Journal of Magnetic Resonance 301 (2019) 1–9

Contents lists available at ScienceDirect

Journal of Magnetic Resonance

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

Spin-Scenario: A flexible scripting environment for realistic MR simulations

Yan Chang [a,b], Daxiu Wei [c,*], Huihui Jia [d], Cecilia Curreli [a,e], Zhenzhou Wu [a], Mao Sheng [d], Steffen J. Glaser [f,*], Xiaodong Yang [a,*]

[a] Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Sciences, 215163 Suzhou, China
[b] University of Chinese Academy of Sciences, China
[c] Physics Department and Shanghai Key Laboratory of Magnetic Resonance, East China Normal University, 200062 Shanghai, China
[d] Department of Radiology, Children's Hospital of Soochow University, 215025 Suzhou, Jiangsu, China
[e] Munich School of Engineering, Technical University of Munich, 85748 Garching, Germany
[f] Department of Chemistry, Technical University of Munich, 85748 Garching, Germany

ARTICLE INFO

Article history:
Received 4 January 2019
Revised 29 January 2019
Accepted 30 January 2019
Available online 13 February 2019

Keywords:
Pulse sequence programming
Optimal control
Spin dynamics
TensorFlow
Time-frequency spectrogram

ABSTRACT

In this paper we present a new open source package, Spin-Scenario, aimed at developing an intuitive, flexible and unique scripting framework able to cover many aspects of simulations in both MR imaging and MR spectroscopy. For this purpose, we adopted the Liouville space model as the standard computing engine and let the consequent computational burden be afforded by parallel computing techniques. Benefitting from the powerful Lua scripting language, the pulse sequence programming syntax was specially designed to offer an extremely concise way of scripting. Moreover, the built-in dataflow graph based optimal control scheme enables an efficient optimization of shaped pulses or multiple cooperative pulses for real-life experiment evaluations. As the name states, the users are expected to be able to realize their creative ideas like a scenarist that creates a scenario script and looks at the spin actors acting accordingly. The validation of the framework was demonstrated with several examples within MR imaging and MR spectroscopy. Spin-Scenario is available for download at https://github.com/spin-scenario.

1. Introduction

Numerical simulation of spin dynamics has become a powerful tool for a variety of MR studies, including design of pulse sequences, experimental validations, educational purposes and many others. Depending on the field of applications, those existing tools can be classified basically into two categories: MR imaging [1–4], which uses numerical Bloch equation simulators, and MR spectroscopy [5–8], which uses either analytical or numerical simulators based on quantum mechanics.

With several assumptions, the Bloch's semi-classical theory is pretty simple and efficient for MRI simulations regarding a system of independent proton isochromats. However, it has certain limitations as physics beyond the standard Bloch equation is not yet taken into account [9]. For instance, in order to include the kinetic effects of diffusion and flow (see the Bloch-Torrey-Stejskal (BTS) equation) an extension of Bloch's theory is required, while a further extension of the BTS equation for multi-pool model of spin packets is necessary in order to include the effects of magnetization transfer. Also, simulations of higher spins ($I > 1/2$) are just beyond the limits of the Bloch equation, though the higher spin imaging has nowadays attracted much attention. For example, the lithium-based MRI [10] enables a completely non-invasive visualization and characterization of the changes that occur on battery electrodes and in the electrolyte. The sodium-based MRI [11] can provide direct insights into the metabolic activity and cellular integrity of tissues – in particular in the human brain. It also seems that the emerging MR methodologies, especially in the field of medicine, are blurring the boundaries between MR imaging and MR spectroscopy. E.g., magnetic resonance spectroscopic imaging (MRSI) [12], combining both spectroscopic and imaging methods to produce spatially localized spectra from within the sample or patient, is now playing a vital role in the study of metabolic changes in brain tumors, Alzheimer's disease and other diseases affecting the brain. Compared with the Bloch equation, the Liouville-von Neumann equation offers a general formalism for spin dynamics, based on quantum mechanics, which is able to broadly describe experiments either in MR imaging or

[*] Corresponding authors.
E-mail addresses: dxwei@phy.ecnu.edu.cn (D. Wei), glaser@tum.de (S.J. Glaser), xiaodong.yang@sibet.ac.cn (X. Yang).

https://doi.org/10.1016/j.jmr.2019.01.016
1090-7807/© 2019 Elsevier Inc. All rights reserved.

spectroscopy. A representative example is the Spinach [8,13] library, with the quantum mechanical simulations in Liouville space, the package is extremely versatile including spatially encoded NMR, diffusion processes, and even hydrodynamics.

Another crucial issue for MR simulations is the pulse programming environments, whose intuition and flexibility are essential. ODIN [1] offers researchers a flexible way to develop pulse sequences with low-level custom C++ classes, method which usually requires basic object-oriented programming experience and potentially leads to more lines of code. JEMRIS [3] utilizes XML tree to represent a pulse sequence, the user can define the loops at branch nodes and the actual pulses at leaf nodes respectively. Note that nodes definitions must be bounded to the tree structure, which, to some extend, decreases the readability of the resulting pulse sequence. SequenceTree [14] also benefits from the hierarchical tree concept: it shows the sequence tree, parameters and pulse diagram by a QT-based graphical user interface, which provides a visual and intuitive environment to nonprogrammers, being particularly useful for the training and education purposes.

The ultimate aim of the Spin-Scenario library is to provide an intuitive, flexible and unique scripting environment for both MR imaging and spectroscopy. To achieve this objective, first we internally use the Liouville space model for the spin dynamics calculations. The consequent time-consuming simulations can be greatly accelerated by taking advantage of parallel computing techniques. Inspired by design philosophy in the game industry, the kernel functionality of Spin-Scenario is written in C++ for high performance, whereas the experimental scenarios are created via a customized scripting language with elegant simplicity. As a powerful, embeddable and fast scripting language, Lua [15] has been widely used in many industrial applications, especially in games. The Spin-Scenario extension with Lua bindings offers a straightforward scenario scripting environment in a natural manner. An introduction to Lua provides a simple way to get the stage ready for the play and, more importantly, builds a whole new, flexible and user-friendly pulse programming environment that provides researchers all functionalities for rapid-sequence prototyping.

2. Methods

2.1. Physical background

In the Liouville space, the motion of the density operator in the presence of relaxation has the following form

$$\frac{\partial \hat{\rho}}{\partial t} = \left(-i\widehat{L} + \widehat{R} \right) \hat{\rho}, \tag{1}$$

in which $\widehat{L}$ is the commutation superoperator corresponding to the system Hamiltonian, and $\widehat{R}$ is referred to as the relaxation superoperator. The relaxation parameters for tissues (e.g. T_1 and T_2 for the longitudinal and transverse relaxation time respectively) and molecules (dipolar, CSA, quadrupolar, etc.) can be embodied in the relaxation superoperator $\widehat{R}$.

