of Bloch-Torrey [53, 186] and Liouville - von Neumann [26, 161] formalisms in favour of the Fokker-Planck equation [81]. The primary factors that have facilitated this transition are the dramatic recent improvement in the speed and capacity of digital computers, the emergence of transparent and convenient sparse matrix manipulation methods in numerical linear algebra [214], and the recent progress in matrix dimension reduction in magnetic resonance simulations [197, 205].

In the more distant perspective, high-fidelity simulations that are free of significant approximations are expected to play an increasing role in magnetic resonance experiment design and in subsequent data processing. As optimal control theory [215, 216] illustrates, the understanding of the detailed dynamics taking place in a particular experiment is increasingly hard to achieve; it is slowly being replaced by the understanding of the factors generating the dynamics. Mathematically, this corresponds to the well-known relationship between a Lie group – usually an exceedingly complicated object – and the Lie algebra of its generators, which is fundamentally easier to understand and interpret. For complicated systems and processes, numerical simulation is the only practical way of connecting the two.

Conclusions and future work

It was previously impossible to simulate magnetic resonance experiments involving complex molecules and three-dimensional diffusion and flow MRI / NMR. Singlet state imaging, metabolite imaging, pure shift NMR, and ultrafast NMR all required quantum mechanical treatment of spin and classical description of spatial dynamics at the same level. This work puts such simulations within reach: Fokker-Planck formalism can deal with quantum mechanical treatment of spin and classical treatment of spatial dynamics simultaneously. This formalism is particularly convenient because stationary spatial dynamics generators (diffusion, stationary flow, fixed frequency off-resonance pulses, *etc.*) are represented by constant matrices that are more efficient from the programming point of view than the time-dependent Hamiltonians in the Liouville - von Neumann formalism.

This is a major improvement on the state of the art – the currently dominant Bloch-Torrey formalism cannot handle spin interactions, and the Liouville - von Neumann equation has difficulties with spatial dynamics. Fokker-Planck formalism is free of such limitations. This is demonstrated here using spatially encoded magnetic resonance spectroscopy of large spin systems distributed and moving in three spatial dimensions. Such systems can now be simulated in reasonable time, and essentially without approximations. Singlet imaging is one example; it requires shaped pulses, gradients, diffusion, flow, quantum mechanical spin evolution and complicated relaxation theory simultaneously. Further examples include ultrafast [94], spatially encoded pure-shift [176], and diffusion [217] NMR spectroscopy where spatial dynamics is inextricably linked to multi-spin processes.

The principal limitation of the formalism and the software presented in this work is the mismatch between the level of mathematical formalism that is used, and the level that is taught to the practitioners of NMR and MRI. The core insights that have made this work possible come from Lie groups and from the theory of spectral methods for partial differential equations. The equations are simple once written, but they are not easy to derive. The greatest challenge would therefore be in finding simple explanations that would allow chemists to use their intuition in the same way as they use it with the Bloch sphere and the product operator formalism.

The mathematics and the programming are now done, and the future work is mostly in the applications. The following general areas appear promising:

1- Sophisticated multi-spin effects in MRI applications are expected to become popular in the near future [218-220]. Attempts at writing in-house simulation packages are being made every year, but this is the first time when the complexity of the task is such that the *auteurs* might be put off and start using *Spinach*.

2- Spatial encoding in NMR spectroscopy is becoming increasingly popular [221]. While ultrafast variants of NMR correlation experiments will always be a niche market because of their sweep width limitations, spatial encoding in pure shift and diffusion spectroscopy will become more and more popular [222]. Again, this work is the only general simulation infrastructure in existence.

3- Studies of spatially distributed and MRI-detected kinetics of complex metabolites [223] will necessitate detailed theoretical modelling of the pulse sequences, as well as fitting of sophisticated models to the experimental data. In almost any scenario, the use of simulation techniques will increase.

4- Bringing the simulation infrastructure into physical units. Much of the current body of magnetic resonance simulation tools work in dimensionless units that may be hard to relate to the real world - for example, the unit part of the density operator is routinely ignored and the magnetisation part is scaled. Moving to software packages that make no simplifying assumptions would benefit the field.

