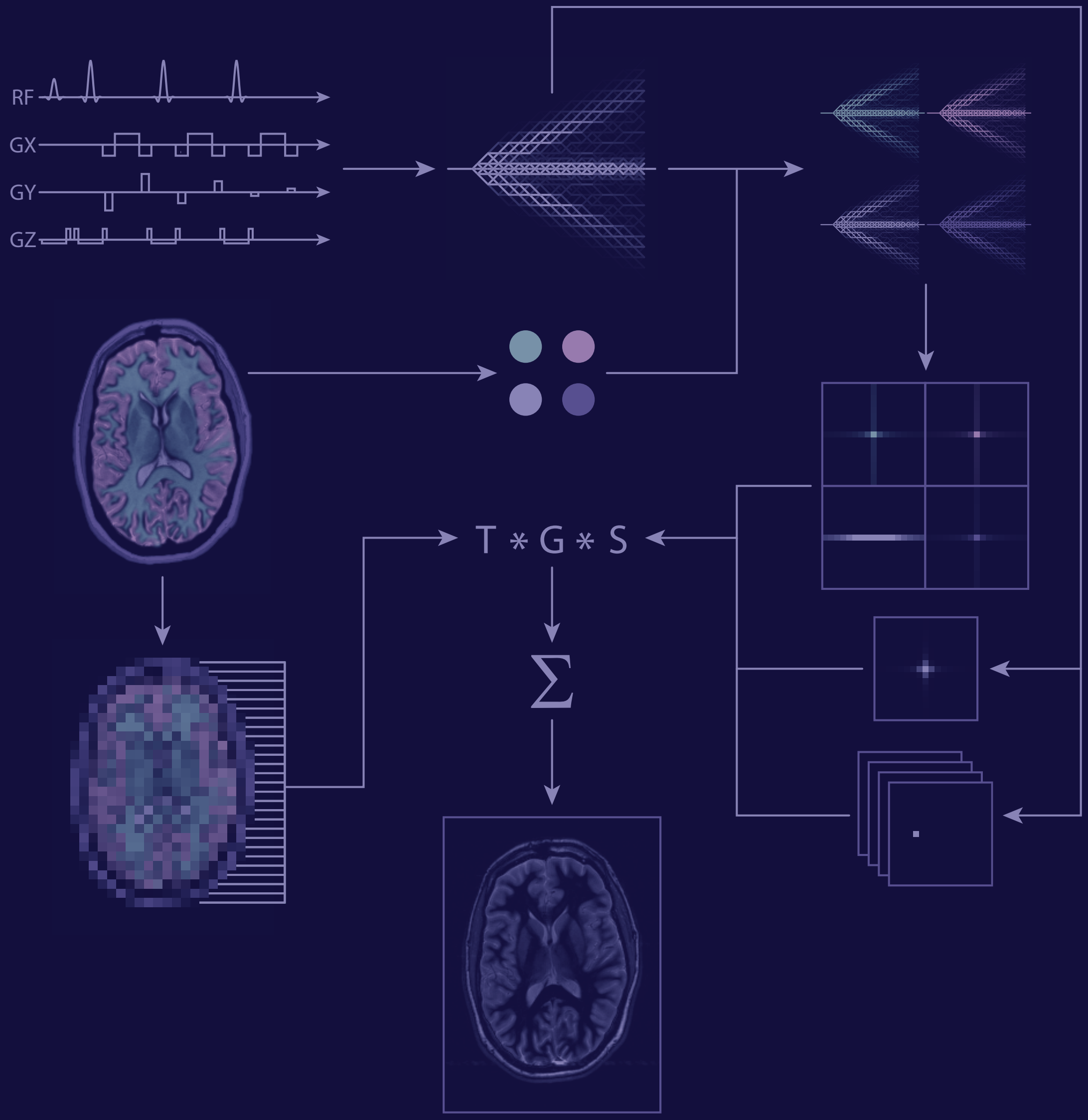
Efficient Simulation of Magnetic Resonance Imaging

Cristoffer Cordes



OPTIMUS

Universität Bremen

Fraunhofer MEVIS

Efficient Simulation of Magnetic Resonance Imaging

Dissertation
submitted to the Faculty of Physics
University of Bremen, Germany
for the degree of
Doctor of Natural Sciences (Dr. rer. nat.)

Cristoffer Cordes

Referees:
Prof. Matthias Günther
Prof. Tobias Preußer

Bremen 2015

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

Cordes, Cristoffer:
Efficient Simulation of Magnetic Resonance Imaging
ISBN 978-3-86376-052-6

Dissertation
zur Erlangung des Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.).
Als Dissertation an der Universität Bremen angenommen.

1. Gutachter: Prof. Dr. Matthias Günther, Universität Bremen
2. Gutachter: Prof. Dr. Tobias Preusser, Jacobs University
Datum des Promotionskolloquiums: 18. November 2015

Diese Dissertation ist elektronisch publiziert:
http://nbn-resolving.de/urn:nbn:de:gbv:46-00104889-13

Alle Rechte vorbehalten
1. Auflage 2016
© Optimus Verlag, Göttingen
© Coverbild: Cristoffer Cordes
URL: www.optimus-verlag.de
Printed in Germany
Papier ist FSC zertifiziert (holzfrei, chlorfrei und säurefrei, sowie alterungsbeständig nach ANSI 3948 und ISO 9706)

Abstract

Efficient Simulation of Magnetic Resonance Imaging

Simulation of Magnetic Resonance Imaging (MRI) is based on the Bloch equation. Solving the Bloch equation numerically is not difficult, but realistic imaging experiments bear a high computational burden. The prevalent approach requires highly resolved sampling of the object and an individual calculation of each sample's behavior for the duration of the imaging sequence. The intrinsic complexity of this approach can be remedied through more computational resources or by application-specific approximations, which both restrict the usefulness and versatility of MRI simulation.

This work presents methods that simplify the problem by exploiting hardware restrictions and the structure of common MRI sequences while not enforcing any approximations. The presented simulation strategies use the reoccurrence of radiofrequency pulses, partial availability of analytical solutions, a reformulation of the problem in Fourier space and finally an inclusion of the reconstruction process to perform MRI simulation in image space, titled Sequence Response Kernel approach.

The algorithmic efficiencies of the methods are investigated and applied to realistic imaging experiments. The particular properties and potential of the algorithms are exemplified, with an emphasis on the Sequence Response Kernel approach and its applications. Depending on the simulation problem, choosing the optimal strategy can greatly decrease computational effort. The Sequence Response Kernel approach opens a new perspective of MRI simulation that can be understood intuitively, even without knowledge of MR physics, and augment image processing techniques.

Table of Contents

CHAPTER 1

Introduction

1.0 Introduction

Since its beginning in the 1970s, Magnetic Resonance Imaging (MRI) has been a very active research topic and is now a decisive part of clinical routine. It is a medical imaging technique which is based on manipulating strong magnetic fields and radio frequency fields to influence the collective magnetization of hydrogen nuclei.

MRI is arguably the best tool for clinical tomography. It methodologically bears no risks for the patient; most notably it does not require ionizing radiation as X-ray based imaging methods do. It can be used to reliably acquire anatomical images of high quality, and even properties of the object that are beyond anatomy. Many techniques can be developed and tested using the same hardware and without harming the imaged person.

MRI simulation is a helpful tool for the understanding and the development of MRI-related methods. It can be used to test hypotheses before initiating lengthy experiments, to optimize certain method parameters or to post-process acquired image data without the imperfections of a physical experiment.

The general theory of the relevant MRI effects can be grasped quickly and the underlying mathematical model, the Bloch Equation, formulates well-posed problems, can be solved accurately, and be implemented easily. The problem is however, that the simulation needs to be performed at a high resolution, employing elaborate numerical integrations that in general cannot be accelerated.

This complication is usually approached in one of two ways—either with computational power or with model simplifications. The computational power approach preserves the general applicability of the method but is infeasible for realistically scaled experiments with contemporary hardware. Simplifications are often valid and legitimate, and sometimes even enable analytical investigation—but they are usually highly specialized, hard to extend and may ignore even basic effects.

This work aspires to find a good compromise, and to build a bridge between the general but computationally exhaustive methods and drastic simplifications. It uses simplifications that are valid for almost all clinically relevant MR sequences, carefully investigates the requirements, consequences, and limits of the used simplifications and thereby provides algorithms for optimally efficient simulation.

The introduction chapter (Chapter 1) briefly establishes the Magnetic Resonance effects that are necessary to understand MRI. It then describes MRI itself and MRI pulse sequences that are used in this work. The next section provides the mathematical fundamentals of the MRI simulation methods. The introduction chapter concludes by defining the goals of efficient MRI simulation.

The methods chapter (Chapter 2) elaborates the simulation techniques, which are the focus of this work. More specifically, the techniques are investigated in the context of performing full virtual MR imaging experiments with high accuracy and efficiency. The final technique produces byproducts that are of great value to simulation-related tasks which are defined in the last section of that chapter.

The results chapter (Chapter 3) mostly follows the method chapter's structure and supports the aspects of the individual techniques. It also provides examples and comparisons with physical measurements where applicable. Effort estimations are an additional focus, in which the techniques are considered with respect to realistic settings.

The discussion chapter (Chapter 4) illustrates the technical details of the methods in a more general scope. The chapter then commences by presenting its impact, feasible extension and possible application areas.

1.1 Magnetic Resonance Imaging fundamentals

Even though MRI is directly enabled by quantum mechanical properties, an understanding of quantum mechanics is virtually irrelevant for almost all MRI techniques. The abundance of nuclei in any MRI experiment permits a classical description. A comprehensive explanation of the nuclear magnetic resonance phenomena can be found in [26], and their relations to MRI are emphasized in [41].

Further quantum mechanical interpretations of MRI experiments have led to many myths and misunderstandings [17]. In the context of this work, the Bloch Equations can be considered as axioms, ignoring the quantum mechanical motivation completely.

1.1.1 Bloch equation

The Bloch equation [4] describes the behavior of macroscopic nuclear magnetization $\overrightarrow{M} = (M_x, M_y, M_z)$ of an object:

$$\frac{\partial}{\partial t}\overrightarrow{M} = \gamma\overrightarrow{M} \times \overrightarrow{B}(t) - \frac{M_x\overrightarrow{e_x} + M_y\overrightarrow{e_y}}{\text{T2}} + \frac{(M_0 - M_z)\overrightarrow{e_z}}{\text{T1}} \tag{1.1}$$

The gyromagnetic ratio γ is a physical constant of the nucleus. This work exclusively considers hydrogen nuclei, for which $\gamma \approx 42.58\,(2\pi)\,\mathrm{MHz\,T^{-1}}$.

The z-component of the magnetic field contains a term B_0 which is constant in time and influences the equilibrium magnetization M_0, which is aligned with B_0.

The relaxation times T1 and T2 depend on the chemical properties of the object and will be explained in the following paragraphs.

Objects are assumed to not interact directly. An object that has inhomogeneous relaxation times or experiences a different magnetic field can be split into sub-objects that are treated by the Bloch equation independently. A very small object, such as a small volume element that can be considered homogeneous, is called a spin packet.

1.1.1.1 Larmor equation

For T1, T2 $\rightarrow \infty$ and $B = (0, 0, B_0)$, the Bloch equation produces the Larmor equation:

$$\omega_0 = \gamma B_0. \tag{1.2}$$

The Larmor equation defines a frequency ω_0, the Larmor frequency, at which the magnetization vector rotates around the z-axis. This process is called precession. The x and y components of the magnetization are called the transverse components, and often times more conveniently expressed using one complex value $M_\perp = M_x + iM_y$. The z-component is called the longitudinal magnetization and is unaffected by the Larmor equation.

1.1.1.2 Relaxation

If a spin system is perturbed from its thermal equilibrium, then it returns to this state over time. This process can be separated into two independent sub-processes: The recovery of longitudinal magnetization—T1 relaxation, and the decay of transverse magnetization—T2 relaxation.

T1 Relaxation

The longitudinal relaxation is dominated by the deposition of energy into the environment, and it is thus also called spin-lattice-relaxation. It mainly depends on the external magnetic field and the inner motion of the molecules. It can be phenomenologically modeled as an exponential decay towards the equilibrium with decay rate $\mathrm{T1}^{-1}$.

T2 Relaxation

The transverse magnetization decays due to interactions of the individual spins. This process does not transfer energy to the environment, and is thus called spin-spin-relaxation. It is largely independent of the external magnetic field and can be phenomenologically modeled as an exponential decay with decay rate $\mathrm{T2}^{-1}$.

1.1.1.3 T2′ and T2*

Microscopic magnetic field inhomogeneities, denoted ΔB_0, are usually assumed to be Cauchy-Lorentz distributed. This directly translates to a Cauchy-Lorentz distribution of the Larmor frequencies, which causes a dephasing of the individual spin packets that is observable as a further exponential decay of the net signal. The total decay rate of the net transverse signal is called T2*, which can be separated into the contribution of temporally static magnetic field inhomogeneities and the other effects by using the following formula:

$$\frac{1}{\mathrm{T2*}} = \frac{1}{\mathrm{T2}} + \frac{1}{\mathrm{T2'}} = \frac{1}{\mathrm{T2}} + \gamma \Delta B_0 \tag{1.3}$$

The main reason why this distinction is relevant, is that the decay caused by T2′ can be reversed. If the magnetization of all spin packets is rotated around a transverse axis by 180°, then the phase of each spin packet is inverted. The phase accumulation that happened before the rotation reverses, until the spin packets are refocused when the dephasing and rephasing times are equal, at which time they create a so called spin echo [15]. The experiment will be outlined later (see Section 1.2.2.1).

1.1.1.4 Biological tissues

Biological tissues have distinct T1 and T2 constants. In theory, $\mathrm{T2} \leq 2\,\mathrm{T1}$, but $\mathrm{T2} \leq \mathrm{T1}$ is never violated under clinical conditions [37]. Within this work however, this relation is sometimes ignored to emphasize the mathematical properties of the underlying methods. Some of the most important MRI parameters for some clinically relevant tissues are illustrated in Table 1.1.

1.1.2 MRI

A Magnetic Resonance Imaging setup can manipulate the magnetic field in specific ways, thereby specializing the Bloch equation.

Tissue type	proton density	T1	T2	T2*
CSF	1	2569	329	58
Grey matter	0.86	833	83	69
White matter	0.77	500	70	61
Fat	1	350	70	58
Muscle / Skin	1	900	47	30
MS Lesion	0.76	752	237	204

Table 1.1: Relaxation times of some brain tissues at 1.5 T [6]. Further information about typical relaxation times can be found in [38] and [12].

Larmor precession of spin packets can be measured with a suitable coil. To enable Larmor precession, spin packets need to be driven out of their equilibrium state. This can be achieved by applying a magnetic field with transverse components. If this magnetic field oscillates with Larmor frequency, then this perturbation can manipulate the magnetization effectively due to resonance. Such electromagnetic pulses are called radio frequency (RF) pulses.

Spin packets need to be encoded spatially to yield information that can be used to produce images with spatial correspondence. A spatially varying longitudinal component of the magnetic field equally varies the Larmor frequency. By applying linear slopes of longitudinal magnetic field in different directions, a spin packet obtains phase and frequency information that depends on its location. Such electromagnetic pulses are called gradient pulses.

1.1.2.1 MRI Bloch equation

The MRI setup specializes the Bloch equation:

$$\begin{aligned}
\frac{\partial}{\partial t}\overrightarrow{M}(\overrightarrow{r},t) =& \gamma\overrightarrow{M}(\overrightarrow{r},t) \times \overrightarrow{B}(\overrightarrow{r},t) \qquad (1.4)\\
&- \frac{M_x(\overrightarrow{r},t)\overrightarrow{e_x} + M_y(\overrightarrow{r},t)\overrightarrow{e_y}}{\mathrm{T2}}\\
&+ \frac{(M_0(\overrightarrow{r},t) - M_z(\overrightarrow{r},t))\overrightarrow{e_z}}{\mathrm{T1}}\\
\overrightarrow{B}(\overrightarrow{r},t) :=& (B_0 + B_{\mathrm{inhom}}(\overrightarrow{r}) + B_G(\overrightarrow{r},t))\overrightarrow{e_z} \qquad (1.5)\\
&+ B_{1,x}(\overrightarrow{r},t)\overrightarrow{e_x} + B_{1,y}(\overrightarrow{r},t)\overrightarrow{e_y}\\
B_G(\overrightarrow{r},t) :=& \langle \overrightarrow{G}(t), \overrightarrow{r} \rangle \qquad (1.6)\\
B_{1,x}(\overrightarrow{r},t) :=& T_{\mathrm{RF}}(\overrightarrow{r})B_{\mathrm{RF},x}(t) \qquad (1.7)\\
B_{1,y}(\overrightarrow{r},t) :=& T_{\mathrm{RF}}(\overrightarrow{r})B_{\mathrm{RF},y}(t) \qquad (1.8)
\end{aligned}$$

where:

B_0	=	Main magnetic field
$\overrightarrow{G}(t) = (G_x(t), G_y(t), G_z(t))$	=	Gradient pulses
$B_{\mathrm{RF},x}(\overrightarrow{r},t), B_{\mathrm{RF},y}(\overrightarrow{r},t)$	=	RF pulses
$B_{\mathrm{inhom}}(\overrightarrow{r})$	=	B_z inhomogeneities
$T_{\mathrm{RF}}(\overrightarrow{r})$	=	RF send coil transmissivity

Within the hardware capabilities, the time-dependent real-valued gradient and RF pulse functions can be arbitrarily controlled. The main magnetic field, B_0 inhomogeneities and coil transmissivity are static during the measurement.

The signal acquisition process can be formulated as follows:

$$S(t) = \int_{\Omega} S_{\mathrm{RF}}(\overrightarrow{r})(M_x(\overrightarrow{r}, t) + \mathrm{i}M_y(\overrightarrow{r}, t))\, \mathrm{d}\overrightarrow{r} \tag{1.9}$$

where:

$S(t)$	=	Signal that is measured by one coil
Ω	=	Domain of the experiment
$S_{\mathrm{RF}}(\overrightarrow{r})$	=	RF receive coil sensitivity

This formulation is simplified. Firstly, it does not consider the sampling that is needed to capture the signal. In reality, the signal is sampled—one time point actually corresponds to a time span during which the signal accumulates. However the sample distances are usually small enough to justify this simplification and treat the sample length separately. Choosing a higher sample length influences the measured noise, and proportionally scales the measured signal intensity. Both effects can be applied after calculating the perfect signal. Many further phenomena that are not of main interest for MRI are also ignored, such as concomitant gradient terms and spectroscopic effects [41] [26].

MRI sequence definition
An MRI sequence is the combination of gradient pulse descriptions, RF pulse descriptions and sampling settings—essentially every aspect of the MRI system that can be controlled. An MR sequence usually depends on a collection of parameters, e.g. spatial resolution, slice position, echo time, repetition time. The term protocol is used to describe the combination of an MR sequence blueprint and associated sets of defining parameters.

There is no consistent definition of the term MRI sequence. Within this work, an MRI sequence is defined as an actual description of the waveforms and timing. In contrast to other interpretations of this term, a sequence is different from a sequence type. A sequence type is a more abstract description of the sequence which requires further parameters to be completely defined, such as timing or resolution.

1.1.2.2 MRI setup

The most important hardware components of the MRI setup will be described in the following paragraphs. Components that are not mentioned here are usually not of interest for MR simulation experiments and are thus ignored in this work. There are many other parts essential to a MR scanner, but most of them assume a supporting role that corrects the imperfections of actual physical experiments, such as shim coils. More details about MRI system hardware can be found in [3].

Main coil

The B_0 field strength is proportional to the equilibrium magnetization, which is in turn proportional to the maximum achievable signal. This magnetic field is usually created using a superconducting electromagnet. It needs to be large enough to create a sufficiently homogeneous field within sections of a human body—which is the reason why most systems are designed to fit a patient inside the bore of a cylindrical coil.

Contemporarily used field strengths are $1.5\,\mathrm{T}$, $3\,\mathrm{T}$ and $7\,\mathrm{T}$. The field created by the main coil is always active, always constant and regularly serviced to stay homogeneous.

Gradients

An additional set of coils is used to manipulate the magnetic field spatially. They are supposed to exclusively create a linear slope of B_z in any chosen spatial direction, which in turn results in a slope in Larmor frequency in that direction. It is essential for all MR imaging techniques that these coils have a strong maximum amplitude and can be controlled very fast, because spatial encoding of the signal is governed by the gradient field as will be explained in Section 1.2.1. The point, at which the influence of the gradient field is always zero, is called the isocenter and is usually located near the center of the bore. Within this work, gradient fields are assumed to be perfectly linear. The impact of gradient imperfections is discussed in Section 4.1.1.3.

A contemporary maximum amplitude is $0.043\,\mathrm{T\,m^{-1}}$ with a slew rate (maximum possible change) of $180\,\mathrm{T\,m^{-1}\,s^{-1}}$ [32].

RF send coil

Electromagnetic pulses with a frequency, that is close to the Larmor frequency of spin packets of interest, are used to perturb the spin packets out of their equilibrium state. Unlike former two components, the magnetic field of the RF coils is not supposed to be aligned with the z-axis and is called the B1 field. Such a perturbation is usually on the order of $5\,\mathrm{ms}$.

Most applications rely on a homogeneous B1 field. The spatial transmissivity distribution and the resulting local B1 field inhomogeneity can be an error source and image artifact in some applications.

Throughout this work, pulses that are sent via the RF coils are either called RF pulses or pulses for brevity. If the gradient coils are used to apply gradient pulses, then this will be stated explicitly to avoid confusion.

RF receive coil

Precessing spin packets can induce a current which is picked up by the receive coil and then sampled to yield the data of the measurement.

The current that is induced in a coil by a spin packet is weighted by the spatial coil sensitivity profile. Coil arrays are often used to achieve a more homogeneous collective coil sensitivity profile and reduce noise or to use the individual coil signals for more intricate reconstruction purposes.

1.2 MRI pulse sequences

Based on the physical details of MRI sequences that were established in the previous section, actual pulse sequence schemes can be introduced. The k-space concept can help understand sequences and is introduced first. Almost all sequences are created by combining the same basic components, which are described in the second part of this section. Finally, the four common sequence types that are used throughout this work will be defined and explained.

1.2.1 k-space

Many effects in MRI can be explained more easily with the help of k-space, which is introduced in the following paragraphs. Further information can be found in [3].

Let $M_\perp$ be a relaxation-free distribution of transverse magnetization. When a gradient pulse is applied, the slope in magnetic field induces a slope in precession frequency in the same direction, thus $M_\perp$ accumulates a corresponding phase over time:

$$M_\perp(\overrightarrow{r},t) = M_\perp(\overrightarrow{r},0)\mathrm{e}^{\mathrm{i}\varphi(\overrightarrow{r},t)} \tag{1.10}$$

$$\varphi(\overrightarrow{r},t) := \gamma \int_0^t \overrightarrow{r} \cdot \overrightarrow{G}(t')\,\mathrm{d}t'. \tag{1.11}$$

The signal $S(t)$ that can be read by an idealized readout coil is proportional to the accumulated transverse magnetization within its volume of sensitivity Ω:

$$S(t) = \int_\Omega M_\perp(\overrightarrow{r},t)\,\mathrm{d}\overrightarrow{r}. \tag{1.12}$$

This can be expressed more clearly by introducing $\overrightarrow{k}(t)$:

$$S(t) = \int_\Omega M_\perp(\overrightarrow{r},0)\mathrm{e}^{\mathrm{i}\overrightarrow{k}(t)\cdot\overrightarrow{r}}d\overrightarrow{r} \tag{1.13}$$

$$\overrightarrow{k}(t) := \gamma \int_0^t \overrightarrow{G}(t')\,\mathrm{d}t'. \tag{1.14}$$

Alternatively, this can be expressed by utilizing the Fourier transform (Equation 1.20):

$$S(t) = \mathcal{F}(M_\perp)(\overrightarrow{k}(t),0). \tag{1.15}$$

In conclusion: The Fourier transformed transverse magnetization can be read directly by means of the RF receive coil. The integral of the gradient pulse, also referred to as the 0^{th} moment, determines the position at which it is evaluated.

When the gradient amplitude is kept constant over a certain time span at which signal is acquired, then $\overrightarrow{k}(t)$ sweeps over a line segment. This process is referred to as acquiring a line in k-space.

This basic idea is augmented to yield the k-space based simulation method in the next chapter (see Section 2.3).

1.2.2 RF pulses

RF pulses generate a magnetic field that is perpendicular to the main magnetic field. As such, it causes a transformation of a spin packet's transverse and longitudinal components.

Details about RF pulse design can be found in [3]. For the scope of this work, it suffices to examine the sinc pulse (Figure 1.2), which is the classic and most common pulse shape.

RF pulses are often interpreted as rotations around a transverse axis, determined by the so called phase of the RF pulse. The angle of rotation is commonly referred to as the flip angle of the RF pulse. If this effect is supposed to occur only within a certain Larmor frequency band, then the sinc pulse (Figure 1.2) is the classical choice, motivated by its Fourier transform.

Sinc pulse

When the overall effect of an RF pulse on a magnetization vector results in a small flip angle, then the Fourier transform of the pulse shape approximates the resonance offset frequency dependency of the pulse effect, also called pulse profile [3].

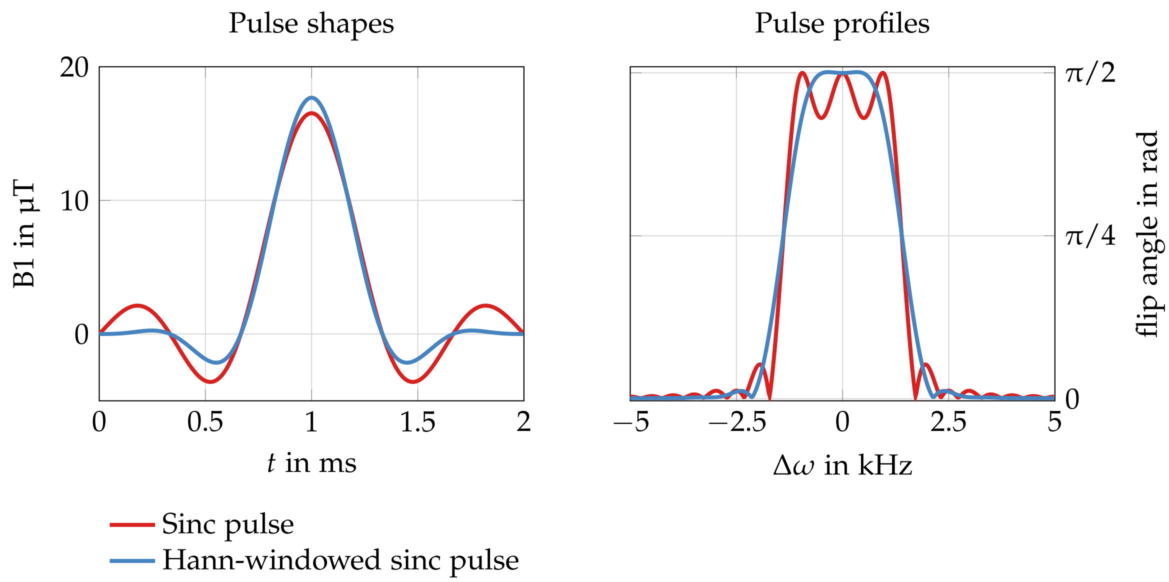


Figure 1.2: Pulse shapes and pulse profiles of a sinc pulse and a Hann-windowed [18] sinc pulse. Both pulses perform well at low-pass excitation. The Hann-windowed pulse produces a smoother profile, but has a wider transition band.

The pulse can be fully characterized by its maximum amplitude, duration, number of side lobes and phase, even though it is often more convenient to define it through flip angle, bandwidth and bandwidth-time-product.

RF pulses can be modulated with a certain frequency, which shifts the pass-band of the pulse from zero to e.g. the Larmor frequency in this case.

Windowed sinc pulse

An RF pulse cannot have infinite duration. Direct truncation of the pulse results in ripples of passband and stopband. One way to improve this profile is to apply a window function to the pulse shape, for example the Hann window [18]. This results in a smoother pulse profile at the cost of a wider transition band.

1.2.2.1 RF pulse usage

RF pulses are often applied with a single specific intention, most commonly excitation, refocusing or preparation. However, in reality, these goals cannot be achieved isolatedly. Therefore, these three terms do not have a clear definition, but ease the comprehension of a sequence.

Excitation

One common intention is to convert longitudinal magnetization into transverse magnetization, so it can be sensed by the acquisition coil. Pulses of this type often involve a very small flip angle, but pulses of up to 90° are also common for certain applications.

Refocusing

A refocusing pulse is employed to invert phases. This can be useful for reversing the T2′ decay, thereby creating a spin-echo, as motivated in Section 1.1.1.3, or to traverse quickly in k-space, which will be explained in the next chapter (see Section 2.3). An ideal refocusing pulse has a 180° flip angle.

Preparation

A preparation pulse is used to modify the magnetization prior to an excitation pulse to emphasize certain properties of the spin packets. One common preparation is inversion recovery—a 180° pulse that inverts longitudinal magnetization. The magnetization then partially decays according to T1, and the magnetization that is then excited carries a T1 weighting. A preparation can also be used to separate or suppress parts of the magnetization. Common applications of this kind of preparation are background suppression and fat saturation [3].

Spin echo

The spin echo pulse configuration (Figure 1.3) is a technique that is used to recover the reversible part of the T2* decay (see Section 1.1.1.3).

After excitation pulse (90°), the spin packets decay according to T2*. After a chosen time TE/2, a second pulse is applied (180°) which reverses the phases of all spin packets, such that they reach coherence again at the echo time TE. Using former terminology, the first pulse is used for excitation, and the second pulse is used for refocusing.

1.2.3 Gradient pulses

The gradient coils can be controlled to produce arbitrary pulse shapes within the hardware limitations, but since a complicated gradient waveform yields a complicated behavior of the individual spin packets, they are mostly applied as trapezoidal pulses. They are thus characterized through amplitude, ramp up time, flat top time, ramp down time.

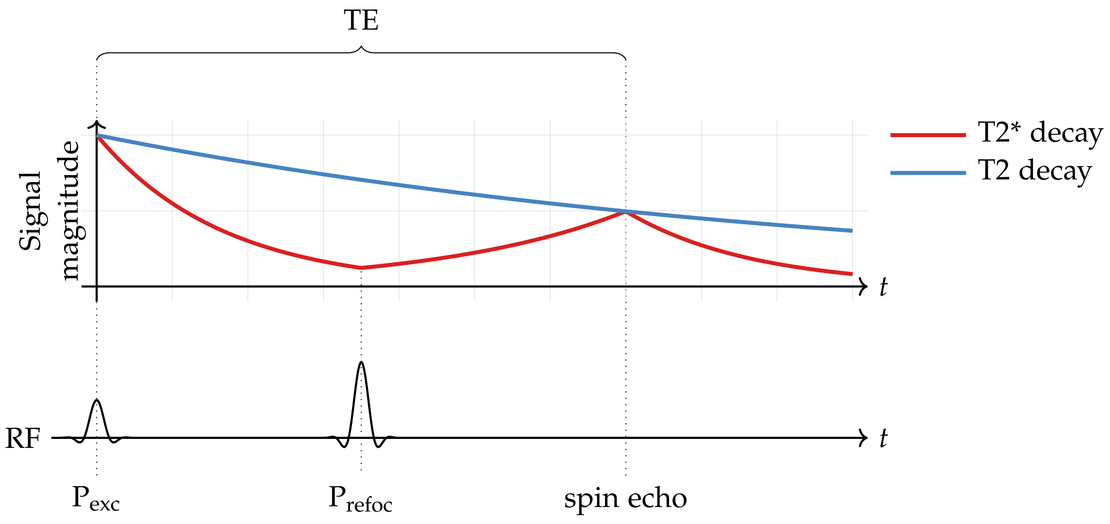


Figure 1.3: The spin echo scheme. The T2′ contribution of the T2* decay can be recovered by employing a refocusing pulse. The T2 contribution of the T2* decay is irreversible.

1.2.3.1 Gradient pulse usage

Gradient pulses are applied with specific intentions. Intentions that are common or particularly relevant for this work are presented in the next paragraphs, followed by a diagram that showcases the most important gradient pulse types in a pulse sequence diagram (Figure 1.4).

Slice selection

Excitation RF pulses are usually designed to excite a specific frequency band. When a gradient pulse is switched on while an RF pulse is applied, the frequency selectivity of the pulse causes only those spin packets to be manipulated that have a Larmor frequency, for which the pulse is selective. Since Larmor frequencies are equal along a plane perpendicular to the gradient pulse's direction, a slice is selected, giving those gradient pulse their functional name—slice selective pulses.

Rewinder

The transverse magnetization after a slice-selective pulse does not possess homogeneous phase. The linear part of this phase inhomogeneity can be corrected for by applying a gradient in the direction opposite to the slice encoding gradient direction. That gradient pulse is called rewinder.

Frequency encoding

As explained in Section 1.2.1, a k-space line is acquired when signal is read while a gradient pulse is active. This is called frequency encoding because each frequency that is present ideally corresponds to spin packets that are on a plane perpendicular to the gradient direction. If the direction of the frequency encoding gradients is constant throughout the sequence, then this direction is called frequency encoding direction.

Prephaser

A frequency encoding gradient is often supported by a prephaser gradient which is applied beforehand to allow the k-vector to pass through zero in that direction during readout.

Phase encoding

Gradient pulses that are applied perpendicular to the frequency encoding direction and slice direction with the intention of spatial encoding are commonly referred to as phase encoding gradients. They are applied before signal is measured.

As explained in Section 1.2.1, such a gradient pulse induces a phase slope, which shifts the k-space line which is later acquired during frequency encoding. The direction that corresponds to the phase encoding gradient pulse is commonly referred to as phase encoding direction.

However, each gradient pulse causes a linear phase slope, thus the literal term has no distinctive properties and does not possess a mutually accepted definition.

Spoiler

After signal is acquired, a gradient pulse with a high 0^{th} moment can be applied to remove the influence of the residual transverse magnetization. The spoiling gradient creates a phase dispersion of at least 4π per voxel. The dispersed magnetization will then ideally produce no signal in further measurements. When spoiling techniques are used, the RF pulse phase is usually varied at each acquisition to further deteriorate the unwanted signal.

Crusher

If an imperfect refocusing RF pulses are used, newly excited magnetization and refocused magnetization can be separated by introducing crushers. Crusher gradients are a pair of two equal gradients placed directly around the RF pulse. The first gradient induces the same effect as a spoiling gradient but the second gradient compensates this effect for refocused spin packets. This effectively suppresses newly created and non-refocused magnetization for the following readout, but potentially reintroduces them in later readouts. Many advanced crushing schemes can be used to select desirable signal contributions [13].

1.2.3.2 Basic line acquisition

An MRI sequence is often visualized by a pulse sequence diagram (Figure 1.4) to reveal the relationship between the individual pulses.

The five time axes correspond to the RF channel, the three gradient channels and one channel that indicates when the analog-to-digital converter (ADC) is configured to read signal.

1.2.3.3 Signal Acquisition

In most cases, signal is acquired at a fixed sampling rate. The analog-to-digital converter then returns a complex number for each sample. The sampling rate can be controlled through the acquisition bandwidth.

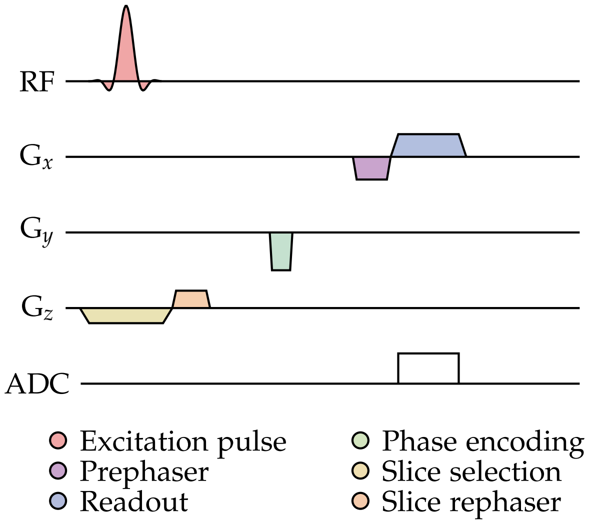


Figure 1.4: Pulse sequence diagram of a simple single-line gradient echo acquisition to showcase basic sequence components.

Basic encoding strategy
The most common image acquisition strategy is to select the slices of interest individually and acquire lines with equal frequency encoding pulse patterns but differing phase encoding gradient pulses. After sufficient k-space lines are acquired, Fourier transforms along phase and frequency direction reconstruct the selected slice.

Reconstruction
The acquired signal needs to be processed to represent information spatially. If the k-space samples are acquired according to a regular grid, then the discrete Fourier transform (Equation 1.20) along the grid dimensions yields data with spatial correspondence, holding intensities proportional to spin packet magnetizations, weighted depending on the sequence type and sequence parameters. Sophisticated reconstructions that are beyond Fourier transform are not immediately relevant for this work and are thus omitted here.

Reordering scheme
In practice, the order in which the lines are acquired often determines image quality. Center lines in k-space determine the contrast of the image while the lines on the boundaries of k-space determine the high-frequency components. It can be expected that the first lines that are acquired, are measured with overall stronger intensity. The fluctuation of line intensities can result in image artifacts. This can be partially avoided by introducing dummy acquisition cycles that are not used for image reconstruction, but this is not always a reasonable strategy.

1.2.4 Common MRI pulse sequences

The sequence building blocks that were introduced in the previous part can be combined in many ways to produce meaningful results. A pulse sequence that acquires k-space lines that are then reconstructed together is called an imaging sequence.

A different imaging sequence that acquires the same k-space lines can produce a substantially different image because timing and used building blocks emphasize particular information about the measured object, such as proton density or relaxation times.

The following examples describe the imaging sequences that are used in this work.

1.2.4.1 Spoiled gradient echo (SPGR)

The spoiled gradient echo sequence acquires one line per excitation pulse. There is only one RF pulse shape used in the sequence. In order to acquire lines fast, the flip angle of this pulse is usually chosen low, because more magnetization is available for the next excitation this way.

At the beginning of each RF pulse, there is no transverse magnetization expected to be present. The RF pulse then excites the magnetization, followed by a line readout centered at TE. After that, the spoiler removes the residual transverse magnetization and the process is repeated for the next line after the repetition time TR has passed. Different image contrasts can be generated using this sequence by controlling the sequence parameters—T1 contrast can be achieved by adjusting TR and flip angle, TE controls the T2* weighting.

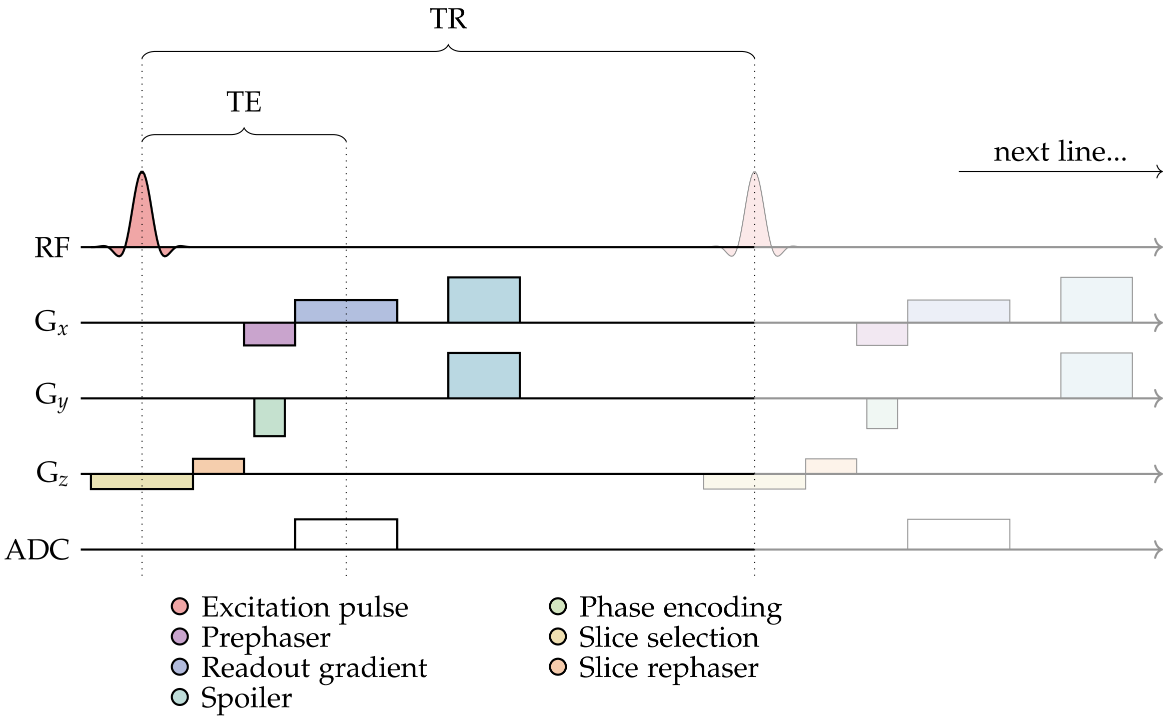


Figure 1.5: Pulse sequence diagram of an SPGR sequence

1.2.4.2 Echo planar imaging (EPI)

The EPI sequence uses only one pulse to acquire the image of a selected slice. It is often used for fast acquisition.

After the excitation pulse, the first line is read as a gradient echo, similar to the SPGR sequence. But instead of spoiling, a short second phase encoding gradient pulse, called blip is applied to read the second line in reverse direction. This scheme continues until a whole image is acquired. In this sequence type, the echo time TE is defined as the temporal distance from excitation to the acquisition of the sample corresponding to $k = 0$.

The echo train length is long in comparison to other sequences. This results in a strong T2* weighting in the image.

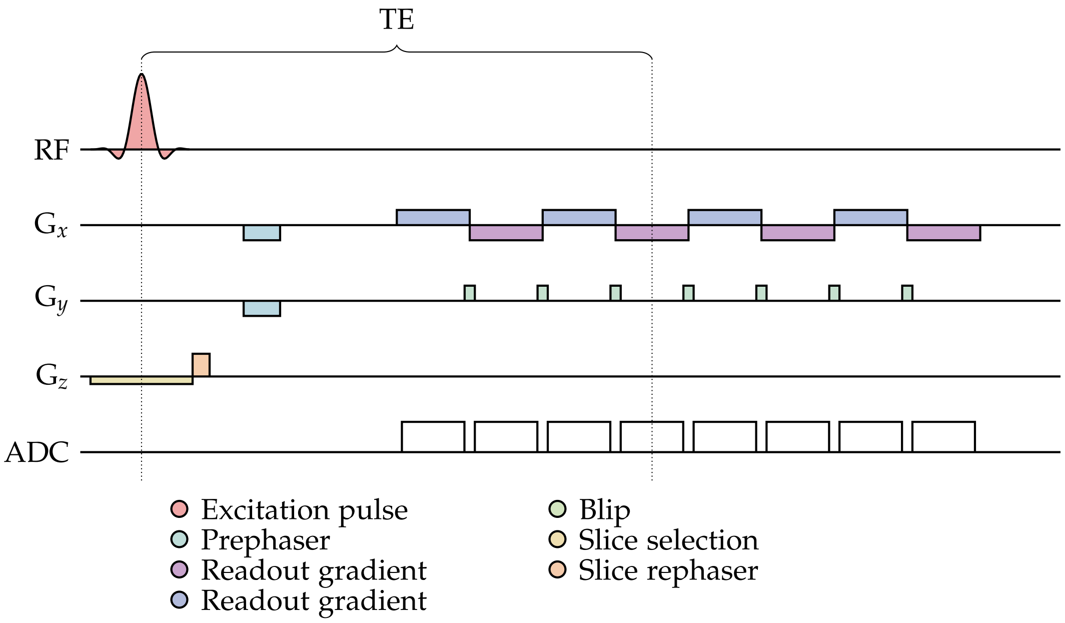


Figure 1.6: Pulse sequence diagram of an EPI sequence

1.2.4.3 Balanced steady state free precession (bSSFP)

The bSSFP sequence acquires one line per excitation pulse, but in between each two pulses, the 0^{th} gradient moment is nulled. The gradients employed for this purpose are called rewinders. This effectively removes all gradient-induced phase accumulation of static tissue, and the residual magnetization is reused in the next acquisition. The readout is placed exactly between two subsequent RF pulses.

Over time, a steady-state will develop, determined by the T1 recovery and the T2 signal loss between the pulses, which in turn determines the contrast behavior. This steady state is determined by the pulse's flip angle and TR.

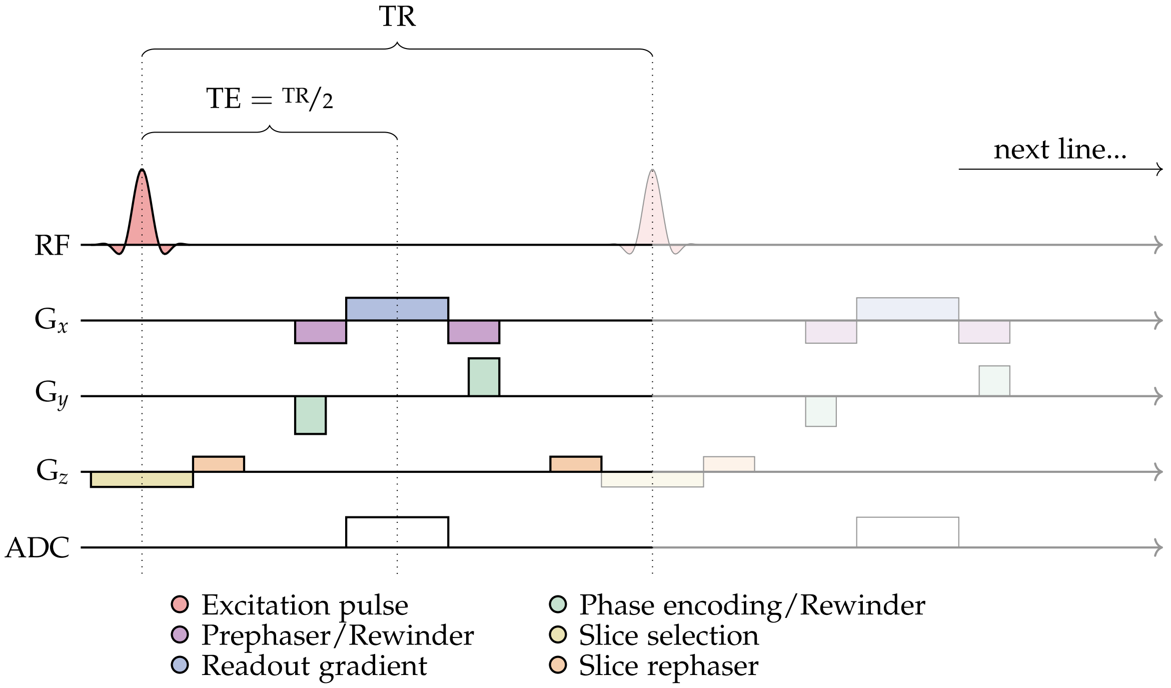


Figure 1.7: Pulse sequence diagram of a bSSFP sequence

1.2.4.4 Turbo spin echo (TSE)

After an excitation pulse, the signal is refocused several times—dictated by the turbo factor—followed by a line readout each time. Crusher gradients around the refocusing pulses are employed to reduce unwanted signal.

TE is often defined in an alternative way. For a fixed reordering scheme, the time between the excitation and the acquisition of the center point in k-space determines the main contrast, and is therefore also called TE. This parameter will be referred to in an unambiguous way in later sections.

In addition to TR, TE and the flip angle of the refocusing pulse, the reordering scheme and turbo factor largely define the contrast behavior, because of the potentially long acquisition trains. If a turbo factor of one is chosen, the sequence is called spin echo sequence.

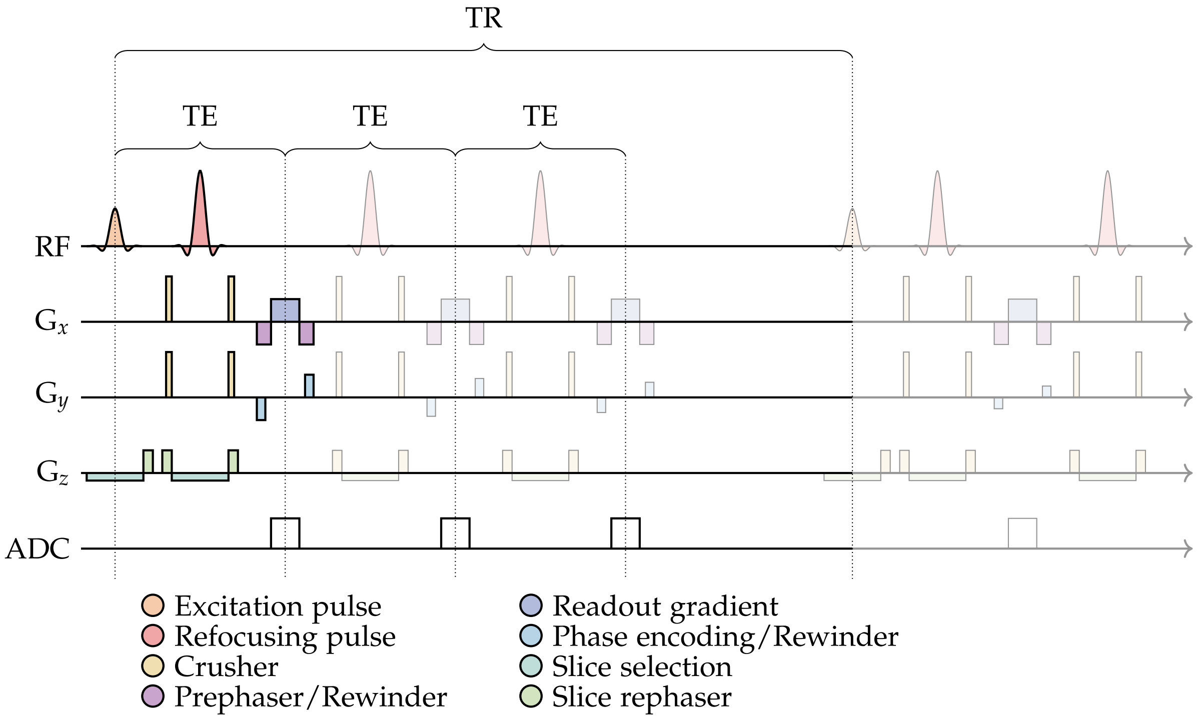


Figure 1.8: Pulse sequence diagram of a TSE sequence

1.2.5 Images

After the signal is acquired and reconstructed, it needs to be displayed to allow for interpretation. To display the reconstructed data, the color of each pixel is determined by the according entry in the matrix through a lookup table (LUT). A good LUT emphasizes the important variations within the matrix and therefore, a certain pixel color has a meaning that depends on the data and the LUT simultaneously. In computed tomography and MRI, the term windowing is commonly employed to express where intensity values are cut off, with a linear interpolation of levels of gray in between.

