Journal of Magnetic Resonance 246 (2014) 79–93

Contents lists available at ScienceDirect

Journal of Magnetic Resonance

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

Computer-intensive simulation of solid-state NMR experiments using SIMPSON

Zdeněk Tošner [a,b,*], Rasmus Andersen [a], Baltzar Stevensson [c], Mattias Edén [c], Niels Chr. Nielsen [a,*], Thomas Vosegaard [a,*]

[a] Center for Insoluble Protein Structures (inSPIN), Interdisciplinary Nanoscience Center (iNANO) and Department of Chemistry, Aarhus University, Gustav Wieds Vej 14, DK-8000 Aarhus C, Denmark
[b] NMR Laboratory, Department of Chemistry, Faculty of Science, Charles University in Prague, Hlavova 8, CZ-128 43, Czech Republic
[c] Physical Chemistry Division, Department of Materials and Environmental Chemistry, Arrhenius Laboratory, Stockholm University, SE-106 91 Stockholm, Sweden

ARTICLE INFO

Article history:
Received 27 May 2014
Revised 8 July 2014
Available online 22 July 2014

Keywords:
SIMPSON
Simulation of solid-state NMR experiments
Fast powder averaging
Cloud computing
Optimal control pulse sequence optimization

ABSTRACT

Conducting large-scale solid-state NMR simulations requires fast computer software potentially in combination with efficient computational resources to complete within a reasonable time frame. Such simulations may involve large spin systems, multiple-parameter fitting of experimental spectra, or multiple-pulse experiment design using parameter scan, non-linear optimization, or optimal control procedures. To efficiently accommodate such simulations, we here present an improved version of the widely distributed open-source SIMPSON NMR simulation software package adapted to contemporary high performance hardware setups. The software is optimized for fast performance on standard stand-alone computers, multi-core processors, and large clusters of identical nodes. We describe the novel features for fast computation including internal matrix manipulations, propagator setups and acquisition strategies. For efficient calculation of powder averages, we implemented interpolation method of Alderman, Solum, and Grant, as well as recently introduced fast Wigner transform interpolation technique. The potential of the optimal control toolbox is greatly enhanced by higher precision gradients in combination with the efficient optimization algorithm known as limited memory Broyden–Fletcher–Goldfarb–Shanno. In addition, advanced parallelization can be used in all types of calculations, providing significant time reductions. SIMPSON is thus reflecting current knowledge in the field of numerical simulations of solid-state NMR experiments. The efficiency and novel features are demonstrated on the representative simulations.

1. Introduction

Numerical simulations are playing an increasingly important role in the development and exploitation of solid-state NMR spectroscopy for identification and analysis of molecular structure and dynamics within broad areas of chemistry, materials science, nanotechnology, molecular biology, biology, and medicine [1,2]. Governed by the development of increasingly powerful computer systems, simulation software packages [3–18], and efficient numerical algorithms for fast numerical simulations [9,12,19–32], the area of applications of solid-state NMR numerical simulations is steadily increasing. Recent interests involve, for example, optimal control [33,34] experiment design [16,35–49] and spectral analysis of very large spin systems [17,27–32] to be performed on a more routinely basis.

The continuous challenge of increasingly complex simulations tasks provides strong impetus for improving the performance (speed, flexibility, and robustness) of numerical simulations including optimization of all elements of the simulation process. This concerns optimization of the "stand-alone" software code for more efficient computation of (i) the fundamental mathematical operations, (ii) efficient powder averaging [50–54], (iii) handling of combinations with analytical coherent averaging [55–58], and (iv) realization of improved performance through parallelization and improvement of the interface and adaptation of the software to contemporary computer platforms for high-performance-computing.

With attention to the widely used open-source SIMPSON (SIMulation Package for SOlid-state NMR spectroscopy) simulation

[*] Corresponding authors. Address: NMR Laboratory, Department of Chemistry, Faculty of Science, Charles University in Prague, Hlavova 8, CZ-128 43, Czech Republic (Z. Tošner).

E-mail addresses: tosner@natur.cuni.cz (Z. Tošner), ncn@inano.au.dk (N.Chr. Nielsen), tv@chem.au.dk (T. Vosegaard).

http://dx.doi.org/10.1016/j.jmr.2014.07.002
1090-7807/© 2014 Elsevier Inc. All rights reserved.

program [7], we will in this article follow the above routes and present a significantly speed-enhanced version of SIMPSON for computer intensive calculations. Our address to different computer architectures includes stand-alone multiprocessor computers, large computer clusters, as well as grids and cloud computing. In addition to adaptation of SIMPSON to such parallel computing platforms, we have also significantly enhanced the native speed of the program by improving the underlying algorithms and by including new features for frequency domain calculations and for rapid computations of powder spectra by the Alderman, Solum, and Grant (ASG) [59] and fast Wigner transform (FWT) [60] interpolation techniques. On the side of optimal control pulse sequence design, significant improvement is achieved by using higher order approximation to gradients [61] and employing efficient limited memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) optimization algorithm [62]. The novel SIMPSON package includes a wide collection of numerical methods for simulation of solid-state NMR experiments, which are released with the GNU license as an open project hosted at https://code.google.com/p/simpson.

2. Enhanced speed through optimization of internal operations

It is obvious that the efficiency of numerical simulations critically depends on the speed of the internal numerical procedures executing key spin dynamics operations. We will focus exclusively on simulations performed in Hilbert space. Given the scaling of matrices with the number of spins, we have a limitation due to computer memory to systems of up to 12–15 spins. Such a barrier can be circumvented to some extent by making use of inherent sparse structure of the matrices (see below) but we still get limited by complexity of solid-state NMR simulations and powder averages, leading to prohibitively long calculations for more than say 15 spins on current computer resources. Here we present improvements of numerous internal procedures relative to the initial versions of SIMPSON leading to orders of magnitude reduction of calculation time.