References

[1] H.C. Torrey, Bloch equations with diffusion terms, Physical review, 104 (1956) 563.
[2] F. Bloch, Nuclear induction, Physical review, 70 (1946) 460.
[3] E.Y. Chekmenev, J. Hövener, V.A. Norton, K. Harris, L.S. Batchelder, P. Bhattacharya, B.D. Ross, D.P. Weitekamp, PASADENA hyperpolarization of succinic acid for MRI and NMR spectroscopy, Journal of the American Chemical Society, 130 (2008) 4212-4213.
[4] M. Goldman, H. Jóhannesson, O. Axelsson, M. Karlsson, Hyperpolarization of 13C through order transfer from parahydrogen: a new contrast agent for MRI, Magnetic resonance imaging, 23 (2005) 153-157.
[5] E.E. de Lange, T.A. Altes, J.T. Patrie, J.D. Gaare, J.J. Knake, J.P. Mugler III, T.A. Platts-Mills, Evaluation of asthma with hyperpolarized helium-3 MRI: correlation with clinical severity and spirometry, Chest, 130 (2006) 1055-1062.
[6] F.A. Gallagher, S.E. Bohndiek, M.I. Kettunen, D.Y. Lewis, D. Soloviev, K.M. Brindle, Hyperpolarized 13C MRI and PET: in vivo tumor biochemistry, Journal of Nuclear Medicine, 52 (2011) 1333-1336.
[7] M.S. Vinding, C. Laustsen, I.I. Maximov, L.V. Søgaard, J.H. Ardenkjær-Larsen, N.C. Nielsen, Dynamic nuclear polarization and optimal control spatial-selective 13C MRI and MRS, Journal of Magnetic Resonance, 227 (2013) 57-61.
[8] I.I. Rabi, Space quantization in a gyrating magnetic field, Physical Review, 51 (1937) 652.
[9] E.M. Purcell, H.C. Torrey, R.V. Pound, Resonance absorption by nuclear magnetic moments in a solid, Physical review, 69 (1946) 37.
[10] P.W. Atkins, R.S. Friedman, Molecular quantum mechanics, Oxford university press, 2011.
[11] J.C. Simpson, J.E. Lane, C.D. Immer, R.C. Youngquist, Simple analytic expressions for the magnetic field of a circular current loop, (2001).
[12] E.M. Haacke, R.W. Brown, M.R. Thompson, R. Venkatesan, Magnetic resonance imaging: physical principles and sequence design, Wiley-Liss New York:, 1999.
[13] D.C. Giancoli, Physics: principles with applications, Pearson/Prentice Hall, 2005.
[14] B.R. Martin, G. Shaw, Particle physics, John Wiley & Sons, 2017.
[15] D.E. Woessner, S. Zhang, M.E. Merritt, A.D. Sherry, Numerical solution of the Bloch equations provides insights into the optimum design of PARACEST agents for MRI, Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine, 53 (2005) 790-799.
[16] A. Abragam, H. Carr, The principles of nuclear magnetism, in, AIP, 1961.
[17] J. Keeler, Understanding NMR spectroscopy, John Wiley & Sons, 2011.
[18] S.A. Smith, W.E. Palke, J. Gerig, The hamiltonians of NMR. Part II, Concepts in magnetic resonance, 4 (1992) 181-204.
[19] M.H. Levitt, Spin dynamics: basics of nuclear magnetic resonance, John Wiley & Sons, 2001.
[20] R. Kosloff, Time-dependent quantum-mechanical methods for molecular dynamics, The Journal of Physical Chemistry, 92 (1988) 2087-2100.
[21] Y. Ren, B. Zhang, H. Qiao, A simple Taylor-series expansion method for a class of second kind integral equations, Journal of Computational and Applied Mathematics, 110 (1999) 15-24.
[22] M. Goldman, Formal theory of spin–lattice relaxation, Journal of Magnetic Resonance, 149 (2001) 160-187.
[23] A.G. Redfield, On the theory of relaxation processes, IBM Journal of Research and Development, 1 (1957) 19-31.
[24] M.H. Levitt, Singlet nuclear magnetic resonance, Annual review of physical chemistry, 63 (2012) 89-105.
[25] M. Carravetta, M.H. Levitt, Theory of long-lived nuclear spin states in solution nuclear magnetic resonance. I. Singlet states in low magnetic field, The Journal of chemical physics, 122 (2005) 214505.

[26] R.R. Ernst, G. Bodenhausen, A. Wokaun, Principles of nuclear magnetic resonance in one and two dimensions, Clarendon Press Oxford, 1987.
[27] M. Mehring, Principles of high resolution NMR in solids, Springer Science & Business Media, 2012.
[28] R. Kimmich, Irreducible Spherical Tensor Operators, in: NMR, Springer, 1997, pp. 455-466.
[29] N. Bloembergen, L. Morgan, Proton relaxation times in paramagnetic solutions. Effects of electron spin relaxation, The Journal of Chemical Physics, 34 (1961) 842-850.
[30] R. Freeman, S. Wittekoek, R. Ernst, High-Resolution NMR Study of Relaxation Mechanisms in a Two-Spin System, The Journal of Chemical Physics, 52 (1970) 1529-1544.
[31] M. Carravetta, M.H. Levitt, Long-lived nuclear spin states in high-field solution NMR, Journal of the American Chemical Society, 126 (2004) 6228-6229.
[32] J. Jeener, Ampere international summer school, Basko Polje, Yugoslavia, 197 (1971).
[33] P. Giraudeau, L. Frydman, Ultrafast 2D NMR: an emerging tool in analytical spectroscopy, Annual Review of Analytical Chemistry, 7 (2014) 129-161.
[34] L. Frydman, T. Scherf, A. Lupulescu, The acquisition of multidimensional NMR spectra within a single scan, Proceedings of the National Academy of Sciences, 99 (2002) 15858-15862.
[35] M. Nilsson, G.A. Morris, Pure shift proton DOSY: diffusion-ordered 1 H spectra without multiplet structure, Chemical communications, (2007) 933-935.
[36] J.A. Aguilar, M. Nilsson, G.A. Morris, Simple proton spectra from complex spin systems: pure shift NMR spectroscopy using BIRD, Angewandte Chemie, 123 (2011) 9890-9891.
[37] K. Zangger, H. Sterk, Homonuclear broadband-decoupled NMR spectra, Journal of Magnetic Resonance, 2 (1997) 486-489.
[38] P. Moutzouri, New and improved methods for mixture analysis by NMR, in, University of Manchester, 2018.
[39] D.W. McRobbie, E.A. Moore, M.J. Graves, MRI from Picture to Proton, Cambridge university press, 2017.
[40] S.A. Huettel, A.W. Song, G. McCarthy, Functional magnetic resonance imaging, Sinauer Associates Sunderland, MA, 2004.
[41] J. Hennig, A. Nauerth, H. Friedburg, RARE imaging: a fast imaging method for clinical MR, Magnetic resonance in medicine, 3 (1986) 823-833.
[42] A.S. Merbach, The chemistry of contrast agents in medical magnetic resonance imaging, John Wiley & Sons, 2013.
[43] D.W. McRobbie, E.A. Moore, M.J. Graves, M.R. Prince, MRI from Picture to Proton, Cambridge university press, 2017.
[44] S. Ljunggren, Imaging Methods, Journal of Magnetic Resonance, 54 (1983) 338-343.
[45] P. Mansfield, Multi-planar image formation using NMR spin echoes, Journal of Physics C: Solid State Physics, 10 (1977) L55.
[46] M. Tropper, Image reconstruction for the NMR echo-planar technique, and for a proposed adaptation to allow continuous data acquisition, Journal of Magnetic Resonance (1969), 42 (1981) 193-202.
[47] C. Westbrook, Handbook of MRI technique, John Wiley & Sons, 2014.
[48] J.P. Wansapura, S.K. Holland, R.S. Dunn, W.S. Ball Jr, NMR relaxation times in the human brain at 3.0 tesla, Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine, 9 (1999) 531-538.
[49] J. Bittoun, J. Taquin, M. Sauzade, A computer algorithm for the simulation of any nuclear magnetic resonance (NMR) imaging method, Magnetic Resonance Imaging, 2 (1984) 113-120.
[50] R.M. Summers, L. Axel, S. Israel, A computer simulation of nuclear magnetic resonance imaging, Magnetic resonance in medicine, 3 (1986) 363-376.
[51] M.B. Olsson, R. Wirestam, B.R. Persson, A computer simulation program for MR imaging: application to RF and static magnetic field imperfections, Magnetic Resonance in Medicine, 34 (1995) 612-617.