In MRI, choosing a different set of acquisition parameters also changes the signal intensity of the objects that are measured. The difference of two objects' signal intensity in relation to the image noise is called the contrast to noise ratio (CNR) and describes how well-suited the image is to distinguish the two tissues. Similarly, the signal of a tissue in relation to the noise is called the signal to noise ratio (SNR). Consequently, MRI sequences may produce more signal at the cost of contrast, thereby not providing additional information in comparison to an image that has low signal but a very strong contrast.

As a general rule of this work, images that are displayed in a gray scale correspond to data that can be interpreted as medical images, simulated medical images, or are strongly linked to the LUT of a medical image. The usage of color indicates a more abstract dataset, mostly either by representing a derived parameter, by having axes that do not correspond to spatial dimensions, or by depicting a magnitude that is not easily relatable to pixels of a medical image.

1.3 Mathematical fundamentals

MRI simulation is supported by several mathematical disciplines that are mostly part of the basic university physics knowledge base. This section introduces the most important concepts and notations, supplemented by elaborations of the details that are especially relevant to MRI.

1.3.1 Algorithmic efficiency

Efficiency is a major topic in this work. It will become evident in the later sections, that certain approaches do not scale well or cannot be translated outside of their specialty application onto other relevant scenarios. The basic concepts of algorithmic efficiency and its notation are explained in this section.

The two main resources that are to be considered for a numerical method are the number of basic operations that can be performed in a fixed time span and the amount of data that can be stored and retrieved instantly. But even when the according technological limits of suitable devices are reached, there may be hope for the possible allocation of more resources through parallelization. But a distribution of workload is only possible if the algorithm allows it. An algorithm that requires random access to all previous calculations and needs strictly sequential processing of each data object cannot be distributed.

A slightly more complex input to an algorithm might require an over-proportional increase in computational effort. This is mostly due to the limiting behavior of the algorithm. This means that no matter how the algorithm is implemented, the required computational effort will grow according to a rough estimate that is based purely on the method.

1.3.1.1 Benchmarking

The obvious way to compare two similar, competing algorithms, is to measure the time that each algorithm takes to create the same results under equal circumstances. This direct comparison gives a tangible and validatable result, but it hides the underlying reasons for this time effort. Some special cases or organized input data may singularly be treated better by one version, and the link between the nature of the input data and the required time can only be probed opposed to explained. This method of comparison is also biased by the implementation efficiency of the developer.

Run time comparison can be meaningful when the circumstances, results and input data of algorithms are equal. However, the algorithms in this work utilize further assumptions or simplifications, and thus limit the applicable circumstances or the quality of the results. Furthermore, a different approach may be well-exploitable in some special use cases, but obstructive in others.

1.3.1.2 Time complexity

When a universal Turing machine is considered as the basis of an algorithm—as it is the case for all contemporary computing—then the number of required basic algorithmic operations can be considered a measure of runtime. When an algorithm is theoretically investigated, then this measure can often be estimated in an asymptotic sense as a function of input data size. This concentrates the essential complexity of an algorithm and removes algorithmic steps that are not significant from the theoretical analysis.

The notation used for this analysis is the big O notation, which is part of the Landau notation.

Definition 1. *Let $f, g : \mathbb{R} \to \mathbb{R}$. One writes:*

$$f(x) = O(g(x)) \quad \text{or} \quad f(x) \in O(g(x)) \tag{1.16}$$

if and only if

$$\limsup_{x\to\infty} \left| \frac{f(x)}{g(x)} \right| < \infty. \tag{1.17}$$

This implicitly removes lower order terms and eases comparison. Some examples of relevant complexity classes are illustrated in Table 1.9. Within this work, $x \to \infty$ is always implied when big O notation is used.

The big O notation implies arithmetic operations, for instance:

$$O(n) \cdot O(n) = O(n^2), \quad O(n) + O(n) = 2O(n) = O(n). \tag{1.18}$$

It is also worth noting that the equality symbol is not symmetric in this notation, such that:

$$O(n) = O(n^2), \quad O(n^2) \neq O(n), \tag{1.19}$$

because for any $f(x) \in O(n^2) : f(x) \in O(n)$, but the reverse does not hold. This notation may appear abusive, but it is a very valuable tool to estimate and express the complexity of an algorithm.

Class	Description	Examples
$O(1)$	constant	lookup in a static table, fixed number of operations
$O(\log(n))$	logarithmic	search in a balanced tree
$O(n)$	linear	search in an unsorted list, sum of n elements
$O(n\log(n))$	loglinear	optimal sorting algorithms, FFT
$O(n^2)$	quadratic	direct convolution, discrete Fourier transform
$O(c^n)$	exponential	Creating a balanced tree of depth n

Table 1.9: Common computational complexities that are relevant to this work

In comparison to benchmarking, time complexity analysis reveals the actual reasons for an input-data dependent run time behavior, and is thus especially interesting when the input data has multiple dimensions of unpredictable size. This work exclusively contains time complexity arguments.

Further operations within this work that are reduced to $O(1)$:

- One step of the Cash-Karp method (see Section 1.3.3.1), applied to the Bloch equation (Equation 1.1) (neglecting adaptive step size adjustments)
- Matrix multiplication of 4 × 4 matrices
- Evaluation of basic functions, in particular the complex-valued exponential function

Constants that are relevant either in scale or due to direct comparability properties are not reduced to $O(1)$. This special case applies to operations that are be performed orders of magnitude more often, and to algorithmically expensive operations that have to be performed a fixed, known number of times.

1.3.2 Fourier transform

Fourier analysis assumes an important role in functional analysis and signal processing. This section should not serve as an entry point to Fourier analysis, but instead establish basic notation and reveal insight into advanced aspects that are required for the justification of later techniques (see Sections 2.3.3.3, 2.4.1.2), thereby reducing the Fourier transform basics of this section to a minimum. This section is based on the definitions and theorems of [10], supplemented by their relevance in MRI theory.

For notational purposes and the sake of completeness, the Fourier transform is defined as follows:

Definition 2. *Let* $f : \mathbb{R} \to \mathbb{C}$ *be integrable. The* Fourier transform *is defined as:*

$$\mathcal{F}(f)(\xi) := \tilde{f}(\xi) := \int_{-\infty}^{\infty} f(x)\mathrm{e}^{-2\pi\mathrm{i}x\xi}\,\mathrm{d}x, \quad \xi \in \mathbb{R} \tag{1.20}$$

If $\tilde{f}$ is also integrable, then f can by recovered by the *inverse Fourier transform* according to the Fourier inversion theorem:

$$\mathcal{F}^{-1}(\tilde{f})(x) := \int_{-\infty}^{\infty} \tilde{f}(x)\mathrm{e}^{2\pi\mathrm{i}x\xi}\,\mathrm{d}\xi = f(x), \quad x \in \mathbb{R} \tag{1.21}$$

The elaboration of basic Fourier transform properties, such as linearity and multiplication-convolution duality, is omitted in this work. The discrete Fourier transform will be introduced later in this section (see Section 1.3.2.2) as a special case of the continuous Fourier transform. The only pair of analytical function and its Fourier transform that is relevant to this work are top-hat function $\chi_{[-1/2,1/2]}$ and sinc function,

$$\mathrm{sinc}(x) = \begin{cases} 1 & \text{for } x = 0 \\ \frac{\sin x}{x} & \text{else} \end{cases}, \tag{1.22}$$

which will be used later to define the voxel basis (see Section 2.4).

1.3.2.1 Spaces

The integrability of f alone merely guarantees that the Fourier transform is well-defined, not that it is itself integrable (only $\in L^{\infty}(\mathbb{R})$). The actual space that a physical f resides in has impact on the Fourier transform, which is important for the justification of algorithms and approximations—especially in the k-space formalism (see Section 2.3) and Sequence Response Kernel (see Section 2.4) sections.

Square integrability
Any signal that can be encountered physically is square integrable because a violation of square integrability means that the instantaneous power of the signal is not limited. Since $L_2([a,b]) \subset L_1([a,b])$, the Fourier transform is defined for any function that can correspond to a physical signal.

Periodicity
The special case of periodicity is of great consequence: An $f \in L_2[a,b]$ that continues periodically now possesses a Fourier series decomposition, which means that a full representation of f in terms of countable Fourier coefficients is possible, and the asymptotic behavior of the series (alongside with it's resulting approximation error) can be estimated if more information about f is available.

Bounded domain
The special case of a bounded $f \in L_2[a,b]$ is similarly convenient: f can be continued by 0 outside its domain—corresponding to multiplication with a top-hat-function, or a convolution with a sinc-function of its Fourier representation. The sinc-interpolation yields no extra information content, thus the Fourier series representation suffices to regenerate f.

Without a bounded domain or periodicity argument the Fourier series representation does not apply, and the Fourier transform is not easily approachable with means of numerical analysis.

It is important to keep in mind that a violation of this continuation is a common occurrence in MRI. This violation results in artifacts that originate in Fourier analysis. The most prominent example of this difficulty is the wrap-around artifact, but advanced reconstruction methods also need to consider this violation.

Bounded variation, continuity and differentiability
For an $f \in L_2([a,b])$, the total variation is defined as:

$$V(f) = \sup_{P \in \mathcal{P}} \sum_{i=0}^{n_P - 1} |f(x_{i+1}) - f(x_i)| \tag{1.23}$$

where

$$\mathcal{P} = \{P = \{x_0, ..., x_{n_P}\} \,|P \text{ is a partition of } [a,b],\, n_P \in \mathbb{N}\} . \tag{1.24}$$

A function that has bounded total variation, i.e. $V(f) < \infty$, also has Fourier coefficients that eventually converge to zero ($f \in BV([a,b]) \Rightarrow \tilde{f}(n) = O(1/n)$). Such functions are often used as a theoretical input for MRI related algorithms, for instance in simulation experiments that assume a perfect sphere that is filled with protons homogeneously. If the total variation of the function can be estimated, then this estimate can be used to express the limiting behavior of the Fourier coefficients for $n \to \infty$.

If the function is furthermore absolutely continuous, then the convergence of the Fourier coefficients for $n \to \infty$ is accelerated ($f \in AC([a,b]) \Rightarrow \tilde{f}(n) = o\left({}^{1}\!/\!{}_{n}\right)$), and even more so if the function is continuously differentiable ($f \in C^r([a,b]) \Rightarrow \tilde{f}(n) = o\left({}^{1}\!/\!{}_{n^r}\right)$).

Schwartz space and tempered distributions
The Schwartz space is defined as:

$$S(\mathbb{R}) := \left\{ f \in C^\infty(\mathbb{R}) | f^{(n)}(t) = O\left({}^{1}\!/\!{}_{|t|^k}\right), \forall n, k = 0, 1, ... \right\}. \tag{1.25}$$

It is of particular importance for Fourier analysis because the Fourier transform is an automorphism on this space. The dual space of $S(\mathbb{R})$, denoted by $S'(\mathbb{R})$ is called the space of all tempered distributions on $\mathbb{R}$.

The space of tempered distribution enables the natural definition of the Fourier transform for distributions that are compatible, most importantly the Dirac distribution δ, which is defined through $\delta[f] := f(0)\,\forall f \in S(\mathbb{R})$, or the Dirac comb, which is the infinite sum of equally spaced shifted Dirac distributions.

Tempered distributions and the transition to the discrete Fourier transform
The Dirac comb is the central theoretical building block of digital signal processing, which is why the space of tempered distributions plays such an important role in Fourier analysis. The Dirac comb is the bridge between the discrete Fourier transform and continuous signals, and is used to prove the Nyquist–Shannon sampling theorem.

Interpreting discrete data as a function multiplied with a Dirac comb is always valid and can greatly help to explain phenomena and artifacts that are present when the conditions of Fourier analysis are stretched in theoretical processing of data.

2D Fourier transform
The Fourier transform can be extended to multiple dimensions directly by treating each extra dimension analogously.

1.3.2.2 Discrete Fourier transform

For a sequence $f \in \mathbb{C}^N, f = (f_0, f_1, ..., f_{N-1})$, the Fourier transform $\mathcal{F} : \mathbb{C}^N \to \mathbb{C}^N$ is defined as:

$$\tilde{f}_k := \sum_{n=0}^{N-1} f_n \mathrm{e}^{-\mathrm{i}2\pi \frac{n}{N} k} \tag{1.26}$$

And the inverse Fourier transform is thus:

$$f_k := \frac{1}{N} \sum_{n=0}^{N-1} \tilde{f}_n \mathrm{e}^{\mathrm{i}2\pi \frac{n}{N} k} \tag{1.27}$$

The discrete Fourier transform is connected to the continuous Fourier transform: If f is the result of the sampling (multiplication with an adequate Dirac comb) of a periodic function, then $\tilde{f}$ corresponds to the dually sampled Fourier transform of f. The rules that hold for the discrete Fourier transform can be proven in an elegant way by attributing them to their continuous counterparts.

Fast Fourier transform (FFT)
If N is a power of two, then the discrete Fourier transform can be calculated in $N \log_2(N)$ multiplications, which is substantially superior to the naive N^2 multiplications that are required if the transform is performed directly. If the samples are not acquired in a compatible way, then they are often gridded onto a compatible sampling to make use of this numerical efficiency.

1.3.3 Numerical analysis

Part of the simulation is of course actual computation of the simulated behavior. This section addresses the most relevant topics of numerical analysis for the scope of this work.

1.3.3.1 Initial value problems

The Bloch equation (Equation 1.1) is the mathematical foundation of MRI. In general, it is an ordinary differential equation—or an initial value problem when applied to a spin packet with a defined initial state.

Definition 3. *An* initial value problem *is a differential equation $y'(t) = f(t, y(t))$ with $f : \Omega \to \mathbb{R}^n$, where $\Omega \subset \mathbb{R} \times \mathbb{R}^n$ is open, together with a point $(t_0, y(t_0)) = (t_0, y_0) \in \Omega$.*

Existence and uniqueness of a solution can be affirmed directly for any circumstance that can be created by physical tools in the context of this work through the Picard–Lindelöf [11] theorem which merely requires continuity of f in t and Lipschitz continuity of f in y.

Since there exists no general solution to the Bloch equation [33], an approximation strategy needs to be pursued for the general case. A great choice is the Runge-Kutta method [25], which basically integrates the equation stepwise in temporal direction until the desired time is reached. Such a step is defined by a Butcher tableau, which approximates the solution within this step up to a certain order. In this work, the Cash-Karp method [5] is employed. This method generates two approximations: one of fourth order and one of fifth order. The local approximation error that is estimated based on those two solutions is then used to adapt the step size to a desired accuracy level.

1.3.3.2 Properties of numerical algorithms

The algorithm that is employed to solve the Bloch equation is guaranteed to provide a local approximation to the real solution for each step. But there are errors present at each step, which may cause significant deviations from the final/global solution. These errors can be estimated and are either intrinsic to the algorithm that is employed or to the problem itself.

Stability

Stability is a property that can be applied to the process of solving a differential equation. Basically, a problem is considered numerically stable if small errors in the initial conditions lead to small errors in the result. A non-stable problem would be one for which small discretization or approximation errors cause the final solution to show significantly different behavior. Luckily, this is not a major problem for the Bloch equation because deviations from the equilibrium configuration decay over time. This means that the problem is asymptotically stable for errors in the magnetization state, and stable for errors in the equilibrium magnetization.

Stability often is a problem when dynamics are present on multiple time scales at the same time, which would be the case when the laboratory reference frame is used to solve the Bloch equation because of the Larmor precession. The numerical properties of the specialized Bloch equation (Equation 1.4) will be discussed in the next chapter (see Section 2.1.1).

Condition

Condition is a measure of output variance under input perturbance. The Bloch equation is well-conditioned since the solution constantly decays towards the equilibrium configuration.

Conclusion

The problems encountered in this work are stable, possess unique solutions and do not evoke significant challenges to reach good approximation quality. Therefore they are not inspected further.

Simulation objectives

MRI is a field at which people from many different expertises come together. The formulas and models that form the MRI knowledge-base are the main tool of communication among specialists, and an application or simulation of those models can be seen as a necessary widening of this foundation.

A universally applicable and efficient simulation can help the understanding of MRI through all expertises, most importantly in research, education and clinical application. This topic will be revisited in the last chapter (see Section 4.2), considering the results of this work.

1.4.1 Research

Research is the area at which a simulation has the strongest impact. Each new method needs to be designed, implemented and evaluated. A well-performing simulation is able to test novel methods in an adequate and thorough way.

The most common practice to test a new idea is to first use a simulation that is specialized to the effects relevant to this new method, estimate the results, and then compare those results to physical measurements. The problem with this path is that the validation is only carried out at the very end of this process. Intermediate results cannot be compared because the scanner is bound to physical limitations and the new method needs to be fully implemented prior to the validation.

The simulation that is employed in this scenario is built in a rigid way and most likely reinvented completely due to the lack of simulation standards or optimized approaches. This hinders further development and the investigation of the simulation results in the absence of physical measurements.

1.4.1.1 Common simulation tasks

A simulation that attempts to answer a question motivated by research is usually aimed at a very specific context and initial configuration of the system. It then requires a comparison or trend of competing methods, including error measures and sensitivity analysis.

The actual computation of full realistic images is usually not the focus of the simulation task. The performance of a single module within the sequence is the main interest. This could be the individual pulses, sequence timing or gradient pulse strategy.

A simulation for research purposes needs to be adaptable to a high range of scenarios since research is usually performed for extreme special cases. A high simulation speed or a quick response to a change of the experiment's setting is desirable, but usually not required.

Example questions

The following questions are examples of typical questions that are posed towards a simulation tool.

- How does an idealized spin distribution behave in the presence of a single (RF or gradient) pulse?
  - What is the slice profile of an RF pulse?
  - What are the deviations from the ideal case in the presence of offset frequencies, flow or diffusion?
  - How does B1 inhomogeneity deteriorate the pulse performance?
- What effect does one specific sequence design parameter have on spin packets?
  - How robust is the RF pulse slice profile?
  - How good is the background suppression?
  - How do competing gradient pulse strategies compare?
- How do pulse patterns perform?
  - What are the optimal parameters for the individual pulses?
  - Which timing yields optimal signal quality?
  - What is the influence of Flow or Diffusion?
  - What side-effects are ignored by a competing mathematical simplification?
  - What side-effects can be expected in the final image?

1.4.1.2 Impact of an efficient simulation

Efficient simulation tools are essential for further development of simulation based methods. An efficient and applicable simulation reduces the need for simplified models and can partially replace them with accurate calculations. Simulation based research such as MR Fingerprinting [27] can only be practiced if a proper simulation tool is available.

Implementing and testing a new technique on the MR scanner is time-consuming and lacks debugging means. An approximation of low accuracy or exhaustive simulation time is often seen as an overhead that is of low value to research and development. An efficient simulation can best support the development of new techniques and theories if it is accurate and fast. The simulation can thereby also provide insight into the state of the system, opposed to having to interpret the final result of the measurement process. This added insight and reduced implementation effort can shorten the development cycle and aid the investigation of phenomena that might arise for a new technique.

1.4.2 Education

Concepts and ideas need to be communicated in all scientific fields. Abstract models and formulas can often be grasped more easily if they are supported by examples, applications or metaphors. The MRI learning curve is generally considered to be steep, and much effort is put into explaining it for each knowledge level and educational background. MR effects are often explained in an illustrative way, supported by gestures and 3D hand-waving. The correct medium for this explanatory style is a 3D animation, and there are various attempts to establish a tool of this kind [16] [29]. However, their performance and further applicability is limited and the gap to current research is still wide. Thus the education stops directly after the basic level.

Education as a simulation task should not only be understood as the transfer of basic knowledge. Each new idea that is carried to a person even in the same and advanced scientific field needs a form of education.

1.4.2.1 Common simulation tasks

An educational tool usually does not need to be quantitatively accurate. It needs to show the general behavior and be highly responsive. It should be easy to perform experiments, allow for various different views on the system, and be easy to adapt to a different special case.

Example questions
The following example questions are typical educational tasks:

- How does a small set of spin packets behave as it is exposed to the sequence?
- How can the spin configuration be interpreted, how should it be read?
- Which effects cause the system to behave undesirably?
- Why does a change in the sequence produce the desired result?

1.4.2.2 Impact of an efficient simulation

An efficient MRI simulation that is used as an educational tool serves as an entrance point to research and a communication medium. Ideas should be easy to illustrate and thus educate after development. Ideas are also more trustworthy and testable if they are supported by a transparent, customizable simulation. The transition to research scenarios should be smooth.

1.4.3 Clinical Application

A simulation of the MR-effect in the context of MRI is especially relevant for clinical diagnostics, clinical studies, and the definition of measurement protocols. At all these stages, MRI simulation is rarely used.

The device that is responsible for controlling the MRI system needs to be highly quality compliant. As such, for the image acquisition stage, the application developer needs to work with limited freedom, increased implementation obstacles and confined computational resources. Fully featured simulation is infeasible in this environment. Thusly, specializations, simplified models, or precalculated approximations are necessary.

After the measurement, the data is post-processed to enable a diagnosis. This step has potentially all required liberties and resources, but the patient can be assumed to be not measurable under equal conditions after the post-processing. But even the post-processing step is rarely supported by a simulation due to the high computational effort, required quality compliance, and limited infrastructure at clinical sites.

1.4.3.1 Common simulation tasks

Clinical applications need to be highly reliable and their results need to have a known and high quality. They also have to be fast for post-processing applications and very fast for tasks directly relevant to the acquisition process.

Example questions
The following example questions are related to clinical applications:

- How should the sequence be configured to achieve the best contrast?
- How can differently acquired images be interpreted together?
- How can artifacts be avoided?
- How reliable is the measurement?

1.4.3.2 Impact of an efficient simulation

Simulation within the clinical environment can simplify the measurement process. The sequence can be tuned algorithmically instead of by experience—removing the strain of finding suitable parameters. A simulation tool could also anticipate problems that are likely to occur and provide an option to control those problems directly—in contrast to indirect tweaking of raw sequence parameters.

The post-processing steps could be enhanced by adding knowledge, gained by a virtual simulation of a patient-specific experiment. Image sets acquired with different sequences are commonly interpreted individually, because they are not directly comparable. A simulation can connect the information and derive more robust and physiological parameters from the compound data. It is also feasible to conjoin images acquired with differing sequence, differing hardware specifications, and differing vendors.

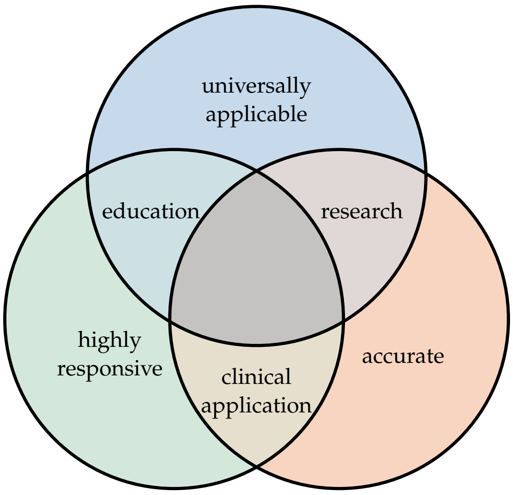


Figure 1.10: Venn-Diagram of MRI simulation requirements and application areas.

CHAPTER 2

Methods

2.0 Introduction

Presented in this chapter are the simulation methods. These methods are the main focus of this work. Each method is investigated and optimized with full and accurate imaging experiment conditions in mind—opposed to the task of simulating only parts of a pulse sequence or neglecting the imaged object completely. Each of the four approaches' sections concludes by presenting the workflow that is needed for a full simulation experiment.

Direct Bloch simulation (see Section 2.1) is the most common and also most general way to perform a simulation experiment. By exploiting MRI sequence properties, the Smart Bloch solver (see Section 2.2) improves the common approach to achieve increased efficiency without sacrificing accuracy. The k-space simulation method (see Section 2.3) is based on the extended phase graphs [19] idea, refined to enable accurate results and to become applicable to a larger scope of problems. The Sequence Response Kernel method (see Section 2.4) further extends the k-space based simulation to simulate directly in image space by strongly decoupling the tissue-dependent and geometry-dependent contributions of the simulation process. The last section of this chapter (see Section 2.5) showcases applications that can be handled through the final approach.

The results chapter (Chapter 3) provides simulation results for each method, further inspects efficiency and illustrates the intrinsic potential and imperfections of the approaches.

Direct Bloch simulation 2.1

The Bloch equation is a simple ODE which can easily and accurately be solved numerically using a Runge–Kutta method.

This section briefly establishes the state of the art. The techniques and numerical details are not of high interest for the following approaches and further development of these approaches is not part of this work, which is why this section is kept short.

2.1.1 Numerical properties

From a numerical point of view, the Bloch equation is very well behaved (also see Section 1.3.3). The magnetization decays exponentially towards its equilibrium state $M = (0, 0, M_0)$.

The Bloch equation describes the behavior of a spin packet. While doing so, it implies that the individual spin packets evolve independently, which is true at the abstraction level at which the Bloch equation holds.

2.1.1.1 MRI-specific properties of the Bloch equation

There are multiple relevant timescales in a common simulation experiment. The main magnetic field of the MRI device induces a Larmor precession which is on a timescale at which no other process is important. This numerical complexity can easily be removed by using a rotating frame of reference with a frequency equaling the Larmor frequency for the main magnetic field. In that frame of reference, spin packets do not experience the B_0 field, transverse fields are modulated with negative Larmor frequency, and all other fields are unaffected. Within this work, all magnetic fields are assumed to be in terms of this rotating frame of reference.

The automatic step size adaptation of the Cash-Karp method can lead to improper results if dynamics are present that occur on a time scale that is orders of magnitude below the current step size—as it is often the case for MRI pulse sequences. Many MRI sequences consist of short time spans during which RF pulses are applied, followed by prolonged silence on all channels, and gradient waveforms with sudden changes. An adaptive step size solver is likely to miss those changes if used directly. It is necessary to choose an adequate minimum step size or alternatively to define time points at which the solver reconfigures its step size.

2.1.1.2 Phantom sampling

Spin packets have to be simulated individually, and choosing a set of spin packets that produces accurate results is no trivial task. Systematic sampling according to a grid is likely to introduce aliasing errors in the signal. Sampling only few spin packets per imaging voxel fails to resolve sub-voxel effects that are common for MRI sequences, and the T2′ effect (see Section 1.1.1.3) requires even more samples according to the offset frequency distribution.

The sampling of spin packets should in general be randomized and highly resolved.

2.1.1.3 Nonexistence of a general analytical solution

Even though the Bloch equation is numerically well behaved, there is no feasible analytical solution for the general problem.. Only special cases with particularly organized patterns can be solved analytically.

In fact, it is possible to find a B-field that produces any spatial/frequency distribution in any finite time span. The approaches to do so are well-known and investigated, for instance concerning the design of volume-selective pulses.

More information about the nonexistence of analytical solutions can be found in [33].

2.1.2 State of the art

The simplicity of the Bloch equation has lead to many implementations of a general simulation. The equation is easy to extend by further physical effects or system imperfections and can be applied to any problem. Many sophisticated solutions embrace this simplicity and concentrate on the framework, extensibility, applications, or parallelization of the simulation code [22] [35] [2] [14] [40].

But this flexibility comes at the price of computational effort. The experiments can only be performed at low resolutions given reasonable time and resource constraints. This limits the significance of the tool to basic verification of sequences and a preparation for non-virtual experiments.

The algorithmic optimization of the mentioned implementations is limited to basic numerical improvements such as the mandatory rotating reference frame or step size awareness. Attempts of numerical optimization are limited to special cases [1]. The most sophisticated improvement is a first-order approximation of the magnetization dispersion within a voxel, which can be found in [21]. The main focus of established software is the inclusion of physical phenomena such as magnetic susceptibility, chemical shift, or customizable virtual coils. The inevitable computational effort is exclusively answered by parallelization.

2.1.3 Algorithmic potential

The great potential that makes it possible to handle the Bloch equation conveniently, is parallelization. The spins do not interact, and the signal accumulation is performed globally. If there is a great number of spin packets that are to be simulated, then the workload can be separated in an arbitrary way.

But this does not remove the need for efficient simulation. While it may be tempting that implementing the Bloch equation alone does not depend on further numerical tweaks to provide accurate results, it is often not possible to choose a number of spin packets that produce sufficiently realistic results.

However, the Bloch equation is easy to extend by further effects such as complicated magnetic fields or spin packet behavior, which makes the direct Bloch simulation a reasonable choice when such features are relevant and need to be investigated transparently.

2.1.4 Full simulation workflow

First, the phantom needs to be prepared because the Bloch equation requires sufficiently many spin packets. It is not only necessary to choose a sufficiently fine sampling of the spin packets, but also to avoid regularity.

The sequence description needs to be translated to the right hand side of the Bloch equation and the times at which signal is acquired need to be extracted. It may also be necessary to deduce hints about discontinuities and sudden changes in the magnetic field that the ODE solver is likely to require, depending on the implementation.

Now the state of a spin packet for a given point in time can be calculated by integrating the ODE. This has to be done for each acquisition time, at which the total signal needs to be weighted and accumulated for each coil.

The signal is then complete, in the form of one complex-valued sample per acquisition time, per coil. It can then be reconstructed using the same algorithms that are employed on the MRI system.

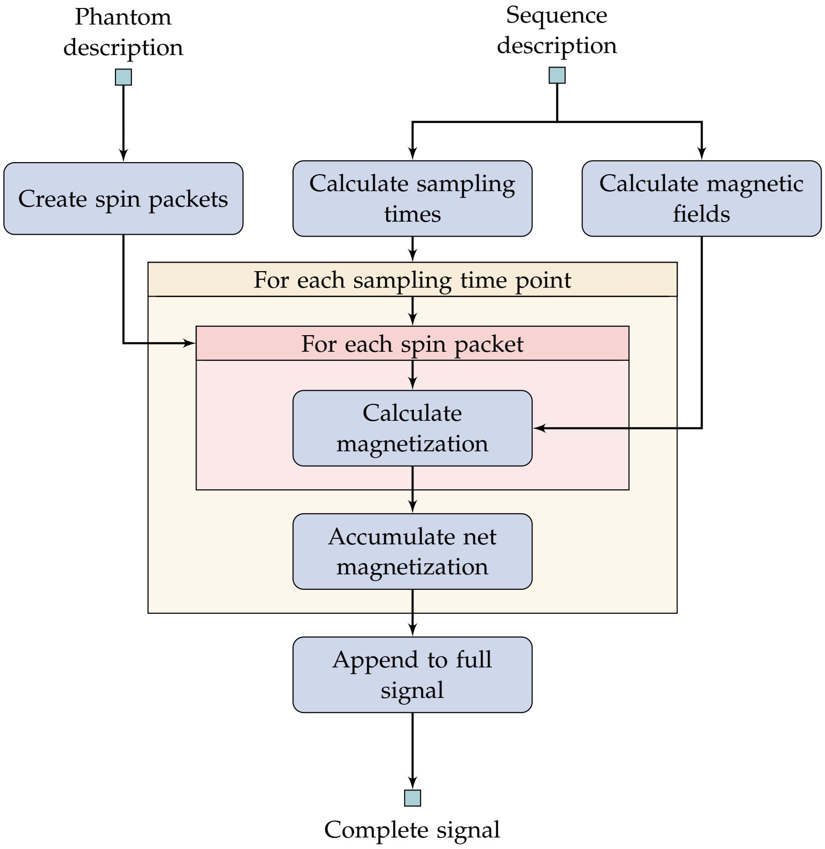


Figure 2.1: Direct Bloch simulation workflow

2.2 Smart Bloch simulation

The hardware setup of an MRI system specializes the Bloch equation, as described in Equation 1.4. Furthermore, the RF pulses that are part of a pulse sequence are often repeated or applied in a similar fashion. These two facts can be exploited to yield accelerated ODE integration with no drawback in accuracy.

This section introduces concepts to apply analytical solutions where possible and re-use simulation results from RF pulse simulation. This is followed by the workflow for full imaging experiment simulation. The benefit and efficiency of this approach will be elaborated in sections 3.2 and 3.5.

2.2.1 Pulse idealization

RF pulses are usually thought of as acting as a rotation on the magnetization within a certain on-resonance bandwidth. This idea is mostly sufficient to understand a sequence, or to work on effects that are not directly affected by imperfection of this assumption.

A solution that is often applied and closer to reality is the so called hard pulse approximation, which basically integrates the Bloch Equation using the forward Euler Method. There are no reasons to prefer this approach to a solution using a more sophisticated solver when it comes to solving the Bloch Equation, except for a byproduct of the hard pulse approximation that allows inversion and can directly generate a pulse shape for a desired pulse profile [31].

The following paragraphs define a workflow that possesses the full exactness of a Direct Bloch simulated pulse, but is also as easy to use as the rotation approximation of RF pulses.

2.2.1.1 Theory

A spin packet can be separated into smaller spin packets. This is particularly useful when the effect of an RF pulse on similar spin packets is of interest.

The Bloch equation (Equation 1.1) is a homogeneous and linear equation, thus the superposition principle holds:

$$f(\overrightarrow{M}) := \frac{\partial}{\partial t}\overrightarrow{M} = \gamma\overrightarrow{M}\times\overrightarrow{B} - \frac{M_x\overrightarrow{e_x} + M_y\overrightarrow{e_y}}{\mathrm{T2}} + \frac{(M_0 - M_z)\overrightarrow{e_z}}{\mathrm{T1}} \tag{2.1}$$

$$f(a\overrightarrow{M^{(a)}} + b\overrightarrow{M^{(b)}}) = af(\overrightarrow{M^{(a)}}) + bf(\overrightarrow{M^{(b)}}) \tag{2.2}$$

Consequently, any spin packet can be decomposed into a linear combination of four basis spin packets. To ease the notation in the following paragraphs, the magnetization vector is supplemented by the spin packet's equilibrium magnetization magnitude to form a vector of dimension four. The basis spin packets are:

$$\overrightarrow{M^{(1)}} = (1,0,0,1), \quad \overrightarrow{M^{(2)}} = (0,1,0,1), \quad \overrightarrow{M^{(3)}} = (0,0,1,1), \quad \overrightarrow{M^{(4)}} = (0,0,0,1). \tag{2.3}$$

Let $\overrightarrow{M} = (M_x, M_y, M_z, M_0)$. Choose $d = M_0 - M_x - M_y - M_z$, then:

$$\overrightarrow{M} = M_x\overrightarrow{M^{(1)}} + M_y\overrightarrow{M^{(2)}} + M_z\overrightarrow{M^{(3)}} + d\overrightarrow{M^{(4)}} \tag{2.4}$$

Or equivalently:

$$\overrightarrow{M} = M_x(\overrightarrow{M^{(1)}} - \overrightarrow{M^{(4)}}) + M_y(\overrightarrow{M^{(2)}} - \overrightarrow{M^{(4)}}) + M_z(\overrightarrow{M^{(3)}} - \overrightarrow{M^{(4)}}) + M_0\overrightarrow{M^{(4)}} \tag{2.5}$$

The evolution of the combined sub-packets when exposed to the Bloch equation will always follow that of the original spin packet.

The choice of, e.g., $\overrightarrow{M^{(1)}} = (1, 0, 0, 1)$ instead of $\overrightarrow{M^{(1)}} = (1, 0, 0, 0)$, is due to the fact that spin packets without equilibrium magnetization would not be physically reasonable, because no spin packet can be in this configuration—which would lead to computational difficulties.

This separation into elemental spin packets could potentially be maintained for the duration of the whole sequence, but that would further increase the numerical effort. In this approach, the separation is only kept during a segment of the sequence, during which pulses are present. After that, the sub-packets are recombined back into the original spin packet:

Let $\overrightarrow{M^{(i)+}}$ denote the magnetization of $\overrightarrow{M^{(i)}}$ after the pulse for $i \in \{1, \ldots, 4\}$. Let

$$\begin{aligned} P &= \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} (\overrightarrow{M^{(1)+}} - \overrightarrow{M^{(4)+}}) & (\overrightarrow{M^{(2)+}} - \overrightarrow{M^{(4)+}}) & (\overrightarrow{M^{(3)+}} - \overrightarrow{M^{(4)+}}) & \overrightarrow{M^{(4)+}} \end{pmatrix} \end{aligned} \tag{2.6}$$

The spin packet $\overrightarrow{M}^- = (M_x, M_y, M_z, M_0)$ will evolve to $\overrightarrow{M}^+ = P \cdot \overrightarrow{M}^-$ after the pulse.

The great efficiency gain of this approach now lies in the fact that once the pulse is simulated for the four elemental spin packets, a spin packet with equal relaxation parameters that experiences the same magnetic field but is oriented a different way, can evolve through the pulse duration in one matrix multiplication.

This potentially very helpful speedup, that does not deteriorate the solution in any way, is ignored by established simulation approaches.

2.2.1.2 Dimensional reduction

All RF pulses can be categorized according to the gradient pulses that are applied simultaneously. Less exotic gradient pulse shapes allow for dimensionality reduction of the degrees of freedom in the spin packet's tissue parameters.

Whenever a dimensionality reduction is feasible, a potentially large number of spin packets share the same pulse effect. If the results of all pulse simulations are stored, then those similar spin packets can directly use these results instead of requiring recalculation.

Non-selective pulses

An RF pulse is called non-selective when no gradient pulse is applied simultaneously. They are moderately uncommon and mostly used for preparation purposes. The term non-selective might be misleading since it still selects a certain frequency band due to its temporal extension, but they are equally selective at any spatial position—coining the term.

If no gradient pulse is present, then the position of the spin packet is irrelevant for the Bloch equation's right hand side. The tissue parameters (e.g. $T1, T2, \Delta\omega$) and the coil transmissivity do of course depend on the position, but two spin packets that possess the same environmental and tissue parameters will experience the same pulse effect.

Slice selective pulses

In this work, a slice selective pulse is defined as a pulse that is applied in combination with a constant gradient pulse for the duration of the pulse. Slice selective pulses are the most common kind of pulse.

A gradient with a constant amplitude and direction results in a constant term in the Bloch equation's right hand side—and this fact can be exploited. The gradient influence on the change in $\frac{\partial}{\partial t}\vec{M}$ is given by:

$$\frac{\partial}{\partial t}\overrightarrow{M_{grad}}(\vec{r}) = \gamma\vec{M} \times \left(\vec{G} \cdot \vec{r}\,\vec{e_z}\right) \tag{2.7}$$

This is an extra offset frequency of the spin packet, that is determined by its position in slice direction. The slice position can be merged with the also constant resonance offset frequency $\Delta\omega$ which is present due to T2′ and the chemical environment of the spin packet. Even though the position of a spin packet is relevant for the pulse effect of slice selective pulses, the equivalent pulse effect is experienced by a spin packet at the isocenter with the corresponding offset frequency. Spin packets of equal parameters except for slice position and offset frequency share a one-dimensional collection of pulse effects.

The directions orthogonal to slice direction still have no influence on the pulse effect.

Variable rate pulses

A pulse is considered variable rate within this work if the gradient pulse applied during the pulse has a constant direction, but not a constant amplitude. Pulses that are commonly referred to as variable rate pulses are often used for safety-compliance reasons because they allow an RF pulse to be of smaller amplitude albeit for the price of a longer duration, effectively lessening the energy decomposition, but still yielding the same effect in slice direction. Pulses with this distinguishing feature are also used for 1D-spatial/spectral selection and not commonly called variable rate pulses, which however does not pose a difference for the analysis carried out in this paragraph. 1D-spatial/spectral selection pulses and variable rate pulses are rarely used and can often be replaced by a combination of slice-selective and non-selective pulses.

The position in slice-direction can now no longer be translated to a constant offset frequency, thus these two dimensions do not share common solutions any more. However, the directions perpendicular to slice direction still have no direct influence in the Bloch equation and can still be ignored.

3D/4D selective pulses

Any pulse that does not fall into either of the former categories is technically selective in multiple spatial dimensions and frequency dimension. And indeed these pulses are mostly used for exactly that purpose. But a well-tailored volume-selective pulse takes a rather long time and puts a high strain on the patient in terms of energy decomposition—and even then they are always followed by complicated side-effects. For these reasons they are often avoided and even in its main area of application, which is spectroscopy, often replaced by simpler pulses.

This is the worst case in terms of reusing calculated pulse effects. The experienced pulse effect depends directly on position in a way that is devoid of a methodological simplification based on above assumptions and since there can only be one tissue property and transmissivity at one position, nothing is won by not calculting the pulse effect for each spin packet individually. However, if the same pulse is applied multiple times and the spin packet parameters are constant, then it can be reapplied without solving the Bloch equation again.

Conclusion

Each gradient complexity comes with a restriction towards dimensional reduction. The most common pulses induce effects can be heavily reduced, leaving only the tissue parameters and pulse transmissivity as degrees of freedom for pulse effect.

This fact allows for a potentially great reduction in simulation time, which will be illustrated in Section 3.5.

2.2.1.3 Instantanization correction

A pulse that is applied using the described method of decomposing and recombining spin packets will also induce an implicit time-step that corresponds to the pulse length. This has some practical complications that can easily be resolved. The simulation methods described in the next sections also rely on this correction. This will be referred to as the instantanization correction.

Correction

The effects that occur during the sequence can be expressed by one matrix multiplication for any start and end time where no RF pulse is in-between, and this matrix is always invertible (see Section 2.2.2).

By applying the effects of gradient pulses and relaxation during the pulse inversely in the correct order around the pulse effect, a new pulse effect matrix $\tilde{P}$ is generated as illustrated (Figure 2.2). This new pulse effect matrix produces equivalent results in the context of the whole sequence simulation but does not imply a temporal jump.

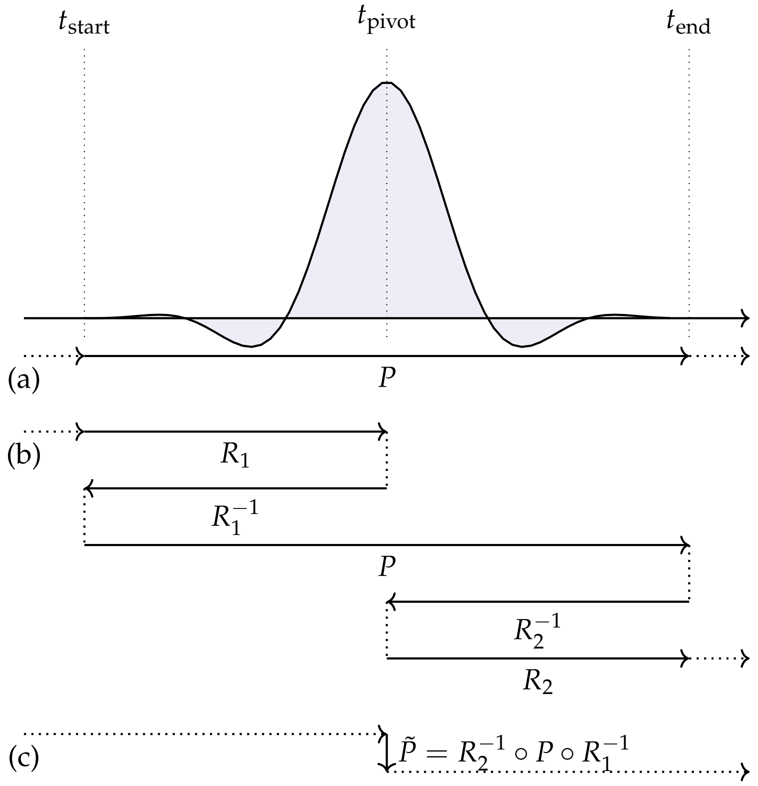


Figure 2.2: Schematic description of pulse instantanization. The uncorrected pulse effect matrix implies a time step (a). It is possible to remove this time step (c), by correcting as illustrated in (b).

After choosing an arbitrary pivot point, e.g., the center of the pulse, the matrix description of gradient and relaxation effects during the pulse to and from the pivot point can be calculated analytically for a given tissue set, yielding R_1 and R_2. The inverse of these matrices corrects for the implicit time steps, such that

$$\tilde{P} = R_2^{-1} \circ P \circ R_1^{-1} \tag{2.8}$$

has the same effect, but does not imply a time step.

Simulations using P and $\tilde{P}$ pulse effects produce equal results, except when evaluated at time points that are within the time span of the pulse. This is acceptable when whole sequences are to be simulated, because no signal is measured during that time.

But even when the simulation of singular pulses are of interest, the instantanization correction can be helpful, because the influence of tissue parameters is then separated from the bias of the pulse duration's influence. This will be shown in the results chapter (see Section 3.2.1).

2.2.2 Magnetization state and signal calculation

The Bloch equation (Equation 1.4) is greatly simplified in the sections of the sequence in which no RF pulse is active. The effects of the M_z component are completely decoupled from the transverse effects, and the equation has an analytical solution. The gradient pulses cause further precession, and the relaxation effects commute with the precession effects.

The gradient pulse and relaxation effects on a spin packet M during the time-interval $[t^-, t^+]$ can be calculated as follows:

$$M_\perp(\overrightarrow{r}, t^+) = M_\perp(\overrightarrow{r}, t^-)\mathrm{e}^{\mathrm{i}\varphi(\overrightarrow{r}, [t^-, t^+])}\,\mathrm{e}^{-(t^+ - t^-)/\mathrm{T2}} \tag{2.9}$$

$$M_z(\overrightarrow{r}, t^+) = M_0 - (M_0 - M_z(\overrightarrow{r}, t^-))\mathrm{e}^{-(t^+ - t^-)/\mathrm{T1}} \tag{2.10}$$

$$\varphi(\overrightarrow{r}, [t^-, t^+]) := \gamma \overrightarrow{r} \cdot \int_{t^-}^{t^+} \overrightarrow{G}(t)\,\mathrm{d}t. \tag{2.11}$$

The 0^{th} moment of the gradient channel $\int_{t_-}^{t_+} \overrightarrow{G}(t)\,\mathrm{d}t$ is the most expensive calculation within the solution, but it is completely independent of the object and can be directly calculated for a given sequence. Further dynamics of B_z within the Bloch equation (Equation 1.4) result in additional phase effects that can be treated independently, or, in case of B_0, remedied by choosing a suitable rotating reference frame.

Since this analytical approach is only possible for sections between the RF pulses, it is most reasonable to choose the integration starting points t_- to be at the end of each pulse, or in case of instantanization correction directly at the pulse pivot points.

For each time point in the sequence outside of RF pulse sections, the spin packet's magnetization can be calculated from its magnetization at the end of the previous pulse in one step with constant run time.

These analytical solutions are only valid for static tissue. Flow and diffusion can also be treated analytically to some degree, as explained in later parts of this work (see Section 4.1.1.1).

2.2.2.1 Signal acquisition

As for the signal acquisition process in the Smart Bloch Solver approach, no improvement over the direct Bloch Solver is pursued. After the magnetizations of all spin packets are calculated, they have to be accumulated for each coil, weighted by the local coil sensitivity.

This process often requires most computational resources, which will be illustrated in Section 3.5.

The following two simulation methods employ techniques to resolve this issue.

2.2.3 Full simulation work-flow

The full workflow for this method is separated into three steps: Pulse effect calculation (Figure 2.3), RF pulse state calculation (Figure 2.4) and signal calculation (Figure 2.5).

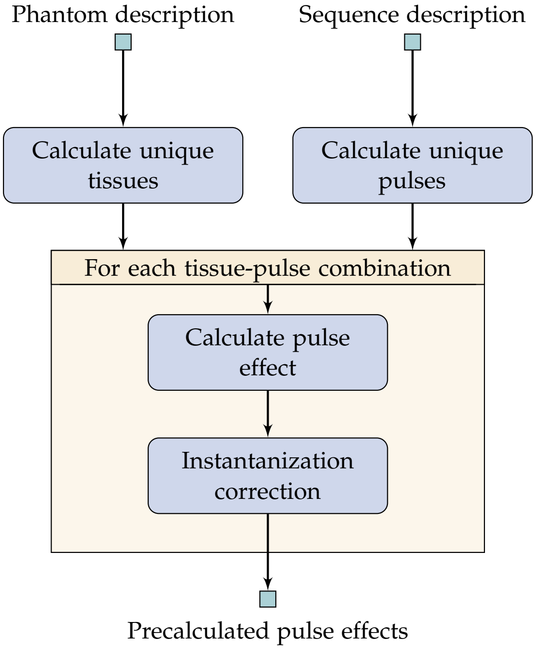


Figure 2.3: Smart Bloch Simulation workflow—Pulse precalculation

The pulse effects need to be calculated for each unique tissue parameter and each unique pulse (Figure 2.3).

The pulses need to be precalculated in ranges according to their relevant effects. These are usually T1, T2, offset frequency and coil transmissivity. However, T1 and T2 are potentially numerically negligible during the RF pulse, reducing the computational effort. On the other hand, pulses that are classified as variable rate or volume selective can be infeasible to precalculate and may rather be calculated on demand.

A great side effect of this precalculation is that pulses which are reused in the sequence only need to be calculated once according to the dimensionality reduction. This also holds for slice selective pulses that are modulated with different resonance frequencies, different phases or at different gradient amplitudes, which basically corresponds to a different slice selection and a rotation of the pule effect. This is applicable to most common pulse sequences, which have no more than a handful different pulses.

After all required pulse simulations have been identified, they have to be performed for the four basis spin packets, corrected, and stored for usage in the next step.