SIMPSON simulations involve extensive use of *matrix–matrix multiplications* and other calculation-intense matrix operations, which now are based on external optimized linear algebra libraries such as the basic linear algebra subprograms (BLAS) [63] and the linear algebra package (LAPACK) [64], which for larger spin systems provide enhanced performance relative to previous implementations even without addressing specific computer architectures. Typically, however, the performance may be further enhanced using mathematical libraries optimized for specific computer architectures, including, for example, the Automatically Tuned Linear Algebra Software (ATLAS) [65] and Math Kernel Library (MKL) [66] for Intel processors, the AMD Core Math Library (ACML) [67] for AMD processors, and the system-distributed Accelerate framework for Apple computers [68]. Such proprietary/commercial libraries, which replace the more general ones, may readily be implemented by the user through compilation and linking of the distributed open-source SIMPSON software.

Next to matrix multiplications the other time-consuming operation is calculation of the propagator, i.e., the *exponential operator* of the Hamiltonian, $U = \exp(-iH\Delta t)$, which is evaluated repeatedly by approximating the time-dependent H as a piece-wise constant function. Evaluation of the matrix exponential may be accomplished in several ways [58,69], and we have implemented and tested a number of these to find the fastest and most robust. We considered (i) matrix diagonalization by eigenvalue decomposition followed by direct exponentialization, (ii) scaling-and-squaring with the Padé approximation, (iii) expansion in a series of Chebyshev polynomials, and (iv) Taylor series expansion. The performance of each method depends on the particular implementation – therefore we will describe them in more detail.

(i) As a consequence of the secular approximation, the total Hamiltonian can always be expressed as a real and symmetric matrix (even for rf pulses, where imaginary parts may be avoided using a z rotation). There are several methods to diagonalize such a matrix. The efficient algorithms first reduce the matrix to a tridiagonal form and then find its eigenvalues and eigenvectors. The most time-wise expensive operation is the first transformation as it scales with the matrix dimension to the third power, while the eigenvalues and -vectors of a tridiagonal matrix are then easier to calculate. In SIMPSON, we have tested the LAPACK drivers *dsyev* and *dsyevr* which differ in their decomposition algorithms [64]. According to LAPACK benchmarking the latter is the fastest algorithm. Finally, the exponentialized diagonal matrix is back-transformed to its original basis through its eigenvectors.

(ii) In some numerical software (e.g., Matlab [70]) the standard algorithm for evaluating the matrix exponential is the scaling-and-squaring method combined with the Padé rational approximation of the exponential function. The Padé approximation, which exploits the ratio of two matrix polynomials, converges very fast when the norm of the matrix has a suitable small value. To meet this criterion the matrix has to be scaled down by a proper factor before the approximation making use of the relation $e^A = (e^{A/\beta})^\beta$. We have adopted the recent version of the algorithm [71] which greatly reduces the need of the scaling, achieving fast convergence for matrix norm of up to about 5.37. In typical SIMPSON simulations the Hamiltonian norm is scaled by a time-step Δt (not necessarily meaning the norm gets smaller than 1, cf. Example 1) and it turns out that additional scaling is practically never necessary. As a consequence, the most expensive step of the whole procedure is calculation of the ratio of two complex matrices in the Padé approximant, implemented as solving a set of linear equations with multiple right-hand sides (*zgesv* LAPACK driver).

(iii) Another way of approximating the exponential function is to use the Chebyshev series expansion, $e^{iAx} = J_o(x) + 2\sum_{k=1}^{\infty} i^k J_k(x) T_k(A)$. In this equation, $J_k(x)$ are Bessel functions of a scalar variable x and $T_k(A)$ are Chebyshev polynomials of a matrix A (see more details in Ref. [72]). As an empirical convergence criterion in the expansion we discard all terms that do not contribute more than 10^{-6}. We note that even though this threshold is larger than machine precision for single-precision floating-point variables, all calculations are done in double precision; working in single precision often leads to higher accumulated errors than the threshold. The range of validity and fast convergence imposes restrictions both on x and A ($-1 < x < 1$, the norm of A be smaller than 1). In some implementations [73,74] it is suggested to scale and shift the eigen-spectrum of the Hamiltonian such that all eigenvalues lie within the $(-1,1)$ interval. The range of eigenvalues can be estimated using the Gershgorin circle theorem [75], and the resulting scaling factor can be absorbed into the x variable provided it still remains smaller than 1. Our experience shows, however, that these conditions are often not satisfied and additional pre-scaling is necessary, in which case one can use the scaling-and-squaring method described above. Another observation shows [76] that the expansion converges even outside the previously defined range – when the order k of the Bessel functions, $J_k(x)$, becomes larger than the argument x. In other words, pre-scaling can be avoided when a sufficiently large number of terms in the expansion is used (higher maximal k) at the cost of longer calculation time. The main advantage of the Chebyshev method lies in its possibility to exploit Hamiltonian sparsity: calculations of the $T_k(A)$ polynomials involve only a single sparse-times-dense matrix multiplication per iteration, which is still beneficial even for relatively high values of k.

(iv) Our last tested method for evaluating the matrix exponential is based on the Taylor series expansion with the same convergence criterion as for Chebyshev. This method also converges very fast provided the norm of the matrix is smaller than 1. Again, scaling-and-squaring can be used but it now represents the main time-consuming step as it involves multiplication of dense complex matrices (propagators).

The tests of our implementations of these approaches to propagator calculations show that the Chebyshev method is the fastest when combined with sparse matrix multiplications (available when SIMPSON is linked with the Intel MKL library). All the methods are now implemented in SIMPSON, and the user can decide which to use by defining it in the `par(method)` parameter. Results comparing the performance on a selected example are given in the sections with examples below. We have chosen diagonalization as the default method, since this provides propagators with high accuracy as needed for the optimal control routines also included in SIMPSON [16]. Specifically, we have chosen the *dsyev* driver as our default method due to its robustness, although dsyevr is slightly faster for large spin systems. Furthermore, at the time of writing, there is a quite persistent bug in the dsyevr driver in some LAPACK distributions (mainly linux) when multi-threading used, preventing the use of this driver on these systems.