In detail the Hamiltonian superoperator $\widehat{L}$ can be written as

$$\widehat{L} = \widehat{H}_0 + \sum_k \widehat{H}_k^{rf}, \tag{2}$$

where $\hat{H}_0$ is the free evolution Hamiltonian, $\widehat{H}_k^{rf}$ are the external parts of the Hamiltonian representing the radio frequency (r.f.) control fields.

2.2. General framework

The Spin-Scenario framework consists of three essential parts from bottom to top: the computing engine layer, the pulse sequence layer and the scenario script layer. The first two layers are both implemented in C++ for optimized performance in kernel calculations as well as highly flexible combination in pulse sequence assembly, while the uppermost layer is implemented in Lua for the ultimate intuitive scripting environment.

The kernel calculations inside the computing engine are based on the Liouville-von Neumann equation in order to make it essentially compatible with different MR applications, and moreover, parallel computing techniques are utilized to satisfy the potential large spin system or huge spin ensemble. The pulse sequence layer is accomplished via introducing the sequence blocks concept. Basically, a pulse sequence can be always broken down into two-level blocks. Level 1 blocks are atomic blocks of r.f. pulses, gradients, delays and signal acquisition modules; level 2 (glue level) blocks are serial or parallel combination of arbitrary components from both levels. In this way, each pulse sequence can be ultimately represented by a high-level serial block. The scenario script layer is designed to enable a high degree of user flexibility. We extend the library with Lua bindings, which provides a straightforward scenario scripting environment in a natural manner.

When an experimental scenario is ready, pulse sequences will be broken into a substantial number of steps, each of which can be described as a fixed interval with constant control variables such as r.f. (amplitude, phase and frequency), gradient (axis and amplitude) and acquisition. The evolution of each step can be further solved with the computing engine. In Spin-Scenario, Google's TensorFlow [16] is also utilized to construct the sequence related computation graph, and launch the forward computation.

The major modules of Spin-Scenario are listed in Table 1.

2.3. Spin system operations

In contrast to simulation packages such as JEMRIS [3] and MRiLab [4], the formation of spin systems in Spin-Scenario is based on quantum mechanics instead of on the semiclassical Bloch approach. Basically, the quantum state of a spin system can be represented by the density matrix or the density operator, which is essentially built based on irreducible spherical tensors. The theory details of the density operator are well described in [5,8].

In Spin-Scenario, a spin system can be easily created with a Lua table, including all necessary parameters such as `spin`, `zeeman`, `jcoupling` and `relaxation`. For example, the ^{13}C labelled alanine can be formed with following code snippet:

```
local sys = spin_system{
spin = "13C 13C 13C",
zeeman = "1 scalar 15.74 kHz
          3 scalar  -4.3 kHz",
jcoupling ="1 2 scalar 54.2 Hz
            2 3 scalar 35.4 Hz"
}
```

Table 1
Major modules within the Spin-Scenario library.

Spin-Scenario library	
Spinsys	Spin system formation based on quantum mechanics
Phantom	Digital phantom models including MIDA brain
Seq	Pulse sequence programming language
Physx	Physical engine doing the kernel calculations
Equip	Interfaces for magnet, gradient coils and Tx/Rx coils
Oc	r.f. pulse optimizations
Utility	Utility tools for interactive IO and visualization

In general, the density operators or state operators can be represented as products of individual spin operators. The `op` and `state` functions are used for representing the density operator and the state operator respectively. The validated strings for each spin are `"I+"`, `"I-"`, `"Iz"` and `"Ie"`. E.g `sys:state{1, "I+", 3, "I-"}` is for the state $I_1^+I_3^-$.

Moreover, taking advantage of the computer algebra system Yacas [17], Spin-Scenario also provides a very practical feature for the construction of more complex operators or states that are a linear combination of arbitrary either spherical or cartesian operators. E.g. `sys:expr_state("2*I1xI3x-0.5*I2xI3y")` is a straightforward generation of the combined state $2I_{1x}I_{3x} - 0.5I_{2x}I_{3y}$. The Yacas integration was fully utilized throughout Spin-Scenario to evaluate those kinds of expression in a natural manner.

For a specific spin system, it is also possible to retrieve its free Hamiltonian, total Hamiltonian and the relaxation superoperator via `sys.L0`, `sys.L` and `sys.R` respectively.

2.4. Pulse sequence programming

2.4.1. Sequence block

The most common used rectangle or hard pulses are realized by `hardRF` block. For example, `hardRF{beta = 60, phase = "-y"}` is a 60° -y-pulse for proton ^{1}H excitation. The pulse width is proportional to that of the standard 90° pulse, which can be globally declared via `pw90{}` in advance. Alternatively, adding an explicit `width` key to the table will achieve a similar block. Besides, the valid phase options are not merely limited to the four quadrant axes, arbitrary angle in degree is also generally supported. For other nuclei, the explicit nuclear isotope such as `channel = "13C"` is required. Moreover, multi-channel excitation pulse can be achieved easily by extending both `channel` and `phase` respectively. For instance, `hardRF{beta = 180, channel ="1H|13C", phase = "x|x"}` gives a refocusing pulse applied on a heteronuclear ^{1}H$-^{13}$C spin system.

Shaped pulses has been widely used to increase excitation bandwidth, achieve desired profile over the entire bandwidth, and improve polarization and coherence transfer efficiencies in numerous NMR experiments [18]. To this end, `shapedRF` was provided as a general interface for both routine patterns (such as Sinc, Gaussian, Rectangle, etc.) and external shape files. E.g. `shapedRF {width = 5.12, step = 256, max_amp =100, pattern = "sinc"}` defines a 5.12 ms 256-step five-lobe Sinc pulse on proton channel with a maximum amplitude 100 Hz, an extra `lobe` item can be added if necessary. It also supports to import pulse shape from external waveform file or complex mathematical expression, just specify `pattern` with the waveform file or the desired expression, and Spin-Scenario will automatically assign each channel with amplitude in Hz and phase in degree respectively. E.g. `shapedRF {channel = "1H|13C", width = 5, pattern = "shape.RF"}` generates a 5 ms simultaneous, two-pulse shaped pulse on the ^{1}H and ^{13}C transmitters from external file. Moreover, in order to characterize shaped pulses, a time-frequency representation algorithm [19] was introduced into the environment, which will be shown later.

The pulsed field gradients are most-often used in MR pulse sequence either for spatial-encoding in imaging or rephasing (selection) and dephasing (elimination) of a particular magnetization transfer pathway in spectroscopy. This library offers `trapGrad`, `shapedGrad` and `exprGrad` to create three-axis gradients from the typical trapezoidal type, custom defined pattern file and mathematical expression individually. Taking trapezoidal gradient as an example, it is efficient to specify both `width` and `area` of a desired gradient, and the lobe shape will be automatically generated according to the peak gradient strength as well as the slew rate of a specific gradient system. There is also a shortcut to build imaging gradient waveforms such as frequency-encoding, phase-encoding and slice selection by assigning an alternative `func` key with `read_out`, `phase_encode`, and `slice_select` respectively.