[52] P. Shkarin, R.G. Spencer, Direct simulation of spin echoes by summation of isochromats, Concepts in Magnetic Resonance, 8 (1996) 253-268.
[53] T.H. Jochimsen, A. Schäfer, R. Bammer, M.E. Moseley, Efficient simulation of magnetic resonance imaging with Bloch–Torrey equations using intra-voxel magnetization gradients, Journal of Magnetic Resonance, 180 (2006) 29-38.
[54] R. Kose, K. Kose, BlochSolver: A GPU-optimized fast 3D MRI simulator for experimentally compatible pulse sequences, Journal of Magnetic Resonance, 281 (2017) 51-65.
[55] R. Kose, A. Setoi, K. Kose, A Fast GPU-optimized 3D MRI Simulator for Arbitrary k-space Sampling, Magnetic Resonance in Medical Sciences, (2018) mp. 2018-0022.
[56] M. Veshtort, R.G. Griffin, SPINEVOLUTION: a powerful tool for the simulation of solid and liquid state NMR experiments, Journal of Magnetic Resonance, 178 (2006) 248-282.
[57] S. Stoll, A. Schweiger, EasySpin, a comprehensive software package for spectral simulation and analysis in EPR, Journal of magnetic resonance, 178 (2006) 42-55.
[58] S. Smith, T. Levante, B.H. Meier, R.R. Ernst, Computer simulations in magnetic resonance. An object-oriented programming approach, Journal of Magnetic Resonance, Series A, 106 (1994) 75-105.
[59] A. Abragam, A. Abragam, The principles of nuclear magnetism, Oxford university press, 1961.
[60] L.J. Edwards, D. Savostyanov, A. Nevzorov, M. Concistre, G. Pileio, I. Kuprov, Grid-free powder averages: On the applications of the Fokker–Planck equation to solid state NMR, Journal of Magnetic Resonance, 235 (2013) 121-129.
[61] I. Kuprov, N. Wagner-Rundell, P. Hore, Polynomially scaling spin dynamics simulation algorithm based on adaptive state-space restriction, Journal of Magnetic Resonance, 189 (2007) 241-250.
[62] I. Kuprov, Polynomially scaling spin dynamics II: further state-space compression using Krylov subspace techniques and zero track elimination, Journal of Magnetic Resonance, 195 (2008) 45-51.
[63] M.C. Butler, J.-N. Dumez, L. Emsley, Dynamics of large nuclear-spin systems from low-order correlations in Liouville space, Chemical Physics Letters, 477 (2009) 377-381.
[64] A. Karabanov, I. Kuprov, G. Charnock, A. van der Drift, L.J. Edwards, W. Köckenberger, On the accuracy of the state space restriction approximation for spin dynamics simulations, The Journal of chemical physics, 135 (2011) 084106.
[65] M. Krzystyniak, L.J. Edwards, I. Kuprov, Destination state screening of active spaces in spin dynamics simulations, Journal of Magnetic Resonance, 210 (2011) 228-232.
[66] H. Hogben, P. Hore, I. Kuprov, Strategies for state space restriction in densely coupled spin systems with applications to spin chemistry, The Journal of chemical physics, 132 (2010) 174101.
[67] L.J. Edwards, D. Savostyanov, Z. Welderufael, D. Lee, I. Kuprov, Quantum mechanical NMR simulation algorithm for protein-size spin systems, Journal of Magnetic Resonance, 243 (2014) 107-113.
[68] I. Kuprov, Diagonalization-free implementation of spin relaxation theory for large spin systems, Journal of Magnetic Resonance, 209 (2011) 31-38.
[69] H. Hogben, M. Krzystyniak, G. Charnock, P. Hore, I. Kuprov, Spinach–a software library for simulation of spin dynamics in large spin systems, Journal of Magnetic Resonance, 208 (2011) 179-194.
[70] I. Newton, The Principia: mathematical principles of natural philosophy, Univ of California Press, 1999.
[71] A. Jameson, Numerical solution of the Euler equations for compressible inviscid fluids, Numerical methods for the Euler equations of Fluid Dynamics, 1 (1985).
[72] G.K. Batchelor, An introduction to fluid dynamics, Cambridge university press, 2000.
[73] D.J. Acheson, Elementary fluid dynamics, in, ASA, 1991.
[74] N.I. Kolev, N.I. Kolev, Multiphase flow dynamics: Fundamentals, Springer, 2005.
[75] C. Foias, O. Manley, R. Rosa, R. Temam, Navier-Stokes equations and turbulence, Cambridge University Press, 2001.
[76] M. Taylor, Partial differential equations II: Qualitative studies of linear equations, Springer Science & Business Media, 2013.