Efficient *reuse of propagators* for time-periodic Hamiltonians has always been a central element in SIMPSON to conduct fast simulations of solid-state NMR spectra. In the original form users were encouraged to prepare a simulation such that it exploits the Hamiltonian periodicity, e.g., through the use of the γ-COMPUTE algorithm [21] following ideal pulse excitation. To further enhance this capability, we have here generalized the γ-COMPUTE algorithm implementation to be operative also following arbitrary preparation steps (e.g., non-ideal pulse or evolution in the indirect dimension of a 2D experiment, in analogy with Ref. [73]). This significantly increases computation speed in many cases of practical relevance. We note that the efficient reuse of propagators is not limited to 1D simulations. Favorable 2D pulse sequences may be prepared rotor synchronized with t_1 increments as integer multiples of a rotor period, again allowing for the reuse of the propagator for the t_1 step.

To improve the impact of efficient reuse of propagators in situations with time-periodic Hamiltonians, a new command

```
acq_block {...}
```

has been implemented. This not only speeds up calculations but also greatly simplifies coding of acquisition part of the SIMPSON input file with arbitrary rf pulsing schemes (*vide infra*). The program itself checks for a periodicity with respect to a pulse sequence, sample spinning, and, if necessary, sampling of free induction decay (FID). The COMPUTE algorithm [19,77] is used to generate the entire

Table 1
Summary of new features, parameters and commands in current version of SIMPSON.

Definitions in the `spinsys` section	
`dipole_ave` N_1 N_2 b_{IS} η α β γ	Allows to define asymmetry parameter η for motionally averaged dipole–dipole interactions
Parameters for the `par` section	
`method` value1 value2 value3 ...	Defines calculation methods for `fsimpson`. Possible values are: `direct`, `idirect` (equivalent) triggers direct evaluation of `pulseq`, possibly with COMPUTE method if `acq_block` is used; `gcompute`, `igcompute` (equivalent) uses γ-COMPUTE method, either in original (version 1) `pulseq` syntax or in new syntax with `acq_block` (which allows for manipulation before acquisition); `time`, `frequency` forces calculations to be done in time or frequency domain respectively (frequency can be used only in combination with `acq_block`, both `direct` or `gcompute`); `block_diag` tries to use block-diagonal Hamiltonians whenever it is possible (block-diagonalization according to total quantum number m_z); `sparse` uses sparse matrix algorithms (requires simpson linked with Intel MKL libraries); `diag`, `dsyevr` propagators get calculated via diagonalization of Hamiltonian using *dsyev* or *dsyevr* LAPACK drivers, respectively; `pade` propagators get calculated using scaling and squaring with Pade approximant; `cheby1`, `cheby2` propagators get calculated via expansion into series of Chebyshev polynomials (`cheby1` uses scaling&squaring to reduce matrix norm, `cheby2` uses shift&scale); `taylor` propagators get calculated via Taylor expansion with scaling&squaring to reduce matrix norm; `ASGinterpolation` triggers ASG powder interpolation method used after spectrum generation; `FWTinterpolation`, `FWT2interpolation` triggers FWT interpolation method to be used just for frequencies, or for both amplitudes and frequencies, respectively (works with γ-COMPUTE only and LEBh/ROSELEBh angle sets); `FWTASGinterpolation`, `FWT2ASGinterpolation` combinations of the interpolation methods
`num_cores` N	Defines number of threads to be used for parallel calculations of parameter averages (mainly crystallite orientations); if not defined then all processor cores are used; number of threads can also be limited by environment variable SIMPSON_NUM_CORES
`crystal_file` *filename* N_1 N_2	Optional arguments N_1 and N_2 define index of the first and the last orientation to be read from the *filename*
`crystal_file` *filename1 filename2*	Syntax used for FWT family of interpolation methods, defines source set of orientations (*filename1*) and larger target set of orientations (*filename2*); at the moment, only Lebedev sets LEBh and ROSELEBh are allowed
`rfprof_file` *filename* N_1 N_2	Optional arguments N_1 and N_2 define index of the first and the last parameter to be read from the *filename* containing rf inhomogeneity profiles
`averaging_file` *filename* N_1 N_2	Optional arguments N_1 and N_2 define index of the first and the last parameter to be read from the *filename* containing set of `spinsys` parameter values over which the calculation should average
`points_per_cycle` N	Used for direct method in frequency domain (COMPUTE algorithm), defines number of points to which the one cycle of acquisition Hamiltonian should be split
`ED_symmetry` *val*	Controls how to deal with possible excitation–detection symmetry: if *val* is −1 then it is not considered, if *val* is 0 then it is internally checked and used, if *val* is 1 then it is not checked and it is directly considered (user responsibility); default is −1
`diag_max_dim`	Determines maximal dimension of a matrix that is allowed to be diagonalized; used for `sparse` algebra and `direct` method where it is possible to avoid propagator diagonalization in FID calculations
`oc_grad_level` N	Defines accuracy level of optimal control gradient, default is 1 but L-BFGS methods requires at least 2 or higher
`oc_method` algorithm	Defines optimal control algorithm, can be `CG` for conjugated gradients (default), or `L-BFGS`
`Commands for the pulseq section`	
`acq_block {...}`	Allows easy definition of pulse manipulations during acquisition; the pulse sequence defined in parenthesis should contain one period of the total Hamiltonian; for γ-COMPUTE the total length has to be the rotor period divided by the number of γ-angles; for COMPUTE the pulse sequence is repeated such that it fills up multiple of rotor periods

FID/spectrum for cases with general time-periodic Hamiltonians, while for static sample calculations and time-constant Hamiltonian, the propagator is reused in its diagonal form. As compared to previous implementations, significant improvements of the computation speed (especially for longer FID's) are obtained.

In addition to the generalized γ-COMPUTE algorithm, we have updated SIMPSON to allow simulation of spectra directly in the frequency domain, and to further reduce the simulation time, we have added an option to efficiently exploit potential symmetries between excitation and detection [77] operators. The syntax for invoking these options is summarized in Table 1.

A major asset in spin dynamics simulations is to facilitate easy and fast *setup and update of Hamiltonians*, including modifications as, e.g., sample rotation changes their anisotropic components during time. For large spin systems, the number of distinct interactions can be quite large, implying that a considerable amount of time is spent on establishing the current Hamiltonian from all interactions. The load of these operations can be released if all anisotropic interactions are combined into five matrices forming the second-rank irreducible tensor operator representation of the total anisotropic interaction Hamiltonian [22] to be propagated in a single step in time during updates. Further simplifications can be introduced by exploiting the fact that interaction Hamiltonian may be described as a real, symmetric matrix and only real (instead of more costly complex) arithmetic operations are used.