The data acquisition block `acq` was designed for signal observation, which simulate the quadrature detection of the receiver in MR scanner. For example, `acq{np = 1024, sw = 1e4}` performs ^{1}H acquisition, in which `sw` specifies the spectral width in Hz, together with number of data points `np`, the total acquisition time can be determined. Similar to the r.f. blocks, explicit nuclear isotope should be appended to `channel` for other nuclei. Furthermore, it is worth emphasizing that Spin-Scenario is able to perform more complicated acquisition, any of interested density states can be specified as the observer. For example, `acq{np = 1024, sw = 1e4, observer = "2*I1zI2z"}` can be used to acquire projection of the evolution state of spin system to the observation state $2I_{1z}I_{2z}$.

Phase cycling are commonly used for either selecting NMR signals that have certain properties of interest, while removing other types of signal or suppressing spurious signals generated by imperfections in the spectrometer hardware. A readable phase cycle table such as `"xy-x-y"` or `{x,y,-x,-y}` can be directly assigned to the `phase` key in `hardRF` and `acq` blocks.

2.4.2. sequence assembly

In Spin-Scenario, each sequence always starts with a keyword `seq`, followed by a pair of braces, containing the sequence body. The sub-blocks are separated by default in the body via commas or semicolons (`s1`). The concurrent or parallel block can be simply assembled through an arithmetic operator `+`. E.g. `rf + gz` defines a selective excitation event, and `gx + gy + gz` defines a simultaneous `X/Y/Z` gradient event. The newly built concurrent blocks can be directly inserted into any position of the pulse sequence (`s2`). The sequence can be also written in multi-line style, which may be intuitive for more sophisticated sequences.

```
local s1 = seq{d1, rf, acq}
local s2 = seq{rf + gz,
               gxPre + gyPre + gzReph,
               gx + acq,
               delay}
```

It should be noted that pulse sequences can be also nested inside any other sequence, since they themselves are essentially serial blocks. This feature allows the reuse of predesigned pulse sequence templates, which may further improve the scripting efficiency.

2.4.3. Arrayed experiments

Arrayed experiments are widely employed in both MR imaging and MR spectroscopy. To this end, a simple and flexible loop syntax is specially designed. Basically, the sequence repetitions include both global and local loops. A specific block followed by a global symbol **#** indicates a series of separate experiments is conducted, with the block taking a set of different values. E.g. `seq{rf90I, tau#, rf180, tau#, rf90, acq}` describes a modified INEPT arrayed experiments where the S-spin signal is observed immediately after it has been transferred from I for a set of varied time in delay block `tau`.

The above idea is also shared with phase cycling for r.f. pulses as well as signal acquisition. Consider, for example, a suitable table for the spin echo experiment `seq{rf90#, t1, rf180#, t2, acq#}` is given in Table 2. The phases for `rf90`, `rf180` and `acq` are `"x"`, `"xy-x-y"` and `"x-x"` respectively, so the total number of steps ($n = 4$) in the phase cycle is dominated by the maximum cycle counts of all relevant blocks, while other blocks with smaller cycle number will repeat themselves during the complete repetitions.

Further, to perform two-dimensional, three-dimensional or higher-dimensional experiments in Spin-Scenario, the arraying concept was generalized by assigning an additional cycle priority behind the symbol **#**. Then the experimental hierarchy can be logically divided into levels like outer level and inner level. Taking a two-dimensional COSY sequence enclosing phase cycling [20] for example, as shown in Fig. 1, each row of the data matrix is the result of a complete set of phase cycled experiments, all with the same value of t_1, but with similar cycling of the phases ϕ_1 , ϕ_2 and ϕ_3 according to the phase Table 2. When acquisition of one row is completed, the variable delay t_1 is changed, and the acquisition procedure is repeated. The kernel script of this experiment is then `seq{rf90#2, t1#1, rf180#2, acq#2}`. The biggest priority number always indicates the innermost loop, which represents the phase cycling in this case.

For local loops definition within each TR repetition, e.g. multi-echo acquisitions in EPI sequence `s3`, the symbol ~ is adopted to represent the repetitive echo-train within the effective TE. Similar to the symbol **#**, the local loop count is automatically determined by the specific varied block `gx`.

```
local s3 = seq{rf + gz,
               gxPre + gyPre# + gzReph,
               delay1,
               (gx + gy + acq)~,
               delay2}
```

The syntax for pulse sequence programming is minimalism but highly flexible. The isolated definition of individual blocks, together with one-line assembly of sequence, offer a WYSIWYG way of understanding the pulse sequence structure.

2.5. Optimal control

The primary goal of optimal control of spin system is usually to maximize fidelity to desired state or unitary using optimized pulse shape, with additional constraints associated with specific experimental requirements (e.g., B0 and rf inhomogeneity, rf power limitation). There have been numerous methods for this purpose, and the algorithms involved are mostly based on gradient methods, such as GRAPE [21] and Krotov [22]. In order to quickly develop different kinds of control strategies, we implemented a novel optimization scheme based on Google's TensorFlow [16], which supports building up computational dataflow graph for optimal control problems as well as evaluating their gradients efficiently via reverse-mode automatic differentiation. The above features are well suited for solving rapid user-defined optimization problem.

Fig. 2 shows the computational network graph for optimal control of spin system in the library. The r.f. control fields are discretized into N steps, each of which is sufficient small time step δt and the corresponding optimization is to minimize the custom cost function. There are many cost function contributions [21,23] such as state transfer Φ_0, unitary transfer Φ_1, r.f. amplitude suppression Φ_2 and intermediate forbidden state Φ_3, a linear combination of them forms the total cost function Φ. For simplicity, the graph only shows the calculation of the loss functions Φ_0, Φ_2 and Φ_3.

Table 2
A four-step phase cycle for a spin echo experiment.

Cycle counter	`rf90#`:ϕ_1	`rf180#`:ϕ_2	`acq#`:ϕ_3
0	0	0	0
1	0	$\pi/2$	π
2	0	π	0
3	0	$3\pi/2$	π

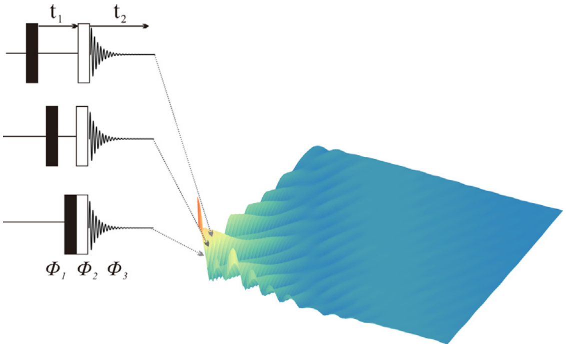


Fig. 1. Illustration of two-dimensional correlation spectroscopy (COSY).