[77] A.N. Brooks, T.J. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, Computer methods in applied mechanics and engineering, 32 (1982) 199-259.
[78] R. Moser, P. Moin, A. Leonard, A spectral numerical method for the Navier-Stokes equations with applications to Taylor-Couette flow, Journal of Computational Physics, 52 (1983) 524-544.
[79] X. He, L.-S. Luo, Lattice Boltzmann model for the incompressible Navier–Stokes equation, Journal of statistical Physics, 88 (1997) 927-944.
[80] P.T. Callaghan, Translational dynamics and magnetic resonance: principles of pulsed gradient spin echo NMR, Oxford University Press, 2011.
[81] I. Kuprov, Fokker-Planck formalism in magnetic resonance simulations, Journal of Magnetic Resonance, 270 (2016) 124-135.
[82] A. Fick, Ueber diffusion, Annalen der Physik, 170 (1855) 59-86.
[83] B. Fornberg, Generation of finite difference formulas on arbitrarily spaced grids, Mathematics of computation, 51 (1988) 699-706.
[84] S. Mazumder, Numerical methods for partial differential equations: finite difference and finite volume methods, Academic Press, 2015.
[85] E.N. Lorenz, Deterministic nonperiodic flow, Journal of the atmospheric sciences, 20 (1963) 130-141.
[86] D. Talay, Numerical solution of stochastic differential equations, (1994).
[87] N. Metropolis, Monte Carlo Method, From Cardinals to Chaos: Reflection on the Life and Legacy of Stanislaw Ulam, (1989) 125.
[88] H.R. Orlande, M.N. Özişik, M.J. Colaço, R.M. Cotta, Finite difference methods in heat transfer, CRC press, 2017.
[89] A. Taflove, S.C. Hagness, Computational electrodynamics: the finite-difference time-domain method, Artech house, 2005.
[90] D. Le Bihan, D. Jones, Diffusion MRI: theory, methods, and applications, in, Oxford University Press, Oxford Google Scholar, 2010.
[91] B.R. Munson, T.H. Okiishi, W.W. Huebsch, A.P. Rothmayer, Fluid mechanics, Wiley Singapore, 2013.
[92] L.N. Trefethen, Spectral methods in MATLAB, Siam, 2000.
[93] G. Pileio, J.-N. Dumez, I.-A. Pop, J.T. Hill-Cousins, R.C. Brown, Real-space imaging of macroscopic diffusion and slow flow by singlet tagging MRI, Journal of Magnetic Resonance, 252 (2015) 130-134.
[94] L. Guduff, A.J. Allami, C. van Heijenoort, J.-N. Dumez, I. Kuprov, Efficient simulation of ultrafast magnetic resonance experiments, Physical Chemistry Chemical Physics, 19 (2017) 17577-17586.
[95] G. Pileio, J.-N. Dumez, I.-A. Pop, J.T. Hill-Cousins, R.C.D. Brown, Real-space imaging of macroscopic diffusion and slow flow by singlet tagging MRI, Journal of Magnetic Resonance, 252 (2015) 130-134.
[96] T.E. Conturo, R.C. McKinstry, J.A. Aronovitz, J.J. Neil, Diffusion MRI: precision, accuracy and flow effects, NMR in Biomedicine, 8 (1995) 307-332.
[97] D. Le Bihan, J.F. Mangin, C. Poupon, C.A. Clark, S. Pappata, N. Molko, H. Chabriat, Diffusion tensor imaging: concepts and applications, Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine, 13 (2001) 534-546.
[98] K. Zangger, Pure shift NMR, Progress in nuclear magnetic resonance spectroscopy, 86 (2015) 1-20.
[99] P.A. Bottomley, Spatial localization in NMR spectroscopy in vivo, Annals of the New York Academy of Sciences, 508 (1987) 333-348.
[100] M. Castillo, L. Kwock, S.K. Mukherji, Clinical applications of proton MR spectroscopy, American Journal of Neuroradiology, 17 (1996) 1-15.
[101] S.J. Sawiak, B. Jupp, T. Taylor, D. Caprioli, T.A. Carpenter, J.W. Dalley, In vivo γ-aminobutyric acid measurement in rats with spectral editing at 4.7 T, Journal of Magnetic Resonance Imaging, 43 (2016) 1308-1312.