Nuclear spin Hamiltonians are inherently sparse in Hilbert space, offering yet another source to enhanced computation speed and lower memory demands. The *sparsity of matrices* is exploited either through block diagonalization or use of dedicated sparse algebra algorithms. During periods without rf irradiation, the total Hamiltonian commutes with the z component of the total angular momentum operator, $F_z = \sum_{i=1}^{n} I_{iz}$, implying that its matrix representation can be *block-diagonalized* according to the total spin quantum number. For heteronuclear spin systems, the Hamiltonian commutes with F_z of each nucleus type separately, enabling even finer block diagonalization. Such effects may straightforwardly be used to reduce the effective matrix sizes leading to faster calculation of propagators. When rf irradiation is used on all channels (i.e., all different spin species), such decomposition is not possible, but obviously important situations exist when rf is only applied to some of the channels. An evident example is NMR signal acquisition under heteronuclear decoupling, where significant reductions in computation time are achieved through block diagonalization. Considering the frequent presence of sparse matrices for, e.g., the initial density matrix, the detection operator, and Hamiltonian of free evolution, it is tempting to assume that specialized sparse matrix operations would provide significant additional speed up. However, the propagators are relatively dense (especially for rf irradiation periods) and the sparsity of matrices is quickly lost during the calculation. Further considerations come from algorithms for efficient NMR acquisition which are based on diagonalization of the total propagator of the period like (γ)-COMPUTE [19,21,77]. The transformation of a potentially sparse propagator to its diagonal form usually leads to very dense matrices of the eigenvectors and, to our knowledge, no efficient algorithm exists to handle such operations. Nonetheless, the feasibility to use sparse matrix routines is present in the current release of SIMPSON when linked with the Intel MKL libraries: It enables mixed use of sparse and dense matrix operations, and it is up to the user's decision to set the maximal sparse matrix dimension that still can be diagonalized (parameter `diag_max_dim`). The use of sparse matrices enables calculations on large spin systems (up to about 15 spin-1/2) that still fit into somewhat enlarged computer memory and would be impossible to do with dense protocols.

Optimal control (OC) tools [35,36] were released in SIMPSON version 2 [16], and have been used extensively to develop novel NMR experiments with improved efficiency [41–44,78–80]. The goal of the SIMPSON implementation of OC was to provide a simple setup for optimization while keeping full flexibility to formulate specific criteria for the target (transfer) function. An exotic example of such OC calculations is the recent work establishing spin-state-selective coherence transfer in rotating solids by means of direct dipole–dipole interaction between carbons (as opposed to *J*-coupling based methods) [44]. The target function for this optimization involved maximization of the desired transfer pathway as well as suppression of several undesired transfers. If one includes also other experimental constraints regarding robustness with respect to, for example, rf inhomogeneity, one is easily faced with rather lengthy calculations. Although SIMPSON's use of multiple computer cores comes as a great help (described below), one should also search for optimization methods that converge faster, using fewer iterations. de Fouquieres et al. [61] recently demonstrated that quasi-Newton optimization strategies reach the final goal faster than the original GRAPE algorithm [35], and even with higher fidelity. By establishing higher order terms in the target function gradient one can get fidelities orders of magnitude better (as it is typically required in quantum information processing applications). We have therefore implemented the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm [62] (using the freely available L-BFGS software package [81]) in the new optimal control tool in SIMPSON, together with the improved precision gradient calculation as prescribed by Eq. (14) in Ref. [61].

3. Fast calculation of orientational averages by using interpolation methods

For simulations involving powdered samples one typically needs to perform the spin dynamics calculation for a large number of crystallite orientations and summing their results to mimic the response of the physical sample and obtain the orientational (powder) averaged NMR response. The overall simulation time then grows linearly with the number of orientations. Thus, even if the computation time is fast for one single crystallite, and the orientational space may be restricted due to symmetries in the NMR responses [54,77], the whole simulation may become very time consuming. Several approaches have been introduced for minimizing the number of sampled orientations, each parameterized by a set of Euler angles, $\Omega = (\alpha, \beta, \gamma)$.

The prevailing strategy is to optimize the sampling of the orientational space using a specific grid of Euler angles. Examples of commonly used schemes are the Zaremba–Conroy–Wolfsberg (ZCW) [50], REPULSION [53], and Lebedev [54,82,83] sets of crystallite orientations. They may be generated for two, ($\Omega = (\alpha, \beta)$), or three Euler angles, depending on the application, which all can be used in SIMPSON.

One may additionally use spectral interpolation to improve the convergence of the orientational averaging. Here the spectrum from the powder is generated by *predicting* the responses based on the calculation of a limited set of orientations. The well-established interpolation method of Alderman, Solum, and Grant (henceforth referred to as ASG) [59] was recently extended to interpolating NMR problems involving non-diagonal Hamiltonians, and complemented with interpolation by fast Wigner transform (FWT) [60]. The FWT strategy interpolates the orientation dependence of the principal characteristics of a spin system response (such as the set of frequencies and amplitudes in the NMR spectrum), calculated for a limited set of orientations and subsequently predicted for a much larger grid. In the present SIMPSON implementation, we restrict ourselves to orientations encoded by two Euler angles ($\Omega = (\alpha, \beta)$), while the averaging over the third (γ) angle is efficiently performed with COMPUTE-based algorithms [21,77]. The details of the FWT protocol are described in Ref. [60]; here, only the most important features are recapitulated.