It is straightforward to define the custom cost function upon the computational graph template by Python. When the final cost function Φ is ready, reverse-mode automatic differentiation of TensorFlow enables time evolution and cost function evaluation by one forward sweep through the graph, and automatic calculation of the full gradients $\partial\Phi/\partial\mathbf{u}$ by one backward sweep. The scheme is particularly convenient for different user-defined optimizations, because only the cost functions are required, without needing to manually derive their corresponding gradients.

2.6. Utility tools

2.6.1. Time-frequency analysis of shaped pulses

Time-frequency analysis comprises those techniques that study a signal in both the time and frequency domains simultaneously, using various time-frequency representations, such as Wigner-Ville, short-time Fourier transform (STFT) and wavelet transforms. The time-frequency analysis of MR pulse shapes can help to understand their underlying mechanisms of action [19]. To visualize

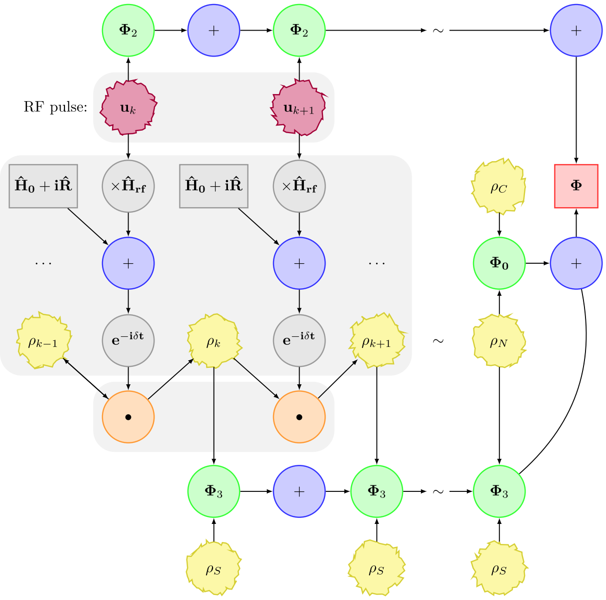


Fig. 2. Graph model for optimal control of spin system. The graph can be resolved into basic operations such as matrix multiplication, addition, matrix exponential, trace, inner product, and absolute value, all denoted by circular nodes. TensorFlow supports directly calculating the backward gradients ($\partial\Phi/\partial\mathbf{u}$). The further optimization can be solved either by TensorFlow itself or third-party libraries such as NLopt.

both the magnitude and phase spectrogam of shaped or composite pulses, we implemented a STFT-based approach, as demonstrated in Eq. (3):

$$\mathbf{STFT}(\tau,\omega) = \int_{-\infty}^{\infty} \mathbf{u}(t)\omega(t-\tau)e^{-j\omega t}dt \tag{3}$$

$\mathbf{u}(t)$ represents the complex pulse shape in which the real and imaginary part are x and y components of r.f. field respectively, and $\omega(t)$ is the window function, commonly a Gaussian window centered around zero. The usage script is readable as `specgram {rf, window = "gauss", wlen = 128, overlap = 0.96, nfft = 1024}`, in which `window` and `wlen` denotes the truncated window shape and its width, `overlap` denotes the amount of overlap between neighbouring windows, and `nfft` is for the number of FFT points.

The tool is especially useful to reveal the hidden structures for the seemingly random but highly optimized pulses, as demonstrated in Section 3.3.

2.6.2. Visualization

It is essential that the Spin-Scenario offers necessary script syntax to help visualizing pulse sequences including their individual blocks, phantom models and simulation results, etc. Most of these are developed based on gnuplot, a famous scientific plotting package, features include 2D and 3D plotting, a huge number of output formats, interactive input or script-driven options, and a large set of scripted examples. Specifically, a unique `plot` function was provided, e.g. `plot(seq)` shows an overall perspective of a specific pulse sequence, while `plot(rf1, rf2, gx)` will display a list of given sub-blocks (such as r.f. and gradient pulses) in detail respectively. Note that in the latter case both order and number of the block inputs are unlimited. The 1D or 2D simulation results can be also visualized with `plot`, which offers user a great deal of flexibility and freedom to produce high quality figures for publication.

3. Examples

3.1. Spectrum of acrylic acid

The Spin-Scenario was developed to provide the community an intuitive, flexible and unique scripting environment for rapid prototyping. For this purpose, the scenario script concept was fully employed. Here we demonstrate a simple spectrum acquisition of acrylic acid, in which the chemical shifts of the three proton nuclei are $\omega_1 = 0, \omega_2 = 88.42, \omega_3 = 214.9$ Hz on a 500 MHz spectrometer, and the coupling constants between protons are $J_{12} = 10.4$ Hz and $J_{13} = 1.2$ Hz and $J_{23} = 17.4$ Hz respectively. In general, a completed spin scenario includes at least three parts: spin system generation, pulse sequence assembly and the final experimental study. The minimal script is as follows:

```
-- scenario A: spin system generation.
B0{"500 MHz"}
local acrylic =spin_system{
    spin ="1H 1H 1H",
    zeeman ="2 scalar 88.42 Hz
             3 scalar 214.9 Hz",
    jcoupling ="1 2 scalar 10.4 Hz
                1 3 scalar  1.2 Hz
                2 3 scalar 17.4 Hz"
}
-- scenario B: pulse sequence assembly.
local rf45 =hardRF{beta =45}
local adc =acq{np =1024, sw =500}

local fid =seq{rf45, adc}
-- scenario C: experimental study.
local result =run{exp =fid, spinsys =acrylic}
```

As long as the spin system and the experimental pulse sequence are well prepared, the simulation task can be done by readily applying `run` command, and then all data will be either preserved in memory for ongoing process or stored into HDF5 files for further processing by Matlab or others.

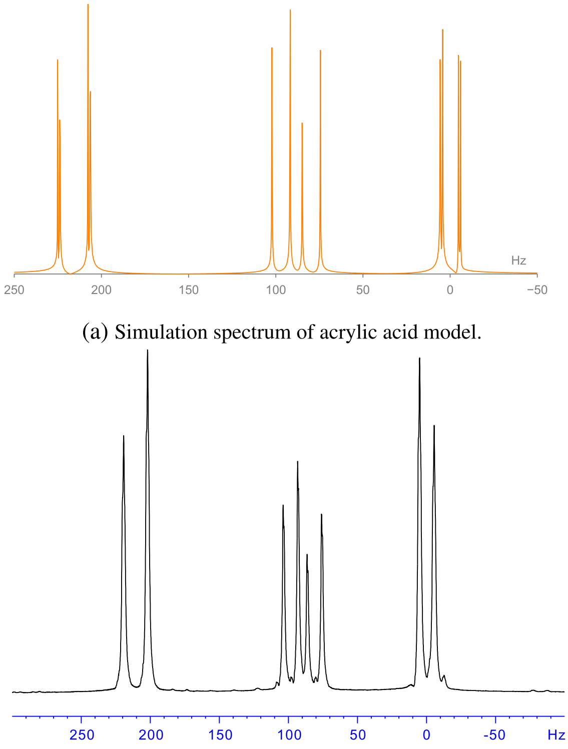
Fig. 3 shows that the simulation spectrum agrees well with experiment result. The doublets generated between spin 1 and spin 3 are imperceptible in the experimental spectrum due to their smaller coupling constant (1.2 Hz) and the relaxation broadening.