[102] R. Richterich, C. IV. a. 1. Pyruvate: Enzymatic determination with lactic dehydrogenase, in: Clinical Chemistry, Karger Publishers, 1969, pp. 219-223.
[103] L. Gladden, Lactate metabolism: a new paradigm for the third millennium, The Journal of physiology, 558 (2004) 5-30.
[104] O. Gruber, A. Santuccione Chadha, H. Aach, Magnetic resonance imaging in studying schizophrenia, negative symptoms, and the glutamate system, Frontiers in psychiatry, 5 (2014) 32.
[105] R. Sener, Proton MR spectroscopy demonstration of taurine peaks in megalencephalic leukoencephalopathy with cysts, Computerized medical imaging and graphics, 27 (2003) 23-26.
[106] B. Leporq, S.A. Lambert, M. Ronot, V. Vilgrain, B.E. Van Beers, Quantification of the triglyceride fatty acid composition with 3.0 T MRI, NMR in Biomedicine, 27 (2014) 1211-1221.
[107] D. Grucker, J. Chambron, Oxygen imaging in perfused hearts by dynamic nuclear polarization, Magnetic resonance imaging, 11 (1993) 691-696.
[108] P. Bhattacharya, E.Y. Chekmenev, W.F. Reynolds, S. Wagner, N. Zacharias, H.R. Chan, R. Bünger, B.D. Ross, Parahydrogen-induced polarization (PHIP) hyperpolarized MR receptor imaging in vivo: a pilot study of 13C imaging of atheroma in mice, NMR in Biomedicine, 24 (2011) 1023-1028.
[109] M. Bak, J.T. Rasmussen, N.C. Nielsen, SIMPSON: a general simulation program for solid-state NMR spectroscopy, Journal of magnetic resonance, 213 (2011) 366-400.
[110] M. Hochbruck, C. Lubich, On Krylov subspace approximations to the matrix exponential operator, SIAM Journal on Numerical Analysis, 34 (1997) 1911-1925.
[111] P. Fernandes, B. Plateau, W.J. Stewart, Efficient descriptor-vector multiplications in stochastic automata networks, J. ACM, 45 (1998) 381-414.
[112] C. Hirsch, Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics, Elsevier, 2007.
[113] J. Jeener, B. Meier, P. Bachmann, R. Ernst, Investigation of exchange processes by two-dimensional NMR spectroscopy, The Journal of chemical physics, 71 (1979) 4546-4553.
[114] R.J. Kee, F.M. Rupley, J.A. Miller, Chemkin-II: A Fortran chemical kinetics package for the analysis of gas-phase chemical kinetics, in, Sandia National Labs., Livermore, CA (USA), 1989.
[115] J. Tu, G.H. Yeoh, C. Liu, Computational fluid dynamics: a practical approach, Butterworth-Heinemann, 2018.
[116] W.W. Hager, Condition estimates, SIAM Journal on scientific and statistical computing, 5 (1984) 311-316.
[117] I.V. Oseledets, Tensor-train decomposition, SIAM Journal on Scientific Computing, 33 (2011) 2295-2317.
[118] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, Annals of Physics, 326 (2011) 96-192.
[119] D. Savostyanov, S. Dolgov, J. Werner, I. Kuprov, Exact NMR simulation of protein-size spin systems using tensor train formalism, Physical Review B, 90 (2014) 085139.
[120] H. Nyquist, Certain topics in telegraph transmission theory, Transactions of the American Institute of Electrical Engineers, 47 (1928) 617-644.
[121] E.O. Stejskal, J.E. Tanner, Spin diffusion measurements: spin echoes in the presence of a time-dependent field gradient, The journal of chemical physics, 42 (1965) 288-292.
[122] P. Stilbs, Fourier transform pulsed-gradient spin-echo studies of molecular diffusion, Progress in nuclear magnetic resonance spectroscopy, 19 (1987) 1-45.
[123] R. Cotts, M. Hoch, T. Sun, J. Markert, Pulsed field gradient stimulated echo methods for improved NMR diffusion measurements in heterogeneous systems, Journal of Magnetic Resonance (1969), 83 (1989) 252-266.
[124] D. Wu, A. Chen, C.S. Johnson, An improved diffusion-ordered spectroscopy experiment incorporating bipolar-gradient pulses, Journal of magnetic resonance, Series A, 115 (1995) 260-264.
[125] R.W. Adams, J.A. Aguilar, J. Cassani, G.A. Morris, M. Nilsson, Resolving natural productepimer spectra by matrix-assisted DOSY, Organic & biomolecular chemistry, 9 (2011) 7062-7064.

[126] Z. Bai, Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems, Applied numerical mathematics, 43 (2002) 9-44.
[127] R.W. Freund, Model reduction methods based on Krylov subspaces, Acta Numerica, 12 (2003) 267-319.
[128] L.J. Edwards, I. Kuprov, Parallel density matrix propagation in spin dynamics simulations, The Journal of chemical physics, 136 (2012) 044108.
[129] R.J. Lipton, R.E. Tarjan, A separator theorem for planar graphs, SIAM Journal on Applied Mathematics, 36 (1979) 177-189.
[130] C. Moler, C. Van Loan, Nineteen dubious ways to compute the exponential of a matrix, SIAM review, 20 (1978) 801-836.
[131] C.G. Xanthis, I.E. Venetis, A. Chalkias, A.H. Aletras, MRISIMUL: a GPU-based parallel approach to MRI simulations, IEEE transactions on medical imaging, 33 (2014) 607-617.
[132] Z. Cao, S. Oh, C.T. Sica, J.M. McGarrity, T. Horan, W. Luo, C.M. Collins, Bloch-based MRI system simulator considering realistic electromagnetic fields for calculation of signal, noise, and specific absorption rate, Magnetic resonance in medicine, 72 (2014) 237-247.
[133] D. Graverondemilly, A. Diop, A. Briguet, B. Fenet, Product-operator algebra for strongly coupled spin systems, Journal of Magnetic Resonance, Series A, 101 (1993) 233-239.
[134] M. Bak, J.T. Rasmussen, N.C. Nielsen, SIMPSON: a general simulation program for solid-state NMR spectroscopy, Journal of Magnetic Resonance, 147 (2000) 296-330.
[135] T. Allman, A.D. Bain, J.R. Garbow, SIMPLTN, a program for the simulation of pulse NMR spectra, Journal of Magnetic Resonance, Series A, 123 (1996) 26-31.
[136] C. Schorn, B.F. Taylor, B.J. Taylor, NMR-spectroscopy: Data acquisition, John Wiley & Sons, 2004.
[137] C. Bengs, M.H. Levitt, SpinDynamica: Symbolic and numerical magnetic resonance in a Mathematica environment, Magnetic Resonance in Chemistry, 56 (2018) 374-414.
[138] K.P. Pruessmann, M. Weiger, M.B. Scheidegger, P. Boesiger, SENSE: sensitivity encoding for fast MRI, Magnetic resonance in medicine, 42 (1999) 952-962.
[139] P. Mansfield, Snap-shot MRI, Physics-Uspekhi, 48 (2005).
[140] D. Sinnaeve, The Stejskal–Tanner equation generalized for any gradient shape—an overview of most pulse sequences measuring free diffusion, Concepts in Magnetic Resonance Part A, 40 (2012) 39-65.
[141] M.G. Myers Jr, D.P. Olson, Central nervous system control of metabolism, Nature, 491 (2012) 357.
[142] J.M. Duarte, H. Lei, V. Mlynárik, R. Gruetter, The neurochemical profile quantified by in vivo 1H NMR spectroscopy, Neuroimage, 61 (2012) 342-362.
[143] A.V.D. Toorn, R.M. Dijkhuizen, C.A. Tulleken, K. Nicolay, T1 and T2 relaxation times of the major 1H-containing metabolites in rat brain after focal ischemia, NMR in Biomedicine, 8 (1995) 245-252.
[144] S. Blüml, Magnetic resonance spectroscopy: basics, in: MR spectroscopy of pediatric brain disorders, Springer, 2013, pp. 11-23.
[145] (!!! INVALID CITATION !!! [141-143]).
[146] F. Howe, R. Maxwell, D. Saunders, M. Brown, J. Griffiths, Proton spectroscopy in vivo, Magnetic resonance quarterly, 9 (1993) 31-59.
[147] G. Brix, S. Heiland, M.E. Bellemann, T. Koch, W.J. Lorenz, MR imaging of fat-containing tissues: valuation of two quantitative imaging techniques in comparison with localized proton spectroscopy, Magnetic resonance imaging, 11 (1993) 977-991.
[148] D.E. Befroy, G.I. Shulman, Magnetic resonance spectroscopy studies of human metabolism, Diabetes, 60 (2011) 1361-1369.
[149] T.A. Bley, O. Wieben, C.J. François, J.H. Brittain, S.B. Reeder, Fat and water magnetic resonance imaging, Journal of Magnetic Resonance Imaging, 31 (2010) 4-18.
[150] F. Del Grande, F. Santini, D.A. Herzka, M.R. Aro, C.W. Dean, G.E. Gold, J.A. Carrino, Fat-suppression techniques for 3-T MR imaging of the musculoskeletal system, Radiographics, 34 (2014) 217-233.