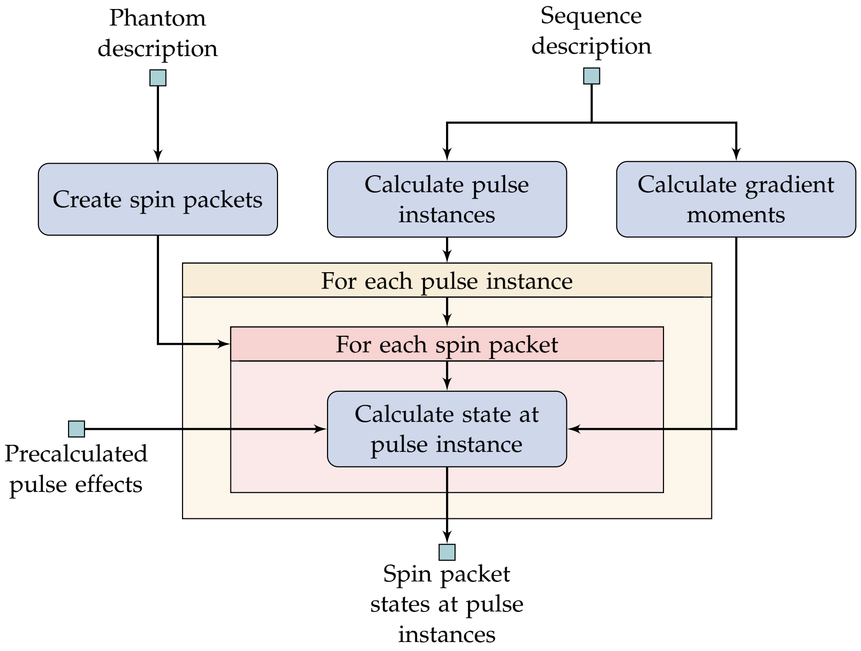

Phantom description
Sequence description
Create spin packets
Calculate pulse instances
Calculate gradient moments
For each pulse instance
For each spin packet
Calculate state at pulse instance
Precalculated pulse effects
Spin packet states at pulse instances

Figure 2.4: Smart Bloch simulation workflow—RF pulse state calculation

The goal of the second step (Figure 2.4) is to calculate the state of the system at the pivot points of the RF pulses.

The individual spin packets need to be generated in the same way as it was the case for the Direct Bloch simulation.

To perform the progression from the state at one pulse to the state at the next pulse, the gradient moments between these two pulse times need to be calculated from the sequence description.

The transition of a spin packet's state at one pulse instance to the state at the next pulse instance requires one analytical step. This step depends on the spin packet parameters and gradient moments in the time span between the two RF pulse pivot points. Afterwards, the correct pulse effect that was calculated in the previous workflow step needs to be retrieved and applied.

The spin packet states are then stored to be used in the next workflow step. The storage is not an essential step since those states can be iteratively processed and then discarded if only the final signal is of interest. But the saving eases parallelization in the following processes, and can also be reused if different coil settings need to be considered.

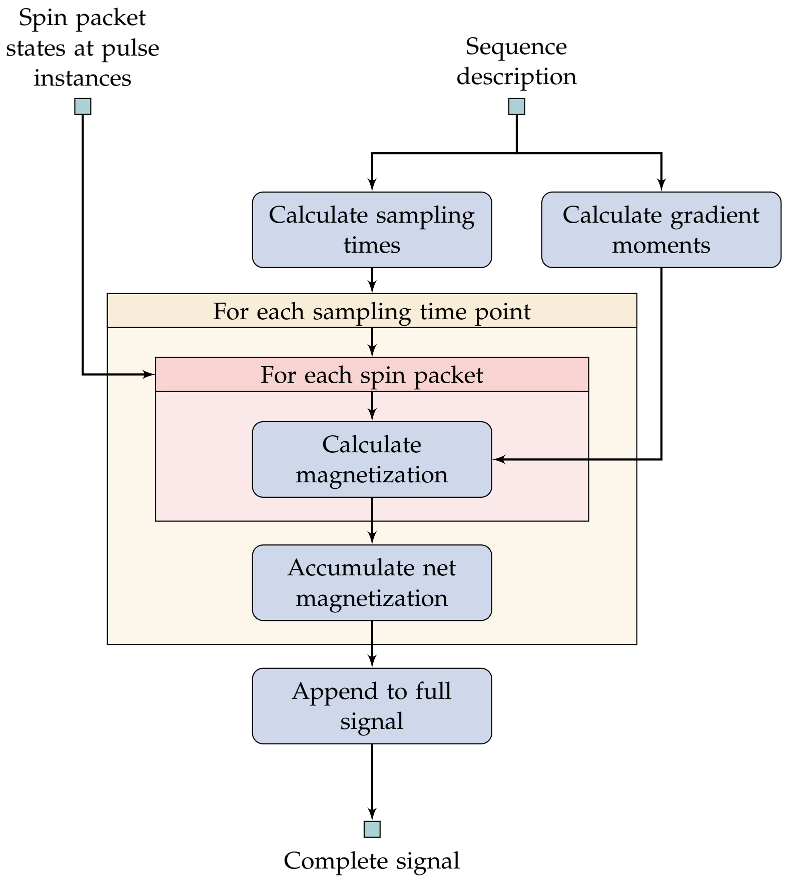


Figure 2.5: Smart Bloch simulation workflow—Signal calculation

In the last step (Figure 2.5) the coil signals are calculated, based on the previously generated system states.

This step requires the gradient moments, measured from each RF pulse instance pivot point to the acquisition times following that RF pulse directly. This enables fast calculation of the system's state at the individual acquisition times.

The signal is then weighted and accumulated in the same way as it was performed for the direct Bloch simulation.

2.3 k-space based simulation

The Direct Bloch simulation and Smart Bloch simulation approaches require a large number of particles that have to be traced individually. However, behavior of the particles is systematically similar due to the MRI Bloch equation (Equation 1.4). Transforming the problem into a suitable frequency space exploits some of this similarity. The transformed problem is in general equivalent and the computational cost may be reduced drastically.

The method described in this section is related to the extended phase graph algorithm [19]. The methodological and functional differences and improvements are elaborated in the later paragraphs of this section (see Section 2.3.4.4).

The method described in this section is only efficient and computationally feasible if phantom and sequence are compatible. The workflow (see Section 2.3.5) and effort estimation (see Section 3.5) will cast light on this issue.

2.3.1 Concept and basis

The original formulation of the extended phase graph algorithm is not used as a starting point, because it includes simplifications that are not reasonable for full MR imaging simulation. Even the basic description of the system state is handled in an alternative way to allow for full accuracy.

The general idea however is the same. RF pulses transform transverse and longitudinal magnetization; gradient pulses and offset frequency act on the magnetization by inducing a phase accumulation which can be seen as a shift of the Fourier transformed transverse magnetization, as foreshadowed by the definition of k-space (see Section 1.2.1).

This and the following part will formally introduce a Fourier transform inspired state system and transfer the results of the Smart Bloch Solver to this new system.

2.3.1.1 Basis definitions

The following paragraphs will introduce a pair of spatial and frequency bases, such that effects can easily be translated between the two, in a way that is equally exact if the object possesses adequate spatial and frequency descriptions. To improve readability, vector arrows are omitted for location, gradient pulse, and frequency vectors.

Spatial domain

The magnetization of a spin packet can be expressed by a four-dimensional vector describing its spatial components and equilibrium magnitude. To express the location of the spin packet, an indicator function $\chi_{r_0}(r)$ can be used which returns one if r and r_0 match, e.g., are in the same voxel, and zero otherwise. In the limiting case of non-discrete resolution, χ degenerates to a δ-distribution. More formally: A spin packet's magnetization can be described by:

$$\begin{pmatrix} M_x^{r_0} \\ M_y^{r_0} \\ M_z^{r_0} \\ M_0^{r_0} \end{pmatrix}_{r=r_0} := M_x^{r_0} \overrightarrow{e_x^{r_0}} + M_y^{r_0} \overrightarrow{e_y^{r_0}} + M_z^{r_0} \overrightarrow{e_z^{r_0}} + M_0^{r_0} \overrightarrow{e_0^{r_0}} \tag{2.12}$$

where $M_x^{r_0}, M_y^{r_0}, M_z^{r_0}, M_0^{r_0} \in \mathbb{R}$, and

$$\overrightarrow{e_x^{r_0}} := \chi_{r_0}(r) \overrightarrow{e_x} \tag{2.13}$$

$$\overrightarrow{e_y^{r_0}} := \chi_{r_0}(r) \overrightarrow{e_y} \tag{2.14}$$

$$\overrightarrow{e_z^{r_0}} := \chi_{r_0}(r) \overrightarrow{e_z} \tag{2.15}$$

$$\overrightarrow{e_0^{r_0}} := \chi_{r_0}(r) \overrightarrow{e_0}. \tag{2.16}$$

This notation then motivates the basis generating set

$$\mathbb{B}_r := \left\{ \begin{pmatrix} M_x^{r_0} \\ M_y^{r_0} \\ M_z^{r_0} \\ M_0^{r_0} \end{pmatrix}_{r=r_0} : M_x^{r_0}, M_y^{r_0}, M_z^{r_0}, M_0^{r_0} \in \mathbb{R} \qquad r_0 \in \Omega_r \subseteq \mathbb{R}^3 \right\}, \tag{2.17}$$

which can be used to describe the magnetization state of a spatially resolved object that is compatible with the choice of Ω_r and χ.

In particular, the equilibrium state of an object $A \in \operatorname{span}(\mathbb{B}_r)$ with proton density $\rho(r)$ is given by:

$$A_0 := \sum_{r_0 \in \Omega_r} \begin{pmatrix} 0 \\ 0 \\ \rho(r_0) \\ \rho(r_0) \end{pmatrix}_{r=r_0} \tag{2.18}$$

Frequency domain

Likewise, under the assumption that the phantom that is to be simulated has a proton density that can be represented in terms of the Fourier basis, any magnetization can be described in frequency domain, which uses wavenumbers to express the spatial variation instead of voxels:

$$\begin{pmatrix} M_v^{k_0} \\ M_w^{k_0} \\ M_s^{k_0} \\ M_t^{k_0} \end{pmatrix}_{k=k_0} := M_v^{r_0} \overrightarrow{e_v^{k_0}} + M_w^{r_0} \overrightarrow{e_w^{k_0}} + M_s^{r_0} \overrightarrow{e_s^{k_0}} + M_t^{r_0} \overrightarrow{e_t^{k_0}} \tag{2.19}$$

where $M_v^{k_0}, M_w^{k_0}, M_s^{k_0}, M_t^{k_0} \in \mathbb{C}$, and

$$\begin{aligned}
z &= |z|\,\mathrm{e}^{\mathrm{i}\theta} = a + b\mathrm{i} \in \mathbb{C} && (2.20)\\
z\overrightarrow{e_\perp} &:= a\,\overrightarrow{e_x} + b\,\overrightarrow{e_y} && (2.21)\\
z\overrightarrow{e_v^{k_0}} &:= z\mathrm{e}^{\mathrm{i}(k_0\cdot r)}\overrightarrow{e_\perp} = |z|\left(\cos(k_0 r + \theta)\,\overrightarrow{e_x} + \sin(k_0 r + \theta)\,\overrightarrow{e_y}\right) && (2.22)\\
z\overrightarrow{e_w^{k_0}} &:= \bar{z}\mathrm{e}^{-\mathrm{i}(k_0\cdot r)}\overrightarrow{e_\perp} = |z|\left(\cos(k_0 r + \theta)\,\overrightarrow{e_x} - \sin(k_0 r + \theta)\,\overrightarrow{e_y}\right) && (2.23)\\
z\overrightarrow{e_s^{k_0}} &:= \frac{1}{2}\left(z\mathrm{e}^{\mathrm{i}(k_0\cdot r)} + \bar{z}\mathrm{e}^{-\mathrm{i}(k_0\cdot r)}\right)\overrightarrow{e_z} = |z|\cos(k_0 r + \theta)\,\overrightarrow{e_z} && (2.24)\\
z\overrightarrow{e_t^{k_0}} &:= \frac{1}{2}\left(z\mathrm{e}^{\mathrm{i}(k_0\cdot r)} + \bar{z}\mathrm{e}^{-\mathrm{i}(k_0\cdot r)}\right)\overrightarrow{e_0} = |z|\cos(k_0 r + \theta)\,\overrightarrow{e_0}\,. && (2.25)
\end{aligned}$$

This suggests the following generating set:

$$\mathbb{B}_k := \left\{ \begin{pmatrix} M_v^{k_0} \\ M_w^{k_0} \\ M_s^{k_0} \\ M_t^{k_0} \end{pmatrix}_{k=k_0} : M_v^{k_0}, M_w^{k_0}, M_s^{k_0}, M_t^{k_0} \in \mathbb{C} \qquad k_0 \in \Omega_k \subseteq \mathbb{R}^3 \right\} \tag{2.26}$$

Note that there might be multiple ways to decompose an arbitrary magnetization in terms of $\mathbb{B}_k$ because:

$$\begin{aligned}
z\overrightarrow{e_v^{k_0}} &= \bar{z}\overrightarrow{e_w^{-k_0}} && (2.27)\\
z\overrightarrow{e_s^{k_0}} &= \bar{z}\overrightarrow{e_s^{-k_0}} && (2.28)\\
z\overrightarrow{e_t^{k_0}} &= \bar{z}\overrightarrow{e_t^{-k_0}} && (2.29)
\end{aligned}$$

But this redundant choice of representatives is convenient in later calculations, and it is not necessary for this work to have a unique representation in terms of $\mathbb{B}_k$.

The equilibrium state of an object $A \in \operatorname{span}(\mathbb{B}_k)$ with Fourier transformed density $\tilde{\rho}(k) = \mathcal{F}(\rho(r))$ can be expressed by:

$$A_0 = \sum_{k_0 \in \Omega_k} \begin{pmatrix} 0 \\ 0 \\ \tilde{\rho}(k_0) \\ \tilde{\rho}(k_0) \end{pmatrix}_{k=k_0} \tag{2.30}$$

2.3.2 Magnetization state and signal calculation

The following paragraphs will deduce the evolution of each basis element when exposed to the simulation experiment, reusing the results from the Smart Bloch Solver approach. It will be shown that each effect can also be described in full exactness using the new magnetization distribution generating set.

2.3.2.1 Gradient and offset frequency effects

To keep track of the magnetization during an MRI sequence, it is necessary to know how an arbitrary element of $\mathbb{B}_k$ behaves when a gradient pulse is applied. These effects will be investigated, assuming the absence of all other effects.

Spatial domain

In order to understand how a gradient pulse alters the magnetization in the spectral domain, it is useful to first formulate its effect in terms of the spatial domain's basis.

The following notation is useful to express the effect of a gradient pulse G:

$$\overrightarrow{\Delta k} = \overrightarrow{\Delta k}(t_-, t_+) := \gamma \int\limits_{t_-}^{t_+} \overrightarrow{G(t)}\, \mathrm{d}t \tag{2.31}$$

As a gradient pulse is applied, the Larmor frequencies of the underlying spin packets follow a slope in the corresponding direction, and the phase gain of a spin packet at position r can be calculated by evaluating $\Delta\varphi(r) = \Delta\varphi(r, t_-, t_+) = \Delta k \cdot r$. This was explained in Section 2.2.2.

Let

$$M_- := \begin{pmatrix} M^{r_0}_{x,-} \\ M^{r_0}_{y,-} \\ M^{r_0}_{z,-} \\ M^{r_0}_{0,-} \end{pmatrix}_{r=r_0} = \left(M^{r_0}_{x,-} + \mathrm{i}M^{r_0}_{y,-}\right) \overrightarrow{e^{r_0}_{\perp}} + M^{r_0}_{z,-}\overrightarrow{e^{r_0}_{z}} + M^{r_0}_{0,-}\overrightarrow{e^{r_0}_{0}} \tag{2.32}$$

with $\overrightarrow{e^{r_0}_{\perp}} = \chi_{r_0}(r)\overrightarrow{e_{\perp}}$. We then apply this phase gain to calculate M_+, the element's state after the gradient pulse:

$$\begin{aligned} M_+ &= \left(M^{r_0}_{x,-} + iM^{r_0}_{y,-}\right) \mathrm{e}^{\mathrm{i}\Delta k \cdot r} \overrightarrow{e^{r_0}_{\perp}} + M^{r_0}_{z,-}\overrightarrow{e^{r_0}_{z}} + M^{r_0}_{0,-}\overrightarrow{e^{r_0}_{0}} \\ &= \begin{pmatrix} Re\left(\left(M^{r_0}_{x,-} + \mathrm{i}M^{r_0}_{y,-}\right) \mathrm{e}^{i\Delta k \cdot r}\right) \\ Im\left(\left(M^{r_0}_{x,-} + \mathrm{i}M^{r_0}_{y,-}\right) \mathrm{e}^{i\Delta k \cdot r}\right) \\ M^{r_0}_{z,-} \\ M^{r_0}_{0,-} \end{pmatrix}_{r=r_0} \end{aligned} \tag{2.33}$$

Frequency domain

The easiest way to investigate the behavior of the elements of $\mathbb{B}_k$ is to express them in their spatial representation, apply the effect of the gradient pulse and translate them back into elements of $\mathbb{B}_k$ afterwards.

For an element

$$\begin{aligned} M_- : &= \begin{pmatrix} M^{k_0}_{v,-} \\ M^{k_0}_{w,-} \\ M^{k_0}_{s,-} \\ M^{k_0}_{t,-} \end{pmatrix}_{k=k_0} \\ &= \left(M^{k_0}_{v,-} \mathrm{e}^{\mathrm{i}(k_0 \cdot r)} + \overline{M^{k_0}_{w,-}} \mathrm{e}^{-\mathrm{i}(k_0 \cdot r)} \right) \overrightarrow{e_\perp} + M^{k_0}_{s,-} \overrightarrow{e^{k_0}_s} + M^{k_0}_{t,-} \overrightarrow{e^{k_0}_t}, \end{aligned} \tag{2.34}$$

applying the spatially dependent phase gain yields

$$\begin{aligned} M_+ &= \left(M^{k_0}_{v,-} \mathrm{e}^{\mathrm{i}(k_0 \cdot r)} + \overline{M^{k_0}_{w,-}} \mathrm{e}^{-\mathrm{i}(k_0 \cdot r)} \right) \mathrm{e}^{i(\Delta k \cdot r)} \overrightarrow{e_\perp} + M^{k_0}_{s,-} \overrightarrow{e^{k_0}_s} + M^{k_0}_{t,-} \overrightarrow{e^{k_0}_t} \\ &= \begin{pmatrix} 0 \\ 0 \\ M^{k_0}_{s,-} \\ M^{k_0}_{t,-} \end{pmatrix}_{k=k_0} + \begin{pmatrix} M^{k_0}_{v,-} \\ 0 \\ 0 \\ 0 \end{pmatrix}_{k=k_0+\Delta k} + \begin{pmatrix} 0 \\ M^{k_0}_{w,-} \\ 0 \\ 0 \end{pmatrix}_{k=k_0-\Delta k} . \end{aligned} \tag{2.35}$$

This result illustrates one power of the k-space formalism approach: The intensities of all components are constant, the gradient pulse effect is a wavenumber shift of the transverse components of the element. This shift is the same for all elements since it does not depend on k_0.

The phase accumulation due to offset frequency can be treated analogously to the gradient directions. The offset frequency dimension of the object's magnetization distribution experiences a virtual gradient that is constant throughout the experiment. A more formal explanation requires an extension of Ω_k to allow for non-trivial offset frequency distributions, but this generalization does not introduce further dynamics and is thus not carried out.

2.3.2.2 Relaxation effects

Relaxation is the second MRI phenomenon that needs to be investigated. Fortunately, the precession and relaxation effects commute, allowing us to apply both effects independently of each other in the absence of other effects. At this stage, an object with homogeneous relaxation times is assumed.

Spatial Domain

In the absence of external magnetic fields, the Bloch equations can be solved easily. For an initial spin packet

$$M_- := \begin{pmatrix} M^{r_0}_{x,-} \\ M^{r_0}_{y,-} \\ M^{r_0}_{z,-} \\ M^{r_0}_{0,-} \end{pmatrix}_{r=r_0} \tag{2.36}$$

the final state is given by:

$$M_+ = \begin{pmatrix} M_{x,-}^{r_0}\,\mathrm{e}^{-\Delta t/\mathrm{T2}} \\ M_{y,-}^{r_0}\,\mathrm{e}^{-\Delta t/\mathrm{T2}} \\ M_{0,-}^{r_0} - \left(M_{0,-}^{r_0} - M_{z,-}^{r_0}\right)\mathrm{e}^{-\Delta t/\mathrm{T1}} \\ M_{0,-}^{r_0} \end{pmatrix}_{r=r_0} \tag{2.37}$$

Frequency Domain
The relaxation effects are independent of position, they therefore translate directly to frequency domain. For an initial configuration

$$M_- := \begin{pmatrix} M_{v,-}^{k_0} \\ M_{w,-}^{k_0} \\ M_{s,-}^{k_0} \\ M_{t,-}^{k_0} \end{pmatrix}_{k=k_0} \tag{2.38}$$

the final state is given by:

$$M_+ = \begin{pmatrix} M_{v,-}^{k_0}\,\mathrm{e}^{-\Delta t/\mathrm{T2}} \\ M_{w,-}^{k_0}\,\mathrm{e}^{-\Delta t/\mathrm{T2}} \\ M_{s,-}^{k_0} - \left(M_{t,-}^{k_0} - M_{s,-}^{k_0}\right)\mathrm{e}^{-\Delta t/\mathrm{T1}} \\ M_{t,-}^{k_0} \end{pmatrix}_{k=k_0} \tag{2.39}$$

2.3.2.3 Pulse effects

As described in Section 2.2.1.3, each pulse effect can be expressed by a matrix multiplication for each spin packet. For the sake of simplicity, pulses are assumed to be non-selective in this section. This restriction will be lifted in Section 2.3.4.1. One detail of the pulse description, which is of particular importance for this simulation method, is that pulse effect operation occurs instantaneously and thus does not imply a time step. This greatly simplifies the calculations.

Spatial Domain
Let $P = (p_{ij})_{i,j}$ describe the corrected pulse effect. Then, for an initial magnetization

$$M_- := \begin{pmatrix} M_{x,-}^{r_0} \\ M_{y,-}^{r_0} \\ M_{z,-}^{r_0} \\ M_{0,-}^{r_0} \end{pmatrix}_{r=r_0} \tag{2.40}$$

the magnetization after the pulse is given by:

$$M_+ = \left(P \begin{pmatrix} M_{x,-}^{r_0} \\ M_{y,-}^{r_0} \\ M_{z,-}^{r_0} \\ M_{0,-}^{r_0} \end{pmatrix} \right)_{r=r_0} \quad , \quad P = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.41}$$

Details and explanations can be found in the previous section (see Section 2.2.1).

Frequency Domain

In frequency domain, there is a similar matrix that also expresses the pulse effect and only mixes components with equal or inverse wavenumber. The calculations in this sections are the main justification for choosing the Frequency domain generating set in a redundant way, because it simplifies the calculations of pulse effects drastically.

The following approach will be used to find a matrix $Q = (q_{ij})_{i,j}$ analog to P:

$$\begin{aligned}
z\overrightarrow{e_v} &\mapsto (q_{11}z)\overrightarrow{e_v} + (q_{21}z)\overrightarrow{e_w} + (q_{31}z)\overrightarrow{e_s} && \text{(I)} \\
z\overrightarrow{e_w} &\mapsto (q_{12}z)\overrightarrow{e_v} + (q_{22}z)\overrightarrow{e_w} + (q_{32}z)\overrightarrow{e_s} && \text{(II)} \\
z\overrightarrow{e_s} &\mapsto (q_{13}z)\overrightarrow{e_v} + (q_{23}z)\overrightarrow{e_w} + (q_{33}z)\overrightarrow{e_s} && \text{(III)} \\
z\overrightarrow{e_t} &\mapsto (q_{14}z)\overrightarrow{e_v} + (q_{24}z)\overrightarrow{e_w} + (q_{34}z)\overrightarrow{e_s} + z\overrightarrow{e_t} && \text{(IV)}
\end{aligned} \tag{2.42}$$

The wavenumbers have been omitted for readability. The shorthand of this section is:

$$\overrightarrow{e_v} := \overrightarrow{e_v^k}, \quad \overrightarrow{e_w} := \overrightarrow{e_w^k}, \quad \overrightarrow{e_s} := \overrightarrow{e_s^k}, \quad \overrightarrow{e_t} := \overrightarrow{e_t^k}. \tag{2.43}$$

The most straightforward way to calculate these factors is to express each left hand side using their spatial representation, apply P, and reorder the terms. It would also be possible to use an alternative spatial representation and linearity arguments to derive this matrix, but the required algebraic effort is similar and calculations are not as tangible.

Investigation of 2.42(I):

Find q_{11}, q_{21}, q_{31}, such that: $z\overrightarrow{e_v} \mapsto (q_{11}z)\overrightarrow{e_v} + (q_{21}z)\overrightarrow{e_w} + (q_{31}z)\overrightarrow{e_s}$

$$\begin{aligned}
&z\overrightarrow{e_v} \\
=&|z|\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)\overrightarrow{e_x} + |z|\sin(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)\overrightarrow{e_y} \\
\mapsto&|z|\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{11}\overrightarrow{e_x}+p_{21}\overrightarrow{e_y}+p_{31}\overrightarrow{e_z}) + |z|\sin(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{12}\overrightarrow{e_x}+p_{22}\overrightarrow{e_y}+p_{32}\overrightarrow{e_z}) \\
=:&\ \mathbf{T}+\mathbf{L}
\end{aligned} \tag{2.44}$$

where **T** contains the transverse components and **L** contains the longitudinal components.

$$\begin{aligned}
\mathbf{T} &= |z|\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{11}\overrightarrow{e_x}+p_{21}\overrightarrow{e_y})+|z|\sin(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{12}\overrightarrow{e_x}+p_{22}\overrightarrow{e_y}) \\
&= |z|\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{11}+\mathrm{i}p_{21})\overrightarrow{e_\perp}+|z|\sin(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{12}+\mathrm{i}p_{22})\overrightarrow{e_\perp} \\
&= |z|\frac{1}{2}(\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}+\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)})(p_{11}+\mathrm{i}p_{21})\overrightarrow{e_\perp}+|z|\frac{1}{2i}(\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}-\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)})(p_{12}+\mathrm{i}p_{22})\overrightarrow{e_\perp} \\
&= \frac{1}{2}|z|\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}(p_{11}+\mathrm{i}p_{21}-\mathrm{i}p_{12}+p_{22})\overrightarrow{e_\perp}+\frac{1}{2}|z|\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}(p_{11}+\mathrm{i}p_{21}+\mathrm{i}p_{12}-p_{22})\overrightarrow{e_\perp} \\
&= \frac{1}{2}z\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r})}(p_{11}+\mathrm{i}p_{21}-\mathrm{i}p_{12}+p_{22})\overrightarrow{e_\perp}+\frac{1}{2}\bar{z}\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r})}(p_{11}+\mathrm{i}p_{21}+\mathrm{i}p_{12}-p_{22})\overrightarrow{e_\perp} \\
&= \frac{1}{2}z(p_{11}+\mathrm{i}p_{21}-\mathrm{i}p_{12}+p_{22})\overrightarrow{e_v}+\frac{1}{2}z(p_{11}-\mathrm{i}p_{21}-\mathrm{i}p_{12}-p_{22})\overrightarrow{e_w}
\end{aligned} \tag{2.45}$$

$$\begin{aligned}
\mathbf{L} &= |z|\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{31}\overrightarrow{e_z})+|z|\sin(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{32}\overrightarrow{e_z}) \\
&= |z|\frac{1}{2}(\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}+\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)})p_{31}\overrightarrow{e_z}+|z|\frac{1}{2i}(\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}-\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)})p_{32}\overrightarrow{e_z} \\
&= |z|(p_{31}-\mathrm{i}p_{32})\frac{1}{2}\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}\overrightarrow{e_z}+|z|(p_{31}+\mathrm{i}p_{32})\frac{1}{2}\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}\overrightarrow{e_z} \\
&= \frac{1}{2}z(p_{31}-\mathrm{i}p_{32})\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r})}\overrightarrow{e_z}+\frac{1}{2}\bar{z}(p_{31}+\mathrm{i}p_{32})\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r})}\overrightarrow{e_z} \\
&= z(p_{31}-\mathrm{i}p_{32})\overrightarrow{e_s}
\end{aligned} \tag{2.46}$$

Thus:

$$\begin{aligned}
q_{11} &= \frac{1}{2}(p_{11}+\mathrm{i}p_{21}-\mathrm{i}p_{12}+p_{22}) \\
q_{12} &= \frac{1}{2}(p_{11}-\mathrm{i}p_{21}-\mathrm{i}p_{12}-p_{22}) \\
q_{13} &= p_{31}-\mathrm{i}p_{32}
\end{aligned} \tag{2.47}$$

Investigation of 2.42(II):

Find q_{12}, q_{22}, q_{32}, such that: $z\overrightarrow{e_w} \mapsto (q_{12}z)\overrightarrow{e_v}+(q_{22}z)\overrightarrow{e_w}+(q_{32}z)\overrightarrow{e_s}$ (mostly analog)

$$\begin{aligned}
&z\overrightarrow{e_w} \\
=&|z|\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)\overrightarrow{e_x}-|z|\sin(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)\overrightarrow{e_y} \\
\mapsto&|z|\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{11}\overrightarrow{e_x}+p_{21}\overrightarrow{e_y}+p_{31}\overrightarrow{e_z})-|z|\sin(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{12}\overrightarrow{e_x}+p_{22}\overrightarrow{e_y}+p_{32}\overrightarrow{e_z}) \\
&=: \mathbf{T}+\mathbf{L}
\end{aligned} \tag{2.48}$$

where **T** and **L** contain the transverse and longitudinal components respectively.

$$\begin{aligned}
\mathbf{T} &= |z|\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{11}\overrightarrow{e_x}+p_{21}\overrightarrow{e_y}) - |z|\sin(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{12}\overrightarrow{e_x}+p_{22}\overrightarrow{e_y}) \\
&= |z|\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{11}+\mathrm{i}p_{21})\overrightarrow{e_\perp} - |z|\sin(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{12}+\mathrm{i}p_{22})\overrightarrow{e_\perp} \\
&= |z|\frac{1}{2}(\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}+\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)})(p_{11}+\mathrm{i}p_{21})\overrightarrow{e_\perp} - |z|\frac{1}{2i}(\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}-\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)})(p_{12}+\mathrm{i}p_{22})\overrightarrow{e_\perp} \\
&= \frac{1}{2}|z|\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}(p_{11}+\mathrm{i}p_{21}+\mathrm{i}p_{12}-p_{22})\overrightarrow{e_\perp} + \frac{1}{2}|z|\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}(p_{11}+\mathrm{i}p_{21}-\mathrm{i}p_{12}+p_{22})\overrightarrow{e_\perp} \\
&= \frac{1}{2}z\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r})}(p_{11}+\mathrm{i}p_{21}+\mathrm{i}p_{12}-p_{22})\overrightarrow{e_\perp} + \frac{1}{2}\bar{z}\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r})}(p_{11}+\mathrm{i}p_{21}-\mathrm{i}p_{12}+p_{22})\overrightarrow{e_\perp} \\
&= \frac{1}{2}z(p_{11}+\mathrm{i}p_{21}+\mathrm{i}p_{12}-p_{22})\overrightarrow{e_v} + \frac{1}{2}z(p_{11}-\mathrm{i}p_{21}+\mathrm{i}p_{12}+p_{22})\overrightarrow{e_w}
\end{aligned} \tag{2.49}$$

$$\begin{aligned}
\mathbf{L} &= |z|\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{31}\overrightarrow{e_z}) - |z|\sin(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{32}\overrightarrow{e_z}) \\
&= |z|\frac{1}{2}(\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}+\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)})p_{31}\overrightarrow{e_z} - |z|\frac{1}{2i}(\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}-\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)})p_{32}\overrightarrow{e_z} \\
&= |z|\frac{1}{2}\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}(p_{31}+\mathrm{i}p_{32})\overrightarrow{e_z} + |z|\frac{1}{2}\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}(p_{31}-\mathrm{i}p_{32})\overrightarrow{e_z} \\
&= \frac{1}{2}z(p_{31}+\mathrm{i}p_{32})\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r})}\overrightarrow{e_z} + \frac{1}{2}\bar{z}(p_{31}-\mathrm{i}p_{32})\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r})}\overrightarrow{e_z} \\
&= z(p_{31}+\mathrm{i}p_{32})\overrightarrow{e_s}
\end{aligned} \tag{2.50}$$

Thus:

$$\begin{aligned}
q_{21} &= \frac{1}{2}(p_{11}+\mathrm{i}p_{21}+\mathrm{i}p_{12}-p_{22}) \\
q_{22} &= \frac{1}{2}(p_{11}-\mathrm{i}p_{21}+\mathrm{i}p_{12}+p_{22}) \\
q_{23} &= p_{31}+\mathrm{i}p_{32}
\end{aligned} \tag{2.51}$$

Investigation of 2.42(III):

Find q_{13}, q_{23}, q_{33}, such that: $z\overrightarrow{e_s} \mapsto (q_{13}z)\overrightarrow{e_v} + (q_{23}z)\overrightarrow{e_w} + (q_{33}z)\overrightarrow{e_s}$

$$\begin{aligned}
&z\overrightarrow{e_s} \\
=&|z|\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)\overrightarrow{e_z} \\
\mapsto&|z|\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{13}\overrightarrow{e_x}+p_{23}\overrightarrow{e_y}+p_{33}\overrightarrow{e_z}) \\
=:&\ \mathbf{T}+\mathbf{L}
\end{aligned} \tag{2.52}$$

where **T** contains the transverse components and **L** contains the longitudinal components.

$$\begin{aligned}
\mathbf{T} &= |z|(p_{13}\overrightarrow{e_x} + p_{23}\overrightarrow{e_y})\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta) \\
&= |z|(p_{13}+\mathrm{i}p_{23})\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)\overrightarrow{e_\perp} \\
&= |z|(p_{13}+\mathrm{i}p_{23})\frac{1}{2}(\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}+\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)})\overrightarrow{e_\perp} \\
&= |z|(p_{13}+\mathrm{i}p_{23})\frac{1}{2}\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}\overrightarrow{e_\perp} + |z|(p_{13}+\mathrm{i}p_{23})\frac{1}{2}\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}\overrightarrow{e_\perp} \\
&= (p_{13}+\mathrm{i}p_{23})z\frac{1}{2}\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r})}\overrightarrow{e_\perp} + (p_{13}+\mathrm{i}p_{23})\bar{z}\frac{1}{2}\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r})}\overrightarrow{e_\perp} \\
&= (p_{13}+\mathrm{i}p_{23})z\frac{1}{2}\overrightarrow{e_v} + (p_{13}-\mathrm{i}p_{23})z\frac{1}{2}\overrightarrow{e_w}
\end{aligned} \tag{2.53}$$

$$\begin{aligned}
\mathbf{L} =& |z|p_{33}\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)\,\overrightarrow{e_z} \\
=& p_{33}z\,\overrightarrow{e_s}
\end{aligned} \tag{2.54}$$

Thus:

$$\begin{aligned}
q_{31} &= \frac{1}{2}(p_{13}+\mathrm{i}p_{23}) \\
q_{32} &= \frac{1}{2}(p_{13}-\mathrm{i}p_{23}) \\
q_{33} &= p_{33}
\end{aligned} \tag{2.55}$$

Investigation of 2.42(IV):

Find q_{14}, q_{24}, q_{34}, such that: $z\,\overrightarrow{e_t} \mapsto (q_{14}z)\,\overrightarrow{e_v} + (q_{24}z)\overrightarrow{e_w} + (q_{34}z)\,\overrightarrow{e_s} + z\,\overrightarrow{e_t}$ (mostly analog)

$$\begin{aligned}
& z\,\overrightarrow{e_t} \\
=& |z|\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)\overrightarrow{e_0} \\
\mapsto& |z|\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{14}\overrightarrow{e_x}+p_{24}\overrightarrow{e_y}+p_{34}\overrightarrow{e_z}+\overrightarrow{e_0}) \\
=:& \mathbf{T}+\mathbf{L}
\end{aligned} \tag{2.56}$$

where **T** contains the transverse components and **L** contains the longitudinal components.

$$\begin{aligned}
\mathbf{T} &= |z|\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{14}\overrightarrow{e_x}+p_{24}\overrightarrow{e_y}) \\
&= |z|(p_{14}+\mathrm{i}p_{24})\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)\overrightarrow{e_\perp} \\
&= |z|(p_{14}+\mathrm{i}p_{24})\frac{1}{2}(\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}+\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)})\overrightarrow{e_\perp} \\
&= |z|(p_{14}+\mathrm{i}p_{24})\frac{1}{2}\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}\overrightarrow{e_\perp}+|z|(p_{14}+\mathrm{i}p_{24})\frac{1}{2}\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)}\overrightarrow{e_\perp} \\
&= (p_{14}+\mathrm{i}p_{24})z\frac{1}{2}\mathrm{e}^{\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r})}\overrightarrow{e_\perp}+(p_{14}+\mathrm{i}p_{24})\bar{z}\frac{1}{2}\mathrm{e}^{-\mathrm{i}(\overrightarrow{k}\cdot\overrightarrow{r})}\overrightarrow{e_\perp} \\
&= (p_{14}+\mathrm{i}p_{24})z\frac{1}{2}\overrightarrow{e_v}+(p_{14}-\mathrm{i}p_{24})z\frac{1}{2}\overrightarrow{e_w}
\end{aligned} \tag{2.57}$$

$$\begin{aligned}
\mathbf{L} =& |z|\cos(\overrightarrow{k}\cdot\overrightarrow{r}+\theta)(p_{34}\overrightarrow{e_z}+\overrightarrow{e_0}) \\
=& p_{34}z\,\overrightarrow{e_s}+z\,\overrightarrow{e_t}
\end{aligned} \tag{2.58}$$

Thus:

$$\begin{aligned}
q_{31} &= \frac{1}{2}(p_{14}+\mathrm{i}p_{24}) \\
q_{32} &= \frac{1}{2}(p_{14}-\mathrm{i}p_{24}) \\
q_{43} &= p_{34} \\
q_{44} &= 1
\end{aligned} \tag{2.59}$$

Assembly

$$\begin{aligned}
Q &= \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & p_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
&= \frac{1}{2}\begin{pmatrix} p_{11}+\mathrm{i}p_{21}-\mathrm{i}p_{12}+p_{22} & p_{11}+\mathrm{i}p_{21}+\mathrm{i}p_{12}-p_{22} & p_{13}+\mathrm{i}p_{23} & p_{14}+\mathrm{i}p_{24} \\ p_{11}-\mathrm{i}p_{21}-\mathrm{i}p_{12}-p_{22} & p_{11}-\mathrm{i}p_{21}+\mathrm{i}p_{12}+p_{22} & p_{13}-\mathrm{i}p_{23} & p_{14}-\mathrm{i}p_{24} \\ 2(p_{31}-\mathrm{i}p_{32}) & 2(p_{31}+\mathrm{i}p_{32}) & 2p_{33} & 2p_{34} \\ 0 & 0 & 0 & 2 \end{pmatrix}
\end{aligned} \tag{2.60}$$

Note that

$$q_{11}=\overline{q_{22}},\quad q_{21}=\overline{q_{12}},\quad q_{13}=\overline{q_{23}},\quad q_{31}=\overline{q_{32}},\quad q_{41}=\overline{q_{42}},\quad q_{14}=\overline{q_{24}}, \tag{2.61}$$

but exploiting this property does not yield much for the applications of this work.

Now, for an initial state

$$M_- := \begin{pmatrix} M^{k_0}_{v,-} \\ M^{k_0}_{w,-} \\ M^{k_0}_{s,-} \\ M^{k_0}_{t,-} \end{pmatrix}_{k=k_0} \tag{2.62}$$

the state after the pulse is given by:

$$M_+ = \left(Q \begin{pmatrix} M^{k_0}_{v,-} \\ M^{k_0}_{w,-} \\ M^{k_0}_{s,-} \\ M^{k_0}_{t,-} \end{pmatrix} \right)_{k=k_0} \tag{2.63}$$

Algorithmically, it is reasonable to treat elements that are the result of an RF pulse effect separately. This is due to algorithmic simplifications that cannot be pursued otherwise, as will be explained later (see Section 2.3.3.1). The RF pulse effect then becomes a split operation that generates three new elements for each previous one.

Example: Rotations
To further illustrate pulse effects within the k-space based simulation method, they are exemplified for some basic rotations. The pulse effects $R_x(\alpha), R_y(\alpha), R_z(\alpha)$ that correspond to a rotation of α around the three axes in spatial domain are:

$$R_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.64}$$

$$R_y(\alpha) = \begin{pmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_z(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Using the relationship of pulse effects in frequency and spatial domain (Equation 2.61), they can be transformed into pulse effects that act on frequency basis vectors:

$$Q(R_x(\alpha)) = \begin{pmatrix} \cos^2\frac{\alpha}{2} & \sin^2\frac{\alpha}{2} & -\frac{1}{2}\mathrm{i}\sin\alpha & 0 \\ \sin^2\frac{\alpha}{2} & \cos^2\frac{\alpha}{2} & \frac{1}{2}\mathrm{i}\sin\alpha & 0 \\ -\mathrm{i}\sin\alpha & +\mathrm{i}\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.65}$$

$$Q(R_y(\alpha)) = \begin{pmatrix} \cos^2\frac{\alpha}{2} & -\sin^2\frac{\alpha}{2} & \frac{1}{2}\sin\alpha & 0 \\ -\sin^2\frac{\alpha}{2} & \cos^2\frac{\alpha}{2} & \frac{1}{2}\sin\alpha & 0 \\ -\sin\alpha & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Q(R_z(\alpha)) = \begin{pmatrix} \mathrm{e}^{\mathrm{i}\alpha} & 0 & 0 & 0 \\ 0 & \mathrm{e}^{-\mathrm{i}\alpha} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

These can be concatenated to yield a rotation around an arbitrary transverse axis, which is particularly relevant to MRI because most idealized pulses are defined in these terms:

$$
\begin{aligned}
Q(R_\varphi(\theta)) &= Q(R_z(\varphi))\, Q(R_x(\theta))\, Q(R_z(-\varphi)) \\
&= \begin{pmatrix} \cos^2\left(\frac{\theta}{2}\right) & \mathrm{e}^{2\mathrm{i}\varphi}\sin^2\left(\frac{\theta}{2}\right) & -\frac{1}{2}\mathrm{e}^{\mathrm{i}\varphi}\mathrm{i}\sin(\theta) & 0 \\ \mathrm{e}^{-2\mathrm{i}\varphi}\sin^2\left(\frac{\theta}{2}\right) & \cos^2\left(\frac{\theta}{2}\right) & \frac{1}{2}\mathrm{e}^{-\mathrm{i}\varphi}\mathrm{i}\sin(\theta) & 0 \\ -\mathrm{e}^{-\mathrm{i}\varphi}\mathrm{i}\sin(\theta) & \mathrm{e}^{\mathrm{i}\varphi}\mathrm{i}\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
\end{aligned}
\tag{2.66}
$$

This is in agreement with the Echo Pathway Segment transition factors of the extended phase graph algorithm [34], except for a different scaling factor of the longitudinal components.

2.3.2.4 Signal Acquisition

Signal acquisition can be simulated quickly with the k-space formalism, in contrast to the Direct Bloch simulation and Smart Bloch simulation approach.

Spatial Domain
In an MRI experiment, signal acquisition corresponds to measuring the total transverse magnetization of the object. Physical coils have a spatial sensitivity that determines how much of the signal at a specific location is acquired by the coil. However, by compensating the density of the static object (see Section 4.1.1.1 for a discussion of non-static objects) accordingly, an experiment using a physical coil can be reformulated as an experiment using an ideal coil. It thus suffices to investigate ideal coils only, which measure transverse magnetization independent of its location.

In other words: the total signal S at any point in time is given by:

$$
S(t) = \sum_{r_0 \in \Omega_r} M_x^{r_0}(t) + \mathrm{i} M_y^{r_0}(t) \tag{2.67}
$$

Even though the process of signal calculation is simple, it is the computationally most costly part of the Smart Bloch simulation as mentioned before (see Section 2.2.2.1) and illustrated later (see Section 3.5.3).

Frequency Domain
The sum of all voxels in Spatial Domain corresponds to evaluating the Fourier transformed transverse magnetization distribution at zero:

$$
S = M_v^0(t) + \overline{M_w^0}(t) \tag{2.68}
$$

Depending on the implementation, the number of elements that have wavenumber zero may be high. If the object has a non-discrete set of components that describe the initial state and elements with equal wavenumbers are not merged, then the number of elements with wavenumber zero grows exponentially with the number of RF pulses. This will be explained in a later part of this section (see Section 2.3.3.1).

2.3.2.5 Summary and terminology

In conclusion, the initial state of the system corresponds to the Fourier transformed proton density. At each RF pulse, the intensities of basis elements of equal or inverse wavenumber mix. If the initial state consists of only one basis element, then it is split into three sub states at each pulse. In this work, every element that is created this way is called Echo Pathway Segment. Tracing only one basis element is sufficient, which will be explained in the next parts of this section (see Section 2.3.3.1).

The offset frequency and gradient pulses that are applied between the pulses shift the transverse segments and do not affect the longitudinal segments. As the Echo Pathway Segments reside as a specific type, they experience either T1 or T2 relaxation. The graph that is generated this way is called the Echo Pathway Tree.

Any transverse Echo Pathway Segment that originated from a $k = 0$ basis element and later assumes wavenumber zero, produces a so called echo at the corresponding time point. If offset frequency induced phase accumulation is considered, assuming wavenumber zero in that direction yields a spin echo. A wavenumber of zero in gradient directions cause a so called gradient echo.

The components of type t are never generated nor destroyed; they stay constant throughout the whole experiment. They restore the equilibrium configuration of the s components as dictated by the T1 relaxation.

The following examples illustrate the k-space algorithm as it is defined thus far.

Three pulse experiment

A three pulse experiment (Figure 2.6) with different temporal RF pulse distances is well suited to illustrate certain simulation effects and limits—which will themselves play a very important role in more involved pulse sequences.

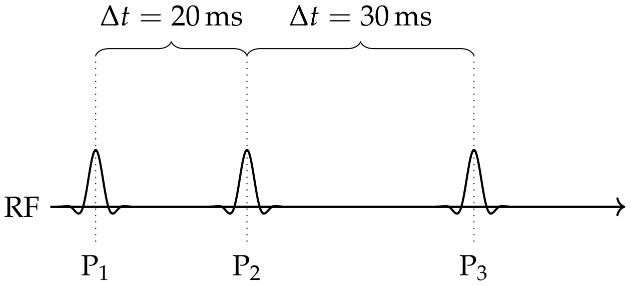


Figure 2.6: Three pulse experiment.

Three pulses that are arranged this way produce a total of six echoes. Echo behavior and Echo Pathway Segment information are often illustrated through extended phase graph diagrams [34]. The following diagram (Figure 2.7) is based on this idea.

The Echo Pathway Tree visualization displays the relationship of the Echo Pathway Segments. The vertical axis corresponds to the wavenumber of the segments, as it would develop in the presence of a constant gradient pulse. Alternatively, this axis can be interpreted as the phase modulation in offset frequency direction. It is also possible to express the phase modulation due to gradient pulses on the y axis. In that case, pathway segments would not be displayed as lines but as curves that increase proportionally to the 0^{th}. A transverse segment that crosses the horizontal axis produces an echo, as explained previously.

Transverse segments are influenced by the change in wavenumber: the wavenumber of v segments increases over time, the wavenumber of w segments decreases. The longitudinal s components are not susceptible to these phase effects and thus stay constant. This diagram assumes one initial Echo Pathway Segment, which is why only one t segment is present. The wavenumber of t segments cannot change, therefore it stays at $k_\omega = 0$.

Along the time axis, segments split each time a pulse is applied. Phase and intensity are not visible in this diagram.

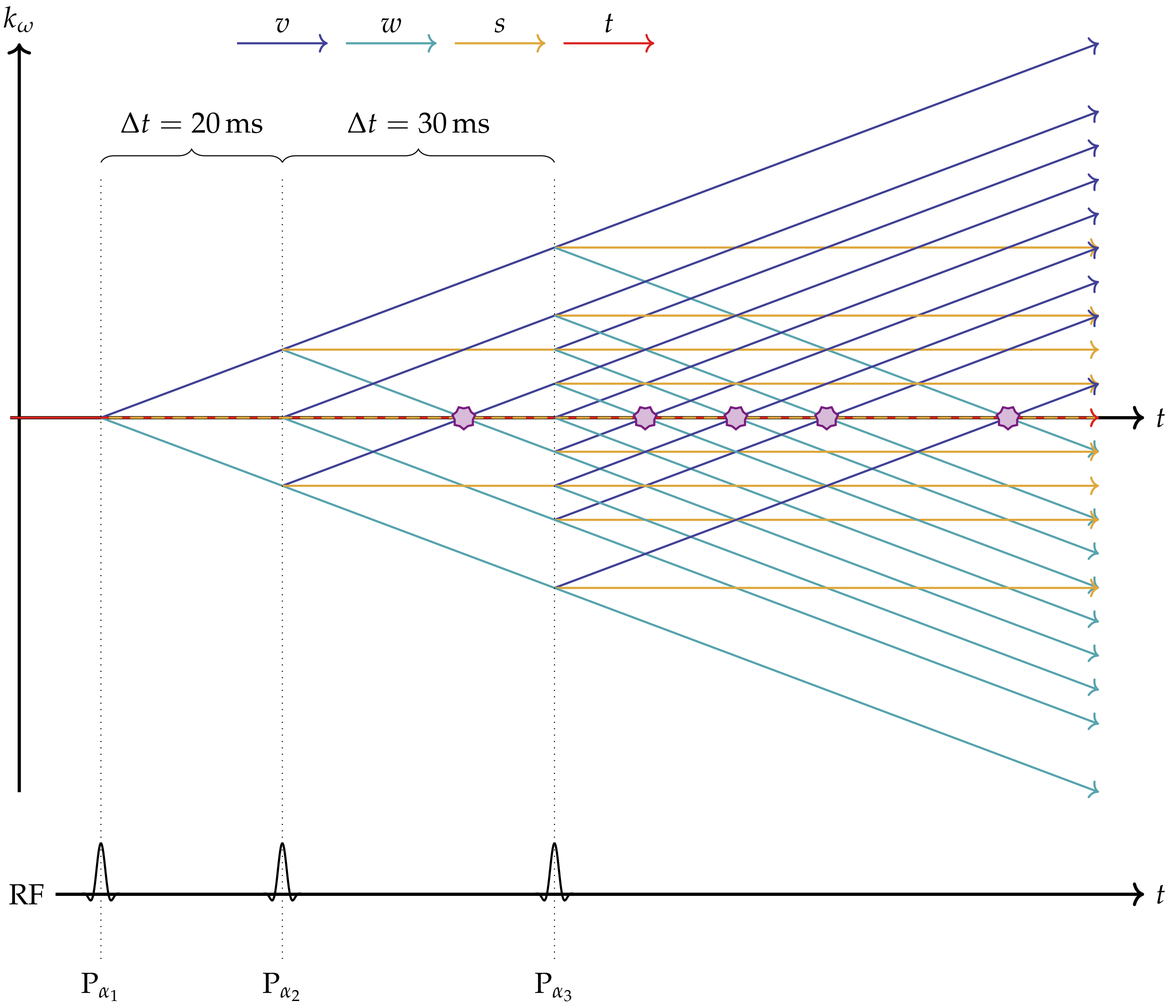


Figure 2.7: Pathway tree visualization of a three-pulse-sequence (see Figure 2.6). The echoes can assume different phases and intensities—depending on the tissue and pulse parameters. The vertical axis denotes the gain in wavenumber of the Echo Pathway Segments, in the presence of a constant background gradient or offset frequency effects. The Echo Pathway Segments split at each pulse and gain wavenumber depending on their type. Echoes are produced when a transverse segment crosses the horizontal axis. The intensity of the echoes or Echo Pathway Segments is not illustrated in this diagram.