3.2. Multidimensional spatially selective excitation MRI

Multidimensional spatially selective pulses [24,25] are used in a variety of MRI applications for excitation, saturation, inversion and refocusing. By simultaneously applying a time-varying selection gradient and an r.f. waveform during excitation, imaging in a reduced field of view can be realized. The gradients define a trajectory through k-space, and, by depositing r.f. energy at discrete locations in k-space, a specific excitation pattern in the spatial domain is produced [26,27]. For the purpose of demonstration, a two-dimensional selective excitation pulse is adopted to excite a cylinder slice within the MIDA phantom [28], of which the voxel- and the surface-based versions of the models are freely available to the scientific community. The gird size of the phantom is 480∗480∗350, with a spatial resolution of 0.5 mm.

Generally, necessary global declarations for the system limits such as static field strength, peak gradient strength and the slew rate should be firstly added to the script.

```
B0{"3T"}
peak_grad{40} -- mT/m
slew_rate{200} -- T/m/s
```



(a) Simulation spectrum of acrylic acid model.

(b) Experimental spectrum of acrylic acid.

Fig. 3. Experimental and simulation of acrylic acid.

For the k-space trajectory, a constant-angular-rate spiral that ends at the origin is chosen to refocus the slice automatically. This k-space trajectory can be written as [25]

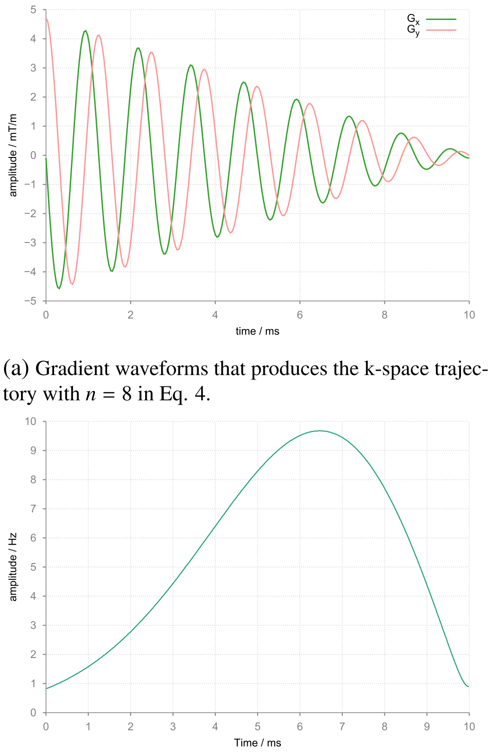
$$\begin{aligned}\mathbf{k}_x(t) &= A\left(1-\tfrac{t}{T}\right)cos\,\tfrac{2\pi nt}{T}\\ \mathbf{k}_y(t) &= A\left(1-\tfrac{t}{T}\right)sin\,\tfrac{2\pi nt}{T}\end{aligned} \qquad (4)$$

where n represents cycle number in the spiral in duration T, and A determines the size of the spiral in spatial frequency. The corresponding gradient waveforms can be calculated by $\mathbf{G}(t) = \frac{1}{\gamma}\dot{\mathbf{k}}(t)$, so it is straightforward to generate the two gradient waveforms via `exprGrad` as below, the gradient shapes of 10 ms with $n = 8$ are shown in Fig. 4(a).

```
local gxRf = exprGrad{axis ="X", width =8, expr =
    "D(t) 1e6/gamma1H*40*(1-t/10)*Cos(2*Pi*8*t
    /10)"} -- mT/m
local gyRf = exprGrad{axis ="Y", width =8, expr =
    "D(t) 1e6/gamma1H*40*(1-t/10)*Sin(2*Pi*8*t
    /10)"} -- mT/m
```

The required selective r.f. waveform is given by $B_1(t) = W(\mathbf{k}) \mid \gamma\mathbf{G}(t) \mid$, in which $W(\mathbf{k})$ is a spatial frequency weighting function whose Fourier transform is the desired localization. For this case, a circularly symmetric Gaussian function $e^{-\kappa^2(k_x^2+k_y^2)/A^2}$ can be used, where κ determines the spatial resolution of the selective volume. The r.f. waveform is further scaled to the desired flip angle. The final selective pulse can be directly created by `shapedRF`:

```
local rf =shapedRF{width =10, step =500, beta=20,
    pattern ="40/10*Exp(-2^2*(1-t/10)^2)*Sqrt((2
   *Pi*8*(1-t/10))^2+1)"}
```



(a) Gradient waveforms that produces the k-space trajectory with $n = 8$ in Eq. 4.

(b) The two-dimensional selective r.f. pulse that will produce a cylindrical Gaussian weighting of k space when applied with the gradient waveforms in 4a.

Fig. 4. The two-dimensional spatial selective r.f. pulse and the correlative gradient waveforms for the spiral k-space trajectory.

The two-dimensional selective excitation pulse sequence is illustrated in Fig. 5 and its vital script is as below:

```
seqParam{fov ="240*240", matrix ="256*256"}
local SpSel =seq{rf + gxRf + gyRf, d1, rf180 +
    gzSlice, gxPre + gyPhase#, gxRead + adc, d2}
local result =run{exp =SpSel, phantom ="mida.h5"}
```

The procedure is similar to the previous spectrum acquisition, except for specifying a `phantom` model instead of the `spinsys`.

When the two-dimensional selective excitation (`rf + gxRf + gyRf`) is applied to the x and y axes, a cylinder along the z axis will be excited. A slice of this cylinder can be further selected using a slice-selective 180° refocusing pulse (`rf180 + gzSlice`), and finally the localized volume can be imaged by conventional imaging sequence. A comparison of images produced by both the spatial selected sequence and the spin echo sequence are described in Fig. 6.

3.3. Cooperative pulses optimization for pseudo-pure state preparation

Cooperative pulses [29] are simultaneously optimized pulses that compensate each others imperfections, a promising alternative to conventional phase cycles in many multi-scan experiments. Similar to the previous single pulse optimization scheme, we have implemented a multi-scan cooperative pulses optimization scheme for general coupled spin system.