[151] A. Haase, J. Frahm, W. Hanicke, D. Matthaei, 1H NMR chemical shift selective (CHESS) imaging, Physics in Medicine & Biology, 30 (1985) 341.
[152] K. Stott, J. Keeler, Q.N. Van, A. Shaka, One-dimensional NOE experiments using pulsed field gradients, Journal of Magnetic Resonance, 125 (1997) 302-324.
[153] G. Yan, T. Zhang, Z. Dai, M. Yi, Y. Jia, T. Nie, H. Zhang, G. Xiao, R. Wu, A Potential Magnetic Resonance Imaging Technique Based on Chemical Exchange Saturation Transfer for In Vivo γ-Aminobutyric Acid Imaging, PloS one, 11 (2016) e0163765.
[154] R.A. De Graaf, In vivo NMR spectroscopy: principles and techniques, John Wiley & Sons, 2013.
[155] J.-N. Dumez, J.T. Hill-Cousins, R.C. Brown, G. Pileio, Long-lived localization in magnetic resonance imaging, Journal of Magnetic Resonance, 246 (2014) 27-30.
[156] C. Laustsen, G. Pileio, M.C. Tayler, L.J. Brown, R.C. Brown, M.H. Levitt, J.H. Ardenkjaer-Larsen, Hyperpolarized singlet NMR on a small animal imaging system, Magnetic resonance in medicine, 68 (2012) 1262-1265.
[157] G. Pileio, Relaxation theory of nuclear singlet states in two spin-1/2 systems, Progress in nuclear magnetic resonance spectroscopy, 56 (2010) 217-231.
[158] A.D. Bain, J.S. Martin, FT NMR of nonequilibrium states of complex spin systems. I. A Lionville space description, Journal of Magnetic Resonance (1969), 29 (1978) 125-135.
[159] G. Pileio, S. Bowen, C. Laustsen, M.C. Tayler, J.T. Hill-Cousins, L.J. Brown, R.C. Brown, J.H. Ardenkjaer-Larsen, M.H. Levitt, Recycling and imaging of nuclear singlet hyperpolarization, Journal of the American Chemical Society, 135 (2013) 5084-5088.
[160] G. Pileio, M. Carravetta, M.H. Levitt, Storage of nuclear magnetization as long-lived singlet order in low magnetic field, Proceedings of the National Academy of Sciences, 107 (2010) 17135-17139.
[161] L. Stables, R. Kennan, A. Anderson, J. Gore, Density matrix simulations of the effects of J coupling in spin echo and fast spin echo imaging, Journal of magnetic resonance, 140 (1999) 305-314.
[162] H.Y. Carr, E.M. Purcell, Effects of diffusion on free precession in nuclear magnetic resonance experiments, Physical review, 94 (1954) 630.
[163] S. Meiboom, D. Gill, Modified spin-echo method for measuring nuclear relaxation times, Review of scientific instruments, 29 (1958) 688-691.
[164] A. Allerhand, E. Thiele, Analysis of Carr—Purcell Spin-Echo NMR Experiments on Multiple-Spin Systems. II. The Effect of Chemical Exchange, The Journal of Chemical Physics, 45 (1966) 902-916.
[165] G.S. Uhrig, Keeping a quantum bit alive by optimized π-pulse sequences, Physical Review Letters, 98 (2007) 100504.
[166] E.R. Jenista, A.M. Stokes, R.T. Branca, W.S. Warren, Optimized, unequal pulse spacing in multiple echo sequences improves refocusing in magnetic resonance, The Journal of chemical physics, 131 (2009) 204510.
[167] J. Du, X. Rong, N. Zhao, Y. Wang, J. Yang, R. Liu, Preserving electron spin coherence in solids by optimal dynamical decoupling, Nature, 461 (2009) 1265.
[168] D.J. Szwer, S.C. Webster, A.M. Steane, D.M. Lucas, Keeping a single qubit alive by experimental dynamic decoupling, Journal of Physics B: Atomic, Molecular and Optical Physics, 44 (2010) 025501.
[169] S. Schirmer, J. Phillips, Comparison of CPMG and Uhrig Dynamic Decoupling (UDD) for tissue refocusing in MRI, arXiv preprint arXiv:1705.03292, (2017).
[170] Z.D. Pardo, G.L. Olsen, M.E. Fernandez-Valle, L. Frydman, R. Martinez-Alvarez, A. Herrera, Monitoring mechanistic details in the synthesis of pyrimidines via real-time, ultrafast multidimensional NMR spectroscopy, J Am Chem Soc, 134 (2012) 2706-2715.
[171] M. Gal, T. Kern, P. Schanda, L. Frydman, B. Brutscher, An improved ultrafast 2D NMR experiment: towards atom-resolved real-time studies of protein kinetics at multi-Hz rates, Journal of biomolecular NMR, 43 (2009) 1-10.
[172] P. Schanda, E. Kupce, B. Brutscher, SOFAST-HMQC experiments for recording two-dimensional heteronuclear correlation spectra of proteins within a few seconds, Journal of biomolecular NMR, 33 (2005) 199-211.