The Echo Pathway Segments that are generated possess different intensities depending on the tissue parameters and pulse settings. A pulse with a low flip angle favors the segments that stay of the same type and otherwise mostly mix transverse with longitudinal components. A pulse with a high flip angle is mostly refocusing, meaning that it transforms transverse segments into each other. It does not favor the mix of transverse and longitudinal components.

There is however no defining property of excitation or refocusing pulses, as also mentioned in Section 1.2.2.

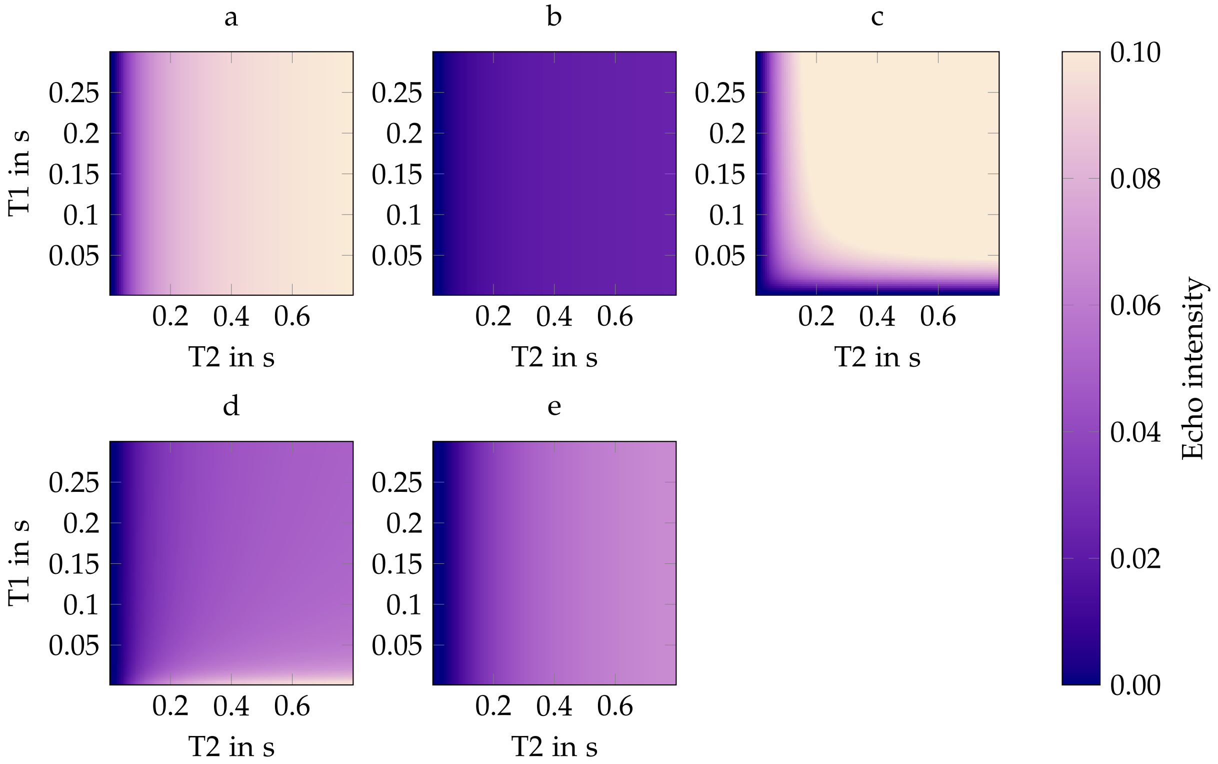


Figure 2.8: Echo intensities of a three-pulse-sequence (Figure 2.6), using 60° pulses, ordered by their time of occurrence, depending on relaxation times (from a to e). The color value represents the echo intensity based on an initial Echo Pathway Segment of magnitude one.

The diagram in Figure 2.8 illustrates the echo intensities depending on the relaxation time of the object. The first (a), second (b), and last (e) echoes are purely T2-dependent because all of their pathway segments are transverse. The pathway of the third (c) echo underwent T2 and T1 decay. Therefore its signal shrinks as T1 or T2 decrease. The pathway of the fourth (d) echo experienced T1 recovery between the first two pulses. For that reason, this echo increases in signal as T1 decreases. Using pulses with different flip angles scales the echo intensities differently, but does not change the T1 and T2 dependence of the echoes.

The following diagram (Figure 2.9) shows the pathway tree of the three pulse experiment when pulses are used that have emphasized excitation or refocusing behavior.

The echo formation is the same, but the intensities of the pathways depend on the RF pulse properties. Some echoes are generated by pathways with low intensity and are thereby suppressed.

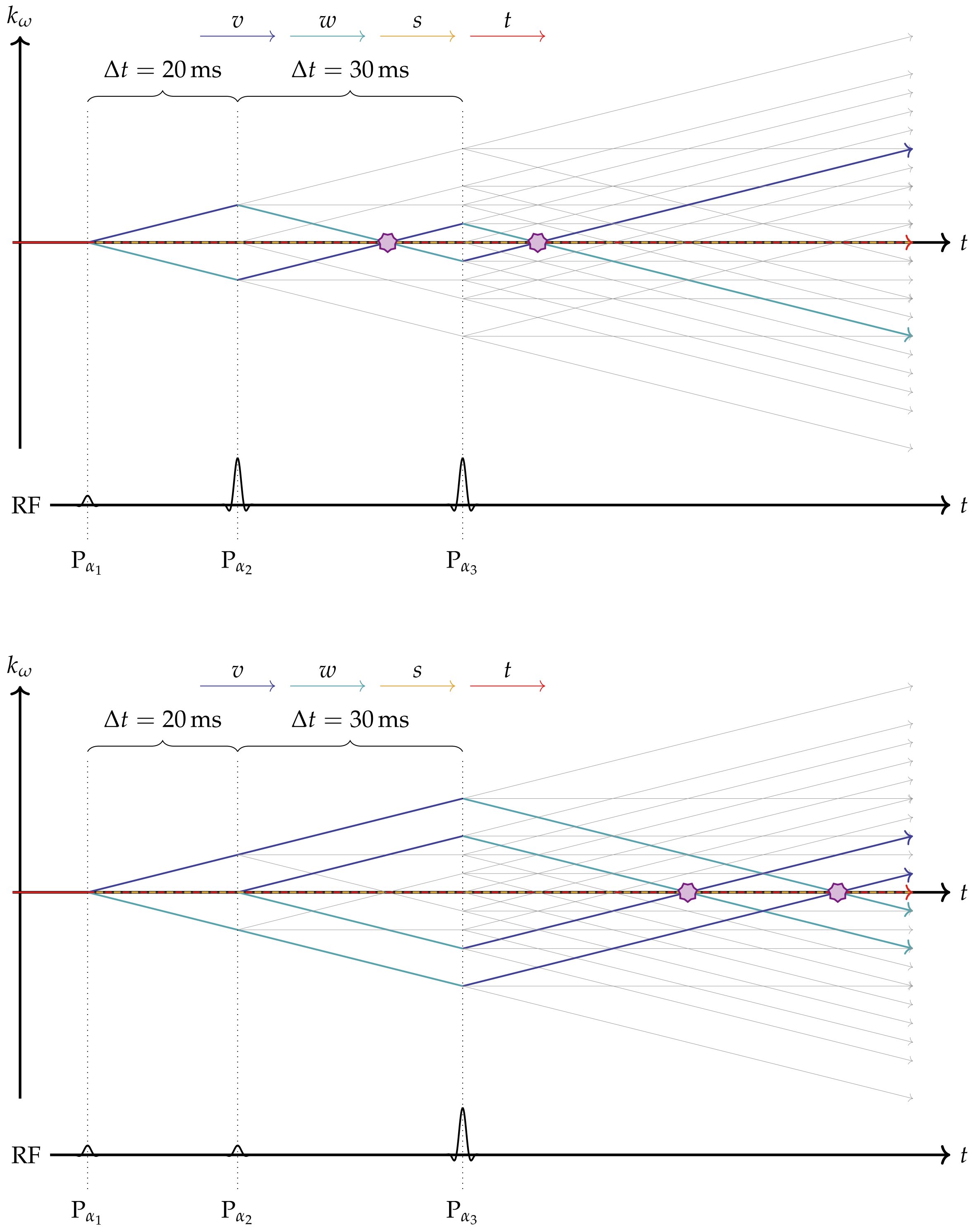


Figure 2.9: Pathway tree visualization of the three pulse sequence (Figure 2.6) with two different sets of pulses. The first variation contains one excitation pulse followed by two refocusing pulses, the second variation contains two excitation pulses followed by one refocusing pulse. Negligible pathways are drawn without color, negligible echoes are not drawn.

Example pathway emphasis

The pathway emphasis of the three pulse example (Figure 2.9) carries over to imaging sequences. The following diagram (Figure 2.10) shows the simplified emphasis for a TSE and an SPGR sequence (see Figures 1.8 and 1.5).

The turbo spin echo sequence refocuses the pathways after a single excitation pulse multiple times, creating spin echoes in the middles between two refocusing pulses. All other Echo Pathway Segments can be neglected and do not contribute to the useful signal.

The gradient echo sequence does not foresee spin echoes. Consequently, the relevant pathways of the pathway tree do not cross the horizontal axis. However, the spatial encoding gradients are applied in a way that the main signal generating pathway segments, which are indicated by thick colored lines, cross the gradient wavenumber axes. But in this illustration, only the wavenumber effects of a constant background gradient are displayed. The thin colored lines correspond to segments that are spoiled and do not contribute to the useful signal. This means that the wavenumbers corresponding to the segments are big, and the tissue geometry can be expected to have low magnitude at those wavenumbers.

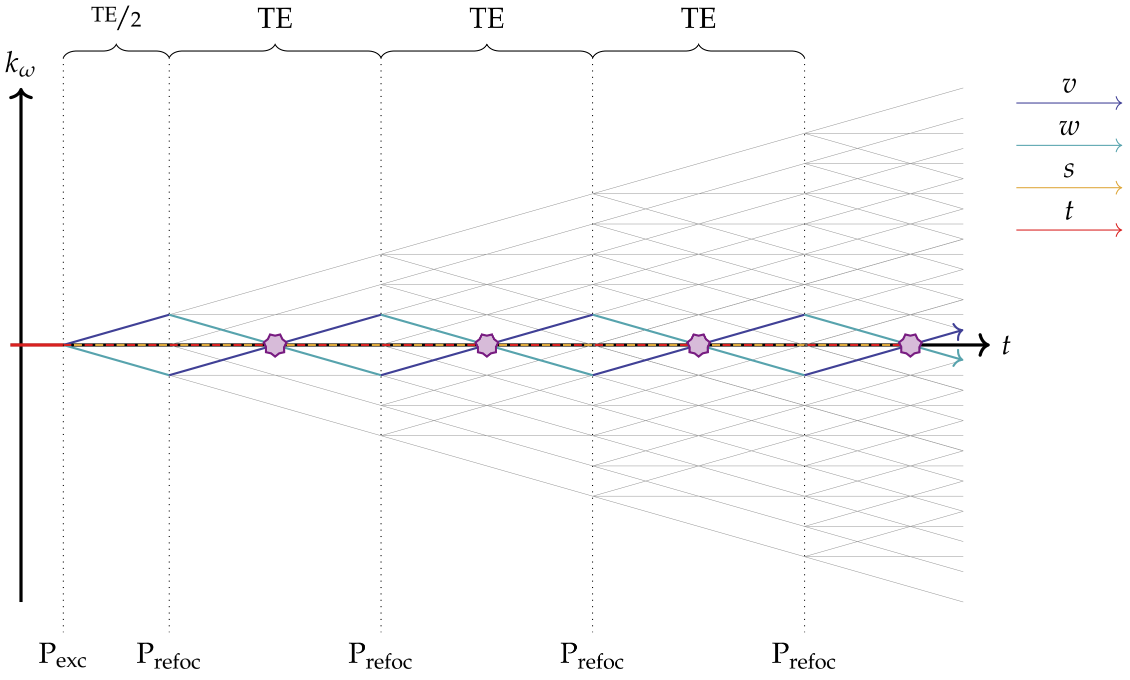
In realistic applications, pulses cannot be perfect excitation or refocusing pulses. Therefore, the pathways that are indicated by gray lines are also present and need to be considered for the calculation of accurate signal.

2.3.3 Simplifications and optimizations

A few properties of the algorithm can be used to reduce the computational effort without reducing the quality of the calculation.

2.3.3.1 Tracing of a single root Echo Pathway Segment

The Echo Pathway Segments at any time point originated from one of the root s elements—it was split at each pulse and modulated its wavenumber between the pulses, thereby following a specific Echo Pathway. If we trace the same pathway starting at a different root element we will always stay parallel to the original pathway, and moreover we will always apply the same linear operators. Therefore it is sufficient to trace one representative root element and apply the corresponding factors to all required parallel elements as needed. If the signal is of interest for the simulation experiment, the only important echo elements that are important are those who pass through $k = 0$ at the moment of signal acquisition. Through the parallelism, the required root pathway intensity contribution that was induced by the phantom shape can easily be found.



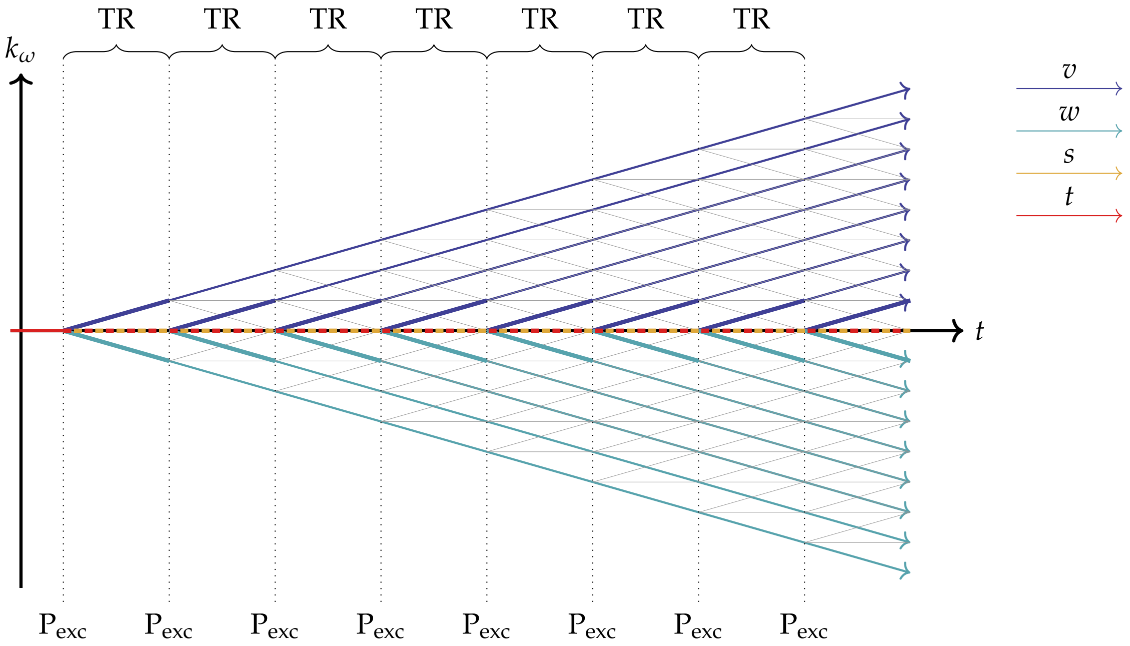


Figure 2.10: Echo emphasis of a TSE and an SPGR sequence.

2.3.3.2 Merging Echo Pathway Segments

Pulse sequences often follow a repeating pulse succession with equal delays between the pulses. Consequently, it is a likely scenario that multiple pathways happen to fall on top of each other, meaning they possess the same wavenumber, are of the same echo type and originate from the same root element. In those cases, Echo Pathway Segments can be merged.

Pathway merging is not necessary and even produces a significant inconvenience: Without merging, each pathway carries a unique history with unique time spans spent as either segment type. The relaxation effects then do not need to be applied in the Echo Pathway Tree generating process, and any combination of relaxation properties can be simulated without regenerating the tree.

The times spent as either pathway type are mostly different and the resulting function of T1 and T2 cannot be expressed with fewer parameters than the separate pathways would need.

Applying the relaxation effects before each pulse is valid since the decay is memoryless, and each pathway segment behaves the same way independent of its previous pathway, thus enabling the merging process. The advantage of this strategy is, that instead of having to trace an exponentially growing number of pathways, the number of pathways might only grow linearly. Because of this decisive advantage, pathway merging is used in this work whenever possible.

2.3.3.3 Discarding Echo Pathway Segment

Not all pathways are equally important. Estimating the importance of each pathway and ignoring certain ones can greatly improve performance. Pathways that do not intermix decay exponentially with at least T1, at best T2. If no merging is pursued, then the pathway intensity will be $e^{t_{\mathrm{T1}}/\mathrm{T1}+t_{\mathrm{T2}}/\mathrm{T2}}$, where t_{Ti} are the times spent as the respective segment relaxation type.

In addition to this exponentially decaying segment amplitude, pathways may carry a high wavenumber—meaning that the corresponding signal that the coil measures originates from a position in k-space which is far away from the center of k-space. k-space falls off rather quickly, e.g. with $O(1/|k|)$ (see Section 1.3.2.1) for a bounded variation object geometry.

Yet, segments with a high wavenumber can be refocused if the sequence is structured that way, and their contribution to the main segments can be of importance.

When pathway merging is performed and the merging relationship is supposed to be applied to different tissue types, then the argument for discarding a pathway needs to be reasonable for the relevant range of tissue types while also considering phantom properties. An importance score that ranks the segments should be a combined weighting of the pathways wavenumber and echo intensities for different relevant tissue types.

2.3.3.4 Templatization

Another effort reduction strategy in pathway merging is that the merging relationship is independent of T1 and T2. If a pathway tree needs to be regenerated with different tissue parameters, then the candidates that are suitable for merging do not need to be determined again but can be reused, which basically saves a considerable number of sort operations of all pathways for the discarding process. A useful side effect of the templatization of pathway trees is that echo intensities can be directly compared, which is otherwise complicated if the merging process is performed in a potentially very different way for each tissue parameter set.

2.3.4 Specifics

The following paragraphs briefly address further details of the k-space approach.

2.3.4.1 Arbitrary pulses

Selective pulses are not easily incorporated into the k-space formalism. They can be realized by separating space into compartments with constant RF pulse influence, e.g. sub-slices for a slice-selective pulse. The alternative to that approach would be to do all pulse calculation in the spatial domain and use a Fourier transform at each RF pulse to perform the pulse effect calculation—which would introduce a high computational burden, has poor error behavior and introduces new systematic inaccuracies.

The spatial dimensions that suffer from this inconsistent pulse effect should be sampled with a density that is applicable to the problem at hand. In case of a slice-selective pulse, the slice profile needs to be resolved with the same accuracy as would be required by the direct Bloch simulation method. Each sub-slice then possesses its own pulse effect that is global for the whole sub-slice.

As soon as the pulses become selective in multiple dimensions, multidimensional sampling is required. In the extreme case of a 4D selective pulse, each voxel has it's own sub-segment and the simulation quality degenerates. In this case, the Smart Bloch solver yields equivalent results.

2.3.4.2 T2′

The T2′ effect is based on the offset frequency distribution of the object. This distribution can be treated as a spatial direction along which a constant gradient pulse is active. This is because the phase evolution in offset frequency is in agreement with the formula for the gradient evolution (Equation 2.9).

If the frequency of a specific tissue follows the commonly assumed Cauchy-Lorenz distribution, then the corresponding T2′ decay contribution can be applied analytically at each signal acquisition time.

In theory, each offset frequency is linked to a different pulse effect—suggesting a separation into sub-classes along the frequency direction. In practice however, this effect plays a very minor role for commonly employed RF pulses and tissues that are of interest to clinical MRI. This is illustrated in the results chapter (Figure 3.4).

2.3.4.3 Inhomogeneous tissue properties

The relaxation effects depend on T1 and T2, which are in turn location-dependent. A location-dependent effect cannot be realized easily in the frequency approach because it generally requires a convolution which makes the approach infeasible.

Since the spin packets do not interact and contribute to the total signal independently, the simulation phantom can be separated into components of different tissue properties that are then treated independently. This is very compatible with phantoms that consist of a low number of different tissues with constant tissue properties each. An experiment that introduces a phantom with voxel-unique tissue properties is numerically inconvenient to handle.

The best candidate for a phantom that undergoes a k-space simulation approach is an analytical one consisting of a low number of sub-phantoms with different tissue properties each. Each sub-phantom should be efficient to evaluate at any point k-space, which may be two or three-dimensional depending on the pulses used.

If the phantom originates from a voxel image, but still has discrete tissue types, then it can still be used but needs to be interpolated because it will likely not be evaluated at the exact corresponding k-space position. If a linear interpolation can be assumed in spatial domain, then the operation that needs to be performed in frequency space is a sinc interpolation, or a convolution with the Kaiser-Bessel function for an approximate solution [20].

The impact of the object description will be discussed for all simulation methods in Section 4.1.4.

2.3.4.4 Relation to the classic k-space formalism and extended phase graph approaches

The main ideas of this simulation method are related to the extended phase graph algorithm. The recent review paper by Weigel et al. [39] provides a broad overview of state-of-the-art phase graph approaches and related methods. The following comparisons are subject to the papers and algorithms mentioned therein.

The first main advantage of the method in this work is the employment of pulse effects that are the result of Bloch simulations of the individual pulses for each isochromat class. Alternative implementations exclusively use approximations in the form of instantaneous rotations. This difference does not change the result drastically when applied to classical EPG problems, but it changes the context in which the algorithm is reasonable to pursue and allows it to answer more general questions. The approach of this section aims at exactness that is algorithmically equal or superior to Bloch equation integration. This unique focus has led to further algorithmical implications that are not present in existing EPG approaches.

Common approaches do not attempt to be applicable to arbitrary pulse sequences due to the exponential growth of states that need to be traced. As a result, strong timing requirements are hardcoded and the development is shifted towards investigating the result for sequences that are especially compatible with the EPG approach. The methods of this work are designed to be generally applicable. This motivated the development of Echo Pathway Segment discarding functionality and more concise segment merging formulations, which are not required for classical EPG problems and not part of common EPG implementations.

Calculating the signal of a shaped object for an arbitrary echo formation is not a standard EPG task. Most of the established codebase is either used exclusively to calculate the echo intensities, or rarely [28] [34] used to apply sampling schemes of very simple sequences with fixed and simplified echo formations [30]. All approaches even restrict the signal calculation to the main segment and ignore the secondary effects. This work combines both sides, and introduces means to perform the calculations in a way that is computationally manageable and produces accurate results—through pathway tree templatization, segment importance measure and the exactness of the segment calculation algorithm.

The pathway tree templatization also emphasizes the decoupling of tissue geometry and tissue-dependent echo features which ultimately spawned the Sequence Response Kernel simulation method that is introduced in the next section. But even the templatization in this method defines a two-step process for signal calculation that is completely missing in existing implementations. This approach is designed to treat heterogeneous tissues and reuse as many calculations as possible if new geometries or even new tissue types are introduced.

In conclusion, this approach is an extension of classical EPG approaches with a focus on accuracy and applicablity to a larger set of problems. The differences start at the fundamental state space definition to permit exact solutions and also introduces a more general workflow to allow for manageable calculations for arbitrary pulse sequences. Established extended phase graph algorithms are seen as a fundamentally different tool that is used to provide answers to a distinct set of problems; the k-space based simulation method of this work is a natural improvement of the Direct Bloch simulation or Smart Bloch simulation.

2.3.5 Full simulation workflow

The full workflow consists of three steps: The pulse precalculation step (Figure 2.3), the template generation (Figure 2.11), and the signal calculation (Figure 2.12). The first step is shared with the Smart Bloch simulation, and is thus omitted here.

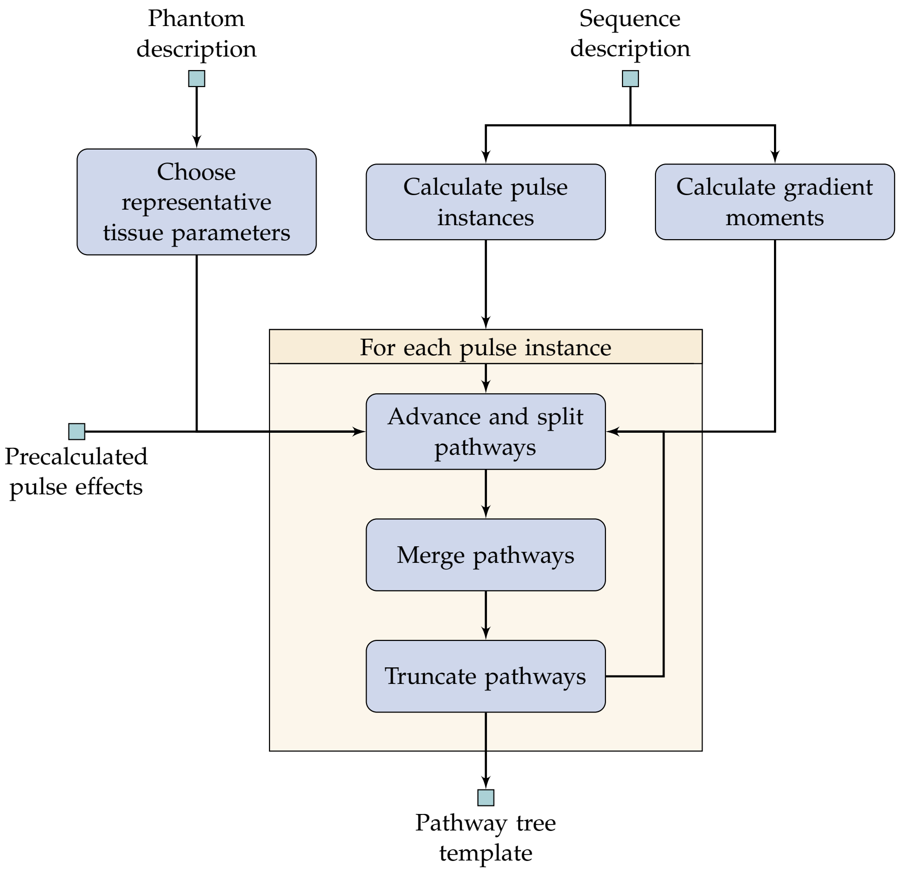

Phantom description
Sequence description
Choose representative tissue parameters
Calculate pulse instances
Calculate gradient moments
For each pulse instance
Advance and split pathways
Precalculated pulse effects
Merge pathways
Truncate pathways
Pathway tree template

Figure 2.11: k-space based simulation workflow—Echo Pathway Tree template generation

The goal of the first step unique to the k-space based simulation (Figure 2.11) is to generate an Echo Pathway Tree template. The state of the system is traced in terms of Echo Pathway Segments. These segments are qualitatively equal for any tissue, but weighted differently according to the tissue parameters. This step prepares the data structures in a way that allows fast echo calculation for any tissue, based on the qualitative template.

One or more representative tissue parameters need to be chosen for the template generation process (see Section 2.3.3.4). These should be close to reasonable tissue parameters because segments that are irrelevant and thus discarded in the template will also be ignored for the signal calculation in the next step.

The pathway tree is initialized with one segment that is then advanced, split, merged, and truncated at each RF pulse instance. The advancing and splitting is done according to Section 2.3.2.5 and relies on the gradient moments between the RF pulses and the precalculated pulse effects. Merging and splitting needs to be performed as explained in Section 2.3.3.

The template holds the information about merging and trunctation. These two processes are computationally exhaustive and do not need to be redone for each tissue. All that remains for calculating the correct Echo Pathway Segment magnitudes is to follow the template with different tissue parameters.

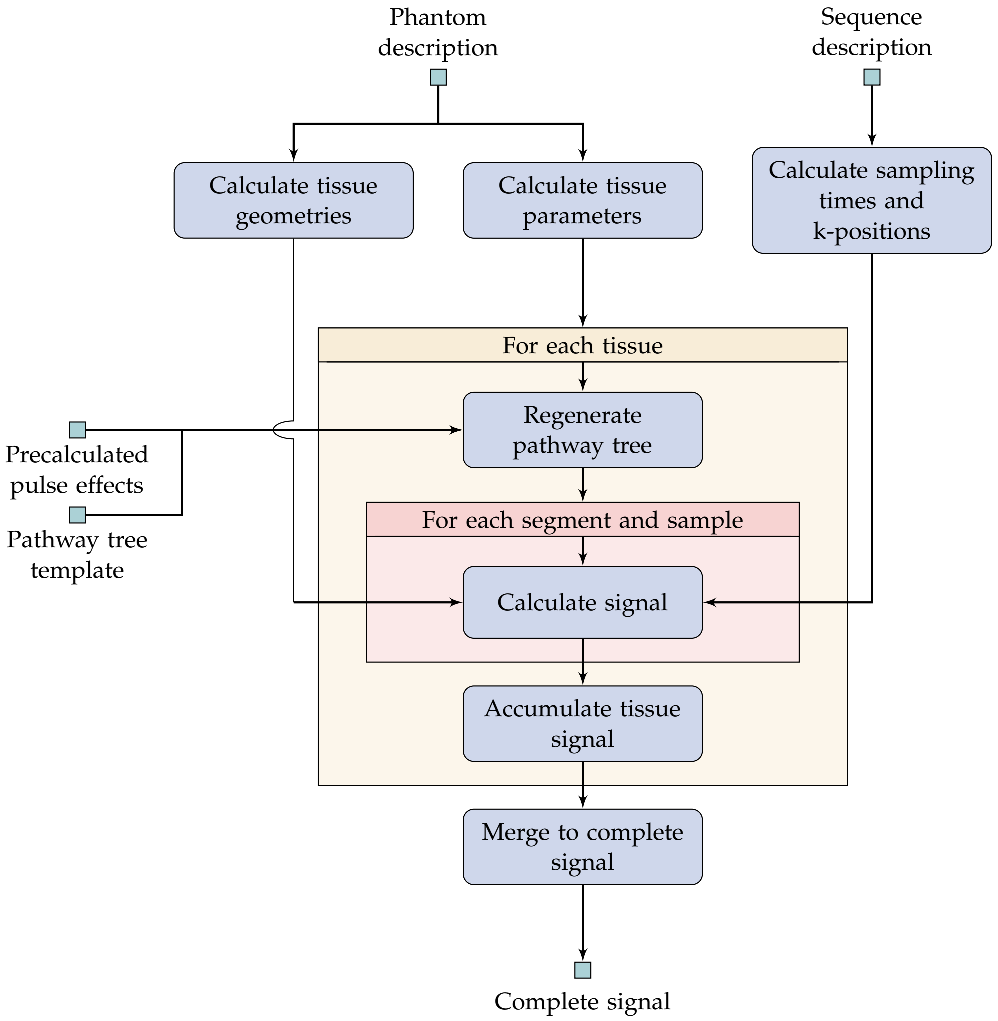


Figure 2.12: k-space based simulation workflow—Signal calculation

The actual tissue parameters and tissue geometries are not introduced to the simulation until the workflow step (Figure 2.12). The pathway tree of each individual tissue class is required. This can be calculated by following the template.

At each acquisition, the contribution of each relevant Echo Pathway Segment is added to the signal. This requires an evaluation of the tissue's Fourier transformed spin density at a position dictated by the Echo Pathway Segment properties.

At the end of the process, the signals of the individual tissue classes are accumulated to produce the total signal.

2.4 Sequence Response Kernel simulation

The approach presented in this section further processes intermediate results from the k-space based simulation. Under assumptions mainly aimed at the reconstruction process, the simulation can be separated into geometry dependent and tissue type dependent contributions, thereby being pushed into image space.

Investigating the simulation in image space has several theoretical and practical advantages. Firstly, image space is more intuitive and closer related to the applications of MRI than k-space or signal itself. It is more relevant to identify and quantify artifacts in image space, since that is where artifacts are most relevant. Features such as blurring or ringing are easily visible in image space, but are rather concealed and interwoven effects in signal space.

Secondly, since the effects of geometry and signal are separated until after they have reached image space, they can be exploited by image processing algorithms. This is further elaborated in the next section (see Section 2.5).

Notably, no simplifications are assumed, thus full numerical accuracy can still be maintained with this approach—even though it is reasonable to sacrifice some of the accuracy for speed, which is directly controllable in this approach.

2.4.1 Concept and theory

The Echo Pathway Tree provides a different perspective on MRI signal. By introducing the reconstruction process it becomes apparent that the image intensity of a physical voxel can be expressed as a convolution of several sparse point spread functions.

2.4.1.1 Prerequisites

The first helpful property of MR imaging is linearity. The signals of two sub-objects do not directly influence each other. Consequently, simulating each object individually and accumulating all signals at the end is a valid approach.

If the reconstruction process exclusively consists of linear operators, then these objects can also be handled individually in image space—which is a desirable property of an imaging technique because the presence of one object should not influence any other object.

The most common operators used for image reconstruction are linear, e.g. Fourier transform, sampling, windowing. Counterexamples are baseline fitting, field of view restriction or calculating the absolute value of a complex-valued image. The latter is commonly used, but fortunately mostly at the end of the reconstruction pipeline and not computationally exhaustive—therefore using this as a post-processing step after the Sequence Response Kernel techniques is unproblematic.

The following parts of this section will assume that the reconstruction process consists of merely a Fourier transform. Further linear operators can be introduced into this concept, but as soon as the multiplication in signal space does not have an easily evaluable counterpart in the corresponding intermediate reconstruction space, the separation of the various effects that influence signal intensity no longer holds, degenerating the Sequence Response Kernel approach and making it computationally infeasible.

2.4.1.2 Image function formulation

The signal of the object when exposed to the MRI sequence can be separated into a sum of the signals of each voxel.

$$S(t) = \sum_{r_0 \in \Omega_r} S_{r_0}(t) \tag{2.69}$$

The total signal of each voxel can itself be separated into the signal that is introduced from the individual Echo Pathway Segments.

$$S_{r_0}(t) = \sum_{p \in \text{Segments}} c_p(t) \tilde{\rho}_{r_0}\big(\Delta k_p(t)\big) \tag{2.70}$$

Where $c_p(t)$ is the intensity and phase of the segment, $\Delta k_p(t)$ is the position in k-space of the segment and $\tilde{\rho}_{r_0}$ is the Fourier transform of the voxel's geometry, located at r_0.

When introducing the 2D discrete Fourier transform, each point of interest in k-space used for the reconstruction is acquired at one point in time, thus a change of variables $t = t(k_x, k_y)$ is possible. The image intensity equation then reads:

$$I(x,y) = \mathcal{F}_{2D}\left(\sum_{p \in \text{Segment Groups}} c_p(k_x,k_y) \tilde{\rho}_{r_0}\big(\Delta k_p(k_x,k_y)\big) \right)(x,y) \tag{2.71}$$

And by using common properties of the Fourier transform:

$$I(x,y) = \sum_{p \in \text{Segment Groups}} \mathcal{F}_{2D}\Big(c_p(k_x,k_y)\Big) * \mathcal{F}_{2D}\Big(\tilde{\rho}_{r_0}(\Delta k_p(k_x,k_y))\Big)(x,y) \tag{2.72}$$

Furthermore, in case of rectilinear grid voxel correspondence, the basis functions of the physical voxels are the equal except for discrete shifts—which can be exploited as well. The Fourier transform of a shifted voxel $\tilde{\rho}_{r_0}(k)$ in terms of an unshifted voxel with the same geometry $\tilde{\rho}_0(k)$ is: $\tilde{\rho}_{r_0}(k) = \tilde{\rho}_0(k) \cdot \mathrm{e}^{-ir_0 \cdot k}$. And inserting this relation into the image equation yields:

$$T := \mathcal{F}_{2D}\Big(c_p(k_x,k_y)\Big) \tag{2.73}$$

$$G := \mathcal{F}_{2D}\Big(\tilde{\rho}_{r_0}(\Delta k_p(k_x,k_y))\Big) \tag{2.74}$$

$$S := \mathcal{F}_{2D}\Big(\mathrm{e}^{-\mathrm{i}r_0 \cdot \Delta k_p(k_x,k_y)}\Big) \tag{2.75}$$

$$I(x,y) = \sum_{p \in \text{Segment Groups}} \Big(T * G * S\Big)(x,y) \tag{2.76}$$

Even though convolution is usually a computationally exhaustive process if done directly, the nature of well-behaved MRI sequences produces sparse kernels, making direct convolution a reasonable approach:

T - Tissue kernel

The first kernel accounts for the Echo Pathway Segment magnitude and phase. From a sequence development point of view, it is most desirable to have constant Echo Pathway Segment intensities because non-constant segment intensities amplify or attenuate certain parts of k-space thereby creating artifacts.

When investigating sequences that cause the magnetization to reach a steady state, it might be possible that this steady state is not reached while acquiring the first lines, in case of rectilinear sampling. This will emphasize certain frequencies in phase encoding direction and is also an undesired artifact in phase encoding direction.

If the time to acquire one line is high in comparison with T2, the signal acquired at the end of the line will be less than the signal acquired at the beginning of the line—this will cause blurring in frequency encoding direction.

Both effects cannot be easily compensated during the reconstruction process because they would require a deconvolution based on known tissue parameters. Therefore, most sequences are designed to yield homogeneous signal and thus sparse tissue kernels.

G - Geometry kernel

The second kernel accounts for the voxel geometry. If chosen correctly, one voxel in image space corresponds to one voxel in physical domain.

Because of the nature of sampling a continuous function and applying the Fourier transform, slight artifacts might be introduced, mainly because the object is not replicated beyond the field of view (see Section 1.3.2.1). But beyond that, the geometry kernel should have only one nonzero entry.

Alterations from the correct sampling required for the Fourier transform will also cause effects in this kernel.

S - Shift kernel

The third kernel accounts for the shift of a voxel. Any sequence imperfections related to rectilinear resolution will be reflected in this kernel. Those are not present in properly developed sequences, which will yield trivial shift kernels.

2.4.1.3 Kernel Illustration

The following figures provide basic examples for kernels and the image assembly process.

The images in Figure 2.13 illustrate basic properties of exemplary tissue and geometry kernels. The tissue kernel experiences a blurring in horizontal direction due to the T2 decay that occurs during a readout. The vertical direction is blurred because the steady state is reached slowly over time.

The geometry kernel does not distinguish between the directions because the resolutions and discretization artifacts are equal in both directions.

The intensity distribution plot shows that the kernels can be heavily truncated. A truncation limit of ten suffices to include all entries with a value of at least 1% of the maximum kernel value.

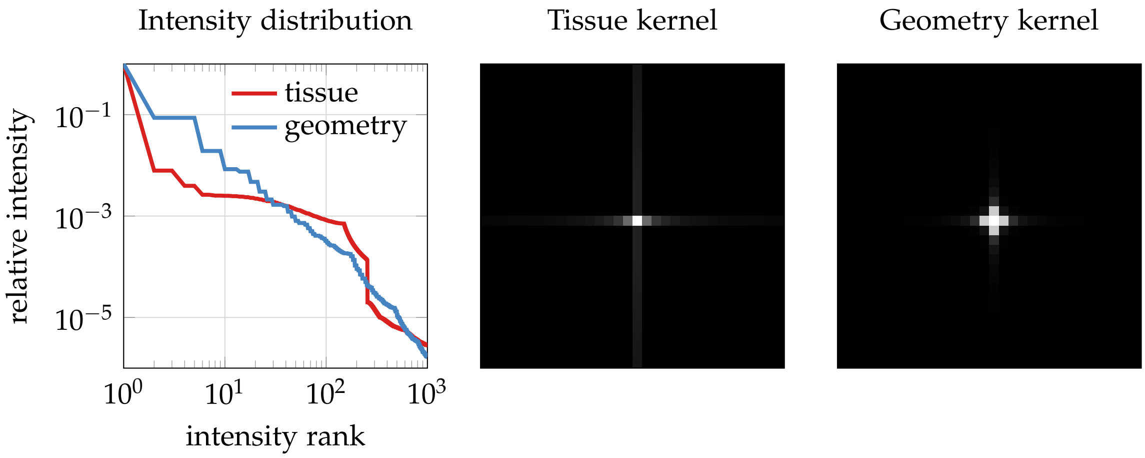


Figure 2.13: Example tissue and geometry kernels and their intensity distribution. The corresponding sequence is defined in Section 3.4.1. The tissue parameters are T1 = 388 ms, T2 = 155 ms. The intensity distribution diagram shows the ordered magnitudes of the 1000 biggest kernel image values.

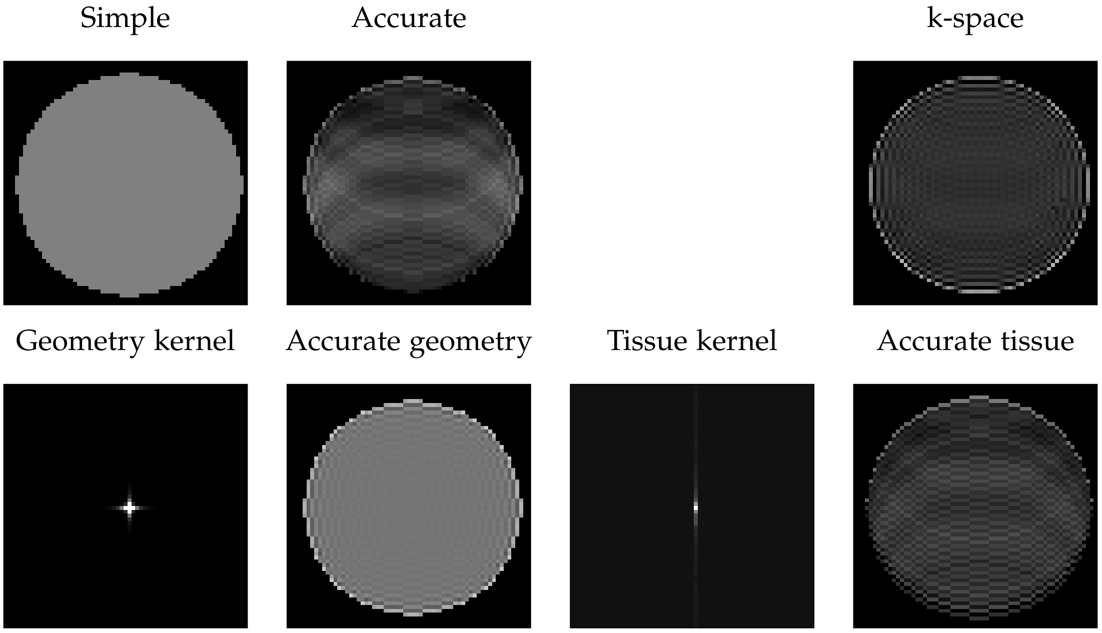


Figure 2.14: Illustration of kernels and assembled images for a SPGR sequence (Figure 1.5). The relevant sequence parameters are: TR = 50 ms, TE = 8 ms, flip angle = 10°, resolution = 64 × 64. The simulated relaxation times of the phantom are: T1 = T2 = 1.3 s.

The illustrations in Figure 2.14 depict the basic process of image simulation. Truncating both kernels to size one yields a rough picture (Simple). Not truncating any kernels yields an image that shows the imperfections of the sequence (Accurate). The k-space simulation result (k-space) cannot be expected to produce the same results because of the discretization assumptions that are made in the Sequence Response Kernel approach.

Truncating the tissue kernel heavily but using the full geometry kernel produces an image that illustrates the errors that are introduced by incorrectly expecting the object to repeat itself periodically for each field of view step (Accurate geometry).

Truncating the geometry kernel heavily but keeping full tissue kernels produces an image that isolatedly displays the influence of signal behavior imperfections (Accurate tissue).

2.4.2 Algorithmic details

Some details of the approach will be elaborated in the following paragraphs.

2.4.2.1 Echo Pathway Segment grouping

The algorithm requires a correspondence of nominal k-position and time. Since k-space was likely acquired using multiple RF pulses, the Echo Pathway Segments have to be grouped together by identifying segments that behave similarly for the whole sequence.

In order to use the Sequence Response Kernel approach efficiently, it is necessary to have tissue kernels that originate from segments with actually similar or at least comparable amplitudes. Thus it is very counterproductive if significantly different segments are incorporated into one tissue kernel.

Furthermore, the sparsity of the geometry kernel breaks down if the object's Fourier transform is not sampled adequately, e.g. due to a collection of Echo Pathway Segments that are not all in agreement with the sampling intended and assumed by the reconstruction process. The effects on the shift kernel are similar to those on the geometry kernel.

The grouping is usually not complicated but needs to be adjusted to suit the sequence. For example, for a sequence with rectilinear sampling and lines acquired in a repeating fashion, segments can be grouped according to their wavenumber before the phase encoding step. For radial sampling with radial spoiling, this approach might not be viable and grouping via pathway amplitudes should be considered instead.

Theoretically, if all calculations are performed with perfect accuracy, the grouping has no influence on the final image because of the linearity of the Fourier transform—therefore any grouping can be justified. However, as stated before, some groups might perform poorly in practice.

2.4.2.2 Tissue parameter influence

The tissue kernel is the only kernel that depends on the Echo Pathway Segment intensity. The parameters required to fully define the tissue in the Response Kernel approach are the same as the segment's intensities, and the dynamics are similarly complex. However it is conceptionally easier to interpolate convolution kernels in image space than interpolating Echo Pathway trees and therefore easier to handle for image calculation.

2.4.2.3 Shift basis

In order to fully describe the physical object, the algorithm requires shifts to any position in the image. However shifts can be concatenated; therefore it is more practical to work with a reasonable basis of shifts that can be combined to produce an arbitrary shift in few steps. A reasonable approach is to use a binary basis in the two image directions, reducing the effort from $x \cdot y$ to $\log_2(x \cdot y)$ required kernels.

2.4.2.4 Non-ideal coils

The approach can easily be extended to allow for coils with spatial sensitivity. The position of the physical voxel induces a weighting that needs to be respected before adding the basis element contribution to the coil's image. Imperfect transmissivity can be treated as a tissue parameter.

Further system imperfections and dynamic object properties will be discussed in Section 4.1.1.1.

2.4.3 Full simulation workflow

The full simulation workflow of the Sequence Response Kernel approach requires six steps. The first two have already been introduced in the previous sections, namely the pulse precalculation and Echo Pathway Tree template generation. The next three steps do not have a specific order and are responsible for the calculation of the tissue (Figure 2.15), geometry and shift kernels (Figure 2.16) respectively. The last step (Figure 2.17) assembles these kernels to produce the final image.

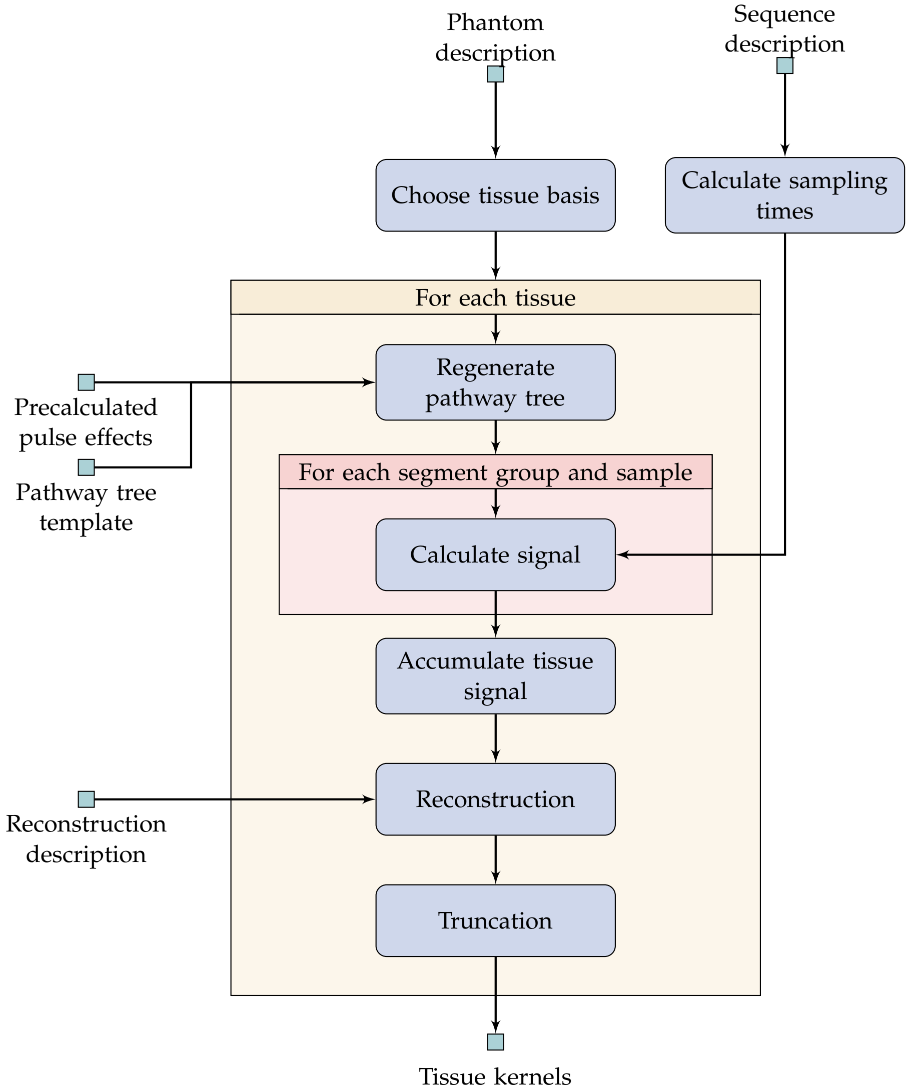


Figure 2.15: Sequence Response Kernel simulation workflow—Tissue kernel calculation

The tissue kernel generation step (Figure 2.15) resembles the signal calculation step of the k-space simulation workflow except for a few major differences:

1. The phantom that is used for this experiment is not yet known. In place of the known tisssue classes, a tissue basis needs to be chosen.
2. The signal of the Echo Pathway Segments is calculated similar to the k-space workflow step, but missing the tissue geometry influence.