To demonstrate, we optimized a 7-scan cooperative pulses for three-qubit pseudo-pure state preparation in [30]. The pseudo-pure states have been widely used as initial states in NMR quantum information processing experiments. As the starting point, the fidelity of the initial state preparation is critical for quantum computation. For a three spin system (spins denoted as I_1, I_2 and I_3), the traceless part of the initial thermal equilibrium state is $I_{1z} + I_{2z} + I_{3z}$, and the projection operator I_i^α $(I_i^\alpha = \frac{1}{2}\mathbf{1} + I_{iz}$, $i = 1, 2, 3)$ is corresponding to a pseudo-pure state, the desired preparation state is then $I_1^\alpha I_2^\alpha I_3^\alpha$, which can be expressed as a sum of following eight terms: $I_{1z} + I_{2z} + I_{3z} + 2I_{1z}I_{2z} + 2I_{1z}I_{3z} + 2I_{2z}I_{3z} + 4I_{1z}I_{2z}I_{3z} + \frac{1}{2}\mathbf{1}$.

Here we choose ^{13}C labeled alanine as the sample for the experimental pseudo-pure state preparation. The chemical shifts of the three carbon nuclei are $\omega_1 = 15.74, \omega_2 = 0, \omega_3 = 4.302$ kHz on a

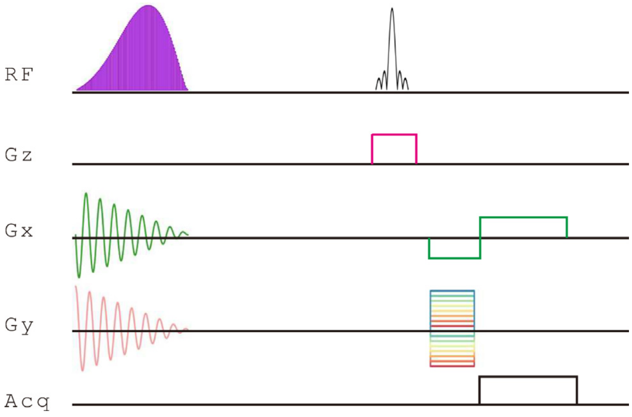


Fig. 5. The two-dimensional selective excitation pulse sequence.

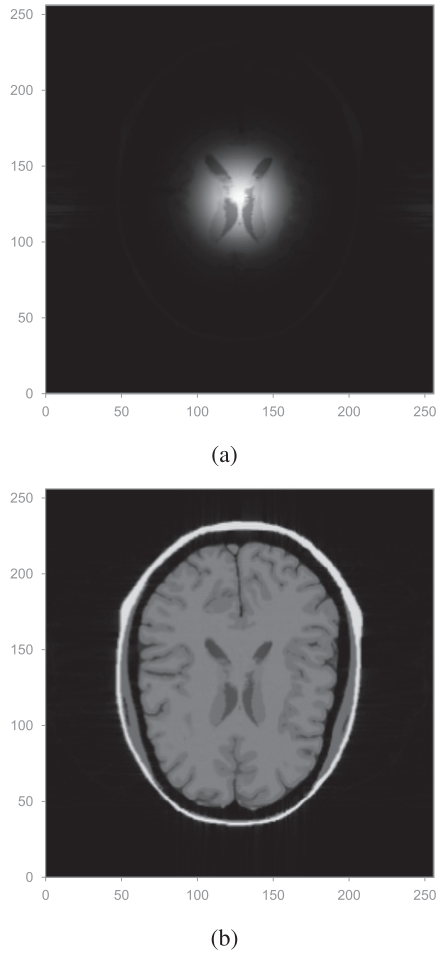


Fig. 6. Images resulted from the spatial selected pulse sequence and the conventional spin echo pulse sequence. (a) The two-dimensional selective excitation had a duration of 10 ms, and a peak gradient amplitude of 4.5 mT/m. The r.f. was scaled to produce a 20° tip angle. The field of view is 24 cm, and the diameter of the selected volume is approximately 6 cm. (b) T_1-weighted image generated by the conventional spin echo pulse sequence (TE = 10 ms, TR = 350 ms) with the same field of view.

500 MHz spectrometer. The coupling constants between carbons are $J_{12} = 54.2$ Hz and $J_{23} = 35.4$ Hz. By setting the pulse duration and the number of pulse scans, a set of COOP pulses can be readily optimized with the following script snippet.

```
local sys = spin_system{
    spin ="13C 13C 13C",
    zeeman ="1 scalar  15.74 kHz
             3 scalar -4.302 kHz",
    jcoupling ="1 2 scalar 52.2 Hz
                2 3 scalar 35.4 Hz"
}
local oc = multi_rf optimizer(sys)
local multi_rf =oc:optimize{
    ncoop =7, width =18.5, step =1850,
    init_state ="I1z+I2z+I3z",
    targ_state ="I1z+I2z+I3z+2*I1zI2z+2*I1zI3z
                         +2*I2zI3z+4*I1zI2zI3z"}

oc:projection{init_state ="I1z+I2z+I3z", rf =
    multi_rf, observ_states ={"I1z", "I2z", "I3z"
    , "2*I1zI2z", "2*I1zI3z", "2*I2zI3z", "4*
    I1zI2zI3z"}}
```

We also provides a similar `projection` function to observe the evolution trajectory components of the average normalized state $\bar{\rho}(t)$ within the cooperative pulses, clearly indicating that the desired average pseudo-pure state is prepared, as shown in Fig. 7.

The optimized COOP pulses shown in Fig. 8 seem to be random but highly modulated. With the benefit of `specgram` function, it is obvious that the COOP pulses contain three frequency positions corresponding to the chemical shifts of each spin, as shown in Fig. 9.

4. Discussion

Spin-Scenario internally used quantum formalism of spin dynamics to make it compatible with different applications. The major cost of the simulations is dominated by the processing of time evolution of the spin system, and thus normally approximating the matrix exponential $e^{-i\left(\widehat{L}+i\widehat{R}\right)\Delta t}$ for the propagators at each time step. However, the exponential of a sparse matrix is generally full, so it is not practical especially when the sparse Hamiltonian matrix is large. In Liouville space, we directly calculate the action of matrix exponential on the state vector $e^{-i\left(\widehat{L}+i\widehat{R}\right)\Delta t}\rho$, which leads to considerable cost reduction. Moreover, for cases like constant Hamiltonians during acquisition, the intermediate states on the equally spaced grid of acquisition points can be efficiently produced [31], thus it could reduce computation load by a substantial margin.

To compare the performance of the above time evolution algorithms, benchmark tests were done with a FID acquisition sequence for a set of 1H spin systems ranging from 1 spin to 7 spins. All tests were performed on a PC with Intel Xeon(R) CPU E5-2690 v3, 2.60 GHz, 64 GB RAM, running Ubuntu 16.04. Fig. 10 indicates that the computational time can be dramatically reduced with the two improvements even only one thread was used through out all the tests.

The package also supports computing with OpenMP parallelism, which can further accelerate the computing within a multi-core PC. Fig. 11 shows another benchmark testing on spin echo sequence for 20 thousand 1H spins using the OpenMP parallelism, it should be noted that the linear speedup is difficult to achieve because more threads introduce a certain amount of overhead.