An orientation dependent function $f(\Omega)$ can be expressed as a series of reduced Wigner functions (i.e., spherical harmonics, $Y_{Jq}(\Omega)$) of increasing spherical rank J [54],

$$f(\Omega) = \sum_{J=0}^{J^f_{max}} \sum_{q=-J}^{J} f_{Jq} Y_{Jq}(\Omega) \tag{1}$$

As opposed to previous interpolation methods (such as ASG) that directly target the frequencies and amplitudes of the NMR spectrum, the FWT-interpolated functions $f(\Omega)$ represent the eigenvalues of the propagator over the Hamiltonian period and associated time-periodic functions, from which the frequencies and amplitudes are constructed for all NMR peaks in the spectrum. The Appendix of Ref. [60] outlines an efficient option for implementing FWT within the γ-COMPUTE protocol, the key-point being that all time-consuming spin dynamics calculations are restricted to each orientation of a *small source* set S^S_Ω of N_S orientations, from which the coefficients f_{Jq} are extracted and subsequently used to predict the values of $f(\Omega)$ from a much larger *target* set S^T_Ω of N_T orientations. This approximates the NMR spectrum associated with the large set, thereby *ideally* providing a reduction in the computation time by the factor N_T/N_S. The FWT implementation in SIMPSON is essentially identical to that described in Ref. [60]; only in relatively exotic cases of explicit amplitude interpolations with a *large* number of sidebands, a minor loss of computation efficiency may occur.

The maximum spherical rank J^f_{max} in the expansion in Eq. (1) dictates the "difficulty" of the orientational averaging problem, where "challenging" cases are normally associated with a high J^f_{max} value [54]. A set of orientations S^s_Ω that integrates *all* $Y_{Jq}(\Omega)$ functions with spherical ranks $J \leqslant J^f_{max}$ produces an *exact* orientational average. Only Gaussian spherical quadrature (GSQ) grids achieve this task, by using a set S^s_Ω associated with $J^s_{max} \geqslant J^f_{max}$. While in practice $J^s_{max} < J^f_{max}$, the higher the rank (J^s_{max}, which is dictated by the size of the source set S^s_Ω) of the grid of orientations employed, the better the approximation of the interpolated function [60].

Thus the FWT interpolation is most suited for GSQ grids, with those of Lebedev ("LEB") [82,83] being the hitherto most efficient schemes thereof. Also, these sets of orientation may be extended to integrate essentially arbitrarily high spherical ranks ("^{ROSE}LEB") [84], thereby offering the necessary precision for broad powder lineshapes. While FWT interpolation usually requires a smaller set of orientations to reach convergence relative to its ASG counterpart, the computational speed may be enhanced further by merging the two methods: first FWT interpolation is used to generate a set of amplitudes and frequencies that are subsequently processed with the ASG protocol, resulting in the FWTASG hybrid interpolation protocol [60].

Note that whereas there are no fundamental limitations in the choice of source set S^S_Ω, current SIMPSON implementations require the use of GSQ (e.g., LEB or ^{ROSE}LEB) grids together with the γ-COMPUTE method; this combination allows for significantly faster calculations compared with using arbitrary (non-GSQ) sets [60]. The transforms are efficiently evaluated with the NFFT library [85,86]. It is generally sufficient to interpolate the eigenvalues of the propagator accumulated over the total Hamiltonian period, from which the spectral frequencies are derived for each orientation S^T_Ω, while the amplitudes are estimated approximately by their values at nearest neighbor orientation from the source set [60]. This provides a substantial time-saving, as the number of NMR amplitudes for all sidebands is much larger than the number of their frequency counterparts [60]. In SIMPSON it is possible to activate the desired interpolation protocol when specifying the calculation method within the `par` section of the SIMPSON input file (see Table 1 and examples below).

Provided that the orientation-dependence of the propagator eigenvalues/vectors may be tracked properly, both the ASG and FWT procedures are applicable for simulations involving non-diagonal Hamiltonians [60]. Unfortunately, there is no general and fool-proof solution to this problem. One option is discussed in Ref. [60]. The present SIMPSON implementation differ from this by using a straightforward sorting of the eigenvalues and -vectors resulting from the eigen-decomposition routines (*zgeev* LAPACK driver). Since the eigenvalues are complex numbers of a unit absolute value (eigenvalues of a unitary matrix) we sort them according to their phase. This approach is aided by exploiting the Hamiltonian block-diagonal structure whenever possible, with each block handled separately, thereby reducing the risk of failure in the eigenvalue tracking. This plain eigenvalue sorting does not provide a fool-proof solution and may lead to faulty interpolation results, especially for small source sets. While there is room for improvements, the current implementation reproduced all examples provided in Ref. [60]; another application to fitting of NMR spectra is presented below. Noteworthy, the frequency-folding problem is automatically handled within all COMPUTE-based algorithms [19,21,77].

4. Enhanced computation through multiple-core parallelization

Following current trends in manufacturing of computers, our aim of optimizing SIMPSON also involves exploiting high-performance computing facilities ranging beyond SIMPSON unit core operations described above. Independently on the underlying software, it is commonly accepted that former strategies relying on faster computers to unleash new and even more intensive calculations will come to an end. Essentially all attention is now turned towards multiple cores, and increased performance now is obtained via parallelization. With this focus, we have introduced parallel features in SIMPSON which exploit efficient handling of data and task level parallelization for fast operation on contemporary multi-node and multi-core computing platforms, ranging from powerful laptops and desktops through clusters and grids to clouds. In the future, calculations on the highly parallel graphics processing units (GPUs) will be very relevant to consider, but at the moment these platforms are very heterogeneous and hardware dependent, why we will postpone address to these to a later stage.

4.1. Multi-core architectures

Nowadays, laptop and desktop computers equipped with *multi-core* (e.g., dual- or quad-core) chipsets prevail the computer market and are obvious first targets of parallel SIMPSON implementations based on multithreading strategy.

A *cluster* computer is composed of a number of computers. Typically, the computing elements are identical and communicate via high-speed ethernet connections, thus providing a tightly coupled homogeneous computing platform. Using software packages like PVM [87] or MPI [88], the cluster can give the illusion that the computer elements function together as a single large virtual machine. SIMPSON support to clusters is based on MPI protocols.