3. The signal of each individual tissue is sampled according to the reconstruction description. This needs to be done per segment group.
4. Each complete signal set is reconstructed to form the raw tissue kernels. These tissue kernels may be truncated to increase performance at the cost of accuracy.

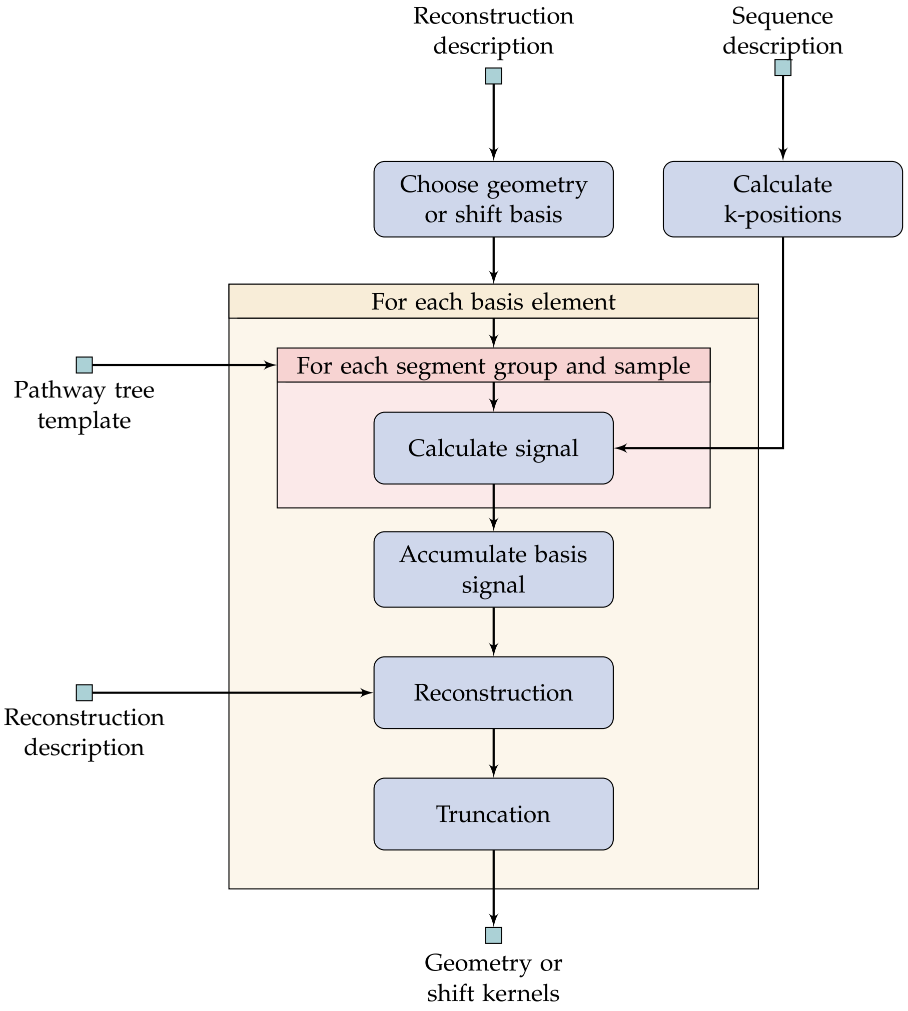


Figure 2.16: Sequence Response Kernel simulation workflow—Geometry and Shift kernel calculation

The structure of the geometry and shift kernel calculation step (Figure 2.16) is similar to the previous one. But instead of tissue parameters, the signal part calculation is performed for each geometry and tissue basis element. This uses the template and the k-offsets of each acquisition time, which were not directly needed for the tissue kernel.

Each geometry or shift contribution of the segment is sampled per segment group, according to the reconstruction description. The reconstruction then produces the raw geometry or kernels, which can then be truncated and stored.

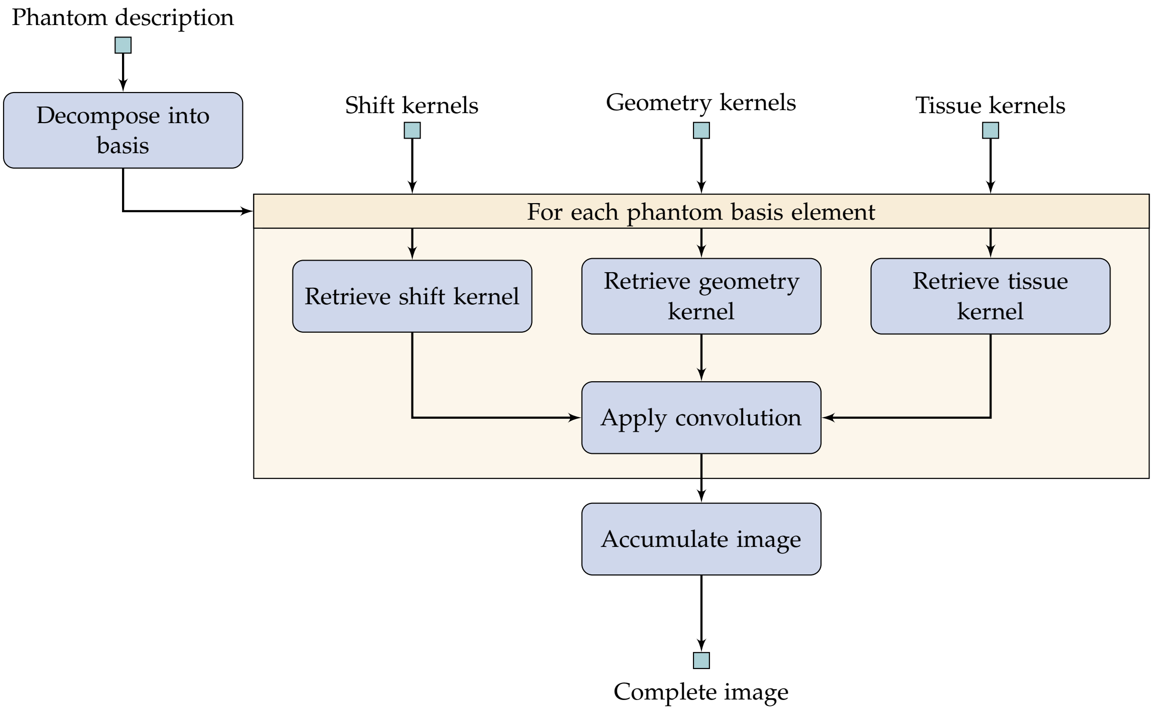


Figure 2.17: Sequence Response Kernel simulation workflow—Image calculation

The final step of the Sequence Response Kernel full simulation workflow is the image assembly (Figure 2.17). For this step, the phantom needs to be decomposed with respect to the chosen geometry and shift basis.

Each element of the decomposed phantom defines its own tissue, geometry and shift kernels that need to be convolved to yield its image contribution. Those kernels were generated in the previous steps and need to be retrieved.

These image contributions can then be accumulated to result in the complete simulated image.

2.5 Derived Algorithms

The Sequence Response Kernel approach enables a different perspective on the way simulation can be used in research, education and clinical applications.

The results and direct discussion of the methods that are presented here will be exhibited in the following chapters (see Section 3.6, 4.1.1.5).

This section showcases the potential of the approach with a focus on the algorithm itself, rather than the results that it provides in this exemplary instance.

2.5.1 Sequence feature extraction

The Sequence Response Kernels contain a lot of information about the sequence. However, the important information of each kernel is hidden inside an image with the full resolution of the MRI sequence. In order to appropriately investigate these kernels, the properties of each kernel need to be expressed in a more intuitive way, which greatly benefits from knowledge about the usual properties of these kernels.

Each sequence feature extraction process is essentially a mapping from the high-dimensional space of all tissue and sequence parameters to a one-dimensional space that has tangible meaning in image space.

The reduction of this mapping to a lower-dimensional domain, for instance containing relaxation time, echo time, or flip angle can then capture the sequence performance with respect to that domain in a way that is very approachable.

2.5.1.1 Formal definition

Let $\mathcal{S}$ be the space of sequence and reconstruction parameters, let $\mathcal{D}(s), s \in \mathcal{S}$ be the sequence and reconstruction information (see Section 1.1.2.1), and let $\mathcal{T}, \mathcal{G}, \mathcal{S}$ be the tissue basis (set of tissue parameters), geometry basis (set of object geometries, i.e. the set containing only one voxel geometry) and shift basis (set of shifts, i.e. powers of two of voxel extents in each direction) respectively.

A function

$$\begin{aligned} f &: \mathcal{S} \times \mathcal{B} \to F \\ \mathcal{B} &\in \{\mathcal{T}, \mathcal{G}, \mathcal{S}\} \\ F &\in \{\mathbb{C}, \mathbb{R}\} \end{aligned} \tag{2.77}$$

is called feature extractor. In other words, a feature extractor maps a combination of sequence and tissue/geometry/shift parameters to a one-dimensional property. Examples of such properties are presented in the following paragraphs.

2.5.1.2 Tissue kernel features

The tissue kernel is the most important, most computationally exhaustive and most interesting Sequence Response Kernel.

Contrast information / signal

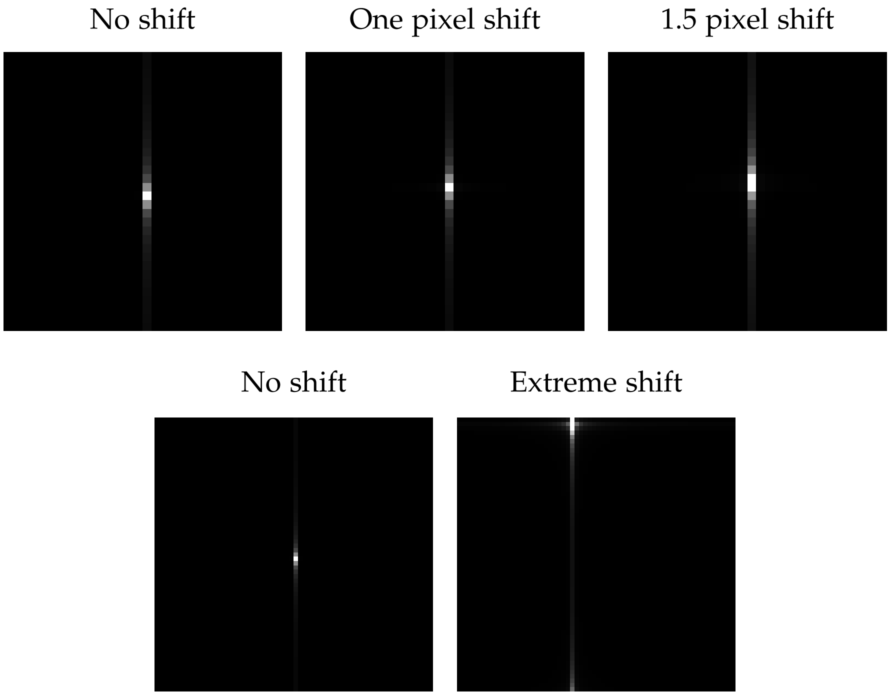
The most relevant part about the tissue kernel is the main signal response of a certain parameter combination. The contrast information is the image intensity or brightness that the sequence translates the physical parameters to. It can be extracted by taking the value of the tissue kernel entries.

When the tissue appears at the position of the image that corresponds to the actual physical position, then this value will be that element of the kernel that does not imply a shift. This is—depending on the implementation of the Fourier transform—either the first or the center sample.

Shift

For tissue parameter sets containing offset frequency, a shift (chemical shift) will occur in frequency encoding direction. The best way to extract this shift is to compute the intensity-weighted average position of all kernel entries.

The shift parameter is prone to Nyquist errors because only integer-valued voxel shifts can be sharply identified in the tissue kernel. Remedies for this problem will be discussed later (see Section 4.1.1.4).



No shift

One pixel shift

1.5 pixel shift

No shift

Extreme shift

Figure 2.18: Shift kernels of an EPI sequence. The first three images are zoomed to illustrate the effects more clearly. The first tissue is on resonance. The second one possesses an offset frequency that corresponds to one pixel—the kernel is practically the same, shifted by one pixel. The third tissue has an offset frequency that induces a 1.5-pixel shift—the image experiences high blurring due to the discretization error. The first image of the second row is the non-zoomed version of the first image, and the last tissue possesses a very high offset frequency—the induced shift is high enough to also affect the phase encoding direction.

Blurring

The T2-decay always induces a signal attenuation at the end of an acquisition period. This corresponds to a Lorenz kernel in image space, which is wide-spread if the decay is prominent, or sharp if the T2-decay is mostly negligible.

In case of single echo acquisition, this Lorenz-curve will be oriented in frequency encoding direction; in case of echo train acquisition, such as EPI, blurring occurs in two directions and for 3D acquisitions, all three spatial dimensions experience blurring.

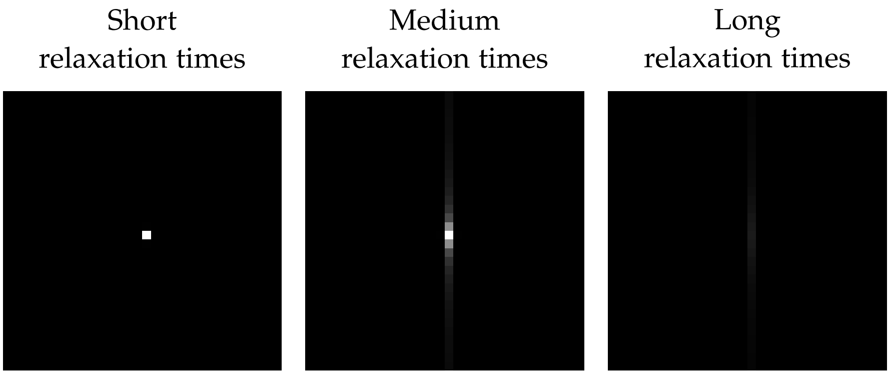


Figure 2.19: The blurring of different tissues for an EPI sequence. High relaxation times induce a negligible blurring. Medium relaxation times induce some blurring and also decrease the highest signal magnitude. Low relaxation times further amplify the effect.

Entropy

The signal development does not usually follow a specific pattern in non-frequency direction. There is no vastly applicable model to describe this process and likewise, this effect can have numerous manifestations in image space.

Calculating the entropy of a kernel provides a measure for the amount of signal that a voxel produces at clearly visible positions, allowing further investigations of artifacts that are not directly related to decay.

Luckily, the effect of reaching a steady state for a sequence of RF pulses is independent of the decay that occurs during the acquisition of one line. Therefore, the dimensions of a 2D image can be investigated independently.

Undesired signal

The relation of contrast information magnitude to the cumulated magnitudes of all other kernel entries is a measure for the undesired signal. This measure is a particularly valuable result of the Sequence Response Kernel approach, because it can not be approached easily with a model approach.

Most prominent values

Reducing a kernel to a fixed number of most important components and omitting the rest is the standard and most versatile approach to deal with kernels, and allows for finely tunable accuracy of image computation. It does not directly yield easily understandable parameters that allow for classification of the sequence.

Image gradient

If the tissue basis parameter ranges are chosen according to a regular grid, then the image gradient operator applied after the contrast information function is defined and yields a derived extractor. It can be understood as a local distinguishability map which is particularly relevant when postprocessing algorithms need to be considered for the given sequence.

2.5.1.3 Geometry and shift kernel features

The geometry and shift kernels are mostly related to sampling of the different Echo Pathways. They can usually be replaced by ideal kernels because they hold few information about the sequence. But there can be exceptions:

Incomplete Pathway groups

Some echoes are not present for all acquired lines. Such incomplete groups will induce artifacts that might need further investigation. They can be extracted by quantifying the ratio of intensity values of the strongest kernel value and all the other values combined. This incomplete Pathway group problem is usually a minor issue since they are commonly linked to minor echoes with low intensities.

Intensity loss

If the resolution used for the geometry kernel is not completely compatible with sequence or reconstruction, artifacts will be apparent for the different shifts—related to Nyquist sampling mismatch characteristics. An inconsistent phase for the highest calculated shift is the best indicator for such problems in imaging plane directions. The consistency of shift and geometry kernels can easily be confirmed by checking against the expected result. The standard deviation of the kernels is a good measure for resolution inconsistency of sequence and reconstruction or an incomplete compatibility of the geometry basis.

2.5.2 Sequence optimization

The sequence characteristics of the previous paragraphs can be used as an input to objective functions that reflect the desired behavior of a sequence. In that case, the feature kernel extractors are restricted to a subspace of sequence parameters and fixed tissue parameters that are to be optimized. Since the kernels are supposedly precalculated, the objective function assembly and the following optimization is detached from the computationally exhaustive simulation steps, and can thus be performed or tweaked even in the presence of limited resources.

2.5.2.1 Examples

Maximization of contrast

If the task is to fine-tune a sequence to provide optimal contrast of two tissues with known physical properties, then the contrast feature of the tissue response kernels can be used almost directly to define the objective function. This objective function is given by:

$$f_{\text{contrast}}(x) = -\left|c_1(x) - c_2(x)\right|, \tag{2.78}$$

where c_1 and c_2 are the contrast maps of the individual tissues and x is an element of the set of relevant sequence parameters.

Maximization of contrast-to-signal ratio
For pratical purposes, it may be more interesting to investigate the contrast-to-signal ratio.

$$f_{\text{contrast to signal}}(x) = -\frac{|c_1(x) - c_2(x)|}{|c_1(x)| + |c_2(x)|}, \tag{2.79}$$

This way, the relative distinguishability is maximized. A tissue combination that produces low signal with a high relative difference is easier to distinguish in practice than a tissue combination that produces high signals with low relative difference.

Minimization of undesired signal
The undesired signal that is caused by the tissue properties corresponds to the sum of all kernel's absolute values except for the center position. This feature cannot be extracted through conventional simulation methods (see Section 2.5.1.2).

$$f_{\text{undesired signal}}(x) = \sum \omega_i u_i(x), \tag{2.80}$$

where u_i are the undesired signals of the tissues of interest, and ω_i are weighting factors.

Simultaneous optimization of run time and contrast
The available kernels can be combined with other measures of sequence quality. Run time is a critical factor for the sequence, and parameters that are key to the run time also often influence contrast. The desired trade-off can be controlled through the weighting factors of the objective function and the optimization is still fast. In this case the objective function is:

$$f_{\text{contrast run time}}(x) = -\left|c_1(x) - c_2(x)\right| + g(t(x)), \tag{2.81}$$

Where t is the runtime and g is a function that applies a weighting.

2.5.3 Model-free parameter mapping

Quantitative MRI is a very active research topic, mainly due to the multitude of available sequences and the lack of a tangible ground truth. It is common practice to choose one sequence, vary one or very few of the sequence parameters and then fit the voxel-wise signal to a simplified signal model.

A set of images acquired with sequences that do not have a common signal model cannot be readily used to extract information simultaneously, and since the different sequence types are likely to be used for the fitting of different parameters, some images may be mostly redundant and scan time could have been saved. The Sequence Response Kernels can assume the role of the common signal model and enable parameter mapping without the need for analytical signal-models.

2.5.3.1 Formal description

Let $\mathcal{I}$ be a K-valued, $N-$dimensional image space, let $I_1, \ldots, I_n \in \mathcal{I}$ be a set of measured images with known sequence and reconstruction descriptions $D_1, \ldots, D_n$, and $\mathcal{D} = \{D_1, \ldots, D_n\}$. Let $\mathcal{P}'^N \subset \mathcal{P}^N$ be a connected subspace of the space of voxelized objects represented by voxel-wise object model parameters. Furthermore, let $\mathrm{Sim} : \mathcal{P}^{\mathcal{N}} \times \mathcal{D} \to \mathcal{I}$ be the simulation.

Objective:
Find an image of parameter vectors $P \in \mathcal{P}'^N$ such that the cumulated residual $r(P)$ of the simulated images $\mathrm{Sim}\,(P, D_1)\,, \ldots, \mathrm{Sim}\,(P, D_n)$ is minimized:

$$r(P) = \sum_{i=1}^{n} \sum_{j=1}^{N} \left| I_i(j) - \mathrm{Sim}(P, D_i)(j) \right|, \tag{2.82}$$

$$\text{find:} \qquad P_{\mathrm{opt}} := \underset{P \in \mathcal{P}'^N}{\arg\min}\, r(P)$$

2.5.3.2 Method

The starting point of the algorithm is a set of images, acquired with given sequences.

First, the parameter fitting domain is determined. This should be done while keeping in mind that fitting in higher dimensions is costly. For the desired fitting domain, Sequence Response Kernels need to be calculated. For algorithmic convenience and illustration purposes, the kernels are truncated to only contrast information. This truncation enables the isolated optimization of each voxel.

For each voxel, the objective function is the sum of the squared differences between simulated image intensity and measured image intensity. The squared differences are calculated in image space.

One evaluation of the objective function is fast. It requires the lookup of the kernel response images at the identical positions, followed by calculating the squared residuals with the image values. Since the kernels are of equal dimensions and the required position is identical, the lookup can be very efficient if the memory/tiles of the images are arranged suitably. The voxel values in the measured images can be kept in memory since they will be constant for the calculation of all the objective function values. In its naïve implementation, this is even more efficient than evaluating a signal model function.

The minimization method must be suitable for the available data and tissue parameter space. Since the contrast values are only available in a discrete space, some basic operations that many methods rely upon—for instance analytical or numerical derivatives—are not readily available. Otherwise however, the objective function is very well suited since it can be expected to be smooth and to have a low number of local minima.

2.5.3.3 Simulated Annealing

Since optimization is not a main topic of this work, a very robust algorithm that can find the optimal solution even under harsh conditions was chosen and is explained very briefly in the following paragraphs. Details are elaborated in the original paper [23]. The Simulated Annealing algorithms is used for all optimization problems in this work.

The general procedure of Simulated Annealing is similar to the hill climbing algorithm. But in order to encourage the algorithm to escape local optima, neighbors that perform worse than the current state are also accepted with a certain probability. This probability or temperature function decreases over time, thereby slowly transitioning the algorithm from random walk to hill climbing.

Simulated Annealing can cope with non-continuous functions or discrete data and is stochastically guaranteed to find the global optimum if given a slowly cooling temperature function, a sufficient number of steps, and a suitable neighbor generating function.

The major drawback of Simulated Annealing is the need for a high number of steps. But in the context of this work, this if of no relevance and even underlines the practicability of the energy function that is to be optimized in the respective optimization problem.

2.5.4 Error characterization

One issue that arises when very different images are combined to extract mutual information is that the shapes of the uncertainties gain importance. The mutual information is likely to be of a higher dimension and as such, standard deviations disregard the shape of the uncertainty. The uncertainty volume may even be non-connected. The shape and volume of this uncertainty also gives hints about what causes the error and what kind of sequence is required to reduce the uncertainty.

2.5.4.1 Formal description

Let $\mathcal{I}$, $I_1, \ldots, I_n$, $\mathcal{D}$, $D_1, \ldots, D_n$, $\mathcal{P}$, $\mathcal{P}'$, P_{opt}, Sim, r be as defined in Section 2.5.3. Let $p(i,P) : \{1,\ldots,N\} \times \mathcal{P} \to \mathcal{P}^N$ be the canonical perturbation of $P_{\text{opt}} \in \mathcal{P}'^N$ in the i^{th} entry, and b :K$\to \{0,1\}$ be an indicator function representing an error threshold.

Objective:
For a given $i \in \{ 1, \ldots, N \}$, approximate

$$E_i = \{x \in \mathcal{P} : f(i,x) = 1\} \tag{2.83}$$

where

$$f(i,x) = b(r(p(i,P)(x))), \tag{2.84}$$

which is a subspace of object model parameter space that satisfies the error condition for the i^{th} object voxel, thus representing its uncertainty.

2.5.4.2 Method

This algorithm is supported by the previous one. The prerequisites and setup of the data is the same up to the definition of the objective function.

The uncertainty volume is the set of parameter combinations that fulfill the uncertainty condition. This could for instance be defined as being within a certain quantile of the possible objective function range—which may be voxel-dependent. If possible, the uncertainty condition should not be a binary function, but a measure of the distance to the threshold to ease the following step.

For each voxel, some initial values are chosen, and Simulated Annealing minimization with the uncertainty condition function is performed.

The parameter space is discretized to allow for volume determination by counting small volume elements that can be assumed to have equal set membership properties.

The final optimization positions are tested for the uncertainty condition, yielding a number of seed points for the next step.

The seed points are then used for an iterative region grow until no more neighbors are left to be tested, resulting in the final uncertainty volume.

The volume is best saved in an octree data structure to avoid unnecessary allocation of memory. It is reasonable to expect that the continuous version of the uncertainty set is locally convex, and the storage of larger blocks of equal set membership is very efficient for the octree data structure.

Another aspect is that the uncertainty condition function results can be cached for Simulated Annealing random walk steps. The Simulated Annealing algorithm is able to cope with discrete spaces, and the function evaluation can be sped up significantly.

2.5.5 Information density optimization

An MRI acquisition protocol is often chosen based on experience. Some specific parameters may be quantified by dedicated sequences with a theoretically evaluated synergistic performance in mind, but that idea cannot be applied to a broader range of possible sequences.

The algorithm explained in the following paragraphs will define a tool to quantize the importance of each individual image in context of the whole protocol and thus allows to omit images that do not yield sufficient extra information with respect to the fitting process employed.

The algorithm leans on the uncertainty volume defined in the previous paragraphs, but could also be built on top of a different error measure.

2.5.5.1 Formal description

Let $\mathcal{I}$, $I_1, \dots, I_n$, $\mathcal{D}$, $D_1, \dots, D_n$, $\mathcal{P}$, $\mathcal{P}'$, Sim be as defined in Section 2.5.3. For an index set $\mathbb{I} \subset \{1, \dots, n\}$ define

$$r(P, \mathbb{I}) := \sum_{i \in \mathbb{I}} \sum_{j=1}^{N} |I_i(j) - \mathrm{Sim}(P, D_i)(j)| \,. \tag{2.85}$$

And define $E_i(\mathbb{I})$ analog to the E_i of Section 2.5.4, where $r(P, \mathbb{I})$ replaces $r(P)$. Furthermore, define a partial order on $I_{\mathbb{I}} := \{I_k : k \in \mathbb{I}\}$ by:

$$I_i \leq I_k \Leftrightarrow |E(I_{\mathbb{I}} \setminus \{I_i\})| \leq |E(I_{\mathbb{I}} \setminus \{I_k\})| \,. \tag{2.86}$$

This expression orders images through the uncertainty volume of all remaining images. The smallest image corresponds to the one that least deteriorates the uncertainty volume when removed from the set of images.

Objective:
Let $\mathbb{I}^0 := \{1, \ldots, n\}$, and $\mathbb{I}^{k+1} := \mathbb{I}^k \setminus \{l_k\}$, where l_k is implicitly defined through $I_k \leq I_i \forall i \in \mathbb{I}^k$. Find $l_0, \ldots, l_{n-1}$.

This sequence describes the resulting order of images when the image that deteriorates the uncertainty volume least is removed iteratively.

2.5.5.2 Method

The uncertainty volume measure can be calculated for any subset of the available images. If one image is removed, then the difference in the uncertainty volume images can be interpreted as a representation of the individual image's contribution to the overall information density.

That way, starting at a well over-determined set of images, the least important image can be eliminated iteratively. This results in a ranking of the images by their order in the elimination chain, which itself can be a helpful indicator for determining the usefulness of an image within the protocol.

CHAPTER 3

Results

3.0 Introduction

In this chapter, the main features of each individual simulation method are revealed. Each of them has its own main area of application, advantage over the other methods and systematic imperfection. Those were qualitatively described in the methods chapter but lack supporting arguments, which are provided here through simulation experiments.

The computational effort that each method requires is extracted for each workflow step, and then put into perspective by applying the estimation to realistic simulation conditions.

Lastly, algorithms that are derived from the Sequence Response Kernel approach are put into action to show the versatility of the approach.

All measured images were acquired on a Siemens Skyra 3 T system [32].

Direct Bloch simulation 3.1

The direct Bloch solver was not developed further. The algorithm is taken as the ground truth for the later methods of increased efficiency. For that reason, simulation results of this approach are not presented as results of this work. Details can be found in [22] [35] [2] [14] [40] and a comparison with existing solvers is omitted due to the availability of analytical solutions and the focus on algorithmic efficiency opposed to benchmarking.

The efficiency is estimated in the following paragraphs as a result of the abstract workflow presented in Section 2.1.4.

3.1.1 Full simulation workflow effort estimation

In any case it is reasonable to choose a stepper that can adapt the integration step size, but there are lower bounds for the step sizes that can be assumed to be respected because the effects that are present are resolved by that step size.

In the absence of RF pulses, thus in the absence of any magnetic field in the transverse plane, there are no effects that occur on a timescale smaller than the smallest of: relaxation, phase gain of the applied gradient and offset frequency. In cylindrical coordinates, the precession effects can be calculated accurately when the steps match the linear segments of the gradient channel. A trapezoidal gradient pulse for example does not require more than four steps to be processed adequately.

The relaxation process is an exponential decay with rates $T1^{-1}$ or $T2^{-1}$. The step size should be smaller than the corresponding time scales of those effects.

In the presence of an arbitrary RF pulse, the behavior of a spin packet is limited by the maximum pulse amplitude. These effects are on the order of the RF pulse's sampling rate.

The signal has to be calculated at every time point at which the sequence is set to acquire it. This often occurs during the flat-top time of the gradients and thus requires further steps that would otherwise not be necessary.

The effort of integrating the Bloch equation is mainly dictated by the complexity and number of pulses which have a large contribution, and by the complexity of the gradient pulse shapes.

In addition to the integration of the Bloch equation, the signal has to be accumulated. Each spin packet needs to be weighted according to the local coil sensitivity and can then be added to the total coil signal. In case of multiple coils, the signal has to be accumulated with multiple respective sensitivities and thus needs to be accumulated once for each coil.

In general, the effort $E_{\text{Direct Bloch simulation}}$, quantized by the number of operations as defined in Section 1.3.1.2 can be estimated as follows:

$$\begin{aligned}
E_{\text{Direct Bloch simulation}} &= O\left(N_{\text{spin packets}} \cdot \; N_{\text{sequence events}}\right) \\
&= O\Big(N_{\text{spin packets}} \cdot \big(N_{\text{gradient channel linear segments}} \\
&\qquad + N_{\text{RF pulse samples}} \\
&\qquad + N_{\text{acquisitions}} \cdot N_{\text{coils}} \\
&\qquad + N_{\text{pulse free time periods}}\big)\Big)
\end{aligned} \tag{3.1}$$

This will be put into the context of full imaging sequence simulation in Section 3.5.

Smart Bloch simulation 3.2

The Smart Bloch simulation introduced concepts to handle the Bloch equation in its most common specialization most efficiently. The simulation approaches that were defined in the methods chapter are supported by analytical arguments and do not require further validation. The practical value of the pulse instantanization will be illustrated in the next paragraphs, followed by arguments about T2′ approximation which are useful for the k-space method. Direct comparison of Smart Bloch simulation and Direct Bloch simulation is omitted because both methods are equally exact by design.

The full workflow effort is estimated for each workflow step, as a result of the workflow description (see Section 2.2.3).

3.2.1 Pulse instantanization

A sinc pulse is often assumed to act uniformly on a selected frequency band. However, the pulse also creates a mostly linear phase dispersion in its pulse profile (Figure 3.1). If the spin packets experienced the frequency band due to a slice selection gradient pulse, then the linear part of the phase dispersion can be corrected by applying the rewinder gradient.

If this pulse effect is calculated using the Smart Bloch Solver but without the instantanization correction, then the solutions will show the same phase behavior as for the pulse without the rewinder gradient. The pulse effects are then hard to analyze and hinder the interpolation capabilities of the solutions.

The corrected $\tilde{P}$ (see Section 2.2.1.3) generates algorithmically equivalent results as the decomposition and recombination part alone, but the interpolatory capabilities of the pulse effect is greatly increased.

The choice of the pivot point for the instantanization correction can be arbitrary and does not even necessarily need to be inside the pulse, but is optimally placed at a point which minimizes the phase dispersion of the corrected pulse profile. In this example this pivot point is chosen to be at the center of the pulse.

The Figure 3.2 illustrates that the relaxation effects that occur during an RF pulse are also more homogeneous and thus easier to interpolate when the correction is employed.

The correction reduces most of the relaxation effects that occur during the pulse, most notably for low T2 values. The amplitude of the corrected result varies less; low T1 appears to be problematic, but tissues for which T1 $<$ T2 are not realistic [37].

3.2.2 T2′ simplification

In theory, the offset frequency that the T2′ of each tissue induces, perturbs the pulse effect that a spin packet experiences. This is particularly costly for the k-space simulation because each different pulse effect requires an individual Echo Pathway Segment calculation. However, in most practical circumstances, the influence of this effect is so

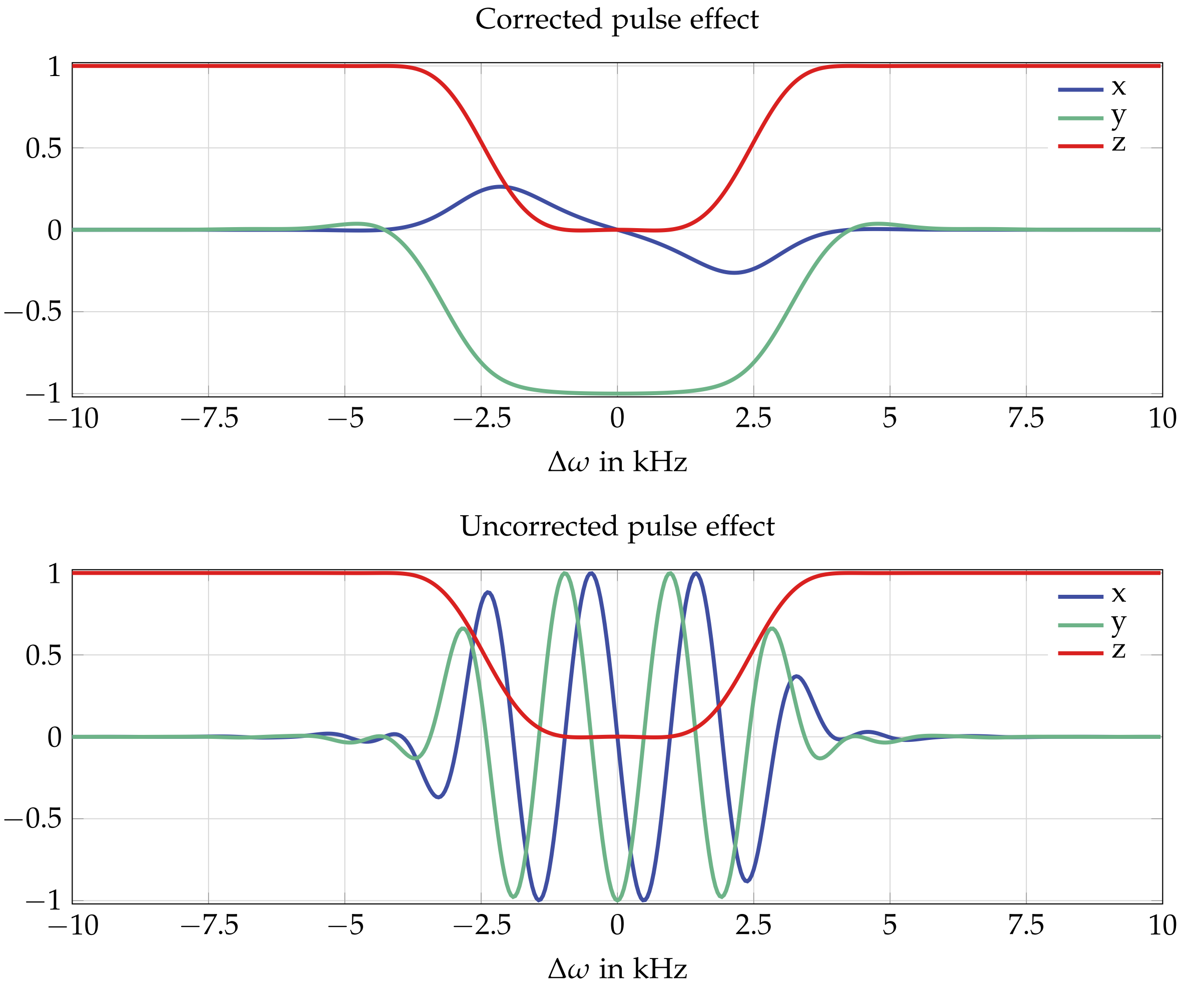


Figure 3.1: Corrected and uncorrected pulse effect of a Hann-windowed sinc (Figure 1.2) pulse with three side lobes and a duration of 2 ms, applied to a spin packet in equilibrium (M_0=1). The correction removes the oscillation and the result can be interpolated easier.

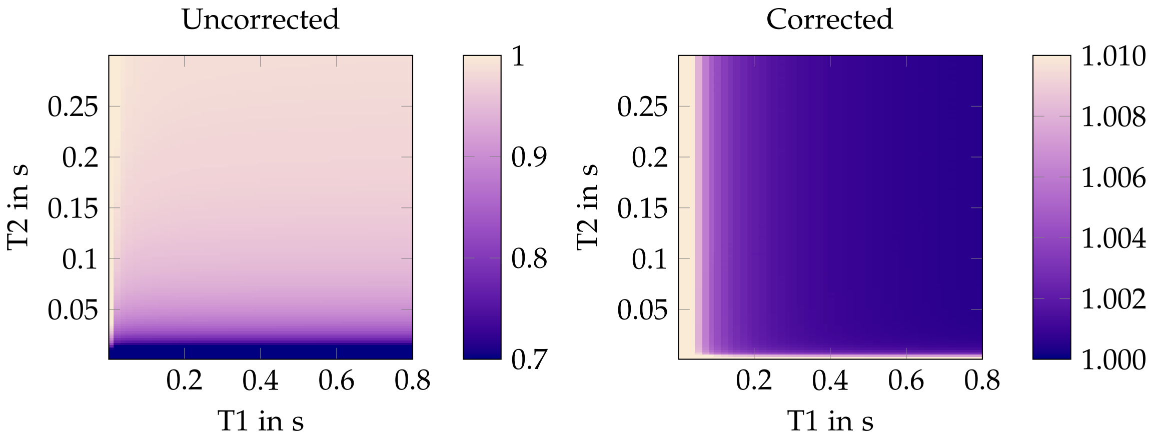


Figure 3.2: Corrected and uncorrected pulse effect of the same pulse used in Figure 3.1, applied to a spin packet in equilibrium (M_0=1). A very long pulse with a duration of 10 ms was chosen to produce extreme results. The color represents the magnitude of the transverse components.

small, that it can be ignored or resolved sparsely. The simulation experiment illustrated by Figure 3.3 casts light on the actual deviation of the magnetization.

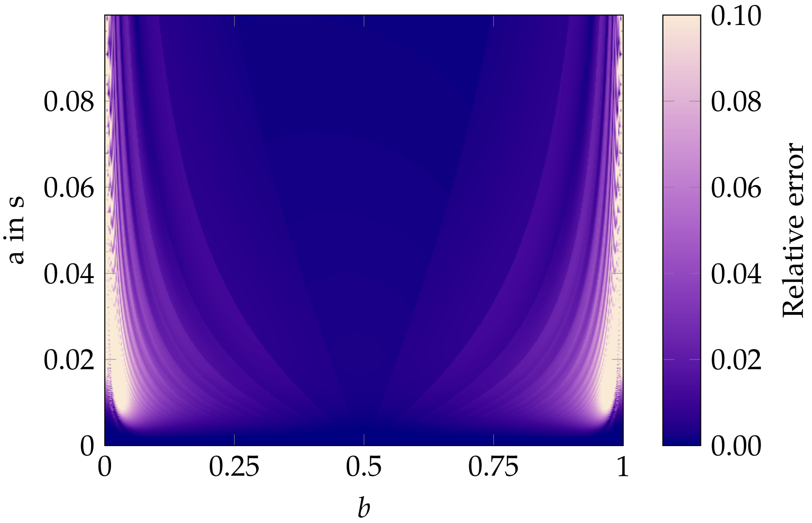


Figure 3.3: Signal errors that result from not respecting the T2′-induced offset frequency in the pulse effects, and instead using a homogeneous pulse effect. Each vertical position on the 2D plane corresponds to a tissue type $a = 2 \cdot \mathrm{T1} = 2 \cdot \mathrm{T2}$. The horizontal axis corresponds to the $T2'$ induced offset resonance distribution. The value at each position is the signal difference magnitude between a spin packet that is simulated respecting/not respecting the resonance offset in the pulse effects. The employed sequence is a spin echo sequence with $\mathrm{TE} = 40\,\mathrm{ms}$ and Hann-windowed sinc pulses with a bandwidth of $3\,\mathrm{kHz}$, $|M_0| = 1$, displayed at $t = \mathrm{TE}$.

Even though all offset frequencies are theoretically present for any T2*, the relative amount of spin packets outside of the pulse's bandwidth is very low for realistic tissues. Furthermore, the strong frequency change at the boundaries of the distribution disperses that magnetization—further weakening the impact of the perturbed pulse effect.

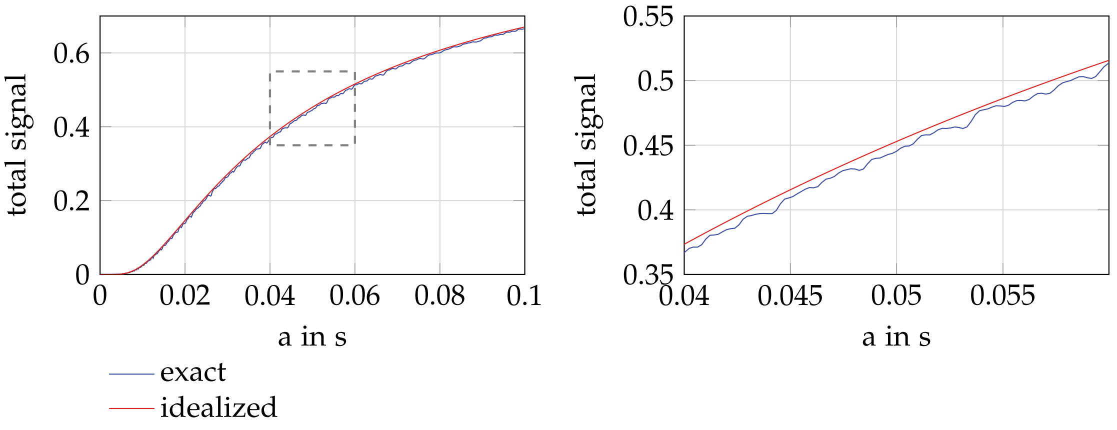


Figure 3.4: Signal errors that result from not respecting the T2′-induced offset resonance frequency in the pulse effects. The setting is the same as in Figure 3.3, but the transverse components of the magnetizations were accumulated in b direction to illustrate the signal error magnitude for that tissue.

The actual tissue-dependent signal difference is negligibly small, as illustrated in Figure 3.4. The errors of the simulation occur at frequencies that are already highly dispersed and barely noticeable in the total signal.

In conclusion, even extreme T2* values usually do not need to be resolved by the pulse effect. The pulse can be assumed to have a constant effect—enabling the analytical handling of the T2′ effect in the k-space simulation approach for common use cases.

3.2.3 Full workflow effort estimation

For the Smart Bloch solver, the numerical effort can be estimated more meaningfully and directly, in terms of the sequence's structure rather than step size estimates. The workflow can be separated into three logical sub-processes with partially separated effort contributions.

Pulse precalculation
A solution for each RF pulse and spin packet parameter combination is required for each of the four basis magnetizations. Depending on the dimensionality reduction that the RF pulse type allows, certain spin packet and pulse parameters may not yield unique pulse effects and can be omitted. The numerical effort for one such integration is on the order of the number of pulse samples at most. Thus for a complex pulse which is 10 ms long, $E_{\text{single pulse simulation}} = 10\,000$ evaluations of the right hand side of the Bloch equation are roughly the worst case. Simple pulses perform better.

The effort of this stage can be estimated as follows:

$$E_{\text{pulse precalculation}} = O\left(E_{\text{single pulse simulation}} \cdot N_{\text{unique pulses}} \cdot N_{\text{unique spin packets}}\right) \quad (3.2)$$

Spin packet states at pulse instances
For each RF pulse instance and each spin packet, the effect of the time between each two pulses has to be applied, followed by the next pulse effect. This takes one operation each, which is greatly accelerated in comparison with the recalculation of each pulse effect for each pulse instance.

The effort of this stage is:

$$E_{\text{packet states at instances}} = O\left(N_{\text{spin packets}} \cdot N_{\text{pulse instances}}\right) \quad (3.3)$$

Signal calculation
For each signal acquisition event and each spin packet, the magnetization at the time of acquisition can be calculated from the magnetization at the previous pulse in one step. After that, the magnetization has to be processed once for each coil.

The effort amounts to:

$$E_{\text{signal calculation}} = O\left(N_{\text{spin packets}} \cdot N_{\text{samples}} \cdot N_{\text{coils}}\right) \quad (3.4)$$

The effort will be put into context and compared with the other methods in a later section (see Section 3.5).

k-space based simulation 3.3

The k-space based simulation makes use of a change in state space, which is not easy to grasp in the context of full image calculation. Therefore, the side-effects from choosing insufficient pathways, or a sequence with poor echo behavior is illustrated in the first section.

Intermediate results of this simulation approach are also of value. Applying the Echo Pathway Tree template to parameter sets can yield information about B1 robustness or the slice profile, as elaborated in the following two sections.

Finally, the effort is estimated for the individual workflow steps that were defined in the methods chapter (see Section 2.3.5).

3.3.1 Pathway tree truncation

The following diagram (Figure 3.5) shows the effect of choosing a low truncation limit for the maximum number of pathways. The phantom used for this experiment is a variation of the Shepp Logan Phantom [12] [24], which is defined as a set of ellipsoids with homogeneous proton density and relaxation times.

Even in this simple case of a very basic sequence, effects are present that cannot be recreated by a conventional Bloch simulation with reasonable effort.

Using only three pathways per pulse yields the same result as the signal equation of the sequence if perfect spoiling is assumed, or alternatively a Bloch approach that is sampled with one spin packet at the center of each voxel. Using the ten most important pathways reveals the extra signal that is created by the spoiled echoes of the previous pulse. The echo intensity is rather high since the flip angle is low, but the k-values of those echoes are also high due to the spoiling. The imperfection of the 4π spoiling in this sequence is a result of the sharp edges of this analytical phantom, which yield a slowly decaying Fourier transformed density that still has considerable amplitudes at the Echo Pathway Segment positions. Figure 3.6 shows the result of an even more imperfect spoiling.

Advancing to 200 pathways per pulse reveals further artifacts. Some of the non-primary pathways are now refocused and merged into the central pathways, which influences the main segments constructively or destructively, depending on the RF pulse phase, which is designed to minimize the unwanted stimulated echo effect by employing RF spoiling (see Section 1.2.3.1). If those effects are to be recreated using the Direct Bloch solver or Smart Bloch solver approach, then a resolution that can comfortably resolve a phase dispersion of approximately 250π per voxel is required.

If an even deeper pathway tree is generated, then the image still changes slightly. Now even the tissues that possess particularly high relaxation can be calculated with accurate signal intensities. A pathway tree that traces 2000 pathway segments per pulse can be considered a full representation for this sequence.

The effects that the k-space simulation reveals in this example are not drastic, which is why they are often overlooked. However, for more extreme sequences, those effects

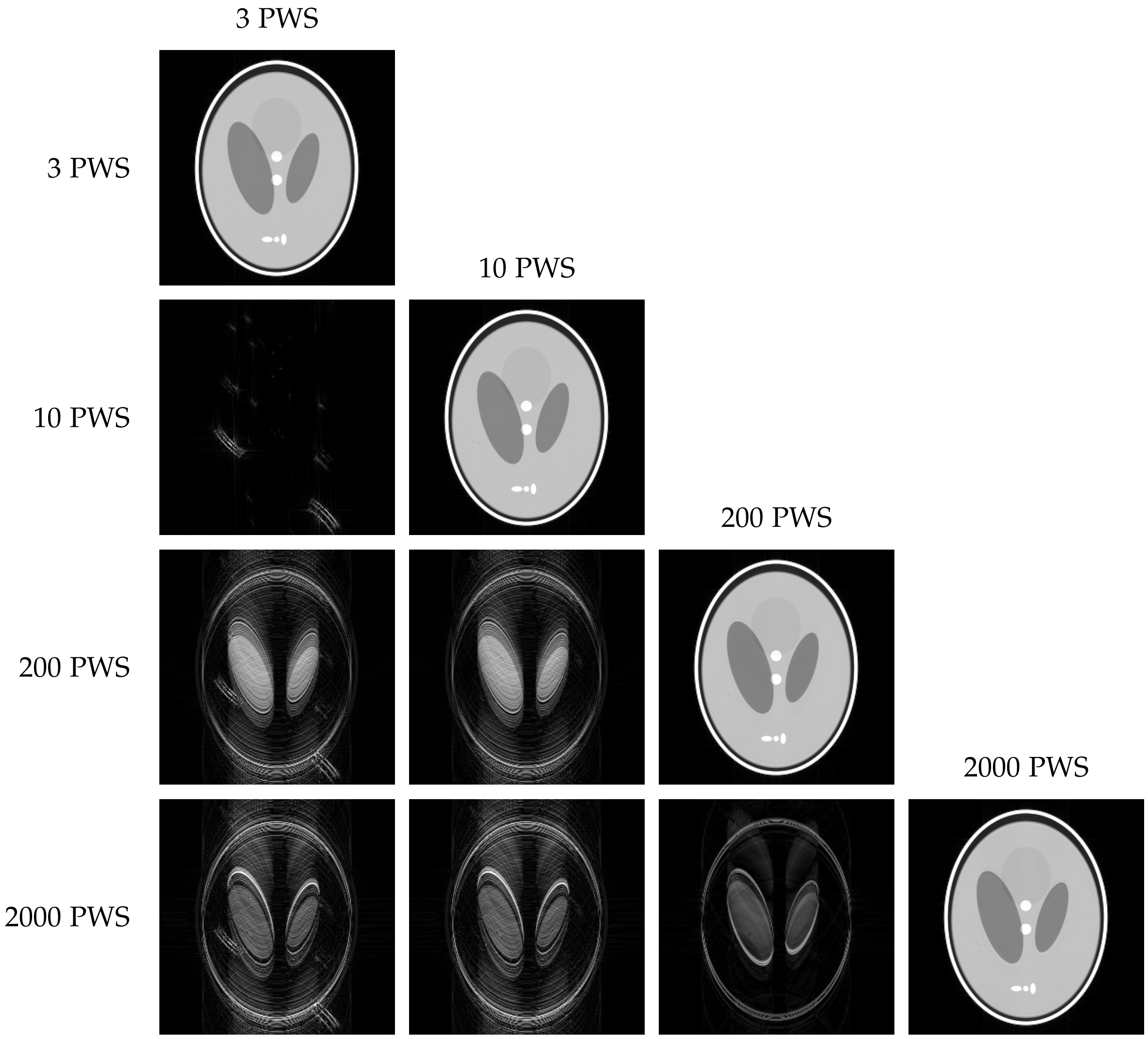


Figure 3.5: Simulation results and difference images of an MRI Shepp Logan Phantom, measured using a spoiled gradient echo sequence (Figure 1.5) with TE $= 8\,\text{ms}$, TR $= 50\,\text{ms}$, 15° flip angle, 4π spoiler. The simulation was performed keeping only the most important 3, 10, 200 or 2000 pathway segments at each step. The differences are emphasized in the difference pictures.

can cause undesired behavior that is hidden if the (Smart) Bloch simulation approach, or a simplified signal model is pursued.