One of major developments of the package is still to improve computational efficiency for larger spin system in NMR and huge spin ensemble in MRI. Advanced improving methods include utilizing massively parallel multiprocessing on GPUs like CUDA and MPI on computer clusters. Furthermore, using the reduced subspace to

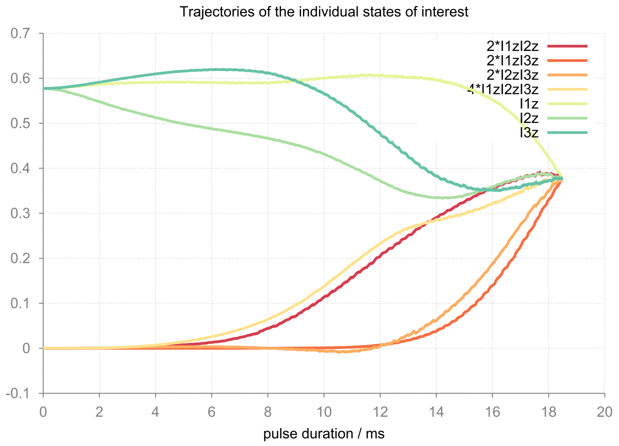


Fig. 7. Magnetization trajectories of the individual states of interest in the average normalized state $\bar{\rho}(t)$. Initially, $\bar{\rho}(0)$ (thermal state) contains only the first three terms with identical coefficient $\frac{1}{\sqrt{3}}$, then their corresponding weights decrease while the weights of the remaining four terms increase, and the weights of all terms approach $\frac{1}{\sqrt{7}}$ at the end of the pulse duration.

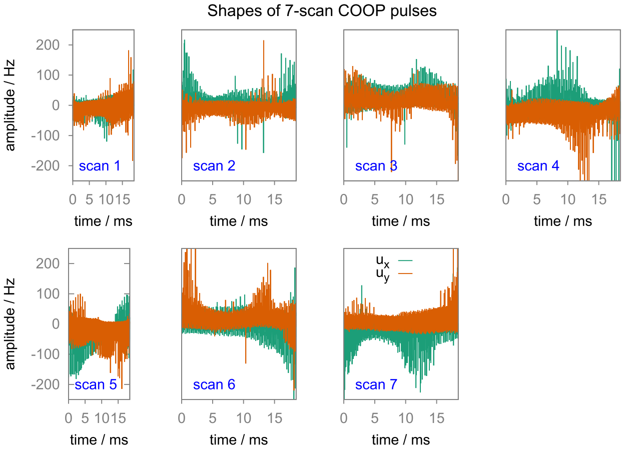


Fig. 8. The time domain shapes of 7-scan COOP pulses (x,y components).

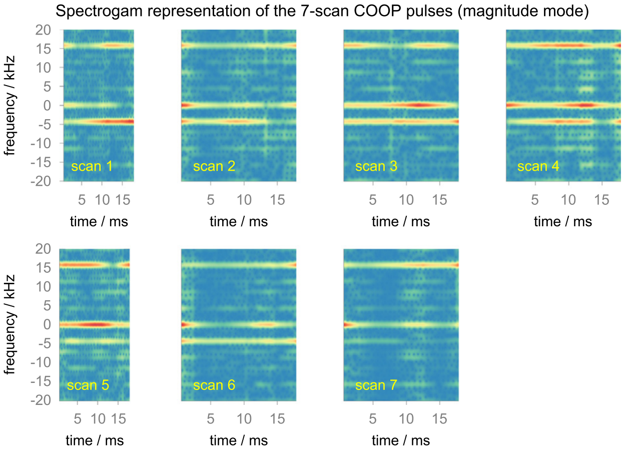


Fig. 9. The time-frequency STFT representation of the 7-scan COOP pulses.

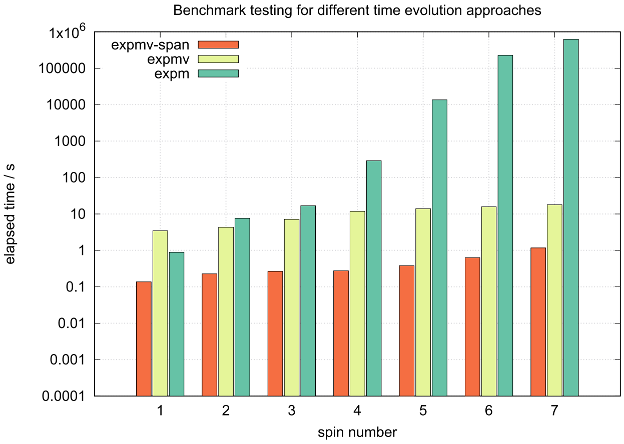


Fig. 10. Benchmark histogram for three time evolution methods (1) conventional matrix exponential (expm); (2) direct calculation of the action of matrix exponential on the state vector (expmv); (3) with addtional intermediate states calculation on the equally spaced grid of acquisition points (expmv-span). The parameters for the FID sequence are 45° flip angle, 4096 sampling points with 10 kHz bandwidth. All the linear algebra of matrix operations are based on Eigen [32], a fast C++ template library.

calculate operators [33,34] could also remarkably decrease the computational time. Other future developments include enriching pulse sequence library, exporting pulse sequences to MR scanner, launching Spin-Scenario cloud version, etc.

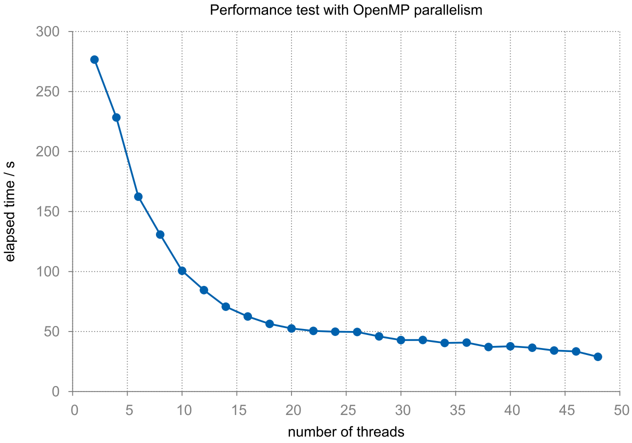


Fig. 11. The speedup performance with different thread number. The parameters for the SE sequence are TR 100 ms, TE 10 ms, 256 sampling points with 25 kHz bandwidth.

5. Conclusions

In this paper, a package Spin-Scenario is described which provides a unique scripting environment to cover many aspects of MR simulations in both imaging and spectroscopy. The experimental scenarios can be conveniently written and simulated within a Lua-based script, which includes lots of commonly used modules such as spin system definition, pulse sequence programming, single or multi-pulse optimization, and shaped pulses characterization. Specially, the pulse sequence programming syntax is easy, intuitive and flexible for rapid-sequence prototyping. In combination with the general Liouville space computing model, this package can be very powerful and productive for the research community.