[173] P. Giraudeau, L. Frydman, Ultrafast 2D NMR: an emerging tool in analytical spectroscopy, Annual review of analytical chemistry, 7 (2014) 129-161.
[174] R. Schmidt, L. Frydman, New spatiotemporal approaches for fully refocused, multislice ultrafast 2D MRI, Magn Reson Med, 71 (2014) 711-722.
[175] K. Zangger, H. Sterk, Homonuclear Broadband-Decoupled NMR Spectra, Journal of Magnetic Resonance, 124 (1997) 486-489.
[176] L. Paudel, R.W. Adams, P. Kiraly, J.A. Aguilar, M. Foroozandeh, M.J. Cliff, M. Nilsson, P. Sandor, J.P. Waltho, G.A. Morris, Simultaneously enhancing spectral resolution and sensitivity in heteronuclear correlation NMR spectroscopy, Angewandte Chemie, 52 (2013) 11616-11619.
[177] P. Giraudeau, S. Akoka, Sensitivity losses and line shape modifications due to molecular diffusion in continuous encoding ultrafast 2D NMR experiments, Journal of Magnetic Resonance, 195 (2008) 9-16.
[178] Y. Shrot, L. Frydman, The effects of molecular diffusion in ultrafast two-dimensional nuclear magnetic resonance, The Journal of chemical physics, 128 (2008) 164513.
[179] I. Kuprov, Fokker-Planck formalism in magnetic resonance simulations, Journal of magnetic resonance, 270 (2016) 124-135.
[180] P.A. Hardy, R.M. Henkelman, J.E. Bishop, E.C.S. Poon, D.B. Plewes, Why fat is bright in rare and fast spin-echo imaging, Journal of magnetic resonance imaging, 2 (1992) 533-540.
[181] A.M. Stokes, Y. Feng, T. Mitropoulos, W.S. Warren, Enhanced refocusing of fat signals using optimized multipulse echo sequences, Magnetic Resonance in Medicine, 69 (2013) 1044-1055.
[182] S.J. Nelson, J. Kurhanewicz, D.B. Vigneron, P.E. Larson, A.L. Harzstark, M. Ferrone, M. van Criekinge, J.W. Chang, R. Bok, I. Park, Metabolic imaging of patients with prostate cancer using hyperpolarized [1-13C] pyruvate, Science translational medicine, 5 (2013) RA108.
[183] L. Frydman, A. Lupulescu, T. Scherf, Principles and features of single-scan two-dimensional NMR spectroscopy, J Am Chem Soc, 125 (2003) 9204-9217.
[184] S. Yang, J. Lee, E. Joe, H. Lee, Y.-S. Choi, J.M. Park, D. Spielman, H.-T. Song, D.-H. Kim, Metabolite-selective hyperpolarized 13C imaging using extended chemical shift displacement at 9.4 T, Magnetic resonance imaging, 34 (2016) 535-540.
[185] P.K. Senanayake, N.J. Rogers, K.L.N. Finney, P. Harvey, A.M. Funk, J.I. Wilson, D. O'Hogain, R. Maxwell, D. Parker, A.M. Blamire, A new paramagnetically shifted imaging probe for MRI, Magnetic resonance in medicine, (2016).
[186] D.V. Nguyen, J.-R. Li, D. Grebenkov, D. Le Bihan, A finite elements method to solve the Bloch–Torrey equation applied to diffusion magnetic resonance imaging, Journal of Computational Physics, 263 (2014) 283-302.
[187] H. Benoit-Cattin, G. Collewet, B. Belaroussi, H. Saint-Jalmes, C. Odet, The SIMRI project: a versatile and interactive MRI simulator, Journal of Magnetic Resonance, 173 (2005) 97-115.
[188] D.A. Yoder, Y. Zhao, C.B. Paschal, J.M. Fitzpatrick, MRI simulator with object-specific field map calculations1, Magnetic Resonance Imaging, 22 (2004) 315-328.
[189] J.S. Petersson, J.O. Christoffersson, K. Golman, MRI simulation using the k-space formalism, Magnetic Resonance Imaging, 11 (1993) 557-568.
[190] T. Stöcker, K. Vahedipour, D. Pflugfelder, N.J. Shah, High-performance computing MRI simulations, Magnetic Resonance in Medicine, 64 (2010) 186-193.
[191] P.T. Callaghan, A Simple Matrix Formalism for Spin Echo Analysis of Restricted Diffusion under Generalized Gradient Waveforms, Journal of Magnetic Resonance, 129 (1997) 74-84.
[192] C. Cai, M. Lin, Z. Chen, X. Chen, S. Cai, J. Zhong, SPROM – an efficient program for NMR/MRI simulations of inter- and intra-molecular multiple quantum coherences, Comptes Rendus Physique, 9 (2008) 119-126.
[193] A.D. Fokker, Die mittlere Energie rotierender elektrischer Dipole im Strahlungsfeld, Annalen der Physik, 348 (1914) 810-820.
[194] M. Planck, Über einen Satz der statistischen Dynamik und seine Erweiterung in der Quantentheorie, Sitzungsber. Kön. Preuss. Akad. Wiss., (1917) 324-341.