Figure 3.6 shows variations of the same simulation experiment with adjusted parameters that show extreme cases of the previously described effects.

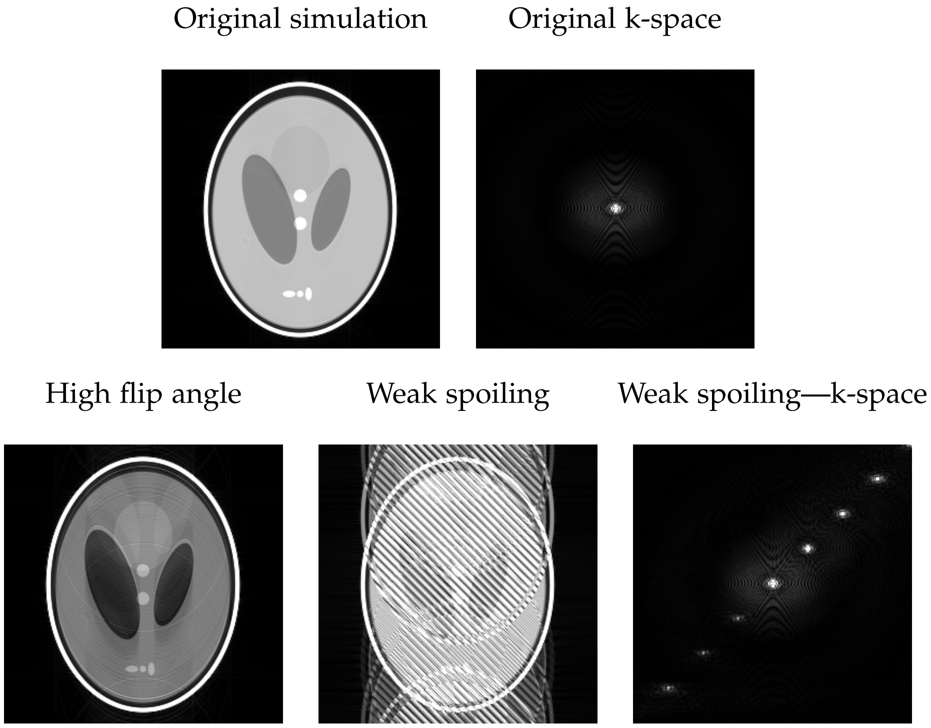


Figure 3.6: The same simulation experiment as in Figure 3.5, using 2000 pathways per pulse. The first row shows the same image as the previous figure, alongside with it's magnitude k-space image. On the second row, the flip angle is changed to 40° for the first image, and spoiling is set to $\pi/8$ for the other two images.

The 40° flip angle provokes a stronger mixing of the segments and therefore inconsistent echo properties.

The weak spoiling causes a significant amount of misplaced signal to still be present. This is clearly visible in k-space as replications of the original k-space in spoiling direction. The reconstructed image shows shifted replications of the original images with superposed high-frequency waves in spoiling direction. The shift in y-direction is due to the different phase of RF pulse and acquisition that is chosen for each line acquisition.

These effects appear obvious in these artificial examples, but under non-virtual experimental conditions, the source of the undesired effects is often hidden, unclear and impossible to isolate. A simulation is a natural choice to identify the source of these problems, but if the sources are of a similar form as here, then the Direct Bloch simulation or Smart Bloch simulation is a poor choice, while the k-space formalism is very efficient.

3.3.2 TSE main pathway intensity

The k-space formalism is particularly interesting for sequences with complex pulse schemes and non-trivial gradient moments between the pulses. For those sequences it can predict signal that cannot feasibly be resolved by the Direct Bloch simulation or Smart Bloch Simulation. This example (Figure 3.7) illustrates the applicability of the method by investigating different crushing schemes for a TSE sequence.

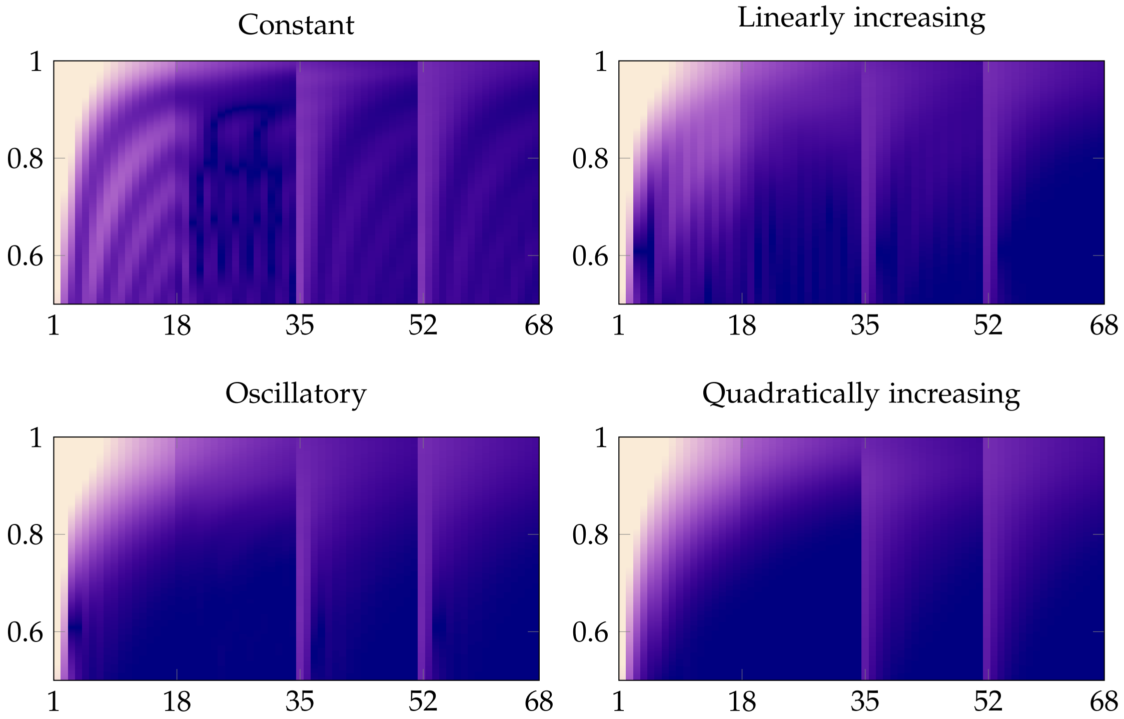


Figure 3.7: Echo magnitudes of the main echo of a TSE sequence for different crushing schemes, depending on the B1 transmissivity. The sequence and relaxation parameters of this experiment are $20 \cdot \mathrm{TE} = \mathrm{T1} = \mathrm{T2}$ (spacing of the refocusing pulses), $\mathrm{TR} = \mathrm{T1} = \mathrm{T2}$ and a turbo factor of 16. The horizontal axis represents the pulse instance, the vertical axis represents the relative B1 transmissivity with respect to the intended magnitude.

The crushing technique employs a pair of strong equal gradient pulses that are applied directly before and after a refocusing pulse. This separates the Echo Pathway Segments in the direction of the gradient pulses (see Section 1.2.3.1). Depending on the crushing scheme, segments can interfere constructively or destructively. In these examples, the moments of the crushing gradients are multiples of a basis moment, which suffices to rule out influence of the crushed segments on the main signal. The schemes of this example are: constant $(1, 1, 1, 1, \ldots)$, linearly increasing $(1, 2, 3, 4, \ldots)$, oscillatory $(1, -1, 1, -1, \ldots)$, and quadratically increasing $(1, 4, 9, 16, \ldots)$. Further details of the crushing schemes can be found in [13].

The quadratically increasing crushers show the clearest, most predictable behavior since no pathway segments can fall on top of each other—the main pathway decays with $T2$ after each excitation pulse. The system is not in a steady state at the first and second excitation pulse, but the third and fourth excitation pulses are applied to a mostly identical system.

The oscillating crushing scheme performs almost as good as the quadratically increasing one. One great advantage of this crushing scheme is that it does not reach hardware limits with an increasing number of refocusing pulses—opposed to the previous crushing scheme.

The linearly increasing crushers cause a mixing of the pathways starting at each fifth refocusing pulse. The signal is considerably higher, but the decay is less intuitive to understand. The signal behavior at an RF pulse transmissivity of 1 is the same for all crushing schemes because the pulses are perfect 180° pulses. But a transmissivity of 70 % or less would result in an image with blurring that is hard to grasp visually, because the lines in k-space show complicated signal fluctuations. It also takes slightly longer for this configuration to reach a steady state.

The last sequence configuration employs constant spoiler gradients. The overall signal magnitude is highest, but a lot of stimulated echoes are produced. The patterns of those stimulated echoes in tissue and parameter space are hard to describe and produce artifacts in the resulting images which are hard to understand.

Conclusion

The k-space formalism is a great tool for understanding MRI sequences in a way that is different to the standard Bloch simulation approaches. The methods that are presented in this work provide an efficient way to investigate echo behavior—most notably through the use of a template pathway tree—while still maintaining full accuracy.

3.3.3 SER slice profile

Simultaneous Echo Refocusing [13] is a technique that relies heavily on Echo Pathway analysis. This experiment (Figure 3.8) investigates a SER sequence with parameters similar to those used for the TSE sequence in the previous section. The only structural difference between these two sequences, that is important for the k-space formalism, is that instead of one excitation pulse, two excitation pulses are employed. These pulses excite different slices, yet the refocusing pulses are selective to both. This leads to an intricate slice profile.

The Echo Pathway Segment mixing and decay effects are qualitatively similar to those of the TSE sequence. The general signal shape agrees with [13] and additionally reveals the echo behavior over time and over multiple shots.

Conclusion

The k-space formalism defined in this work can be used effectively to investigate experiments for which the exact pulse dynamics are of interest. This goes beyond the contemporarily used extended phase graph implementations, as discussed in Section 2.3.4.4.

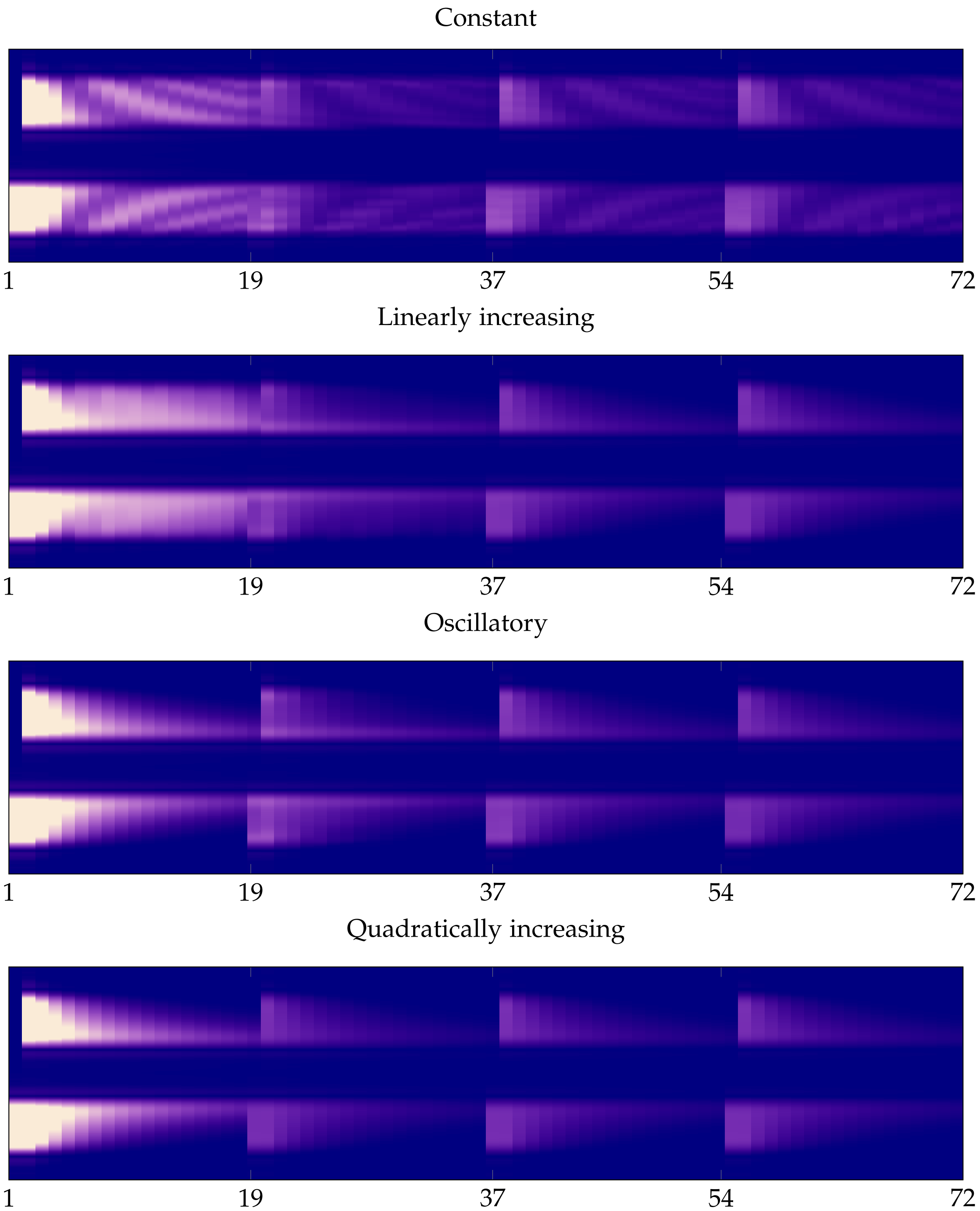


Figure 3.8: Main echo intensities of a SER sequence. The vertical axis describes the slice position for a constant transmissivity of 1. All other settings are as in Figure 3.7. The individual images correspond to different spoiling schemes—as in Figure 3.7. The excitation pulses have a bandwith-time product of 5, the refocusing pulses have a bandwidth-time product of 30. Both are Hann-windowed sinc pulses. The bands of the excitation pulses are chosen to touch, the band of the refocusing pulses are chosen to embrace the excitation bands exactly.

3.3.4 Full workflow effort estimation

In contrast to the previous method, the effort required to simulate a sequence using the k-space formalism approach heavily depends on sequence and phantom properties. Typically, a pathway tree template is created first by generating initial trees, discarding segments according to a chosen importance score. Then this template is applied to all required tissue parameters, which is a discrete number that depends on the phantom. For each tree, the signal is then evaluated and added to the total signal. These three stages are preceded by the pulse effect precalculation of the Smart Bloch solver.

Echo Pathway Tree template

Each important Echo Pathway Segment needs to be traced. In the worst case scenario, each pulse triples the number of segments. The number of pulses itself is usually of the order of the number of lines in the acquired image. This is not computationally feasible, so the maximum number of segments at each pulse should always have a hard limit. In the best case scenario, which is equidistantly positioned pulses with constant gradient moments in between, the number of segments present can be merged drastically to a linear growth.

The tree is generated iteratively, starting with one initial pathway. At any pulse, the pathways need to first undergo all processes that occur in between the two pulses, then each pathway splits into three children with split factors determined by the same pulse precalculation as the Smart Bloch Solver approach. This is basically one operation per segment.

The effort of splitting all segments at a pulse is:

$$E_{\text{segment splitting}} = O\left(N_{\text{segments before pulse}}\right) \tag{3.5}$$

The segments need to be merged. Therefore they have to be sorted:

$$E_{\text{pre-merge sorting}} = O\left(N_{\text{new segments}} \cdot \log(N_{\text{new segments}})\right) \tag{3.6}$$

After merging, the score of each segment needs to be calculated and the worst-performing segments need to be eliminated:

$$E_{\text{pre-elimination sorting}} = O\left(N_{\text{merged segments}} \cdot \log(N_{\text{merged segments}})\right) \tag{3.7}$$

This can be simplified:

$$E_{\text{single pulse succession}} = O\left(N_{\text{segments present at pulse}} \cdot \log(N_{\text{segmetns present at pulse}})\right) \tag{3.8}$$

For the worst case of no possible merging and no truncation limit, this amounts to:

$$\begin{aligned} E_{\text{pathway template}} &= O\left(\sum_{n=0}^{N_{\text{pulses}}} 3^n \cdot \log(3^n)\right) \\ &= O\left({}^3/_4\left(2 \cdot 3^{N_{\text{pulses}}} N_{\text{pulses}} - 3^{N_{\text{pulses}}} + 1\right)\log(3)\right) \\ &= O\left(3^{N_{\text{pulses}}} N_{\text{pulses}}\right) \end{aligned} \tag{3.9}$$

Which is worse than exponential and cannot be handled computationally for realistic sequences.

If a truncation limit is enforced and the sequence is ill-suited, such that this limit is reached almost immediately, the effort can be estimated as follows:

$$\begin{aligned} E_{\text{pathway template}} &= O\left(\sum_{n=0}^{N_{\text{pulses}}} N_{\text{truncation limit}} \cdot \log(N_{\text{truncation limit}})\right) \\ &= O\left(N_{\text{pulses}} \cdot N_{\text{truncation limit}} \cdot \log(N_{\text{truncation limit}})\right) \end{aligned} \tag{3.10}$$

Thus the effort scales linearly with the number of pulses, while the truncation limit can be adjusted to achieve an acceptable computation time.

If the merging can be performed with optimal—but common—effectiveness, the number of segments present increases linearly. In that case:

$$\begin{aligned} E_{\text{pathway template}} &= O\left(\sum_{n=0}^{N_{\text{pulses}}} 3n \cdot \log(3n)\right) \\ &= O\left(\sum_{n=0}^{N_{\text{pulses}}} n \cdot \log(n)\right) \\ &= O\left({N_{\text{pulses}}}^2 \cdot \log(N_{\text{pulses}})\right). \end{aligned} \tag{3.11}$$

In practice, the effort of a pathway template creation is between former two estimates and will fall back to the truncation-limited effort progression eventually. This ensures that the effort eventually scales linearly with the number of pulses, but it is important to keep in mind that the quality of the pathway tree template depends on the preservation relevant Echo Pathway Segments. This quality is likely to suffer if the truncation-limited progression is reached early.

Echo Pathway Tree regeneration

After the template is calculated, trees can be generated efficiently for any tissue parameter set. Each Echo Pathway Segment is calculated by combining its parents at their state right before the respective pulse with their according pulse transition factors. Then the effects between the pulses should be calculated for that segment for further usage required by the next pulse. In the regeneration phase, each segment needs to be operated on only once. Thus:

$$E_{\text{pathway regenerations}} = O\left(N_{\text{unique tissues}} \cdot N_{\text{all segments}}\right) \tag{3.12}$$

Signal calculation

At every sample, the available transverse segments need to be evaluated at their corresponding wavenumber to calculate the signal. For that it is reasonable to again discard any Echo Pathway Segment that is too far away from the k-space origin to produce relevant signal. It is very unlikely that there are more than a handful of relevant segments. In most cases there is only one. It is important to note that discarding further segments for the signal calculation is reasonable—this is because segments that are far away from the center could very well be refocused and contribute to the intensities of the main segments in the future, but are dormant for some signal acquisitions.

If the relevant segments are marked in the pathway tree template, then calculating the signal of one sample requires only one evaluation of the phantom's k-space geometry at the corresponding position—per tissue type:

$$E_{\text{signal calculation}} = O\left(N_{\text{unique tissues}} \cdot N_{\text{relevant segments}} \cdot N_{\text{acquisitions}}\right) \tag{3.13}$$

This potentially minor effort is the main argument for choosing the k-space formalism.

Further information about the computational effort will be elaborated later (see Section 3.5).

3.4 Sequence Response Kernel simulation

This section showcases Sequence Response Kernel simulations of full images, compared with physical measurements. This is followed by the last simulation workflow effort estimation, based on the workflow defined in Section 2.4.3.

Further Sequence Response Kernel-related examples and results can be found in the derived algorithms section of this chapter (see Section 3.6).

3.4.1 Image assembly—contrast tubes

The following experiment (Figure 3.10) displays the speed-accuracy trade-off of the Sequence Response Kernel approach. Six cylindrical gel phantoms were measured using a set of spin echo sequences to determine their characteristic T1, T2 and proton density values. Afterwards, they were measured using a bSSFP sequence (Figure 1.7) to be compared with simulation results.

Position	PD	T1 in ms	T2 in ms
top left	1	222	42
top right	1	88	61
middle left	1	378	63
middle right	1	284	85
bottom left	0.4	2087	2372
bottom right	1	486	157

Table 3.9: Average tissue parameters of six gel phantoms.

The simulation was performed using tissue parameter ranges sufficient to cover the measured gel phantom parameters. The first simulation was performed with heavily truncated kernels of size one to reflect the basic image contrast. The second image assembly used tissue kernels that were big enough to hold the full relevant signal.

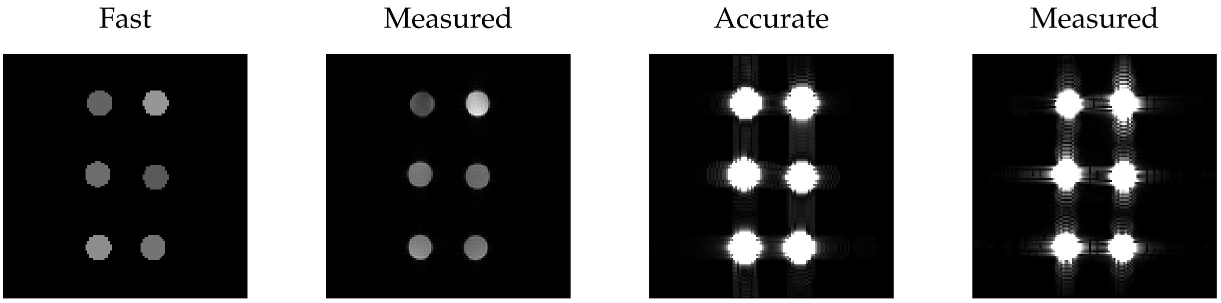


Figure 3.10: Simulated and measured gel phantoms. The fast simulation and accurate simulation are using the single most important kernel value, and the 128 most important kernel values respectively. The measured images show the same measured gel phantoms with adjusted windowing to allow for comparison. The sequence used for this image is a bSSFP sequence with $TR = 9.6\,\text{ms}$, flip angle $= 44°$, 20 averages.

The image contrast can be recreated with the heavily truncated kernels. This calculation took 6 ms on a contemporary desktop computer. The larger kernels can produce some artifacts that are similar to those seen in the measured image. However, one major source of the image artifacts are the partial volumes that are not respected by this geometry basis choice. This calculation took 44 s, which is on the order of the run time of the sequence.

3.4.2 Image assembly—brain

This example (Figure 3.11) is based on a parameter map that is generated in a later section (see Section 3.6.3). The parameter map was then used to simulate four TSE sequences using heavily truncated kernels of size one. The perceptual tissue contrast of the images can be recovered adequately and with great computation speed.

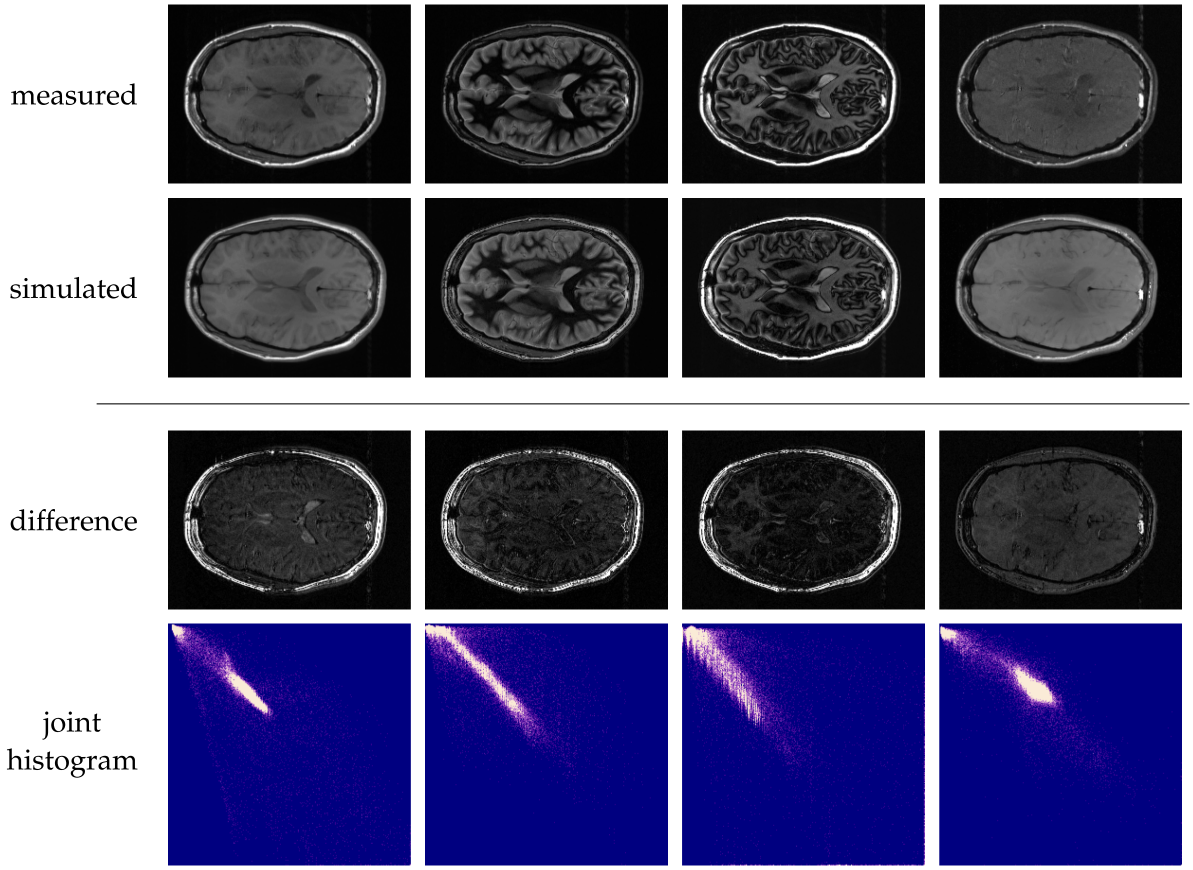


Figure 3.11: Four pairs of measured and simulated TSE images of a brain. The sequence parameters can be found in Table 3.29 at index 9, 18, 19 and 26. The transfer functions of the images, including the difference images, are equal.

The errors are pronounced at the skull area, mostly due to fat, which requires offset frequency consideration which is not employed in the parameter mapping step. The sinus sagittalis, seen near the boundary of the brain on the right side of the image, introduced flow artifacts on some acquired images. They can be seen as stripes in phase encoding (vertical) direction. The effects of movement between the individual images deteriorates the fitting quality at sharp edges, such as at the skull.

The other errors are likely due to the over-determinedness of the parameter extraction phase. The fitting errors are investigated further in a following section (see Section 3.6.4).

3.4.3 Full workflow effort estimation

The Sequence Response Kernel simulation workflow is based on the Echo Pathway Tree as well, so it shares the effort of pathway tree template generation and pathway tree regeneration with the k-space approach.

3.4.3.1 Kernel calculations

Each kernel calculation requires a sampling of the signal contribution of each element to signal space:

$$E_{\text{signal accumulation}} = O\left(N_{\text{elements}} \cdot N_{\text{pathway segment groups}} \cdot N_{\text{acquisitions}}\right) . \tag{3.14}$$

Where the number of elements is either tissue, geometry or shift basis size. The pathway grouping which is required in the pathway tree template generation step does not introduce significant overhead.

Next, the image of each Echo Pathway Segment group for each parameter set needs to be reconstructed individually. In the context of this work, the effort for this step is:

$$E_{\text{reconstructions}} = O\left(N_{\text{elements}} \cdot N_{\text{segment groups}} \cdot N_{\text{acquisitions}} \cdot \log(N_{\text{acquisitions}})\right) , \tag{3.15}$$

which is mainly dictated by the Fourier transform.

The truncation step requires another sorting of all kernel values of each image. This process is as exhaustive as the reconstruction:

$$E_{\text{truncation}} = O\left(N_{\text{elements}} \cdot N_{\text{segment groups}} \cdot N_{\text{acquisitions}} \cdot \log(N_{\text{acquisitions}})\right) . \tag{3.16}$$

Tissue kernel calculation

For each parameter combination of the tissue basis, the pathway tree has to be regenerated the same way as in the k-space approach:

$$E_{\text{pathway tree regenerations}} = O\left(N_{\text{tissue basis}} \cdot N_{\text{all segments}}\right) \tag{3.17}$$

The rest of the effort is the same as all other kernel efforts with

$$N_{\text{elements}} = N_{\text{tissue basis}} \tag{3.18}$$

Geometry kernel calculation

The geometry basis does not require more than one element if the voxel basis is chosen. This basis does not require a pathway tree template regeneration since the geometry basis is independent of the tissue types.

$$N_{\text{elements}} = 1 \tag{3.19}$$

Shift kernel calculation
The shift kernel basis size introduces a trade-off between the number of convolutions that are required in the image generation process and the number of elements that need to undergo the signal contribution kernel extraction process. Multiple convolutions of shift kernels also potentially introduce new errors. Any basis size between one, the number of image voxels and even beyond that are feasible, but a good choice is:

$$N_{\text{elements}} = \log(N_{\text{image resolution}}) \tag{3.20}$$

3.4.3.2 Image assembly

The decomposition of the phantom into basis functions should be easy if the basis is chosen in a compatible way. The number of basis elements that result from this decomposition should be on the order of the image dimension in case of a voxel basis. It may be smaller if many voxels are empty or it could be larger if there is a fuzzy relationship between the tissue types and the image voxels. For each such phantom component, the convolutions have to be performed. For a general convolution followed by another truncation:

$$E_{\text{conv}}(T_1, T_2) = O(T_1 \cdot T_2 \cdot \log(T_1 \cdot T_2)) \tag{3.21}$$

Where T_1, T_2 are the respective truncation limits (not relaxation times).

Even though the convolution operation is commutative, convolution order matters. The shift kernels are usually the sharpest and a low truncation limit does not deteriorate the result. On the other hand, the tissue kernel holds most of the artifact information and requires the highest truncation limit. It should be applied at the end. In most cases, the order shift → geometry → tissue is the best choice. The composite effort of one phantom component is:

$$\begin{aligned} E_{\text{component processing}} = O(&T_{\text{shift}} \cdot N_{\text{shifts}} \cdot E_{\text{conv}}(T_1, T_1) \\ &+ E_{\text{conv}}(T_1, T_{\text{geometry}}) + E_{\text{conv}}(T_2, T_{\text{tissue}})) \end{aligned} \tag{3.22}$$

Where T_1 and T_2 are the truncation limits that are enforced after the individual convolution steps.

The full image assembly requires:

$$E_{\text{image assembly}} = O(N_{\text{components}} \cdot E_{\text{component processing}}) \tag{3.23}$$

The effort estimations will be put into relation in the next section (see Section 3.5).

3.5 Effort estimation

The individual simulation methods perform well at distinct sub-tasks and suffer from their respective weaknesses. In this section, the effort estimations of the previous sections are used to reveal the bottlenecks of the full imaging sequence simulation for five exemplary settings.

3.5.1 Sequences

Sequence parameters that do not influence the computational effort are omitted in the description. Sequence descriptions can be found in Section 1.2.4.

Sequence	Configuration
EPI	• standard pulse • 64x64 resolution • single slice
SPGR	• standard pulse • RF spoiling • 128x128 resolution • single slice
TSE	• standard pulse • turbo factor 7 • 32 excitations • RF spoiling • 256x224 resolution • single slice
bSSFP 1	• standard pulse • 128x128 resolution • single slice
bSSFP 2	• 32 slices • (otherwise the same as bSSFP 1)

Table 3.12: Sequences used for effort estimation.

3.5.2 Method effort magnitude

The effort magnitudes, as outlined in Section 1.3.1.2, do not allow for a direct interpretation. As elaborated in Section 1.3.1, the O-notation can only describe the asymptotic behavior of an algorithm. Factors between different sub-tasks do not necessarily translate directly to run time. For instance, each operation that requires constant time is considered to be equivalent. This means that a data lookup, followed by a 4×4 matrix multiplication, then followed by one evaluation of the exponential function is considered as costly as one floating point multiplication—even though the latter operation is about fifty times faster.

However, a subtask that is clearly dominant by orders of magnitude, or a subtask that has constant effort for all sequences is a strong argument for this kind of comparison.

Direct Bloch simulation

The calculations in this paragraph are based on the workflow of estimate (see Section 3.1.1). One hundred spin packets per voxel are assumed to suffice to remove simulation artifacts and are thus chosen as the number of spin packets per imaging voxel.

number of operations	EPI	SPGR	TSE	bSSFP 1	bSSFP 2
sim. of gradient events	$3.18 \cdot 10^{8}$	$3.15 \cdot 10^{9}$	$2.04 \cdot 10^{10}$	$3.77 \cdot 10^{9}$	$3.22 \cdot 10^{12}$
sim. of RF pulses	$3.28 \cdot 10^{9}$	$1.68 \cdot 10^{12}$	$1.17 \cdot 10^{13}$	$1.68 \cdot 10^{12}$	$1.72 \cdot 10^{15}$
signal accumulation	$1.68 \cdot 10^{9}$	$6.71 \cdot 10^{9}$	$3.29 \cdot 10^{11}$	$6.71 \cdot 10^{9}$	$2.75 \cdot 10^{13}$
sim. of pulse-free periods	$4.10 \cdot 10^{5}$	$4.19 \cdot 10^{8}$	$2.94 \cdot 10^{9}$	0.00	0.00
total	$5.27 \cdot 10^{9}$	$1.69 \cdot 10^{12}$	$1.21 \cdot 10^{13}$	$1.69 \cdot 10^{12}$	$1.75 \cdot 10^{15}$

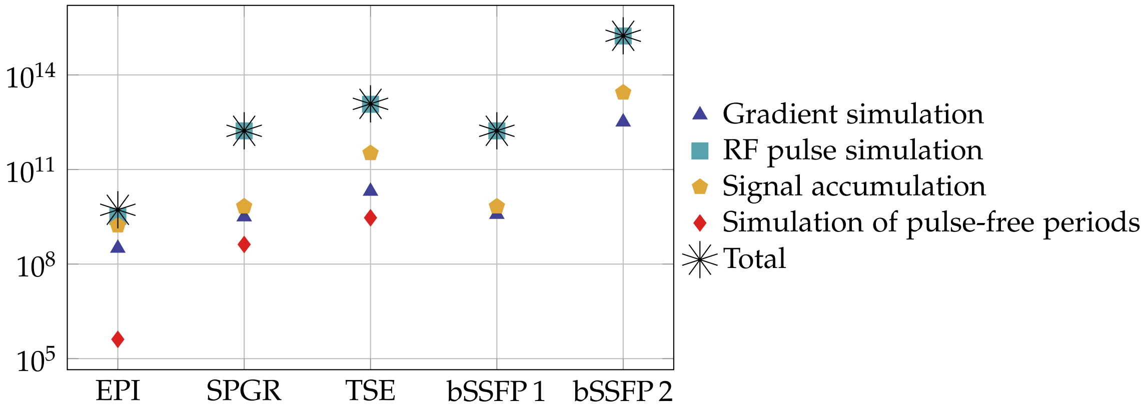


Figure 3.13: Direct Bloch simulation effort overview.

The RF pulse events clearly dominate the whole simulation process. Any solver needs to perform a high number of calculations when the magnetization changes drastically—which is during the RF pulses. Gradient segments are less in number than RF pulse samples and far easier to process in cylindrical coordinates. The simulations of time spans during which no pulses are applied are even easier and usually even fewer in imaging sequences. The bSSFP sequence for instance has none. The signal accumulation is the second most important effort contributor, and even on par with the pulse calculation effort for the EPI sequence.

Smart Bloch simulation

For the Smart Bloch solver, it is assumed that 128 unique tissues are sufficient to resolve the pulse effects. Also, 100 spin packets per imaging voxel are assumed, in agreement with the Direct Bloch solver estimate. The general workflow effort was calculated in Section 3.2.3.

The pulse precalculation effort is equal for the EPI, SPGR and bSSFP 1 sequences because they share the same number of pulses. The SPGR and bSSFP 1 sequences

number of operations	EPI	SPGR	TSE	bSSFP 1	bSSFP 2
pulse precalculation	$4.10\cdot10^{6}$	$4.10\cdot10^{6}$	$8.19\cdot10^{6}$	$4.10\cdot10^{6}$	$2.10\cdot10^{9}$
state at pulses	$4.10\cdot10^{5}$	$2.10\cdot10^{8}$	$1.47\cdot10^{9}$	$2.10\cdot10^{8}$	$2.15\cdot10^{11}$
signal accumulation	$1.68\cdot10^{9}$	$2.68\cdot10^{10}$	$3.29\cdot10^{11}$	$2.68\cdot10^{10}$	$2.75\cdot10^{13}$
total	$1.68\cdot10^{9}$	$2.71\cdot10^{10}$	$3.30\cdot10^{11}$	$2.71\cdot10^{10}$	$2.77\cdot10^{13}$

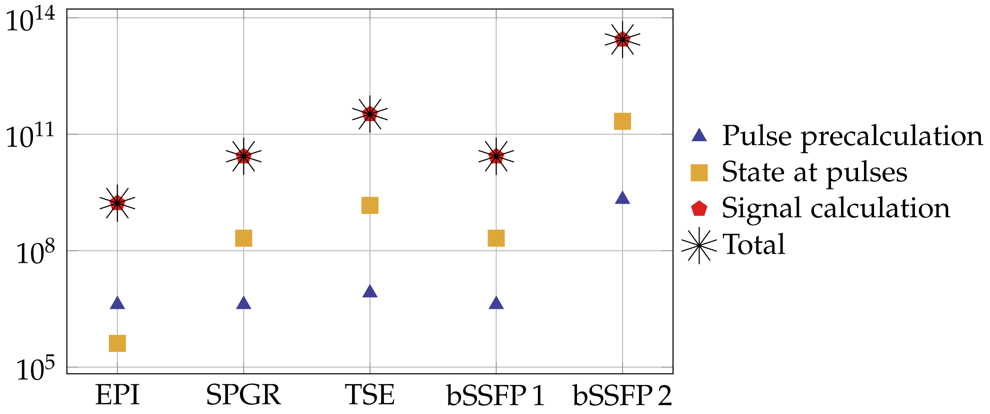


Figure 3.14: Smart Bloch simulation effort.

furthermore share all parameters that are relevant for the simulation effort and thus produce the same result. The second pulse of the TSE sequence doubles the pulse precalculation effort and the extra slice resolution significantly raises the effort for the bSSFP 2 sequence.

Most of the computations are very clearly performed in the signal calculation subprocess—consistently throughout all sequences.

k-space simulation

The k-space simulation in the following calculations enforces a truncation limit of 1000 and uses eight unique tissues. A higher number of tissues is used for the Sequence Response Kernel method in the next paragraph, which shares most of the sub-tasks with the k-space simulation. The workflow effort was estimated in Section 3.3.4.

number of operations	EPI	SPGR	TSE	bSSFP 1	bSSFP 2
pulse precalculation	$2.56\cdot10^{5}$	$2.56\cdot10^{5}$	$5.12\cdot10^{5}$	$2.56\cdot10^{5}$	$8.19\cdot10^{6}$
pathway template generation	3.00	$3.45\cdot10^{4}$	$7.68\cdot10^{5}$	$3.45\cdot10^{4}$	$1.15\cdot10^{7}$
pathway tree regeneration	$2.40\cdot10^{1}$	$1.31\cdot10^{5}$	$2.05\cdot10^{6}$	$1.31\cdot10^{5}$	$9.85\cdot10^{8}$
signal calculation	$6.55\cdot10^{4}$	$2.62\cdot10^{5}$	$9.18\cdot10^{5}$	$2.62\cdot10^{5}$	$2.68\cdot10^{8}$
total	$3.22\cdot10^{5}$	$6.84\cdot10^{5}$	$4.25\cdot10^{6}$	$6.84\cdot10^{5}$	$1.27\cdot10^{9}$

For this method, there is no clear dominance of a specific subtask. The signal calculation effort is mostly as high as the tree regeneration. This means that neither the pulse calculation nor the signal calculation is a bottleneck of the k-space simulation method—in contrast to the spin packet based simulations. The SPGR and bSSFP 1 sequences again yield the same effort.

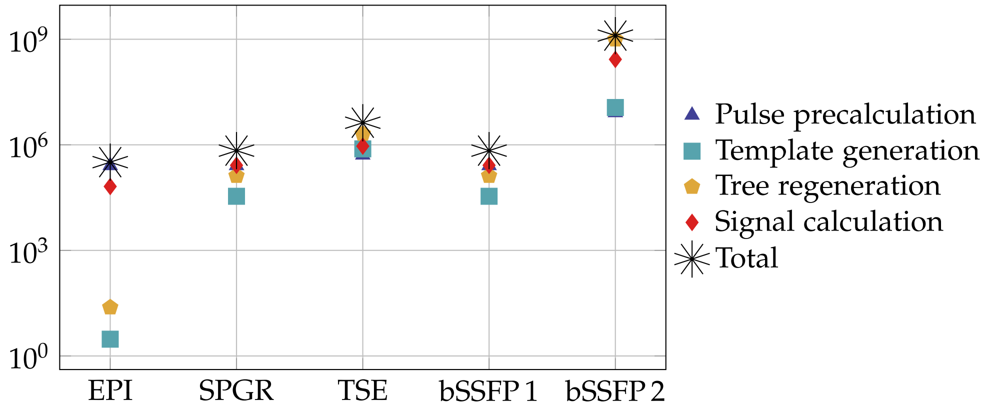


Figure 3.15: k-space simulation effort overview.

Sequence Response Kernel simulation

The efforts of the Sequence Response Kernel simulation are separated into the precalculation and the assembly part because multiple configurations are reasonable for full imaging sequence simulation. The precalculation can be performed sparsely—for eight different tissue configurations—or densely—for 65 536 different tissue configurations. The subsequent assembly step is not affected by this choice. The fast assembly uses completely truncated kernels or features that are extracted from the kernels, while the accurate assembly uses truncation limits of 64, 64 and 128 for shift, geometry and tissue convolutions respectively, as defined in Section 2.4.3 and estimated in Section 3.4.3.

number of operations	EPI	SPGR	TSE	bSSFP 1	bSSFP 2
pathway template generation	3.00	$3.45{\cdot}10^4$	$7.68{\cdot}10^5$	$3.45{\cdot}10^4$	$1.15{\cdot}10^7$
pulse precalculation sparse	$2.56{\cdot}10^5$	$2.56{\cdot}10^5$	$5.12{\cdot}10^5$	$2.56{\cdot}10^5$	$1.31{\cdot}10^8$
pathway tree regeneration sparse	$2.40{\cdot}10^1$	$3.77{\cdot}10^5$	$6.35{\cdot}10^6$	$3.77{\cdot}10^5$	$2.27{\cdot}10^8$
kernel calculation sparse	$7.07{\cdot}10^5$	$3.55{\cdot}10^6$	$1.50{\cdot}10^7$	$3.55{\cdot}10^6$	$3.85{\cdot}10^8$
pulse precalculation dense	$2.10{\cdot}10^9$	$2.10{\cdot}10^9$	$4.19{\cdot}10^9$	$2.10{\cdot}10^9$	$1.07{\cdot}10^{12}$
pathway tree regeneration dense	$1.97{\cdot}10^5$	$1.07{\cdot}10^9$	$1.68{\cdot}10^{10}$	$1.07{\cdot}10^9$	$2.52{\cdot}10^{11}$
kernel calculation dense	$2.21{\cdot}10^9$	$1.01{\cdot}10^{10}$	$3.95{\cdot}10^{10}$	$1.01{\cdot}10^{10}$	$4.28{\cdot}10^{11}$
total sparse	$9.63{\cdot}10^5$	$4.22{\cdot}10^6$	$2.26{\cdot}10^7$	$4.22{\cdot}10^6$	$7.54{\cdot}10^8$
total dense	$4.31{\cdot}10^9$	$1.33{\cdot}10^{10}$	$6.05{\cdot}10^{10}$	$1.33{\cdot}10^{10}$	$1.75{\cdot}10^{12}$

number of operations	EPI	SPGR	TSE	bSSFP 1	bSSFP 2
assembly fast	$4.10{\cdot}10^3$	$1.64{\cdot}10^4$	$5.73{\cdot}10^4$	$1.64{\cdot}10^4$	$5.24{\cdot}10^5$
assembly accurate	$4.67{\cdot}10^{10}$	$2.18{\cdot}10^{11}$	$8.61{\cdot}10^{11}$	$2.18{\cdot}10^{11}$	$9.46{\cdot}10^{12}$

The precalculation effort basically consists of k-space simulations and reconstructions. Both are rather exhaustive and have to be performed for each basis element of the tissue, geometry, and shift basis. The image assembly effort scales directly with the image resolution with a scaling factor of one for fast simulation or $1.14{\cdot}10^7$ for

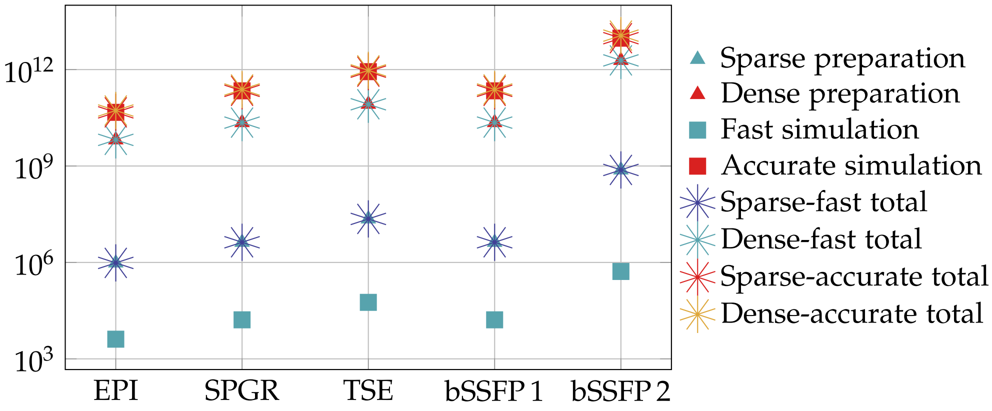


Figure 3.16: Sequence Response Kernel simulation effort overview.

accurate simulation. The full simulation effort is clearly dominated by either of the two processes, depending on the combination.

3.5.3 Conclusion

The direct Bloch simulation is dominated by the RF pulse simulation. The pulse precalculation of the Smart Bloch solver resolves that issue. However, both spin packet based solvers require the same high amount of operations for the signal accumulation at each acquisition time. This problem is circumvented by the k-space approach. But now the number of tissues that are present, and the pathway compatibility of the sequence influence the computational cost. The Sequence Response Kernel approach does not remove this effort, in fact the effort of the k-space method is still present and even extended by numerous reconstruction operations. But the Sequence Response Kernel approach separates this effort from the actual image assembly process. The amount of tissues that are present is not relevant for the image assembly step, it only depends on the chosen image assembly parameters and the image resolution.

The transition from Direct Bloch solver, to Smart Bloch solver, to k-space simulation, to Sequence Response Kernel simulation each removes the most computationally inconvenient sub-process to the point at which the computation time can be controlled and adjusted directly. However, the error nature and intrinsic inaccuracies of the methods are different, which is why the more advanced method may not always be the best choice.

This section only covered the use case of full image simulation. This may not always be of interest for a simulation experiment. A more detailed discussion from task perspective (see Section 4.1.2) and guidelines for selecting the optimal method (see Section 4.1.6) are part of the next chapter.

3.6 Derived algorithms

The byproducts of the Sequence Response Kernel approach can be used for algorithms that use precalculated simulation results, work on image space, and can be evaluated with a well-configurable accuracy-time trade-off.

This section applies the algorithms described earlier (see Section 2.5).

3.6.1 Sequence feature extraction

The sequence feature extractors (see Section 2.5.1) return a value that is meaningful in image space for a given sequence, reconstruction, tissue and geometry basis choice. Performing this process on a one-dimensional or two-dimensional parameter range yields images that are intuitive to read and capture the sequence nature.