A *grid* is a distributed large-scale cluster, which incorporates a collection of heterogeneous computational resources. The type of a resource can vary among any type of computer devices with a network interface. A grid can function on a global scale with geographically dispersed resources. Thus, in contrast to a cluster computer, a grid presents a loosely coupled, heterogeneous computing platform. The primary disadvantage of a grid is the potential instability and reduced speed of the network connections, requiring a minimum of inter-node communication. However, the ability to collect any type of resource from multiple domains and put the aggregated computational power on tap for a single application makes it extremely powerful. Hence, a grid works particularly well for multiple-tasked jobs, i.e., jobs that can be divided into many independent parallel computations (*vide infra*) with no or little need for inter-communication. For such systems, parallel SIMPSON jobs can be executed with MPI parallelization, or one can distribute jobs manually among arbitrarily heterogeneous resources.

Cloud computing is a new paradigm for the provision of computing infrastructure and IT services based on internet protocols. A cloud refers to an amorphous, distributed collection of computer resources used in a way that one does not really care where particular applications reside. Although there is a long debate what exactly constitutes a "cloud", it brings several new aspects: (i) the appearance of infinite computing resources quickly available on demand, (ii) the elimination of an up-front commitment by cloud users not buying unnecessary computer clusters, (iii) the ability to pay for use of computing resources on a short-term basis and release them when no longer needed. There is new market for computing resources, providing on-demand services (infrastructure, platform and software), with Amazon's "Elastic Compute Cloud" (EC2) as an example. It is possible to set up a cluster within a cloud specifically tailored for a particular SIMPSON calculation with selected number of nodes and their processor types, with required memory and storage. This may be facilitated using community-developed scripts [89].

4.2. Parallelization of solid-state NMR simulations: crystallites, field variations, and parameter distributions

Addressing solid-state NMR simulations in particular, numerous possibilities exist for parallelization of similar calculations, including powder averaging (same calculation performed for different crystallite orientations), rf and static field inhomogeneity (same calculation differing only in the strengths of the involved rf or static fields), and parameter scans/distributions (same calculation performed with systematic – known variation of parameters passed through SIMPSON's Tcl interface). The latter may, for example, involve a grid scan to examine the effect of internal or external parameters, establish gradients in relation to experiment optimization, or simulate spectra for inhomogeneous samples with distributions of spin-system parameters.

In general, SIMPSON distributes independent calculations using the threads mechanism whenever possible, typically on laptops and workstations. On clusters and grids, the MPI protocol is used. Per default SIMPSON automatically detects number of available cores and uses them all for maximum efficiency. However, the user has the possibility to specify the exact number of cores either by setting the environment variable SIMPSON_NUM_CORES or by setting `par(num_cores)` in the input file. When running SIMPSON with MPI, the number of SIMPSON instances is determined by the standard MPI methods (which may, however, differ slightly for different flavors of MPI implementations and job scheduling systems). In the job setup, the user should reserve all cores on requested processors as the SIMPSON automatically use them all, or limit the number of cores as described above. Other combinations may also be useful, for example request all cores of a cluster node to get all of its memory but execute only a single process (thread). The user has also an option to manage the details of the distribution, which may be advantageous when using heterogeneous or not interconnected resources.

The averaging parameters that are distributed on different cores include crystallite orientations (defined by `par(crystal_file)`), rf field inhomogeneity profile (defined by `par(rfprof_file)`), and spin system parameters like a range of chemical shifts (defined by `par(averaging_file)`). The structure and usage of the first two files has been described previously [7,16], but here implemented such that they are automatically distributed to the available (or specified) cores. The latter file involving spin system parameters is a novel feature which may highly facilitate development and analysis of pulse sequences aimed at robust performance under varying spin system parameters, e.g., when designing pulse sequences which are broadbanded with respect to chemical shift offset, dipole–dipole couplings, or chemical shift anisotropies. An example of such file is given below. The first line in the file defines the parameters using the same notation as for the `fsimpson` command, i.e., name of spin system interaction followed by spin number(s) and value name (`iso`, `aniso`, `eta`, `alpha`, `beta`, and `gamma`). In this way, any value given in the `spinsys` section can be modified and distributed. The file shown below involves two parameters, but any number of parameters can be set as columns in the list. The last column should always contain the weight, giving a relative relevance of each instance (all numbers in the last column should sum up to 1). The chemical shifts can be specified absolutely in Hertz or relatively in ppm (append 'p' after a number). The calculation is performed when all parameter values from one line are set and the result reflects the averaging over the given parameter variation in the whole file.

```
shift_1_iso        dipole_1_ 2_aniso        weight
-10p               -10000                   0.2
+10p               -10000                   0.2
0                  -10000                   0.2
0                  -8000                    0.2
0                  -12000                   0.2
```

As another new feature, it is also possible to limit reading from the file only to specific range of lines. This feature is implemented for crystallite orientations as well and gives the user freedom to distribute calculations manually. When all calculations are collected, the resulting fids can be added using the commands `fload` (to load a fid file) and `fadd` (to add fids) in a separate final run of SIMPSON.

```
par {
  ...
  crystal_file       rep100        1      25
  averaging_file     shifts.dat    3      5
  ...
}
```

In this example, powder averaging will be done only for crystallites 1–25 defined in the file `rep100.cry` and to parameters on lines 3–5 (the header line is not counted) from the file `shifts.dat` (if no range is specified then the whole file is read in).

At present, SIMPSON makes parallelization at a high level with single-threaded calculations of individual FID's, but an alternative strategy would be to make use of multiple cores at a lower level to employ parallelized matrix operations implemented in several linear algebra packages like Atlas, OpenBLAS, or the commercial Intel MKL package, which all can be easily linked to the SIMPSON program. However, in our brief survey, we have not observed significant performance gains for typical calculations with up to about 10–12 spins; it seems wiser to distribute individual FID calculations but all may depend on a specific computer configuration. In this regard, one should also be careful not to overload the processor with more threads than it has cores, which would lead to loss of performance. To balance the number of threads spawned by SIMPSON and the linear algebra library one can, for example, set the environment variable OMP_NUM_THREADS, as it is described in the documentation for Intel MKL and OpenBLAS. Although it is possible to parallelize individual matrix operations using dedicated parallel matrix libraries (e.g., PBLAS or ScaLapack), we have not exploited these since most time-consuming calculations may be parallelized through crystallite distributions as explained above.