[195] H. Risken, The Fokker-Planck Equation, Springer, 1984.
[196] L.J. Edwards, D.V. Savostyanov, A.A. Nevzorov, M. Concistrè, G. Pileio, I. Kuprov, Grid-free powder averages: On the applications of the Fokker–Planck equation to solid state NMR, Journal of Magnetic Resonance, 235 (2013) 121-129.
[197] H. Hogben, M. Krzystyniak, G. Charnock, P. Hore, I. Kuprov, Spinach – a software library for simulation of spin dynamics in large spin systems, Journal of Magnetic Resonance, 208 (2011) 179-194.
[198] G. Moro, J.H. Freed, Efficient computation of magnetic resonance spectra and related correlation functions from stochastic Liouville equations, Journal of Physical Chemistry, 84 (1980) 2837-2840.
[199] A. Polimeno, J.H. Freed, A many-body stochastic approach to rotational motions in liquids, Advances in chemical physics, 83 (1993) 89-89.
[200] R.K. Wangsness, F. Bloch, The dynamical theory of nuclear induction, Physical Review, 89 (1953) 728.
[201] J.H. Freed, G.V. Bruno, C.F. Polnaszek, Electron spin resonance line shapes and saturation in the slow motional region, Journal of Physical Chemistry, 75 (1971) 3385-3399.
[202] D. Goodwin, I. Kuprov, Auxiliary matrix formalism for interaction representation transformations, optimal control, and spin relaxation theories, The Journal of chemical physics, 143 (2015) 084113.
[203] D. Tannor, Introduction to Quantum Dynamics: A Time-Dependent Perspective, in, University Science Books, Sausalito, CA, 2007.
[204] R.B. Sidje, Expokit: a software package for computing matrix exponentials, ACM Transactions on Mathematical Software (TOMS), 24 (1998) 130-156.
[205] L.J. Edwards, D.V. Savostyanov, Z.T. Welderufael, D. Lee, I. Kuprov, Quantum mechanical NMR simulation algorithm for protein-size spin systems, Journal of Magnetic Resonance, 243 (2014) 107-113.
[206] D.M. Brink, G.R. Satchler, Angular momentum, Oxford University Press on Demand, 1993.
[207] T. Liszka, J. Orkisz, The finite difference method at arbitrary irregular grids and its application in applied mechanics, Computers & Structures, 11 (1980) 83-95.
[208] R. Eymard, T. Gallouët, R. Herbin, Finite volume methods, in: Handbook of Numerical Analysis, Elsevier, 2000, pp. 713-1018.
[209] L. Guduff, I. Kuprov, C. van Heijenoort, J. Dumez, Spatially encoded 2D and 3D diffusion-ordered NMR spectroscopy, Chemical Communications, 53 (2016) 701-704.
[210] C. Laustsen, G. Pileio, M.C.D. Tayler, L.J. Brown, R.C.D. Brown, M.H. Levitt, J.H. Ardenkjaer-Larsen, Hyperpolarized singlet NMR on a small animal imaging system, Magnetic Resonance in Medicine, 68 (2012) 1262-1265.
[211] J.-N. Dumez, J.T. Hill-Cousins, R.C.D. Brown, G. Pileio, Long-lived localization in magnetic resonance imaging, Journal of Magnetic Resonance, 246 (2014) 27-30.
[212] A.D. Bain, J.S. Martin, FT NMR of nonequilibrium states of complex spin systems. I. A Liouville space description, Journal of Magnetic Resonance (1969), 29 (1978) 125-135.
[213] G. Pileio, S. Bowen, C. Laustsen, M.C.D. Tayler, J.T. Hill-Cousins, L.J. Brown, R.C.D. Brown, J.H. Ardenkjaer-Larsen, M.H. Levitt, Recycling and Imaging of Nuclear Singlet Hyperpolarization, Journal of the American Chemical Society, 135 (2013) 5084-5088.
[214] J.R. Bunch, D.J. Rose, Sparse matrix computations, Academic Press, 2014.
[215] P. De Fouquieres, S. Schirmer, S. Glaser, I. Kuprov, Second order gradient ascent pulse engineering, Journal of Magnetic Resonance, 212 (2011) 412-417.
[216] I. Kuprov, Spin system trajectory analysis under optimal control pulses, Journal of Magnetic Resonance, 233 (2013) 107-112.
[217] L. Guduff, I. Kuprov, C. van Heijenoort, J.-N. Dumez, Spatially encoded 2D and 3D diffusion-ordered NMR spectroscopy, Chemical Communications, 53 (2017) 701-704.

[218] K. Golman, M. Lerche, R. Pehrson, J.H. Ardenkjaer-Larsen, Metabolic imaging by hyperpolarized 13C magnetic resonance imaging for in vivo tumor diagnosis, Cancer research, 66 (2006) 10855-10860.
[219] H. Gutte, A.E. Hansen, H.H. Johannesen, A.E. Clemmensen, J.H. Ardenkjær-Larsen, C.H. Nielsen, A. Kjær, The use of dynamic nuclear polarization 13C-pyruvate MRS in cancer, American journal of nuclear medicine and molecular imaging, 5 (2015) 548.
[220] J. Kurhanewicz, D.B. Vigneron, J.H. Ardenkjaer-Larsen, J.A. Bankson, K. Brindle, C.H. Cunningham, F.A. Gallagher, K.R. Keshari, A. Kjaer, C. Laustsen, Hyperpolarized 13C MRI: Path to Clinical Translation in Oncology, Neoplasia, 21 (2019) 1-16.
[221] R.J. Ordidge, A. Connelly, J.A. Lohman, Image-selected in vivo spectroscopy (ISIS). A new technique for spatially selective NMR spectroscopy, Journal of Magnetic Resonance (1969), 66 (1986) 283-294.
[222] M. Foroozandeh, G. Morris, M. Nilsson, PSYCHE Pure Shift NMR Spectroscopy, Chemistry–A European Journal, (2018).
[223] C. Juhász, D.C. Chugani, O. Muzik, D. Wu, A.E. Sloan, G. Barger, C. Watson, A.K. Shah, S. Sood, E.L. Ergun, In vivo uptake and metabolism of α-[11C] methyl-L-tryptophan in human brain tumors, Journal of Cerebral Blood Flow & Metabolism, 26 (2006) 345-357.