The sequences that are associated with the following illustration (Figure 3.17) are:

- SPGR (Figure 1.5):
  TE $= 8\,\mathrm{ms}$, TR $= 20\,\mathrm{ms}$, base resolution: 256, flip angle: 15°
- EPI (Figure 1.6):
  TE $= 50\,\mathrm{ms}$, base resolution: 64
- bSSFP (Figure 1.7):
  TE $= 8\,\mathrm{ms}$, base resolution: 256, flip angle: 20°
- TSE (Figure 1.8):
  TR $= 1200\,\mathrm{ms}$, TE $= 8.5\,\mathrm{ms}$, TI $= 100\,\mathrm{ms}$, base resolution: 256, turbo 32, flip angles: 90° (excitation), 120° (refocusing), 180° (inversion recovery)

The following examples (Figure 3.17) illustrate the contrast, undesired signal and gradient features that were explained in Section 2.5.1.2.

The spoiled gradient echo sequence in this configuration is commonly referred to as T1 weighted. And indeed, the contrast information barely changes in T2-direction. However, it does change in the presence of extreme T2 values, which is clearly visible in the image. The error feature is also mostly unaffected by T2, low at small T1 values, maximal at about T1 $= 520\,\mathrm{ms}$. The image gradient image of all sequences is extreme for low T2 values because T2 always strongly determines signal intensity, but it also strongly increases as T1 decreases for this sequence.

The EPI kernels are completely T1 independent. The T2 contrast dependence is also rather low for values above the echo time (T2 $> 300\,\mathrm{ms}$). The error kernel is very pronounced at low T2 values due to the long echo train length.

The contrast information of the bSSFP sequence shows its signature T2/T1 contrast. The error kernel however has high values at low T1 values due to the long time it takes to reach a steady state.

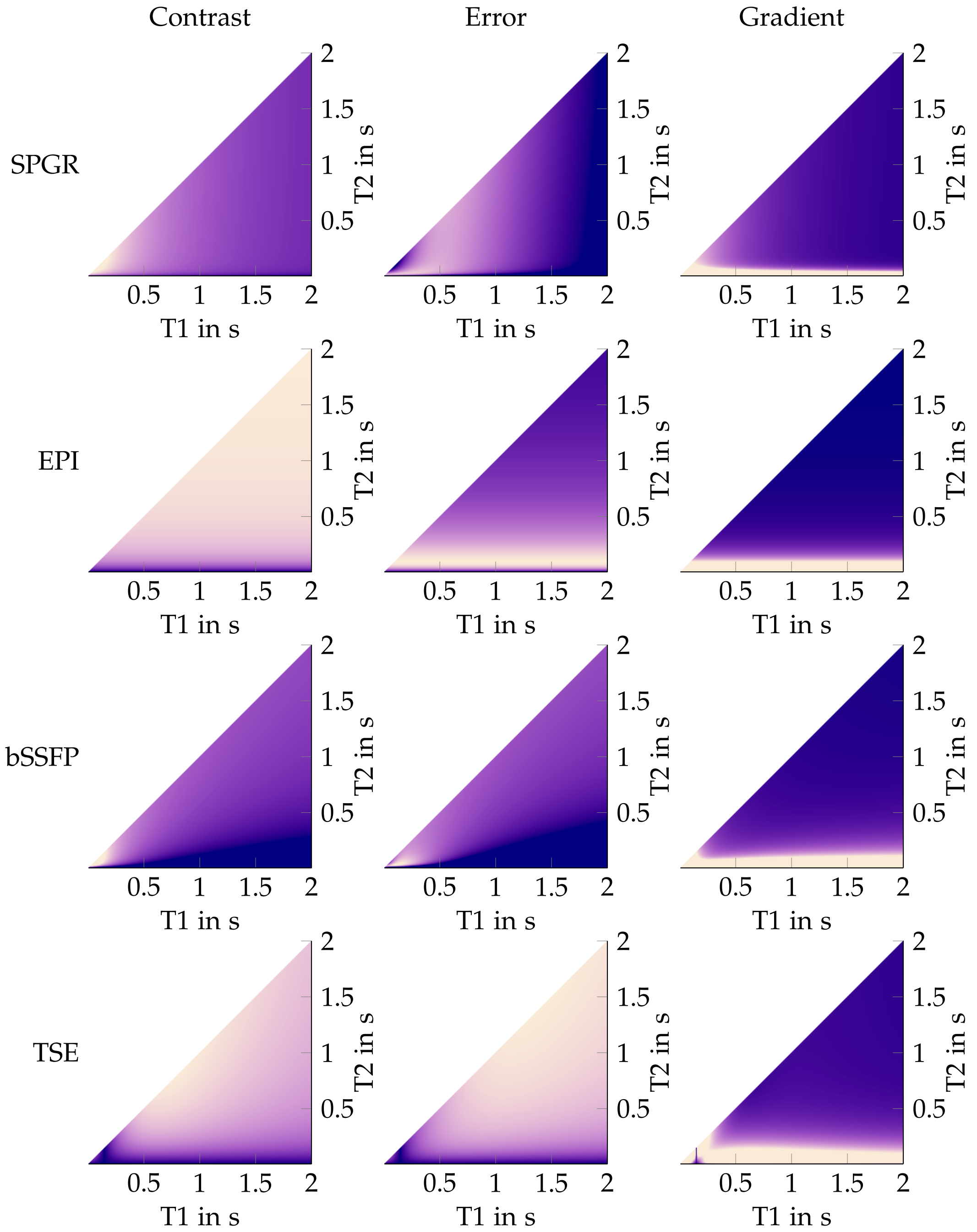


Figure 3.17: Extracted contrast information (signal at the central kernel position), total error (signal outside of the central kernel position) and image gradient magnitude features for some common sequences.

The IR TSE contrast kernel has the expected T1 time at which the recovery cancels out the magnetization (at $\mathrm{T1} = {}^{\mathrm{TI}}/_{\ln(2)} = 144\,\mathrm{ms}$). The TSE kernel behavior is hard to describe with words—supporting the usefulness of this view on a sequence. Interestingly, the image gradient is zero at a specific location on the T1-axis because the signal information has a plateau at this position.

3.6.1.1 Conclusion

Sequences are commonly given terms such as T1 weighted, even though no practical sequence can carry a pure T1 weighting. The actual essence of a sequence's weighting cannot be captured adequately by such decoupled properties. But on the other hand, a high dimensional mapping cannot be grasped directly. These kernel feature images—projections of these mappings—are easy to read and understand, but can still hold the most important information.

3.6.2 Sequence optimization

As described in Section 2.5.2, the kernels can also be used for a different purpose—as objective functions for sequence parameter optimization. In this example, kernel features were calculated for a small set of tissue parameters and reasonable ranges of flip angle and TR of a bSSFP sequence.

The objective function can then be assembled almost instantly—using nothing but a few standard image arithmetic operations. The tissues used for the following sequence optimizations are listed in Table 3.18.

Description	T1 in ms	T2 in ms	proton density
Tissue 1	800	80	1.0
Tissue 2	500	70	1.2
Tissue 3	500	70	0.87

Table 3.18: Tissue parameters used for the sequence optimization examples.

3.6.2.1 Maximization of contrast

Following Equation 2.78, the contrast objective function is the absolute difference between the two contrast kernels. This optimization is performed for two tissue combinations and illustrated in Figure 3.19.

The first objective function (a) has a clear optimum at a small TR and relatively high flip angle. The second example (b) shows a local optimum which is near the optimum of the first example, but the global optimum is at a lower flip angle and a substantially higher TR.

Even though Tissue 2 and Tissue 3 only differ in proton density, the resulting objective function and optima are very diverse.

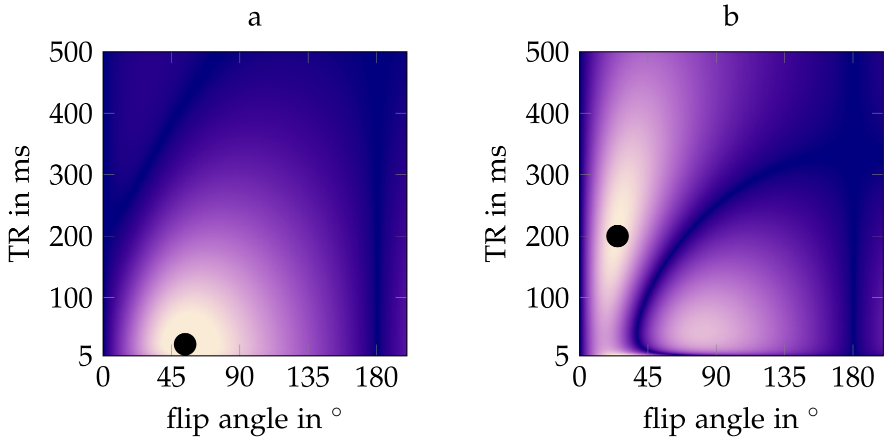


Figure 3.19: The contrast between hypothetical tissues. (a): Tissue 1 and Tissue 2. (b): Tissue 1 and Tissue 3. The optimal setting is highlighted.

3.6.2.2 Maximization of contrast-to-signal ratio

Following the objective function defined in Equation 2.79, the relative signal difference is optimized and illustrated in Figure 3.20.

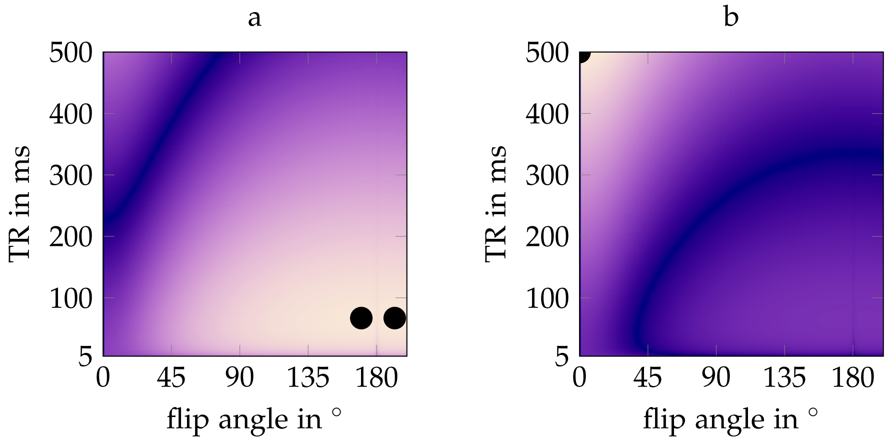


Figure 3.20: The contrast-to-signal ratio between the same hypothetical tissues. (a): Tissue 1 and Tissue 2. (b): Tissue 1 and Tissue 3. The optimal settings are highlighted. The first example has two global optima.

This example uses the same tissue configurations as the previous example, but the contrast-to-signal objective function produces fundamentally different results.

The first optimum (a) is shifted to a substantially higher TR and flip angle. Since all objective functions are axially symmetric around 180° flip angle due to the nature of the bSSFP sequence, a second global optimum is visible within the ranges of the diagram.

The second objective function (b) is optimal at minimum flip angle and maximum TR. This result is plausible because a small flip angle and long TR $= 1/2$ TE combination yields a high distinguishability in proton density and T2. However, the small flip angle also limits the total signal, which is why this result is not feasible.

3.6.2.3 Minimization of undesired signal

Following Equation 2.80, weighted total absolute error kernels were used as an objective function, illustrated in Figure 3.21.

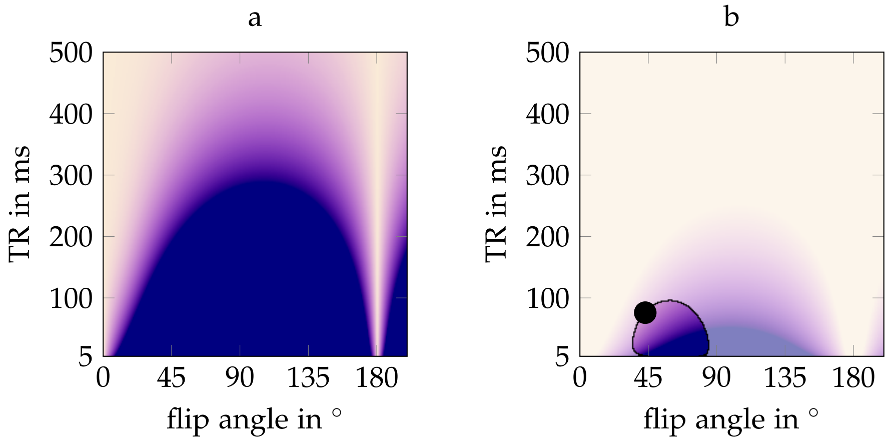


Figure 3.21: The negative undesired signal. (a): Equally weighted Tissue 1 and Tissue 2. (b): Equally weighted Tissue 1 and Tissue 2, restricted to the domain that corresponds to at least 85 % of the optimal objective function value of Figure 3.19a.

For this example, the undesired signal information of Tissue 1 and Tissue 2 is equally weighted. The two diagrams show the same objective function with different LUTs.

The undesired signal is self-evidently zero whenever the total signal equals zero, which is at 0° and 180° flip angle for this sequence type. It is therefore not reasonable to directly optimize this feature directly.

In the second image (b), an area is highlighted which corresponds to at least 85 % of the optimal value of Figure 3.19a. Restricting the optimization of undesired signal to this area yields a reasonable result at its boundary. This optimum has a lower flip angle and higher TR in comparison with Figure 3.19a.

3.6.2.4 Simultaneous optimization of run time and contrast

As defined in Equation 2.81, the objective functions can be enhanced by introducing further influencing factors, in this case the sequence run time, illustrated in Figure 3.22.

For this example, run time is directly proportional to TR and has a linear influence on the objective function.

As the time influence factor is increased, the optimum sequence parameters from Figure 3.19b=3.22a first switch to the formerly local optimum (b) until they eventually become optimal at the lowest possible TR (c).

3.6.2.5 Conclusion

Through the extracted kernels, sequence parameters can be optimized without sequence-specific simplifications and without the need for computationally exhaustive simulations during the optimization process. The objective function can be adjusted easily and sequences that use different approaches to acquire the image could be compared.

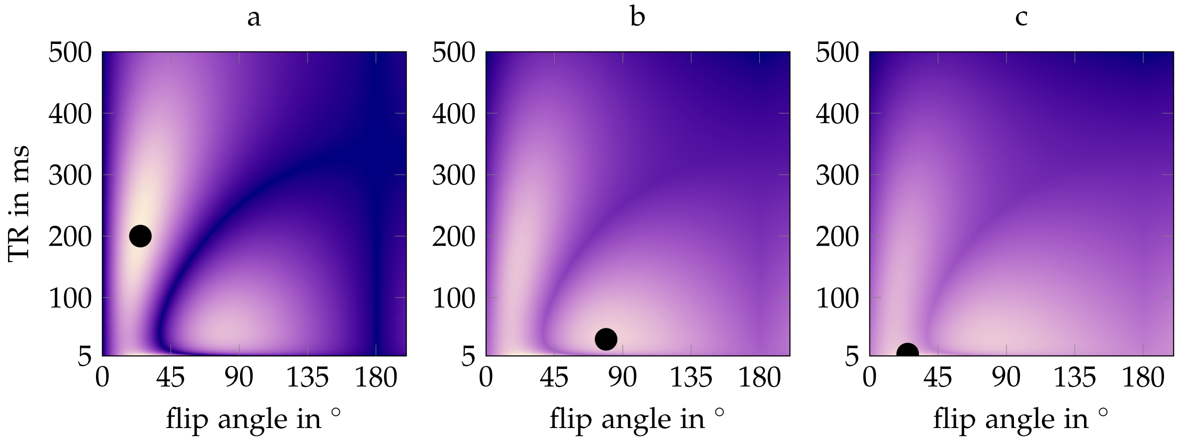


Figure 3.22: Contrast optimization similar to Figure 3.19b, extended by a sequence run time penalty in (b) and (c).

3.6.3 Model-free parameter mapping

A set of 30 brain images were acquired using a set of TSE sequences with diverse settings (Table 3.29). The sequence parameters were not chosen according to a particular scheme, but following intuition with the goal of producing perceptually heterogeneous contrasts.

Sequence Response Kernels were then calculated for a reasonable parameter range and the standard geometry basis. For this parameter-mapping experiment, they were truncated to size one.

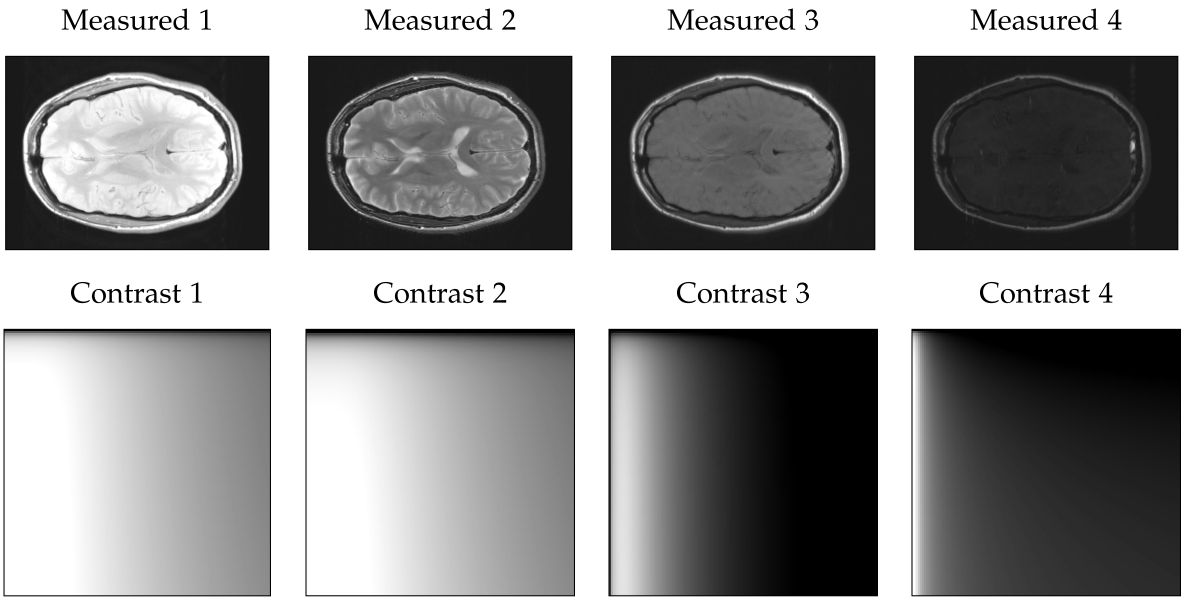


Figure 3.23: Acquired images and extracted contrast feature maps in the ranges T1 $\in [0\,\text{s}, 4\,\text{s}]$, T2 $\in [0\,\text{s}, 4\,\text{s}]$. All images are rendered using the same LUT.

Together with the proton density as a proportionality factor, the extracted kernels form a 3D signal model for each image. These signal models are then combined to form a voxel-wise objective function (see Section 2.5.3). In this example, the Simulated

Annealing algorithm was used for the optimization (see Section 2.5.3.3). The result is shown in Figure 3.24.

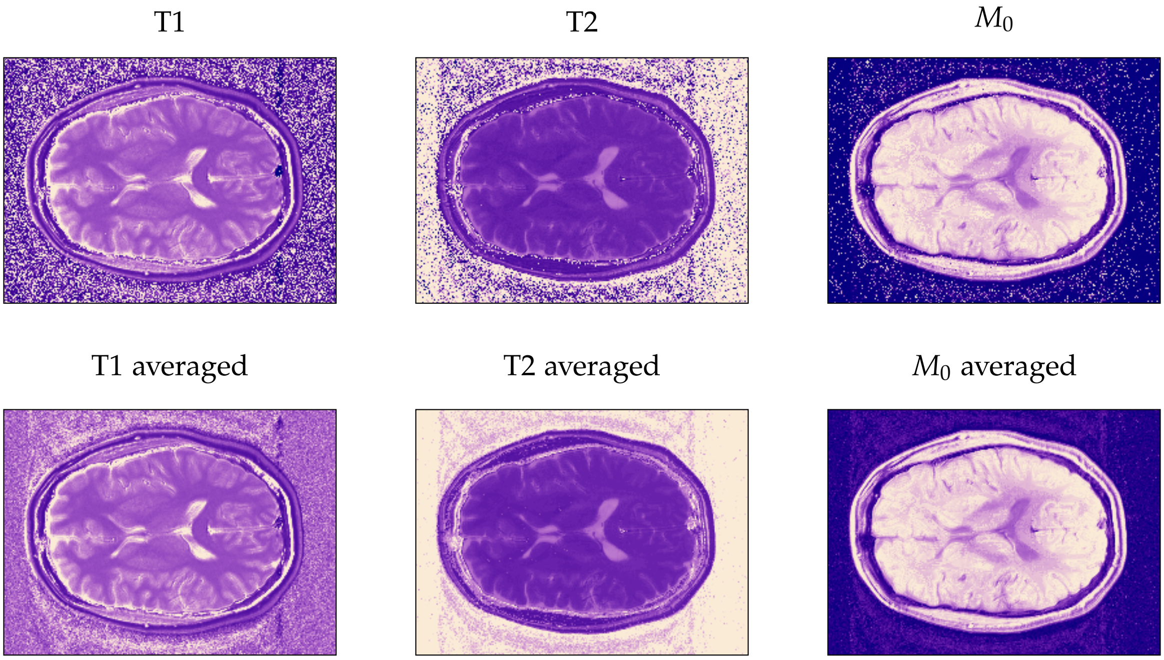


Figure 3.24: The result of the parameter mapping process (top row). Forty averages of the parameter mapping (bottom row).

Outside of the object, there are some voxels that have a substantial proton density, but those voxels also possess a very low T2 value. This is an expected ambiguity. Those parameter combinations are not biologically meaningful, but are in full agreement with the signal models.

Some artifacts in the phase encoding direction (top/down) can also be found. They are a direct consequence of the image artifacts that some of the images experienced. Particularly noteworthy are the flow artifacts that are induced by the sinus sagittalis. The rest of the results appear reasonable.

The average over forty optimization repetitions reassures the reproducibility. The random noise at the voxels that are poorly distinguishable by the objective function are due to the Simulated Annealing algorithm. The averaging removes this noise and thereby gives hints about the noise properties. Further investigation of the fitting error is carried out in the next section.

A forward simulation back into image space was used as an example in a previous section (Figure 3.11). That forward simulation allows a direct comparison between simulated and measured images. The following diagram (Figure 3.25) displays the cumulated joint histogram of all measured and simulated images. The histogram shows some spread and a multitude of clusters, which is due to the overdeterminedness of the system and the presence of distinct tissue types, but it gives no indication of a bias.

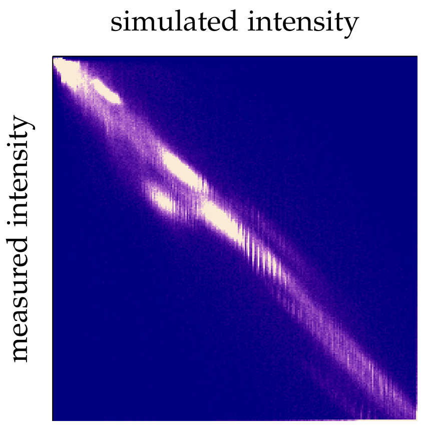


Figure 3.25: Joint histogram of all measured and simulated images

3.6.4 Error characterization

An indicator function can be used to characterize the error volume in tissue model parameter space (see Section 2.5.4). In this example, the indicator function is chosen to represent the subspace for which the objective function value is below the 1 % threshold of possible objective function values.

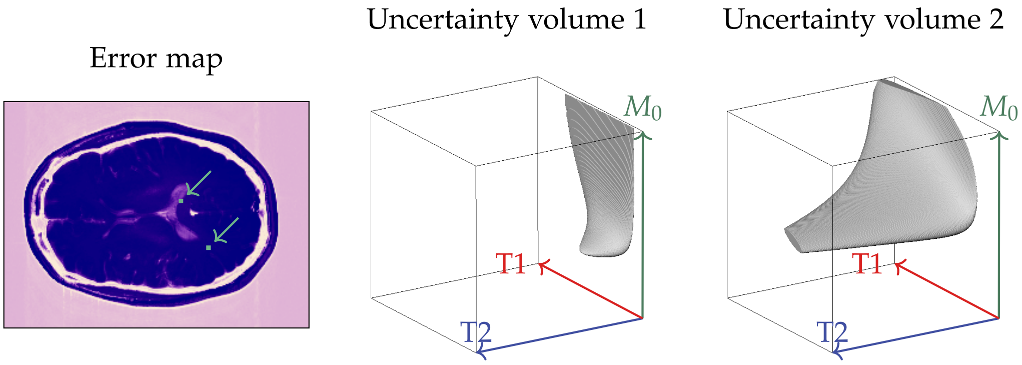


Figure 3.26: Uncertainty volume map and uncertainty shapes. The first image depicts the voxel-wise uncertainty volume. The next two images depict the uncertainty shape within the parameter range box. (uncertainty volume 1) corresponds to the bottom right marked voxel in (error map), (uncertainty volume 2) corresponds to the top left voxel.

The uncertainty volume image highlights the voxels that are hard to parameterize for the objective function. Background and skull are particularly hard to distinguish for the objective function—which is in agreement with Figure 3.11.

The uncertainty shapes at well-behaved positions give hints about parameter combinations that cannot be ruled out easily. Standard deviations that can only approximate boxes or ellipsoids do not suffice.

3.6.5 Information density optimization

As described in Section 2.5.5, the error volumes are used to iteratively remove the image that the fitting objective function least benefits from. The same dataset as in the previous sections was used for this example. The following diagram shows a subset of the results (Figure 3.27).

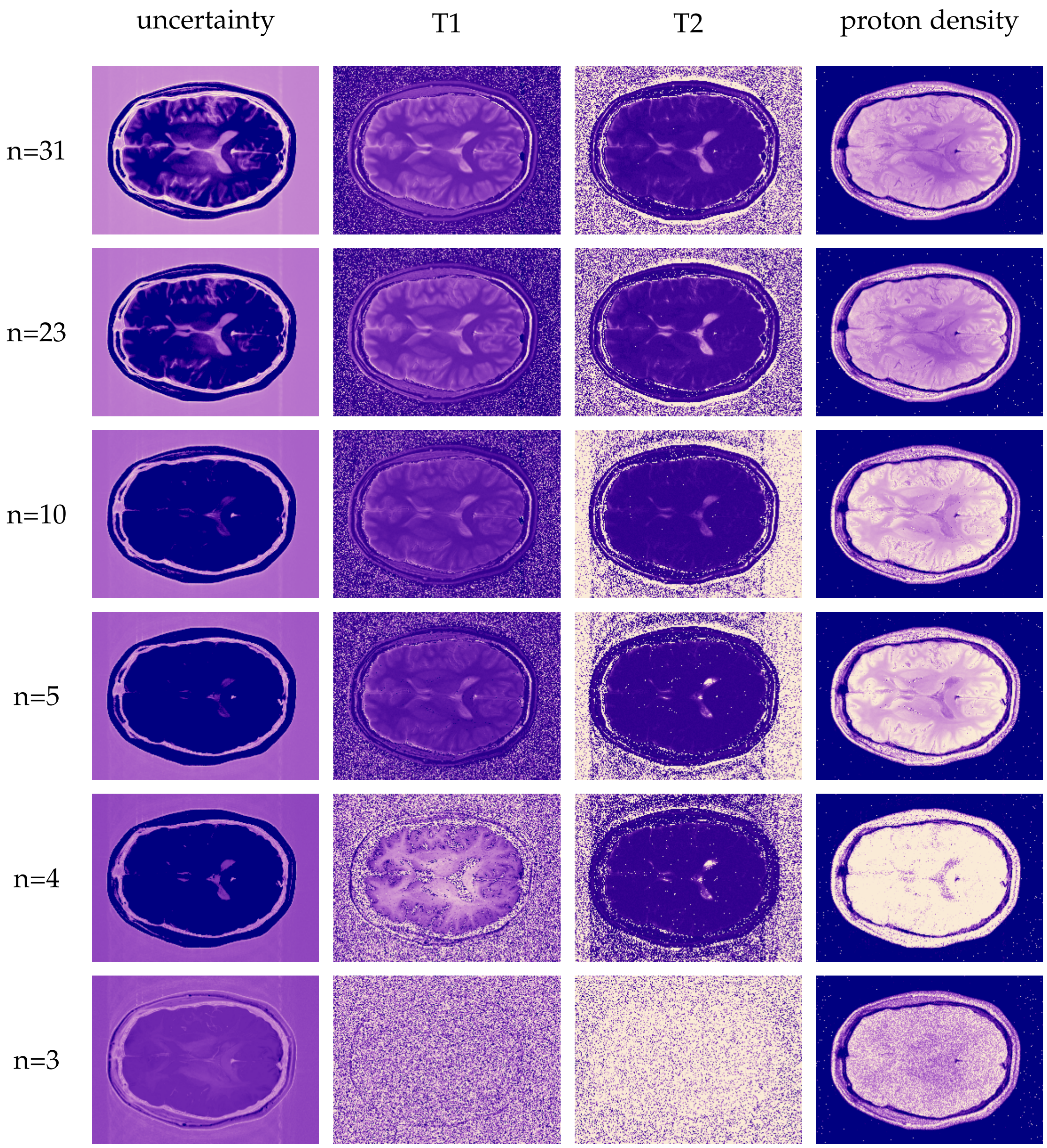


Figure 3.27: Uncertainty volume, fitted T1, fitted T2 and fitted proton density using the most important 31, 23, 10, 5, 4, 3 images according to greedy error volume elimination.

As more sequences are removed, the overall uncertainty decreases at first, but starts to increase at $n = 10$. Likewise, the visual quality of the fit is equal for $n = 31$ and $n = 23$, but starts to deteriorate at $n = 10$—most notably at positions with high relaxation times.

The algorithm favors the elimination of sequences that introduce extreme and error-prone behavior, thereby improving the overall error volume map, thus the first third of the eliminated images did not noticeably change the parameter map. But the algorithm also favors sequences that reveal the properties of tissues with extreme parameters such as the cerebrospinal fluid or the skull. Therefore, elimination of the second third of all sequences does not affect the residuals that are created by the last third—since those last sequences are not particularly sensitive to those extreme tissue parameters. Using the last five remaining sequences still provides a passable parameter map, but further elimination renders the optimization insufficient except for a rough skull segmentation. Figure 3.28 shows simulated images using the parameter maps originating from a reduced set of sequences.

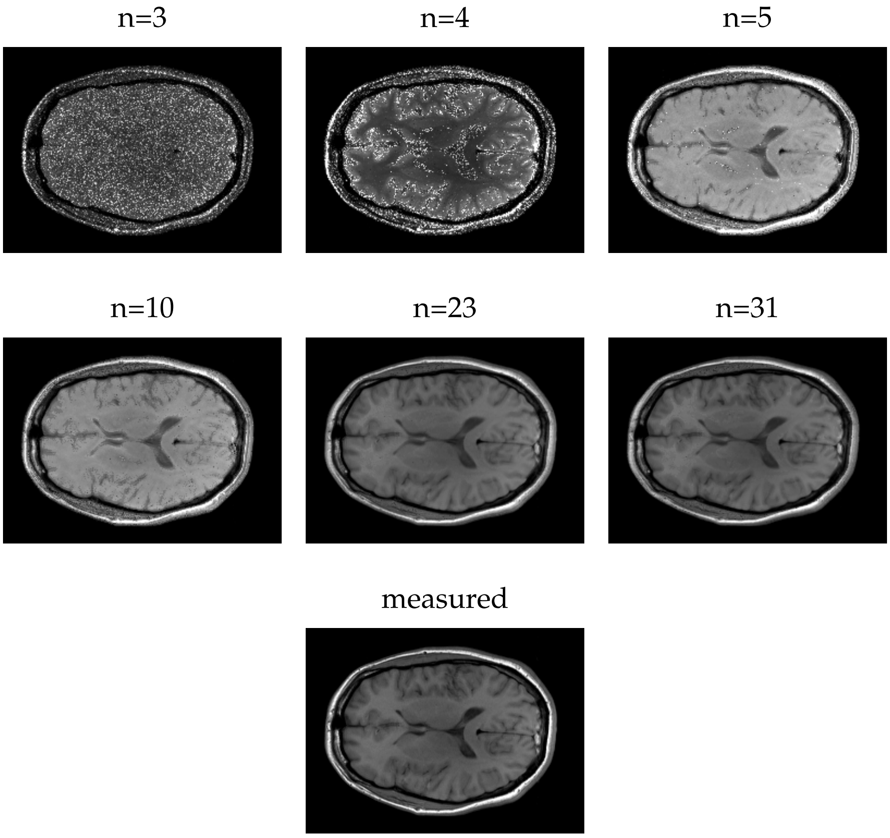


Figure 3.28: Forward simulation of sequence 22 (see Section 3.29) using the 3, 4, 5, 10, 23, 31 most important sequences. The measured image is provided for reference. The LUTs are equal. Forward simulation allows for easy comparison with the simulation to validate the parameter map.

The four most important sequences (see Table 3.29) are indeed remarkably suitable for the parameter mapping process. The two most important ones are mostly equal, except for the maximally different echo time. The third most important sequence introduces inversion recovery, a different turbo factor, short TR, and a different refocusing flip angle. The fourth most important sequence introduces a maximally different TR.

These four sequences are partially similar, so they produce similar artifacts, but also different in the parameters that produce distinguishable contrast.

Number	TF	TE	TR_{refoc}	TR_{exc}	TI	α_{refoc}	IR	Rank
1	34	60	8.57	1200	900	120	yes	29
2	34	60	8.57	1200	900	150	yes	24
3	34	60	8.57	1200	900	180	yes	20
4	34	60	8.57	1200	900	180	no	16
5	34	60	8.57	1200	900	150	no	12
6	34	60	8.57	1200	900	120	no	27
7	7	12	12	3000	100	180	no	22
8	7	12	12	2000	100	180	no	17
9	7	12	12	1000	100	180	no	4
10	7	12	12	500	100	180	no	3
11	7	12	12	200	100	180	no	28
12	7	25	12.5	3000	100	180	no	31
13	7	12	12	3000	100	180	no	21
14	7	37	12.33	3000	100	180	no	26
15	7	50	12.5	3000	100	180	no	19
16	7	62	12.4	3000	100	180	no	23
17	7	75	12.5	3000	100	180	no	25
18	7	87	12.42	3000	100	180	no	30
19	7	12	12	3000	500	180	yes	18
20	7	12	12	3000	700	180	yes	5
21	7	12	12	3000	900	180	yes	8
22	7	8.5	8.5	1200	900	180	yes	15
23	7	8.5	8.5	1200	900	180	yes	10
24	7	8.5	8.5	1200	900	180	yes	11
25	7	8.5	8.5	1200	900	180	yes	13
26	7	8.5	8.5	1200	900	180	yes	24
27	7	12	12	500	100	180	yes	7
28	7	60	8.57	1200	900	180	yes	1
29	7	60	8.57	1200	900	160	yes	2
30	7	60	8.57	1200	900	140	yes	6
31	7	60	8.57	1200	900	120	yes	9

Table 3.29: Table of sequence parameters used in the TSE simulation experiments. The rank reflects the importance of the sequence and was calculated using greedy elimination (see Section 2.5.5.2), a high rank means late elimination. TE corresponds to the time until the center sample is acquired. TF is the turbo factor and the IR column indicates whether inversion recovery was employed. All times are in ms.

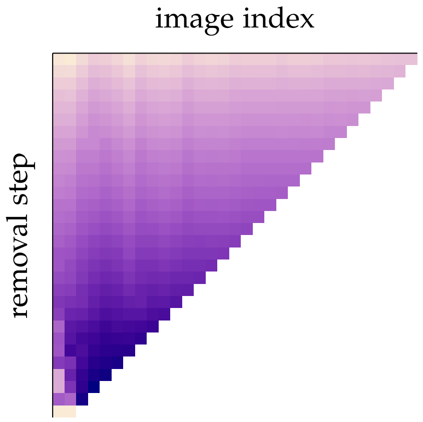


Figure 3.30: Uncertainty volumes as each remaining individual image is left out at each greedy removal iteration. The sequences are ranked by their final importance on the horizontal axis—meaning that each rightmost image is removed. The vertical axis reflects the iteration. The sequences behave mostly consistent for most of the process. The right half of the image does apparently not change in order—the brightness decreases in horizontal and vertical direction. The left half shows clear stripes, meaning that the importance is similar in relation to the other sequences during most of the time. The lower third of the image is less structured because the optimization is pushed beyond its reasonable well-posedness.

3.6.6 Concluding comments

The main argument of this section was that the Sequence Response Kernel approach opens doors to novel algorithms that are applicable to many sequences, easy to understand due to their connection to image-space, and most importantly very compatible with contemporary efficient computer science data structures and algorithms. It is also noteworthy that despite the stochastic nature and the high number of objective function evaluations of the optimization algorithm that was used throughout this section, the computational power of a common scientific workstation was absolutely sufficient. Further discussion of derived algorithms can be found in the following chapter (see Section 4.1.1.5).

CHAPTER 4

Discussion

4.0 Introduction

Now that the simulation methods are described, justified, and exemplified, they can be discussed more generally. The goal of this work is to find efficient and accurate means to simulate full MR imaging experiments. No tool that is currently available is able to handle this task adequately or even attempts to do so. Therefore, the procedures that are defined in the methods chapter should be considered the main result of this work. The common approaches, namely the Direct Bloch equation simulation (see Section 2.1) and extended phase graph algorithm [34], have been used as a starting point and were then further developed to suit the task. That development gave rise to the Sequence Response Kernel approach (see Section 2.4), which is a completely novel and very task-centered way to perform MRI simulation. The potential of this approach was showcased in some derived algorithms (see Section 3.6).

The third chapter's primary function was to support the design choices that were made in the methods chapter. Some simplifications are only reasonable in certain contexts, and barely computationally manageable otherwise. Likewise, some approximations are perfectly reasonable only if used within their limits of applicability. These details were discussed argumentatively in the methods chapter, and then further exhibited through simulation examples in the results chapter. Another part of the third chapter was to estimate the effort of the individual methods and put them into a common context (see Section 3.5).

The first section of this chapter discusses technical aspects of this work. First, further extensions of the algorithms and possible support of additional physical phenomena is discussed (see Section 4.1.1). The next part (4.1.2) focuses on the task perspective and discuss the best method for a typical simulation problem. Afterwards, the influences of the imaging sequence 4.1.3 and the phantom that is used for the experiment 4.1.4 are elaborated. The next part reveals the limits and verification between the methods (see Section 4.1.5). The last part of the section (4.1.6) summarizes the applicability of the individual methods.

The next section of this chapter discusses more general aspects of this work. The simulation objectives that were described in the introduction chapter (see Section 1.4) are revisited, evaluating the impact of the methods of this work. This is followed by an outlook, possible further development, and requirements for optimal integration (see Section 4.3).

Technical discussion 4.1

4.1.1 Further development

The simulation approaches can be supplemented by advanced features to support non-static tissues and some system imperfections. This topic will be addressed briefly in this section, followed by comments about feasible further development of the algorithms that were derived from the Sequence Response Kernel approach.

4.1.1.1 Dynamic object properties

The methods of this work were described for static objects. Certain dynamic properties, such as flow, diffusion and time-dependent tissue parameters are compatible with some of the approaches. The following paragraphs will describe the necessary extensions of the individual algorithms.

To support time-dependent object parameters, the only difference in the Bloch equation (Equation 1.4) for a given spin packet is, that location and tissue parameters need to be time-dependent. This time-dependence does not not affect the solving algorithm.

The Smart Bloch approach takes advantage of the Bloch equation's properties. This is partially hindered by object time-dependent parameters.

The pulse effects need to be calculated according to the new degrees of freedom that the dynamics yield. These new degrees may not be relevant on the timescale of the RF pulses and can potentially be simplified, analogous to the other degrees of freedom (see Section 2.2.3).

The relaxation and phase behavior of a spin packet can still be treated independently because they commute, regardless of their time dependence. However, their computation is more complex. Time-dependent relaxation times require an evaluation of $\int \mathrm{T2}(t)^{-1}\,\mathrm{d}t$ and $\int \mathrm{T1}(t)^{-1}\,\mathrm{d}t$, which may not have an analytical solution for advanced models. Deterministic motion of a spin packet affects the phase accumulation that is induced by the gradient pulses. The required evaluation of $\int r(t) \cdot G(t)\,\mathrm{d}t$ is potentially hard to simplify because it mixes the sequence-specific gradient behavior with the object-specific motion of the spin packet. A polynomial motion model or an approximation by polynomial segments can resolve this problem, but requires higher-order gradient moments, e.g., for uniform flow:

$$\int \overrightarrow{r}(t) \cdot \overrightarrow{G}(t)\,\mathrm{d}t = \int (\overrightarrow{r_0} + t\overrightarrow{v_0}) \cdot \overrightarrow{G}(t)\,\mathrm{d}t = \overrightarrow{r_0} \cdot \int \overrightarrow{G}(t) dt + \overrightarrow{v_0} \cdot \int t\,\overrightarrow{G}(t)\,\mathrm{d}t. \tag{4.1}$$

In case of a non-deterministic motion model, the probabilistic behavior of the spin packets can either be sampled to then be treated deterministically, or the Bloch equation can be extended. The most common extension is the Bloch-Torrey equation [36], which incorporates diffusion effects. The extra terms of the Bloch-Torrey equation can be treated independently of the other effects and evaluated similarly to the phase effects.

Unlike former two methods, the k-space simulation does not apply to spin packets directly, but is applied to sub-objects with similarly treatable behavior. The operators of the Smart Bloch solver that are required for dynamic object properties need to be transformed to be applicable to the set of k-space simulation basis elements. This can be done for polynomial motion models and unrestricted diffusion and has been combined with the extended phase graph approach [39]. However, this may be impractical if the models possess many degrees of freedom. In particular, motion that is defined through a spatially varying flow field is not compatible with the k-space approach, resembling the incompatibilities of arbitrarily varying relaxation times (see Section 2.3.4.3).

The Sequence Response Kernel approach is based on the calculations of the k-space simulation, hence it needs to be prepared for an extended set of parameters to reflect the dynamic object properties. But since polynomial motion models cause a change of phase in signal space that is independent of the observed tissue, it can be expressed as an additional motion-dependent factor in signal space. It thereby becomes an extra convolution kernel in image space with behavior that can be investigated and applied independent of the tissue, geometry, and shift kernels. Similarly, diffusion and further advanced effects may be introduced with ease to the Sequence Response Kernel approach if it can be expressed as a multiplication in signal space. However, if the motion model affects the pulse effects, such as flow perpendicular to slice direction in the presence of slice selective pulses, then the tissue kernels still need to be calculated for the respective combinations of pulse effects.

4.1.1.2 RF imperfections

The methods of this work can be extended to support RF coils with non-homogeneous transmissivities and sensitivities, as defined in the specialized Bloch equation (Equation 1.4).

RF transmissivity can be treated as a tissue parameter throughout all algorithms. The Smart Bloch solver then requires pulse precalculations that cover the present RF transmissivities, the k-space method requires objects that experience homogeneous RF transmissivities, and the tissue kernels of the Sequence Response Kernel approach need to be extended by the according extra dimension.

Inhomogeneous RF sensitivities introduces a weight on each spin packet, depending on its location (see Section 2.2.2.1) for the Direct Bloch solver and Smart Bloch solver. In frequency space, this multiplicative weighting can be obtained by convolving the object geometry with the Fourier transform of the weighting map. Depending on the complexity of the weighting, incorporating the effect each time signal is measured is likely to be a huge strain on computation time. A precalculation of the coil-weighted Fourier transformed proton density of each specific sub-phantom might be the best choice to improve this inconvenience, but imperfect coils are mostly not efficiently suitable for k-space based simulation unless the convolution can be performed analytically. The Sequence Response Kernel approach can introduce sensitivity directly as a scaling factor in image space that depends on the position of the corresponding basis element. This is equivalent to introducing it as a tissue parameter that directly weights the signal, which is independent of the other tissue parameters and can therefore be treated separately.

RF imperfections can be combined with dynamic object parameters, but the dimensionality of the k-space and Sequence Response Kernel algorithms increases correspondingly.

4.1.1.3 Gradient imperfections

If the gradient coils produce a non-linear field, then the image will be distorted accordingly. If this nonlinearity can be corrected for by inversely distorting the object, then all algorithms are still applicable. The Sequence Response Kernel then requires a more intricate shift and geometry basis, and the object needs to be decomposed into distorted volume elements.

Gradient imperfections of this kind can be combined with RF imperfections and dynamic object parameters, if both can be distorted accordingly.

4.1.1.4 Fractional shifts in the Sequence Response Kernel approach

In this work, only integer-valued voxel shifts due to offset frequency were introduced. Non-integer chemical shifts are common, but unfortunately not directly compatible with the Sequence Response Kernel approach. They can be supported by not storing tissue kernels with direct image voxel correspondence, but instead first identifying an optimal fractional shift, for which the entropy of the kernel is minimal. This extra offset information can decrease the necessary kernel size by minimizing discretization errors and could also be used as a kernel feature for chemical shift. Similarly, the Sequence Response Kernel approach can be extended to support voxels that are not in perfect correspondence with the geometry basis.

4.1.1.5 Further development of derived algorithms

The derived algorithms described in Section 2.5 were only briefly exemplified in Section 3.6. They could be further refined to become more robust, more efficient or more applicable to relevant problems. The following paragraphs will address the feasible further development of the individual examples.

Model-free parameter mapping

An optimization algorithm which requires less function evaluations than Simulated Annealing is a natural improvement of the tool.

Furthermore, Sequence Response Kernels can be used for non-single voxel optimization, which is particularly interesting for offresonance frequency influence in a bSSFP sequence, because the steady state develops slowly, and also interesting for sequences with long echo trains that result in intricate blurring. The computational burden of this idea is quite high because the evaluation of the objective function takes disproportionately more time due to memory management inconveniences, because the voxels cannot be fitted individually.

It could also be beneficial to use a subset of the images to fit some of the parameters, a different subset to fit the others, and repeat the processes until all parameters converge. This may treat higher-dimensional tissue model parameter spaces efficiently.

Error characterization

Many indicator functions are feasible. Depending on the application, it may be beneficial to choose an indicator function that is adapted to objective function properties, the reasonable tissues, or the neighboring voxels. These possible choices also motivate more refined fitting techniques.

Information density optimization

The greedy elimination approach is not an algorithm that is guaranteed to reach the best-performing subset of sequences. But on the other hand, testing all possibilities is not computationally manageable. Further numerical analysis could modify the uncertainty measure to be more efficient to evaluate for a given subset of images. It could also be reasonable to define a redundancy metric to assess image pairs based on a ground truth parameter map.

The elimination process could also be modified to insist on certain sequences that are part of a fixed protocol, or to keep sequences with a particularly important kernel feature behavior to ensure certain information, such as structure or T2.

Furthermore, the algorithm was merely applied to a single sequence type in the example of this work. Conjoint applicability beyond the sequence type is one of the main advantages of the Sequence Response Kernel approach, and was not carried out to its full potential in this work.

4.1.2 Task perspective

There cannot be a single simulation method that is a reasonable choice for all simulation tasks. This section casts light on the applicability of each method from task perspective.

4.1.2.1 RF Pulse simulation

When the task is to investigate the effect of a single RF pulse, and in case it is reasonable to assume that the MR physics related parameters influence the actual pulse effect, there is no way around actually solving the Bloch equation with a contemporary ODE solver.

Using the Bloch equation without any simplifications or numerical adjustments other than polar coordinates and a rotating reference frame is perfectly applicable, and even unavoidable for very involved pulses. When the pulses are long, have a big amplitude and have nontrivial gradients applied simultaneously, this might even be the most investigative approach to answer simulation task related questions.

The Smart Bloch solver might be a valuable choice, since it yields additional information without imposing restrictions to the Bloch equation. One added benefit is that the effect is captured in one matrix, opposed to requiring an initial condition (see Section 2.2.1).

The instantanization correction of the Smart Bloch solver (see Section 2.2.1.3) helps compare the dependence of the experiment parameters on the pulse effect. The effects that arise because the pulse has a temporal extension are separated from the effect that the actual pulse application has. Often times, relaxation, flow, diffusion, and slight offset frequencies do not alter the pulse effect significantly and can be ignored or in- and extrapolated, which is more apparent using instantanization (see Figure 3.2).

The other simulation methods are not supposed to investigate the effects of single pulses—they even require the pulses to have no temporal extent and utilize the Smart Bloch Solver approach to correct the physical pulses accordingly.

4.1.2.2 Pulse pattern simulation

Pulse patterns such as the spin echo, three pulse experiment or simple repeating pulse patterns are the prototypes of actual MR sequences and need to be treated differently for efficient and accurate results. In most cases, the actual generated signal is not as important as understanding the magnetization state of the system.

Solving the Bloch equation directly is still a valid approach when certain points of interest are respected and programmed to not be skipped as it would be the case for a naïve implementation using adaptive step size.

There is no reason other than developing convenience to not make use of the Smart Bloch solver approach, when the Bloch equation is used in its MRI formulation (Equation 1.4). The added effort of calculating the pulse effects is at most a factor of four for the time spans of pulse applications, but yields the benefit of potentially reusing simulation results. The stepping effort of the Direct Bloch simulation where no RF pulses are active is replaced by one step while also removing possible integration inaccuracies of the ODE solver. The worst case scenario of arbitrarily changing spin packet parameters can be handled with greater ease in the Smart Bloch Solver environment, as the required integrals can be investigated directly, opposed to multiple times on the fly.

The Smart Bloch Solver still requires single spin packets to be treated individually. In many applications this is not necessary because they form classes of similar behavior. If only one tissue parameter combination is of interest, or if the tissue parameters are a discrete set, and only density, orientation and position differ, then k-space formalism approach may be the best way to understand the RF pulse sequence pattern. The k-space formalism requires the pulse sequence to be well-behaved for echo analysis. In general, the number of Echo Pathway Segments depends exponentially on the number of RF pulses. Likewise, the type of pulses in the sequence might increase the dimension of classes that need to be treated separately, effectively degenerating the efficiency of this approach. When the k-space formalism is used to simulate signal, the spatial information of the resulting image is algorithmically perfect, which is particularly important when the sequence contains crushing or spoiling methods, since they manipulate the magnetization on a sub-voxel basis.

The k-space simulation also provides supplementary information in form of echo behavior, which is in many cases superior to direct spin packet magnetization information.

The Sequence Response Kernel approach is based on the k-space formalism but also requires a reconstruction pipeline. Since pulse patterns do not produce images, the approach cannot be properly used to understand pulse patterns.

4.1.2.3 Imaging sequence simulation

Imaging sequences are simulation tasks that are linked strongest to the goals of MRI. Further details about this task will be explained in later sections of this chapter.