5. Examples

In this section, we demonstrate that the improved version of SIMPSON (i) provides reasonable speed of calculations on a stand-alone computer, comparable with other high-performance simulation software, (ii) uses simple and transparent coding of pulse sequences at all stages of the NMR experiment – both preparation and acquisition, holding up with our premise that programming complexity is simpler or similar to that on a real NMR spectrometer, (iii) is capable of calculating NMR spectra for relatively large spin systems (10–15 spins), and (iv) is parallelized in all of its applications (data fitting and optimal control experiment design) to exploit the advantages of modern multi-core computers.

5.1. Propagator calculation methods

In our first example we compare the various methods implemented in SIMPSON for calculation of propagators. As a test case we selected simulation of a carbon spectrum influenced by one-to-nine additional protons subjected to continuous wave (CW) rf irradiation of amplitude 70 kHz and 15 kHz magic-angle spinning (MAS). The results summarized in Fig. 1 were obtained by averaging of 400 individual propagator calculations performed during the course of the spectrum simulations using the γ-COMPUTE algorithm with a set of 10 REPULSION (α, β), and 8 γ angles, the time step for time discretization was set to 2 μs. Our implementation includes two methods based on Hamiltonian diagonalization using either *dsyev* or *dsyevr* LAPACK drivers (henceforth referred to as dsyev and dsyevr, respectively), Padé approximation (referred to as pade), Taylor series expansion (referred to as taylor), and series of Chebyshev polynomials taking different precautions in order to guarantee convergence: Scaling-and-squaring when the norm of $H\Delta t$ exceeds limit of 1 (referred to as cheby1), and shift-and-scale to limit eigenvalues of the $H\Delta t$ matrix to lay within the interval $(-1, 1)$ (referred to as cheby2). The taylor, cheby1, and cheby2 methods involve evaluation of several terms in the expansions that are accomplished by sparse-times-dense real matrix multiplications. In addition, taylor and cheby1 finish with scaling step involving multiplications of dense complex matrices. It is evident that relative performances are not dramatically different and it is difficult to make generally applicable conclusions. The Padé method is never the fastest while taylor always performs good. It seems beneficial to use dsyev for small spin system while for more spins ($\geqslant 6$) dsyevr performs better. The success of the expansion methods largely depends on the properties of the Hamiltonian, namely its sparsity and the norm as illustrated in Fig. 1b and c. The relative number of non-zero matrix elements markedly decreases with increasing system dimension making sparse matrix algorithms to work increasingly better. On the other hand, the norm of the $H\Delta t$ matrix increases with more interactions involved between more spins, which triggers the scaling-and-squaring mechanism in taylor and cheby1, and increases the number of terms in series of cheby2 (see Fig. 1d, the number of terms remains constant for taylor and cheby1 thanks to the scaling-and-squaring). One may be tempted to fix this problem by decreasing Δt time step but this would in turn lead to evaluation and multiplication of more propagators. Thus, scaling-and-squaring provides a more efficient strategy that saves some operations, provided the approximation that the Hamiltonian is piecewise constant holds within acceptable error bounds when using larger Δt. The method cheby2 avoids the expensive squaring step but needs more iterations for convergence. As this only involves computationally cheap sparse-times-dense matrix operations it still pays off and cheby2 is the fastest method for propagator generation of large spin systems.

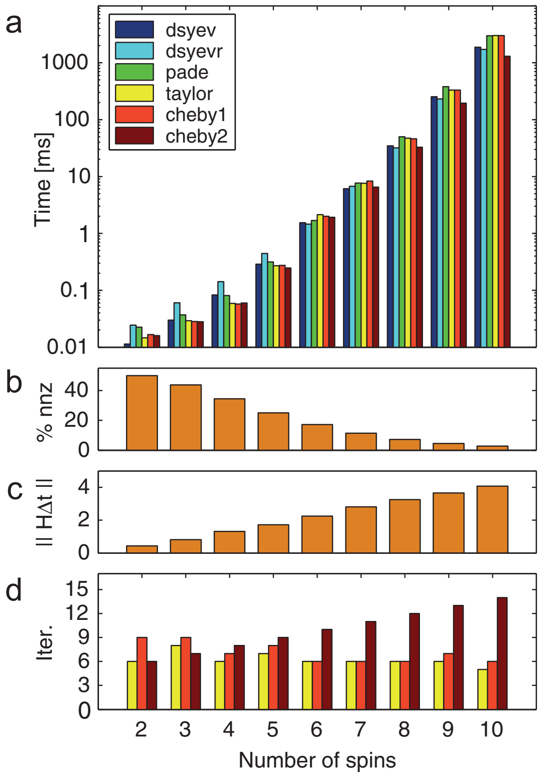

Fig. 1. Performance comparison of different strategies implemented in SIMPSON for propagator calculation (dsyev and dsyevr: diagonalization based approaches using corresponding LAPACK drivers; pade – Padé approximation; taylor: Taylor series expansion; cheby1 and cheby2: expansion in Chebyshev polynomials using scaling-and-squaring and shift-and-scale pre-processing options, respectively) evaluated for different spin systems consisting of one carbon and 1–9 protons (including all dipole–dipole interactions and 70 kHz CW proton decoupling under 15 kHz MAS spinning conditions). (a) Average timing of the methods with dsyev being the fastest for the smallest spin system and cheby2 for large systems. (b–d) Statistics of Hamiltonians and series methods. (b) Percentage of non-zero elements (nnz) within the Hamiltonian decreases with increasing system size. (c) Average norm of the matrix $H\Delta t$ ($\Delta t = 2\ \mu$s) increases with more interactions involved in the Hamiltonian for bigger systems. (d) Number of terms needed for convergence of expansion series methods taylor, cheby1, and cheby2 using the same color coding as in (a). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.2. CW decoupling: single-core performance

In tests of the overall performance of the current SIMPSON version, we performed again a series of CW decoupling experiment calculations with increasing number of protons within the spin system as described above but using 168 (α, β) angles of the REPULSION scheme [53] and 10 γ angles. The spectra consisting 1024 points with a 5 kHz spectral width were generated directly in frequency domain using the γ-COMPUTE algorithm and excitation–detection symmetry assuming ideal carbon excitation. The current version 4 of SIMPSON was linked with MKL version 12.1.2.273, and the calculations were performed twice with and