Spin-Scenario is provided as an open-source project, and will be continuously developed for highly usability as well as high-performance.

Acknowledgments

This work was supported by National Natural Science Foundation of China [11505281, 11675254], General Social Development Project of Jiangsu [BE2017670] and Science and Technology for People's Livelihood of Suzhou [SS201746].

References

[1] T.H. Jochimsen, M. Von Mengershausen, ODIN-object-oriented development interface for NMR, J. Magn. Reson. 170 (2004) 67–78.

[2] W. Overall, J. Pauly, An extensible, graphical environment for pulse sequence design and simulation, Int. Soc. Magn. Reson. Med., Berlin, Germany, 2007.

[3] T. Stöcker, K. Vahedipour, D. Pflugfelder, N.J. Shah, High-performance computing MRI simulations, Magn. Reson. Med. 64 (2010) 186–193.

[4] F. Liu, J.V. Velikina, W.F. Block, R. Kijowski, A.A. Samsonov, Fast realistic MRI simulations based on generalized multi-pool exchange tissue model, IEEE Trans. Med. Imaging 36 (2017) 527–537.

[5] A. Jerschow, MathNMR: spin and spatial tensor manipulations in mathematica, J. Magn. Reson. 176 (2005) 7–14.

[6] M. Veshtort, R.G. Griffin, SPINEVOLUTION: a powerful tool for the simulation of solid and liquid state NMR experiments, J. Magn. Reson. 178 (2006) 248–282.

[7] Z. Tošner, T. Vosegaard, C. Kehlet, N. Khaneja, S.J. Glaser, N.C. Nielsen, Optimal control in NMR spectroscopy: numerical implementation in SIMPSON, J. Magn. Reson. 197 (2009) 120–134.

[8] H. Hogben, M. Krzystyniak, G. Charnock, P. Hore, I. Kuprov, Spinach – a software library for simulation of spin dynamics in large spin systems, J. Magn. Reson. 208 (2011) 179–194.

[9] H. Jara, Theory of Quantitative Magnetic Resonance Imaging, World Scientific, 2013.

[10] S. Chandrashekar, N.M. Trease, H.J. Chang, L.-S. Du, C.P. Grey, A. Jerschow, 7Li MRI of Li batteries reveals location of microstructural lithium, Nat. Mater. 11 (2012) 311–315.

[11] N.J. Shah, W.A. Worthoff, K.-J. Langen, Imaging of sodium in the brain: a brief review, NMR Biomed. 29 (2016) 162–174.

[12] S. Posse, R. Otazo, S.R. Dager, J. Alger, MR spectroscopic imaging: principles and recent advances, J. Magn. Reson. Imaging 37 (2013) 1301–1325.

[13] I. Kuprov, Fokker-planck formalism in magnetic resonance simulations, J. Magn. Reson. 270 (2016) 124–135.

[14] J.F. Magland, C. Li, M.C. Langham, F.W. Wehrli, Pulse sequence programming in a dynamic visual environment: SequenceTree, Magn. Reson. Med. 75 (2016) 257–265.

[15] R. Ierusalimschy, L.H. De Figueiredo, W. Celes Filho, Lua-an extensible extension language, Softw. Pract. Exper. 26 (1996) 635–652.

[16] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: a system for large-scale machine learning., in: OSDI, vol. 16. pp. 265–283.

[17] A. Pinkus, S. Winitzki, J. Niesen, The Yacas Computer Algebra System, 2012.

[18] P.E. Spindler, Y. Zhang, B. Endeward, N. Gershernzon, T.E. Skinner, S.J. Glaser, T. F. Prisner, Shaped optimal control pulses for increased excitation bandwidth in EPR, J. Magn. Reson. 218 (2012) 49–58.

[19] S. Köcher, T. Heydenreich, S. Glaser, Visualization and analysis of modulated pulses in magnetic resonance by joint time–frequency representations, J. Magn. Reson. 249 (2014) 63–71.

[20] M.H. Levitt, Spin Dynamics: Basics of Nuclear Magnetic Resonance, John Wiley & Sons, 2001.

[21] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, S.J. Glaser, Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms, J. Magn. Reson. 172 (2005) 296–305.

[22] I.I. Maximov, Z. Tošner, N.C. Nielsen, Optimal control design of nmr and dynamic nuclear polarization experiments using monotonically convergent algorithms, J. Chem. Phys. 128 (2008) 184505.

[23] N. Leung, M. Abdelhafez, J. Koch, D. Schuster, Speedup for quantum optimal control from automatic differentiation based on graphics processing units, Phys. Rev. A 95 (2017) 042318.

[24] C.J. Hardy, H.E. Cline, Spatial localization in two dimensions using nmr designer pulses, J. Magnet. Reson. (1969) (82) (1989) 647–654.

[25] J. Pauly, D. Nishimura, A. Macovski, A k-space analysis of small-tip-angle excitation, J. Magn. Reson. (1969) 81 (1989) 43–56.

[26] C. Schröder, P. Börnert, B. Aldefeld, Spatial excitation using variable-density spiral trajectories, J. Magn. Reson. Imaging 18 (2003) 136–141.

[27] M.S. Vinding, I.I. Maximov, Z. Tošner, N.C. Nielsen, Fast numerical design of spatial-selective rf pulses in MRI using krotov and quasi-newton based optimal control methods, J. Chem. Phys. 137 (2012) 054203.

[28] M.I. Iacono, E. Neufeld, E. Akinnagbe, K. Bower, J. Wolf, I. Vogiatzis Oikonomidis, D. Sharma, B. Lloyd, B.J. Wilm, M. Wyss, K.P. Pruessmann, A. Jakab, N. Makris, E.D. Cohen, N. Kuster, W. Kainz, L.M. Angelone, Mida: a multimodal imaging-based detailed anatomical model of the human head and neck, PLOS ONE 10 (2015) 1–35.

[29] M. Braun, S.J. Glaser, Cooperative pulses, J. Magn. Reson. 207 (2010) 114–123.

[30] D. Wei, Y. Chang, S.J. Glaser, X. Yang, Cooperative pulses for pseudo-pure state preparation, Appl. Phys. Lett. 104 (2014) 242409.

[31] A. Al-Mohy, N. Higham, Computing the action of the matrix exponential, with an application to exponential integrators, SIAM J. Sci. Comput. 33 (2011) 488–511.

[32] G. Guennebaud, B. Jacob, et al., Eigen v3, http://eigen.tuxfamily.org, 2010.

[33] I. Kuprov, Polynomially scaling spin dynamics ii: further state-space compression using Krylov subspace techniques and zero track elimination, J. Magn. Reson. 195 (2008) 45–51.

[34] A. Karabanov, I. Kuprov, G.T.P. Charnock, A. van der Drift, L.J. Edwards, W. Köckenberger, On the accuracy of the state space restriction approximation for spin dynamics simulations, J. Chem. Phys. 135 (2011) 084106.