The direct Bloch solver is impractical for realistic imaging sequences. Even simplified versions of realistic imaging sequences can only be handled with HPC support. This is mainly due to the high number of required spin packets, the many required signal accumulations and the many integration steps needed.

The Smart Bloch solver can be used whenever the direct Bloch solver is applicable. But the general problem intrinsic to spin packet approximation is the same. The only part that can be remedied by the Smart Bloch solver is the number of steps needed per spin packet for the magnetization calculation. The only reason to fall back to the direct Bloch solver is convenience or necessary inclusion of advanced physical models.

The k-space formalism removes the number of spin packets required for spatial of offset frequency resolution and replaces it with classes of similar echo behavior. This can be a great speedup if the phantom supports it. But each class needs to be treated separately, and the echoes that theoretically need to be accumulated grows exponentially. But it is usually not desired from a sequence development standpoint to have many significantly contributing echoes. In conclusion, the k-space approach can be used most of the time and promises a great speedup.

The Sequence Response Kernel approach transforms the signal to image space and assembles images based on, e.g., a voxel basis. This only works if the k-space formalism is applicable and the echo-behavior is consistent. But if that is the case, then all calculations can be performed in image space with arbitrarily adjustable accuracy. Providing a different view on the sequence and sequence complexity-independent image calculation given sufficient precalculation.

4.1.3 Sequence influences

The nature and patterns of sequences largely influence simulation efficiency. The complexity influence on each simulation method differs—some aspects that do not influence the efficiency of one method might practically break another method. This section addresses imaging sequence properties and resulting method-specific ramifications.

4.1.3.1 Influence of gradient pulses applied during RF pulses

As elaborated in Section 2.2.1.2, gradients that are applied during the RF pulse's duration alter the way spin packets react to the pulse, depending on their location.

The Direct Bloch solver will not be affected by extra pulse complexity since no information about the pulses is required by the method.

The Smart Bloch solver on the other hand attempts to reduce the dimensionality of the pulse effect and will be hindered if complicated gradients are applied. For each direction in which the applied gradient is non-constant, the frequency-position coupling breaks and an extra dimension of pulse simulation is necessary. The efficiency of the Smart Bloch approach decreases that way, yet by design it cannot be considerably worse than the Direct Bloch solver.

The k-space based simulation requires the spin packet classes to possess equal pulse effects and spin packet properties. Firstly, this separates the spin packets along the frequency direction and the selective pulse directions independently. If all pulses are applied without gradients, no additional separation is required. If slice selective pulses are used to select parallel slices, then classes need to be additionally separated along the slice selection direction. However if slice-selective pulses are not parallel, the classes need to be split in multiple directions, which degenerates the k-space approach since each class practically corresponds to a voxel or sub-voxel, which is easier simulated using former approaches. It does not make a difference for the k-space simulation approach whether the pulses have non-constant gradients applied alongside or constant-gradient pulses are applied in different directions, the resulting class separation is the same and will practically break the k-space approach.

The Sequence Response Kernel approach depends on a clear echo structure within the k-space simulation approach. If the sequence contains pulses that are selective in different or multiple directions, then the k-space approach degenerates and the sequence response kernel approach can also not be readily used. Both the k-space method and the sequence response kernel method can potentially still be used if the pulse effects can be idealized, but this requires further application-specific assumptions.

4.1.3.2 Influence of RF pulse duration

If the pulse's duration is at least a tenth of the relaxation times, then it can be considered a long pulse. Long pulses are often connected with elaborate spatial or spectral selection, introducing the problems of the previous paragraphs. But independent of the concurrently applied gradients, the following complications arise.

The direct Bloch solver will be not affected since it does not require any higher level information about the sequence. The step size of the integrator will however suffer for elongated time spans.

The Smart Bloch solver attempts to reuse already simulated pulse effects. This should involve appropriate interpolation where applicable. The relaxation constants are mostly irrelevant for short pulses, but need to be considered for long pulses. Since pulse effects can be reused, the step size problems of the Direct Bloch solver are not significant.

The k-space and Sequence Response Kernel approaches utilize the pulse effects from the Smart Bloch solver. Since the phantom already needs to be separated into classes of differing relaxation times, long pulses do not introduce further algorithmic facets. Heterogeneous pulse effects might emphasize the Echo Pathway transition factors in considerably heterogeneous ways, but the relaxation that occurs between the pulses is likely to overshadow this aspect.

4.1.3.3 Influence of Rapid pulse succession

The spacing between pulses impacts the magnetization state strongly. Pulses that are far apart allow for more relaxation to occur, allowing the system to reach a more stable state. But this stability is usually deliberately sacrificed for better measurement time. The term rapid in this paragraph signifies the relation of pulse distance to relaxation time—a pulse spacing on the order of the relaxation time is considered a rapid pulse succession.

The Bloch solver effort is mostly linked to the number of sequence events, not the spacing of the sequence events—thus making it immune to rapid pulse successions. The accuracy might suffer because the numerical errors have less time to decay between the pulses.

The Smart Bloch solver possesses an essentially equal immunity to pulse spacing, and a slightly superior error behavior since analytical solutions are used between the pulses.

The echo behavior strongly depends on the pulse spacing. Each Echo Pathway decays partially with T1 and T2 after its first split. During that decay, it is further split into three sub-pathways at each pulse. If many pathways stay important and cannot be discarded during the pathway generation process, then the k-space approach becomes inefficient. This is most problematic when the utilized pulses show mixed excitation and refocusing behavior—optimally problematic for 90° flip angles. A hard limit on the number of pathways and reasonable merging should solve this problem for common pulse sequences, but if the amount of discarded pathway energy is not satisfactory, it is likely that additional information about the sequence can improve the simulation quality, e.g. preferred preservation of refocused pathways or removal of all pathways before each excitation. This may appear to be a drastic limitation of the k-space approach, yet the inaccuracies due to higher-order echoes are unapproachable in former two methods for realistic computational conditions.

The Sequence Response kernel calculation is generally as complex as the k-space formalism. The presence of many segments is no theoretical obstacle for the approach, but the segment grouping might be non-trivial for complex echo formation and may need manual adjustments to become reasonable. A clear echo formation and segment grouping is crucial for the Sequence Response Kernel approach and needs to be fulfilled for the simulation method to be numerically feasible. Notably, the image generation process does not suffer from these problems if they can be resolved.

4.1.3.4 Influence of gradient pulses applied between RF pulses

The 0th moment of the gradient pulses that are applied between the RF pulses have the strongest influence on the simulation methods. They can be classified according to their influence as follows:

- Balanced:

  The 0th moment between two pulses equals zero.

- Constant:

  The 0th moment between two pulses is constant.

- Discrete:

  The 0th moment between two pulses is a multiple of a basis moment vector.

- Arbitrary:

  The 0th moment between two pulses does not fall into former categories.

Higher-order moments become relevant when object motion is considered (see Section 4.1.1.1).

The gradient moments do not influence the Direct Bloch solver or Smart Bloch solver, even though the intention behind this method often aims at effects that in turn require a drastic increase in simulation resolution. The k-space and Sequence Response Kernel algorithms on the other hand behave differently. In general, the complexity of the calculations for these two method directly scales with the number of Echo Pathway Segments.

Suitable gradient moments place Echo Pathway Segments on top of each other in terms of wavenumber. This allows for pathway merging—soothing the otherwise exponentially growing complexity. If offset frequency is not of interest, then all pathways will keep the same wavenumber in the balanced case. This echo behavior is not very interesting but might help investigate k-space trajectories when the echo formation is not as relevant. The number of pathway segments stays constant in this case.

If offset frequency is of interest, then balanced gradient moments behave the same way as constant gradient moments if the pulse spacing is constant (see Section 2.3.4.2). In that case, the number of present pathway segments grows linearly.

If the gradient moments are multiples of a basis moment, then the possibility of some segments falling on top of each other persists. This scenario is often present, for instance at sequences that use advanced spoiling or crushing schemes. If the offset frequency is of interest and the pulses are equally distanced nonetheless, then it can be expected that few pathway segment wavenumbers are going to match.

If the gradient moments do not follow one of the former schemes, then it can be expected that no pathways will be mergeable. In that case, the number of pathways grows exponentially and the k-space formalism is only reasonable if the pulses do not succeed rapidly with respect to the relaxation times, because pathway discarding is necessary.

The gradient behaviors may mix—in that case the complexity grows according to the currently employed gradient behavior. Some sequences prepare the magnetization with heterogeneous moments between preparation pulses, but are followed by a very structured readout section. In that case, the number of segments first grows exponentially, but much slower afterwards.

The Sequence Response Kernel approach is again only reasonable if the k-space formalism behaves nicely. Yet the complexity of the calculations does not necessarily carry over to the image space—even though many segments might be needed to calculate the magnetization states accurately, the signal generating Echo Pathway Segments do not need to be plentiful or posses complicated kernel representations. As a consequence, a sequence may be well-treatable in image space using the Sequence Response Kernel approach, even though it required a high computational effort in the precalculation phase.

4.1.3.5 Non-Cartesian readout

The readout is linked to the gradients that are applied in-between the pulses, thus it has no effect on the Direct or Smart Bloch solver.

The k-space based simulation is also not affected by the readout method. The existing Echo Pathway Segments assume different positions in k-space as they are measured.

The Sequence Response Kernel cannot be readily used in case of non-Cartesian readout. The reconstruction is usually not a mostly linear operator, breaking the approach immediately. Further investigation of the reconstruction methods and its effects on image space need to be performed.

4.1.3.6 Multiple slices

If the sequence acquires slices in a successive manner, then it might be most productive to simulate the acquisition of only one slice and then transfer the results. Some methods are more susceptible to this strategy than others. This section assumes the presence of slice selective pulses and the absence of multi-dimensionally selective pulses.

It can only be assumed that the acquired slices look similar if the initial conditions before each slice are equal. This cannot be validated easily using the direct Bloch solver because supplementary information is not readily available due to the generally raw nature of the sequence description.

The Smart Bloch Solver can be modified to help evaluate this requirement. If the magnetization states between two slice acquisitions are sufficiently similar for given respective slices, then it may be reasonable to assume that the slices can be treated independently. This, however, is a very manual approach that requires careful consideration to rule out effects that are hidden, e.g. due to heterogeneous tissue parameters or higher order echoes.

The k-space approach is well-suited to investigate the equality of initial conditions for each slice acquisition: If the Echo Pathway Segments present at the beginning of each new slice acquisition are sufficiently similar for the individual slices, then each slice will be acquired the same way. This has to be tested for pathway trees within reasonable tissue parameter ranges.

The Sequence Response Kernel approach is the most approachable way to test if successive slices are acquired the same way since the comparisons can be performed directly in image space on kernel basis—circumventing the manual definition of similarity criteria on complex data structures in k-space or magnetization. It also directly yields a similarity measure of the slices in image space.

When the independence of slice resolution is proven, it is a good idea to fall back to simulation of single slices or few slices. The generalization back to multiple slices should be done in image space if required—which is easiest using the Sequence Response Kernel approach.

4.1.3.7 Resolution

Increased resolution requires readout modules to evaluate further parts of k-space, and can thereby influence the number of pulses that are required or the resolution at which the simulation has to be performed.

The Direct Bloch solver and Smart Bloch solver require a higher number of spin packets. The higher gradient moments that are linked to the higher wavenumbers in the traversed k-space need to be resolved spatially. The simulation artifacts can be held constant if the number of spin packets per voxel stays the same. The higher number of pulses does not introduce further effects.

The k-space approach does not require a higher resolution due to the different readout because its spatial resolution is intrinsically infinite. But the potentially increasing number of pulses increases the number of Echo Pathway Segments, which may decrease the efficiency or accuracy in case of rapid pulse succession.

The Sequence Response Kernel approach shares its complexity with the k-space approach. But the transition to image space allows for additional resolution-dependency analysis. A fall-back to lower spatial resolutions may be justifiable based on Sequence Response Kernel analysis. Also, the image assembly accuracy can still be controlled freely—independent of the imaging resolution.

4.1.4 Phantom influences

For any MR imaging sequence, the description of the object that is to be simulated may be well or ill-suited for a specific simulation method. This section describes possible natures of phantoms and their impact on the simulation methods.

4.1.4.1 Analytical phantom with discrete tissue properties

An analytical phantom with discrete tissue properties is suited optimally for all simulation methods. The formal conditions can be defined as:

- Only few discrete tissue properties are present.
- The tissue composition at a specific location can be evaluated quickly.
- The Fourier transform of each component density at any frequency can be evaluated quickly.

Phantoms of this kind usually consist of simple geometric objects, e.g. ellipsoids or cuboids. These types of phantoms is of high importance because they are described independent of the image space that is used to acquire the signal. The edges of the sub-objects may mismatch the image resolution and thus generate related artifacts—which is considered in this phantom representation.

The spin packets that are required by the Direct and Smart Bloch solvers can be sampled spatially within the simulation domain in a randomized or resolution-specific way. At each sampled location, a spin packet needs to be spawned for each present tissue component. After the spin packets are selected, the Direct and Smart Bloch approaches do not rely on the phantom description any more, which is unique to these simulation approaches. Especially when the phantom becomes more complex and the evaluation of the geometry ceases to be quick, the Direct and Smart Bloch solvers are mostly unaffected.

The k-space simulation requires the Fourier transformed proton density for each tissue type. Each Echo Pathway Segment that contributes to the signal does so proportional to the echo intensity and the Fourier transformed proton density at the segment's wavenumber. The tissue types are treated independently, thus the effort scales with the number of different tissues. Tissue overlap is irrelevant in contrast to former two solvers. But the Fourier transform can easily become computationally expensive, e.g. by adding multiple ellipsoids to a single tissue class.

The pulses that are present in the sequence might also require a sub-classification along the slice position. This also scales the computational effort linearly.

The Sequence Response Kernel approach requires a set of basis geometries for image calculation. This set of basis geometries might be defined considering the phantom geometry, for example the same set of ellipsoids that are used in the phantom. But it would be closer to an actual application to use a basis that matches the voxel size that is used for the imaging process.

The phantom is otherwise mostly ignored in the precalculation process. The image assembly process however requires the phantom to be sampled in accordance with the basis geometries. All effects that arise from a mismatch of the geometry basis with the phantom components cannot be represented—most prominently edges that do not align with the basis. This cannot be resolved because the basis also needs to be chosen to be well-suited to the sequence resolution to produce sparse kernels. Therefore if those effects are of interest, the Sequence Response Kernel approach needs to be avoided.

4.1.4.2 Fuzzy voxel-based phantom

A phantom may also be defined on a voxel basis while still containing a discrete number of tissue types. This type usually originates from registration of anatomical images. A prominent example is the Brainweb phantom set [6]. For this type of input data, sub-voxel information is lost. The missing partial volume effects cannot be recovered.

In comparison with the analytical phantom:

- Only few discrete tissue properties are present.
- The composition of the tissue at a specific spatial point can be evaluated quickly.
- The Fourier transform of each component density cannot be sampled adequately.

The Direct and Smart Bloch approaches are in no way hindered by this phantom simplification—The Fourier transform is not needed at all.

The k-space approach is problematic for this kind of input data, since the k-space positions that are required to calculate the signal do not necessarily agree with the available k-space positions resulting from the discrete Fourier transform. The phantom needs to be continued reasonably, which can be done by assuming no presence of extra spins packets outside of the field of view, and linear interpolation between the voxels. Even this relatively simple model requires an expensive sinc-interpolation in theory, which can be approximated by falling back to the Kaiser-Bessel window. In conclusion, this type of phantom is mostly incompatible with the k-space approach.

The Sequence Response Kernels on the other hand are a perfect fit for this type of phantom, despite the fact that it is based on the incompatible k-space method. The geometry basis can be chosen to suit the phantom voxels, thereby trivializing the basis decomposition process. The spatial distribution of the phantom only needs to be considered in the image assemply stage. As an added bonus, the use cases that revolve around fuzzy voxel-based phantoms usually consider many phantoms in the same image space and concern images created with a shared sequence. Therefore the precalculations can be reused.

4.1.4.3 Parameter map

This type of phantom is a further generalization:

- There is a large number of tissue properties present.
- The composition of the tissue at a specific spatial point can be evaluated quickly.
- (The Fourier transform of the object geometry does not matter.)

This type of phantom usually arises from a parameter mapping process, which often involves voxel-specific fitting.

The Bloch approaches are again not hindered by this phantom type—however the Smart Bloch solver might require more pulse precalculations if the varying tissue constants are relevantly different.

The k-space approach is not applicable in this case. All available spin packets need to be separated into classes, and those degenerate in the presence of many tissue properties. These classes will not posses a meaningful Fourier transform anymore and can thus not be used for reliable signal calculation.

The Sequence Response Kernel method however may be an optimal choice. If the tissue property ranges are simulated with sufficient resolution, then the virtually infinite tissue properties can be interpolated using a manageable set of simulation results. If the parameter map then furthermore follows a voxel basis, then this basis can be used to the advantage of the method, as explained in the previous phantom type. In fact this approach is so efficient that it can be used for the inverse process as exemplified in the derived algorithms sections of this work.

4.1.4.4 Conclusion

Simulation experiments on imaging level usually involve a phantom of the introduced kinds.

The Direct Bloch solver and Smart Bloch solver are always applicable because they require the least phantom properties.

The k-space approach can easily be broken by even slightly more involved phantoms, which is the main reason why it is not used in scientific practice to simulate natural images. It is a great tool to help understand the workings of a sequence, but it can almost solely cope with abstract phantoms types that can be described comfortably in frequency space.

The Sequence Response Kernel approach shows great potential in terms of phantom compatibility. Firstly, it is very applicable to voxel-wise defined phantoms which comprise the majority interesting phantoms. It can circumvent the phantom geometry restrictions that break the k-space approach by introducing its own set of basis geometries, yet still preserves the echo-based computation efficiency that makes the k-space approach superior to the Bloch methods. Lastly, the need for discrete tissue classes can be partially lifted by employing interpolation.

4.1.5 Limits and verification

Each simulation method has its own strengths and oftentimes there is an obvious candidate for optimal handling of the simulation problem. They follow a rough hierarchy and have overlapping regions of applicability. This is particularly interesting because it enables verifiability and testability of the simulation methods. For this it is necessary to push the methods to their limits to reach a common denominator.

This section will address the places where the methods meet, focusing on the cases that are particularly interesting for cross-validation and testing the limits of the simulation approaches. It also emphasizes the aspects for which there is no single true answer to the problem because the involved methods introduce individual systematic imperfections.

4.1.5.1 Direct Bloch solver

The Bloch equation is the most pure and general description of MR-physics related processes. It is easy to employ, easy to understand and easy to adapt to arbitrary new physical phenomena. It can be considered a ground truth for simulation purposes.

In order to build a solid foundation for more advanced simulation methods, the Bloch equation solver can be tested against analytical solutions that arise from analysis of special pulse sequences. A mere testing against the Hahn Echo experiment or the free induction decay does likely not suffice for the coverage of real-world imaging sequences.

The artifacts of the direct Bloch simulation usually arise from two possible sources:

The first source is the solver itself. The ODE is usually not continuously differentiable, resulting in solution inaccuracies. A pure ODE solver is not readily compatible with the right hand side of common MRI sequences. Multiple timescales are present through prolonged distances between the pulses or long constant gradients. The extra information about the change of pace needs to be communicated to the solver, and this part is very prone to numerical or implementation errors.

The second artifact source is the discretization. The Bloch equation is defined for single spin packets. Discretizing a continuous object insufficiently or inadequately introduces systematic errors that could be hard to isolate. Constant spacing of the samples can easily cause aliasing artifacts. Randomized sampling is a simple fix for this issue, but it hinders reproducibility of the results. But even when the sample distance is chosen to be extremely low, the effects that are induced by the sequence could have an even finer resolution.

Therefore, sampling without considering the scale of the magnetization effects causes systematical errors intrinsic to the approach.

4.1.5.2 Smart Bloch solver

The Smart Bloch solver is essentially an enhancement of the direct Bloch solver. It does not suffer from the ODE solver issues described in former paragraphs since it uses analytical solutions at the problematic sections. The cost of this improvement is that it cannot be used if the analytical solutions for those sections are not available, for instance in the presence of time-dependent relaxation or complicated motion. In these cases an additional model for the Smart Bloch solver would be required.

However, due to the analytical approach and the ceasing approximation influence, testing the Bloch solver against the Smart Bloch solver is sufficient on an elemental basis that consists of the unique building blocks of imaging sequences. If both methods agree on this level, then the results of the Smart Bloch solver can be considered superior to those of the direct Bloch solver.

The discretization issues of the Direct Bloch solver apply in the same way.

4.1.5.3 k-space based approach

Comparing and validating the k-space based approach in a simulation environment cannot be done directly because the calculations are performed in different spaces. But since the derivation of the formalism is based on the Smart Bloch solver, any intermediate result can be imprinted onto sampled spin packets. The waves that are dictated by the Echo Pathway Segments can be superposed, followed by multiplication with the spatially sampled spin packets. The other direction however is not readily possible due to the overdeterminedness of the target space.

Comparing the signal of the different approaches cannot be expected to provide perfectly agreeing data, because the k-space formalism does not suffer from discretization errors. If sampled randomly with a very high number of spin packets, it can be expected that the signals are stochastically equal. And if multiple Echo Pathway Segments are actively contributing to the signal, then the k-space approach is more exact.

This however only holds if the phantom description in spatial and frequency space can be considered equally accurate. The k-space approach limits the phantoms that can be used for simulation. Phantoms that do not possess a Fourier description that can be separated by tissue cannot be handled adequately or suffer from conversion errors.

4.1.5.4 Sequence Response Kernels

The Sequence Response Kernels require a valid k-space simulation algorithm and a compatible reconstruction pipeline. Both can be investigated independently. Both may reveal slight incompatibilities with the assumptions of the approach. The relevant tissue ranges can evoke unsuitable echo behavior for the given sequences and thereby cause unacceptable error properties or insufficient sparsity of the kernels. The reconstruction pipeline might include operators that introduce expensive kernels into the image assembly process, or might even be incompatible altogether.

Those mentioned limits can be easily measured in image space and extracted as fitness factors of the Sequence Response Kernel approach. Together with the approximation level that is chosen for the experiment, the deviation can be evaluated by testing the result against the k-space formalism with a matching geometry and tissue basis.

Further validation is possible by comparing with the Bloch Solver result. This validation is best done using the final images. Interestingly, potential candidate phantoms for this kind of validation are parameter maps with continuous parameter spaces which are not compatible with the k-space formalism. This validation spans over a very wide range of algorithmic steps with no possibility of investigating intermediate results, but the simplicity of the Bloch Solver is a huge argument for the power and usefulness of this alternative validation. However, in addition to the error that is introduced by truncating the kernels, an additional discretization error is present due to the choice of the basis.

4.1.6 Guidelines for selecting the optimal method

The following questions help ruling out specific simulation methods to emphasize the methodological limits of the approaches. The questions are roughly ranked by their ability to distinguish between the applicability of the simulation methods.

Is the experiment based on an extension of the Bloch equation? (As defined in Equation 1.4)

If the Bloch equation needs to be modified to cover a more general case or specialized model, then all higher level methods are potentially broken and need at least a thorough theoretical investigation of the newly introduced features. Adapting the Direct Bloch solver to the deviations of the specific experiment is likely to be the only convenient choice.

Are spin packet properties other than magnetization time-dependent?

Depending on the complexity of the spin packet behavior, the Smart Bloch solver may be partially usable or fully adaptable to the behavior. If the dynamic properties are not relevant on RF pulse timescales, then the RF pulse calculation methods of the Smart Bloch solver can be readily used. For instance, this is the case when flow experiments are investigated where the excitation is not tailored to the expected flow velocities, but rather relevant to the readout module.

The analytical solution of the spin packet behavior between the pulses may be found for slightly advanced time-dependent behavior, such as flow that follows a polynomial function or isotropic diffusion.

Further theoretical investigation is also needed for the k-space and Sequence Response Kernel methods. But even if the operators can be translated to the according space, the k-space approach may still be impractical if the classes of similarly behaved spin packets turn out to be small. The Sequence Response Kernel is still promising if the added complexity can be expressed either as a separate tissue class dimension or as a factor in the signal equation.

Are gradients non-constant in multiple directions during the application of single RF pulses?

Each linearly independent direction adds a degree of freedom to the pulse-effects and also an extra dimension of required echo class separation. The echo class separation dimensions are likely to break the applicability of the k-space formalism and the Sequence Response Kernel approach.

The Smart Bloch Solver is not affected in theory, but the effort reduction for pulse simulation will diminish.

Does the reconstruction include non-linear operators at the beginning of the reconstruction pipeline?

Each reconstruction step requires a careful theoretical analysis to suit the Sequence Response Kernel method. Especially non-linear operators at the beginning of the reconstruction pipeline can make the Sequence Response Kernel approach largely inefficient or even non-applicable. Thus, if advanced reconstruction methods are used, this approach should be avoided.

The other methods produce raw signal and do not include the reconstruction in any way, and are thus unaffected.

Are the gradients that are applied during the pulses always parallel to each other?

If the individual pulses are rather simple, but applied in multiple linearly independent directions, then the same need for high-dimensional echo class separation breaks the k-space and Sequence Response Kernel method.

The Smart Bloch Solver is unaffected.

Is the number of different tissue types big?

As elaborated in the phantom considerations section (see Section 4.1.4), the k-space approach becomes impractical when many different tissue types need to be considered. The Sequence Response Kernel approach can remedy this by interpolation. The Smart Bloch Solver is only slightly affected because the pulse calculations may need to be performed for a larger tissue set.

Are the RF pulses spaced in a complicated way? Is the 0th gradient moment between the individual RF pulses complicated?

Complicated RF pulse spacing is generally a strong indicator for the presence of undesired signal behavior, which affects all methods. The k-space and Sequence Response Kernel methods are particularly error-prone because the echo formation is complex and the relevant signal contributions are spread out widely over the exponentially growing number of Echo Pathway Segments. The Bloch solvers do not introduce further algorithmic complexity, but they may produce simulation artifacts without providing insight into the error sources because they are intrinsically inefficient in the presence of many segments.

As described in Section 4.1.3.4, the gradient moments between the RF pulses affect the merging capabilities of the k-space approach. This leads to complicated echo behavior and therefore the same effects as complicated RF pulse spacing.

Is the distance between the RF pulses short in comparison with the relaxation times? If so: Are the RF pulse flip angles neither small nor big?

If so, the amount of relevant stimulated echo signal can be expected to be rather large. The Bloch solvers are intrinsically ill-suited to cope with stimulated echoes while the number of important segments is likely to be large for the other methods. The k-space or Sequence Response Kernel method should be preferred, even though the computational effort is still high.

Does the phantom have a computationally efficient Fourier representation?

The k-space method relies on many evaluations of the phantom's Fourier transform. If that part is computationally inefficient, then most computation time will be spent evaluating the Fourier transform of the object. This might suggest falling back to the Smart Bloch Solver or advancing to the Sequence Response Kernel approach that do not require the Fourier transform of the phantom.

Summary

The following table (Table 4.1) provides a compact overview of each method's capabilities. Besides this section, supporting information can be found in sections 3.5, 4.1.1, 4.1.3, 4.1.4, 4.1.2 and 4.1.5.

Feature	DB	SB	k	SRK
Full image simulation	✗	✓✗	✓✗	✓
Reuse pulse simulations	✗	✓	✓	✓
Voxel-based phantoms	✓	✓	✗	✓
Geometrically complicated phantoms	✓	✓	✓✗	✓✗
Many different tissues	✓	✓	✗	✓✗
Sub-voxel effects	✓	✓	✓	✗
Effects of higher order echoes	✗	✗	✓	✓
Extensions to the Bloch equation	✓	✗	✗	✗
Intricate image reconstruction	✓	✓	✓	✗
Dynamic object properties	✓	✓	✗	✓✗
Rapid pulse succession	✓	✓	✗	✓✗
2D/3D selective pulses	✓	✓✗	✗	✗
Exchangeable object geometry	✗	✗	✓	✓
Useful intermediate results	✗	✓✗	✓	✓
Effort increases with number of RF pulses	✗	✗	✓	✗
Effort increases with resolution	✓	✓	✗	✗

Table 4.1: Method feature overview—A green check mark ✓ indicates a compatibility or agreement with the corresponding statement, a red cross mark ✗ indicates incompatibility, disagreement or unpracticability. The yellow combination of check and cross mark ✓✗ indicates a partial compatibility, partial agreement or reasonable—but increased—computational effort. The methods are abbreviated by DB (Direct Bloch), SB (Smart Bloch), k (k-space based), SRK (Sequence Response Kernel).

Significance of this work

The applications and derived algorithms that were carried out in this work are proof-of-concepts and prototypes. They are ready to be implemented into a more approachable environment to satisfy the need that was motivated in the first chapter (see Section 1.4). The following paragraphs state the manner in which the methods can be employed.

4.2.1 Research impact

The development of MRI methods is in part hindered by the large effort that needs to be put into a thorough simulation.

The efficiency-focused techniques introduced in this work are relevant for a large span of required accuracies as well as computation speed. In particular, the Sequence Response Kernel simulation approach can be tuned finely. The algorithm can meet any given error tolerance, but also supports fast approximations. The image assembly can be as fast as a lookup-table if needed—which requires the same effort as the image rendering. These two extremes have been present before, as complete Bloch simulations and specialized signal approximations. But it was not possible to arbitrarily choose the degree of exactness or speed. This is particularly interesting for applications that can benefit from multi-scale or refinement options. Many artifacts are too complicated to be treated adequately by a model, yet the application doesn't allow for a full-scale simulation. The proposed approach can solve this problem.

Beyond the Sequence Response Kernel approach, many of the developed workflow steps can be used to aid research tasks: For the Smart Bloch simulation, the pulse instantanization helps understand and estimate pulse effects, the pulse precalculation process reduces computational redundancy, and using analytical solutions when possible increases the numerical accuracy.

The k-space method can now be used with full accuracy, in contrast to contemporary implementations of the extended phase graph algorithm, and can also be applied to arbitrary sequences and can be used with a wider range of software phantoms.

Each of the algorithms derived from the Sequence Response Kernel approach can be developed further to become a powerful tool in quantitative imaging, error analysis and protocol optimization. This further development does not rely on deep insight into the workings of MRI since the Sequence Response Kernels supposedly contain all information needed to interpret the images. It is thus very approachable by specialists of different scientific fields.

4.2.2 Education impact

The transition from signal space to image space can yield a great potential when ideas need to be communicated. Signal amplitude and k-space traversal schemes are two main ways of explaining a sequence. The Sequence Response Kernel approach combines both, adds a thorough theoretical background and then transforms it onto image space. Image space is a more natural and approachable means in MRI since images are the final objective of each acquisition. The suggestive sampling and reconstruction is not left as a thought exercise for the consumer of the education, easing the understanding.

Explaining an MRI sequence through convolutions of images can be an easy entry for developers of image-based algorithms. It is then not necessary to understand MR physics in order to incorporate MR-related artifacts and simulation information into the algorithms, while still maintaining a high numerical accuracy.

Apart from this educational shortcut, parts of the other approaches also have strong illustrative power. The Smart Bloch solver produces the magnetization states at the individual pulses. Given these states, calculating the state at an arbitrary time point is trivial. In addition, even changes to the sequence can be respected quickly if the pulses are precalculated adequately. This can provide a flexible and accurate tool that replaces the commonly employed hand waving when explaining MRI effects.

4.2.3 Clinical impact

Contemporary simulation frameworks are largely incompatible with clinical workflows, but the methods of this work are feasible to be implemented into clinical applications.

The Sequence Response Kernel approach removes the need for a specialized mathematical signal model and enables comparability of the images. It is also based on the full sequence with potentially high numerical accuracy and is thus superior to the quality of an approximate signal model.

It thereby enables highly responsive and reliable simulation support that adds further meaning to acquired images, or provides optimal acquisition parameters.

Some reasonable clinical applications that could be supported by the Sequence Response Kernel method will be elaborated in the next section (see Section 4.3.2).

Outlook

The potential impact of this work has been elaborated in the previous section. The following section discusses the next steps that are necessary to realistically reach those goals.

4.3.1 Context and infrastructure considerations

The establishment of satisfactory and usable simulation tools needs to solve some issues that are explained in the following paragraphs.

4.3.1.1 Unified Simulation Framework

There is an abundance of tools that either answer very specified simulation problems that depend on several problem assumptions, or try to answer a broad range of problems by relying on the largest common denominator. The first kind has a limited range of applicability while the second one is largely inefficient.

For example, pulse simulation has nothing in common with echo calculation. Furthermore, a change in the physical model of the Bloch equation potentially breaks all advanced methods. A unified simulation framework should provide the freedom to adapt the physical description, modify the advanced simulation methods and validate the individual results.

It is important to keep in mind that MRI is a large field and not every researcher has an understanding and functional knowledge of advanced aspects that revolve around MRI simulation, and well-chosen simulation parameters are crucial for satisfactory results.

Certain effects may or may not be of interest, such as pulse simulation opposed to pulse idealization or the sub-slice profile. Many of the resulting simulation parameters can be set directly if the user makes a statement or is asked about such effects. Other parameters, such as the weighting of the Echo Pathway importance function or the number of pathways, cannot be determined automatically. It is merely possible to estimate the error and guess whether this error is acceptable.

A single workflow is not a realistic goal. A good framework should provide tools and carefully provide hints about possible inaccuracies or inefficiencies, as hinted in the method selection guidelines section of this chapter (see Section 4.1.6).

4.3.1.2 General description language

A unified framework needs a general sequence and phantom description language. The level of sequence abstraction that the individual methods need is very different, and the highest level of abstraction might not be available at the time of problem formulation.

Sequence

The Direct Bloch equation simulation requires no knowledge about the sequence except for the magnetic fields that the individual spin packets experience at a given time point. The higher level methods imply gradient linearity, short RF pulses or even Cartesian readout. The user of a unified simulation framework needs to have several points of entry to define the sequence and should be encouraged to define the sequence on an abstract level to enable the potential of more advanced methods.

The nature of the sequence should already suffice to provide hints about the possible complications that arise when the individual methods are used. Without the need of user investigations, some methods can be discarded directly, inefficiencies can be rated and the computational effort can be estimated. This automatic simulation applicability investigation alone can be a huge benefit in sequence development. Most of the higher level methods are designed for sequences that follow a desired behavior, and if the requirements of the higher level methods are stressed, it is likely that the sequence developer introduced changes that give rise to unwanted behavior, e.g. many stimulated echoes or incomplete readout.

There are numerous different MRI sequences, and the advanced methods require a preparation to identify the pulse classes, readout scheme, gradient moments etc. There is currently no MRI sequence description standard—therefore each sequence that is to be simulated needs to be embedded in a compatible way. This is a huge problem that can easily take more time than implementing the simulation procedure itself. And if done inadequately due to intellectual property restrictions or poor reimplementation, the results are invalid.

Phantom and object definition

The object that is to be simulated can also span over multiple levels of abstraction (see Section 4.1.4), and it needs to be transformed into an object that is suitable for the task at hand. Knowledge about the sequence can help determine compatible processing of the phantom. The pulses yield information about the slice positions and required slice profile resolution. The spoiling and crushing gradients can help to decide whether a fine sampling of the spin packets perpendicular to the slice direction is needed.

The phantom may of course also be incompatible with the simulation method, or not suitable. In those cases the phantom abstraction type and sequence should be sufficient to determine the computational effort and expected error range.

4.3.2 Clinically motivated applications

Clinical applications are the strongest motivation of MRI in general. The Sequence Response Kernel approach has great potential for clinical applications, but needs to be adapted further.

4.3.2.1 Automatic sequence characterization

Historically, the development of a sequence begins with an acquisition strategy, for instance gradient echo, spin echo or echo planar imaging, and describes the resulting signal analytically by using individual simplifications. Further development and improvement is usually justified based on those models. Simulation provides a common tool to remove the individuality of the models, and enables cross-comparison of the different sequence types.

The Sequence Response Kernel approach dictates that contrast behavior, blurring and undesired signal can be calculated in the shared image space. A sequence can be characterized this way and yield a concise and understandable description of its properties.

Sequences can be characterized as elaborated in the according section (see Section 2.5.1). With a characterization of that manner, a clinician can quickly grasp the contrast or imperfections of a sequence and compare it to existing ones. This can ease the transition to new methods since comparison with existing methods can be done on that basis rather than through an abstract signal model or a set of some practical examples. Also, virtual experiments can be performed quickly to showcase the differences in an illustrative way.

The development of such a characterization tool requires further research to identify the features of interest, and the implementation of a large set of sequences into the simulation framework.

Reaching a general acceptance of such an automatic and unified characterization is not feasible, since many sequences are designed for a very specific purpose, thus working against the clarity of the characterization, and other sequences are incompatible with the Sequence Response Kernel approach. Realistically, such an application should first be pursued where equal characteristics of multiple sequence are of interest, for instance in manual protocol optimization. This may also lead to the discovery of derived sequence fitness quantities that further perpetuate the understandability of the nature of MRI sequences.

4.3.2.2 Protocol optimization

Nowadays, finding the best sequence parameters is a tedious task, driven by expertise and legacy conventions. This process can in part be replaced by an automatic optimization which is driven by parameters that have a tangible meaning. The clinician could choose features that he aims for—such as contrast between a set of tissues, robustness to chemical shift artifacts or scan time—and receive an optimal configuration of the physical sequence parameters. If a scanner operator wants to change the sequence parameters directly, the simulation can produce feedback in terms of image features and inform about artifacts that are likely to arise. As laid out in the derived algorithms section, sequences can be optimized based on Sequence Response Kernels (see Section 2.5.2).

This process requires a very responsive simulation that is evaluated in image space. The Sequence Response Kernel approach and sequence optimization prototype explained in the previous chapters lay the theoretical basis for this task.

This idea can be used to design protocols that are not a succession of isolated acquisition sets that produce one diagnostically relevant parameter each. The relevant parameters can be derived using a heterogeneous dataset. Each image contributes strongly to those parameters that it provides a high distinguishability for, and weakly to those that it is not sensitive to. This concept is the logical progression of the information density optimization elaborated in the previous chapters. The importance of each individual sequence within the protocol can be spread out, such that more information can be gained in less time while still allowing for robustness against measurement errors if desired.

The implementation of optimization routines that can quickly optimize image features at the scanner can be realized in two ways: Either by a vast preprocessing and storage, or by establishing a connection to an HPC environment. Since this optimization can be done on a per-sequence basis, it can be adapted and quality-assessed individually, hiding the details and algorithmic complexities from the MRI technician.

Sequences that are most promising for this algorithm are those that require trial and error by the clinician to achieve the contrast that is desirable.

4.3.2.3 Information synthesis

The later algorithms in the derived algorithms sections (see Section 2.5) all aim at extracting as much information from an image set as possible, while respecting the nature of the individual sequences. This is a broad and ambitious goal that can best be answered on a use-case basis.

Parameter mapping is a self-evident information synthesis problem. The Sequence Response Kernel approach provides an efficient parameter mapping strategy without the need for a mathematical model, but applications that benefit from parameter mapping usually introduce dedicated sequences that can handle the parameter mapping process particularly well with the aid of a signal model. The real potential of the Sequence Response Kernel approach lies in respecting the parameter-dependence of the sequence effects of sequences that do not have a signal model for that particular parameter. In this situation, the Sequence Response Kernel approach builds the bridge between the sequences opposed to isolating them.

This new comparison tool can also help to harmonize data and protocols of different sequence versions running on different scanners by different vendors. Multi-center studies require a lot of expert input to tweak protocols to realize sufficiently homogeneous conditions at each site. Even though the goal is to achieve similar image features, the homogenization is done by adapting the raw physical sequence parameters. An image-based homogenization loop could solve this complication directly, removing the bias that originates from the physical sequence description.

The approach can also be used to enable a more standardized way of information synthesis. MR images are often interpreted individually. A diagnosis is often times driven by a localized feature on an image that is acquired with one sequence in conjecture with a different feature on a second image. This is either done manually, or by using a simplified mathematical model that was adapted to the approximate sequence class. The Sequence Response Kernel approach motivates means to merge information without the need for a specialized model.

But a direct implementation of this idea is not feasible. The full parameter space is generally of high dimensionality and not computationally manageable. It is more reasonable to use subsets of the image set to fit certain parameters and improve the fit iteratively, fixating the respective parameters of minor influence. This process needs to be adapted to an actual application.

4.3.2.4 Quality and relevance analysis

Quality and relevance analysis is related to the information synthesis idea, which have also been foreshadowed in the derived algorithms sections.

Whenever information is erroneous and redundant, the agreement of the information components can be investigated. Quality assurance, and especially automatic quality assurance is a highly relevant research topic of clinical studies. The error measure of this work (see Section 2.5.4) is a prototype and starting point for simulation-assisted quality assurance. This error measure should be further developed to produce a concise and more generally applicable workflow.

But the further development potential appears promising: Problematic images could be discarded, and repaired or replaced synthetically. Sequences that are not relevant could be replaced by those that maximize information content.

Developing the framework and evaluating the method is feasible, mostly because clinical studies often involve large datasets that the method is applicable to.

4.4 Conclusion

The problem of efficient MRI simulation does not have a single solution. Depending on the circumstances, desired accuracy, and acceptable effort, a compromise has to be found.

The methods of this work were developed to greatly exploit the circumstances while retaining the capabilities of full numerical accuracy. These methods and their auxiliary algorithms show great potential in all areas of MRI simulation. For the case of very compatible circumstances, a completely novel simulation approach was developed that has the potential to introduce MRI simulation easily and accurately in image-based algorithms—fundamentally changing the way MRI simulation can be used.

Intermediate results have been presented at conferences [7] [8] [9], but the true value of this work needs to be exemplified further through clinical applications and connections to specialized research and development tasks.

Acknowledgments

Foremost, I would like to express my gratitude to my advisor Prof. Matthias Günther for the support of my study and research, for his patience, guidance and enthusiasm.

Besides my advisor, I would like to thank Prof. Tobias Preußer for his encouragement and support.

I would also like to thank my colleagues at Fraunhofer MEVIS for giving me the opportunity to spend the past years living the famous *MeVis spirit*.

Bibliography

[1] S. Balac and L. Chupin. Fast approximate solution of Bloch equation for simulation of RF artifacts in Magnetic Resonance Imaging. *Mathematical and Computer Modelling*, 48(11-12):1901 – 1913, 2008.

[2] H. Benoit-Cattin, G. Collewet, B. Belaroussi, H. Saint-Jalmes, and C. Odet. The SIMRI project: a versatile and interactive MRI simulator. *Journal of Magnetic Resonance*, 173(1):97 – 115, 2005.

[3] M. Bernstein, K. King, and X. Zhou. *Handbook of MRI Pulse Sequences*. Elsevier Science, 2004.

[4] F. Bloch. Nuclear Induction. *Physical Review*, 70:460–474, Oct 1946.

[5] J. R. Cash and A. H. Karp. A Variable Order Runge-Kutta Method for Initial Value Problems with Rapidly Varying Right-hand Sides. *ACM Transactions on Mathematical Software*, 16(3):201–222, Sept. 1990.

[6] D. L. Collins, A. P. Zijdenbos, V. Kollokian, J. G. Sled, N. J. Kabani, C. J. Holmes, and A. C. Evans. Design and Construction of a Realistic Digital Brain Phantom. *IEEE Transactions on Medical Imaging*, 17(3):463–468, 1998.

[7] C. Cordes and M. Günther. Real-time MR simulation by combining Extended Phase Graph Diagrams and Bloch Equation Simulations. In *Proceedings of the 30th Scientific Meeting of the ESMRMB*, page 617, 2013.

[8] C. Cordes and M. Günther. Extracting MRI Sequence Response Kernels from Generalized Extended Phase Graph Simulations. In *Proceedings of the 22nd Scientific Meeting of the ISMRM*, page 1510, 2014.

[9] C. Cordes and M. Günther. Uncertainty Volume Analysis—A Measure for Protocol Performance. In *Proceedings of the 23rd Scientific Meeting of the ISMRM*, page 3383, 2015.

[10] G. Folland. *Fourier Analysis and Its Applications*. Pure and applied undergraduate texts. American Mathematical Society, 1992.

[11] O. Forster. *Analysis 2: Differentialrechnung im Rn – gewöhnliche Differentialgleichungen*. Vieweg, 1979.

[12] H. Gach, C. Tanase, and F. Boada. 2D & 3D Shepp-Logan Phantom Standards for MRI. In *Proceedings of the 19th International Conference on Systems Engineering*, pages 521–526, Aug 2008.

[13] M. Günther and D. A. Feinberg. Simultaneous spin-echo refocusing. *Magnetic Resonance in Medicine*, 54(3):513–523, 2005.

[14] H. H. Benoit-Cattin and G. Collewet. Numerical implementation of the Bloch equations to simulate magnetization dynamics and imaging. In *Proceedings of the 14th Scientific Meeting of the ISMRM*, 2006.

[15] E. L. Hahn. Spin echoes. *Physical Review*, 80:580–594, Nov 1950.

[16] L. Hanson. Introduction to Magnetic Resonance Imaging Techniques. Technical report, Danish Research Centre for Magnetic Resonance, 2009.

[17] L. G. Hanson. Is quantum mechanics necessary for understanding magnetic resonance? *Concepts in Magnetic Resonance*, 32A(5):329–340, 2008.

[18] F. Harris. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, 66(1):51–83, Jan 1978.

[19] J. Hennig. Multiecho imaging sequences with low refocusing flip angles. *Journal of Magnetic Resonance*, 78(3):397 – 407, 1988.

[20] J. Jackson, C. H. Meyer, D. G. Nishimura, A. Macovski, et al. Selection of a convolution function for Fourier inversion using gridding. *IEEE Transactions on Medical Imaging*, 10(3):473–478, 1991.

[21] T. H. Jochimsen, A. Schäfer, R. Bammer, and M. E. Moseley. Efficient simulation of magnetic resonance imaging with Bloch-Torrey equations using intra-voxel magnetization gradients. *Journal of Magnetic Resonance*, 180(1):29 – 38, 2006.

[22] T. H. Jochimsen and M. Von Mengershausen. ODIN—object-oriented development interface for NMR. *Journal of Magnetic Resonance*, 170(1):67–78, 2004.

[23] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

[24] C. G. Koay, J. E. Sarlls, and E. Özarslan. Three-dimensional analytical magnetic resonance imaging phantom in the Fourier domain. *Magnetic Resonance in Medicine*, 58(2):430–436, 2007.

[25] W. Kutta. *Beitrag zur näherungsweisen Integration totaler Differentialgleichungen*. B. G. Teubner, 1901.

[26] M. Levitt. *Spin Dynamics: Basics of Nuclear Magnetic Resonance*. Wiley, 2001.

[27] D. Ma, V. Gulani, N. Seiberlich, K. Liu, J. L. Sunshine, J. L. Duerk, and M. A. Griswold. Magnetic resonance fingerprinting. *Nature*, 495(7440):187–192, 2013.

[28] S. J. Malik, F. Padormo, A. N. Price, and J. V. Hajnal. Spatially resolved extended phase graphs: Modeling and design of multipulse sequences with parallel transmission. *Magnetic Resonance in Medicine*, 68(5):1481–1494, 2012.

[29] W. Overall and J. Pauly. An Extensible, Graphical Environment for Pulse Sequence Design and Simulation. In *Proceedings of the 16th Scientific Meeting of the ISMRM*, page 1652, 2007.

[30] J. Petersson, J.-O. Christoffersson, and K. Golman. MRI simulation using the k-space formalism. *Magnetic Resonance Imaging*, 11(4):557 – 568, 1993.

[31] M. Shinnar, S. Eleff, H. Subramanian, and J. S. Leigh. The synthesis of pulse sequences yielding arbitrary magnetization vectors. *Magnetic Resonance in Medicine*, 12(1):74–80, 1989.

[32] Siemens Healthcare. MAGNETOM Skyra. Technical report, Siemens Healthcare, 2010.

[33] M. S. Silver, R. I. Joseph, and D. I. Hoult. Selective spin inversion in nuclear magnetic resonance and coherent optics through an exact solution of the Bloch-Riccati equation. *Physical Review*, 31:2753–2755, Apr 1985.

[34] A. Sodickson and D. G. Cory. A generalized k-space formalism for treating the spatial aspects of a variety of nmr experiments. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 33(2):77 – 108, 1998.

[35] T. Stöcker, K. Vahedipour, D. Pflugfelder, and N. J. Shah. High-performance computing MRI simulations. *Magnetic Resonance in Medicine*, 64(1):186–193, 2010.

[36] H. C. Torrey. Bloch Equations with Diffusion Terms. *Physical Review*, 104:563–565, Nov 1956.

[37] D. D. Traficante. Relaxation. Can T2, be longer than T1? *Concepts in Magnetic Resonance*, 3(3):171–177, 1991.

[38] J. P. Wansapura, S. K. Holland, R. S. Dunn, and W. S. Ball. NMR Relaxation Times in the Human Brain at 3.0 Tesla. *Journal of Magnetic Resonance Imaging*, 9(4):531–538, 1999.

[39] M. Weigel. Extended phase graphs: Dephasing, RF pulses, and echoes - pure and simple. *Journal of Magnetic Resonance Imaging*, 41(2):266–295, 2015.

[40] D. A. Yoder, Y. Zhao, C. B. Paschal, and J. M. Fitzpatrick. MRI simulator with object-specific field map calculations. *Magnetic Resonance Imaging*, 22(3):315–328, 2004.

[41] P. C. L. Zhi-Pei Liang. *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*. Wiley-IEEE Press, October 1999.

Simulation of Magnetic Resonance Imaging (MRI) is based on the Bloch equation. Solving the Bloch equation numerically is not difficult, but realistic imaging experiments bear a high computational burden.

In his dissertation, Cristoffer Cordes presents simulation methods that exploit hardware restrictions and the common structure of MRI sequences while not enforcing any approximations. These strategies use the reoccurrence of radiofrequency pulses, partial availability of analytical solutions, a reformulation of the problem in Fourier space and finally an inclusion of the reconstruction process to perform MRI simulation in image space, titled Sequence Response Kernel approach.

The algorithmic efficiencies of the methods are investigated and applied to realistic imaging experiments. The properties and potential of the algorithms are exemplified, with an emphasis on the Sequence Response Kernel approach.

This book is aimed at physicists and mathematicians with an MRI background.

Printed in Germany
ISBN 978-3-86376-052-6

$ 60,10
CHF 55,10
EUR [A] 44,90
EUR [D] 39,90

9 783863 760526 03990

OPTIMUS