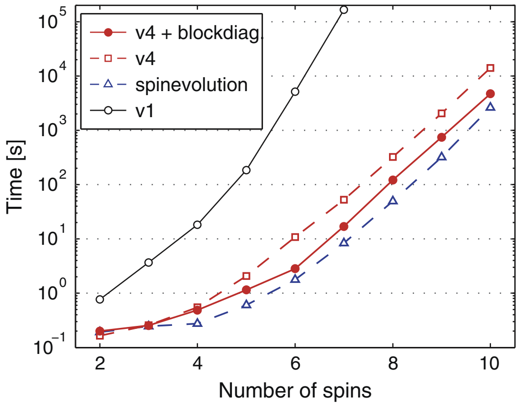

Fig. 2. Computation time of CW decoupling experiment as a function of spin system size. The simulations include a single ^{13}C spin connected to 1–9 ^{1}H spins performed using SIMPSON version 1 (v1, black, open circles), SPINEVOLUTION [14] version 3.4.5 (blue, open triangles), the current version of SIMPSON without making use of block diagonalization (v4, red, open squares), and with taking advantage of block diagonalization (v4+blockdiag., red, filled circles), In all cases the computation was performed on a single core of Intel Xeon processor computer running CentOS 6.3 linux system. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

without Hamiltonian block diagonalization with respect to the ^{13}C spin (it splits the Hamiltonian into two blocks of identical dimension), and are referred to as v4+blockdiag and v4 in Fig. 2 (both using cheby2 propagator method). The original SIMPSON version 1 does not allow for frequency domain calculations nor arbitrary spectral widths and an "as-close-as-possible" setup was used for illustration purposes (referred to as v1). Equivalent calculations were also done with SPINEVOLUTION package of version 3.4.5. All the simulations were performed on a single core (i.e., with no parallelization on any level) of Intel Xeon processor computer running CentOS 6.3 linux operating system.

The wall-clock times for the benchmark calculations are presented in Fig. 2. It documents dramatic speed improvement (several orders of magnitude) of the current version with respect to the original releases of SIMPSON, and a performance similar to SPINEVOLUTION. Importantly, when adding more spins to the simulation, both programs scale essentially identically, with SIMPSON being somewhat slower for larger spin systems. For small spin systems (up to ~5 spins) the differences in performance are marginal and calculations are finished within a second. For larger spin systems, the time spent on matrix manipulations strongly exceeds any other operations, and the simulation time is largely not be affected by startup and other operations requiring small amounts of time. It is claimed that the efficiency of SPINEVOLUTION is derived from approximating the propagators by the Chebyshev series, which scale better with matrix size than diagonalization methods, thanks to exploring the Hamiltonian sparsity. One can inspect the time complexity of the overall calculations on the matrix dimension n in the form $t = n^k$. Doing this, we find an exponent $k = 2.7$ (both for SIMPSON and SPINEVOLUTION), which is indeed smaller than 3 predicted for diagonalization of dense matrices. We note that all of these comparisons are guiding but may be subject to variations across computer architectures and depending on the type of simulation.

5.3. Heteronuclear decoupling methods: novel acquisition commands

In solid-state NMR spectroscopy of organic and biological compounds heteronuclear decoupling is a major issue, which over many years have attracted considerable attention as a means to improve spectra resolution. This has led to the development of many pulse sequences including TPPM [90], SPINAL [91], XiX [92], PISSARRO [93], symmetry-based CN [94] and RN [95] sequences as well as advanced theoretical methods for understanding of their performance and systematic development of such experiments [96–98].

As the effect of heteronuclear decoupling typically depends highly on the spin system, involving effects from both heteronuclear and homonuclear interactions as well as the chemical shift parameters, evaluation of decoupling performance eventually has to rely on experimental comparison, and with the development of efficient numerical simulation programs also on numerical simulations. We here take the latter approach and demonstrate how efficient calculation of decoupling in large spin systems by SIMPSON may add to the discussion of relative performance of different decoupling methods. From the extensive list of heteronuclear decoupling sequences, we have for the purpose of demonstration selected continuous wave (CW) irradiation, two-pulse phase modulation (TPPM) [90], frequency swept TPPM (SWf-TPPM) [99], SPINAL-64 [91], XiX [92], and refocused continuous wave (rCW) [100] decoupling methods. The corresponding pulse sequences, summarized in Fig. 3a, differ in complexity, and some of them are difficult to implement in current NMR simulation software, particularly when addressing synchronization with MAS and detection dwell time periods.

With the new acq_block command, SIMPSON now offers a particularly simple way of implementing rf irradiation during acquisition, which is basically identical to typical implementations on real NMR spectrometers. Pulse sequence events are specified as they come in time, without a need to chop them into blocks of dwell time, as it was necessary in previous versions of SIMPSON. For example, XiX decoupling is coded as follows,

```
acq_block {
  pulse $par(tp) $par(rf) x 0 0
  pulse $par(tp) $par(rf) −x 0 0
}
```

where par(tp) defines duration of the pulse, i.e. 2.85 times a rotor period. SIMPSON itself then checks for periodicity and synchronization, and repeats the sequence as necessary. The FID/spectrum is calculated using the COMPUTE algorithm in time/frequency domain. The algorithm specific parameter defining the number of propagation steps taken within a cycle can be defined in par(points_per_cycle) and is automatically taken into account in splitting the sequence into smaller time intervals. We note that any commands dedicated for pulse sequence section of the SIMPSON input file (pulse, pulseid, delay, offset, pulse_shaped, prop) can be used within the acq_block command. As a second example demonstrating the versatility of the programming, the SPINAL-64 scheme can be nicely implemented using a shaped pulse,

```
acq_block {pulse_shaped $par(tp) $spinal64shape nothing}
```

with all rf amplitudes and phases defined in the shape variable spinal64shape (the keyword nothing is used to define there is no rf irradiation on the second channel – ^{13}C). The shape file merely includes a list of pulse amplitudes and phases as described previously in Ref. [16].

The computation time depends critically on pulse sequence event synchronization. While for CW experiment one can use efficient γ-COMPUTE algorithm, it is not directly possible for the other cases (although we realize one can adjust the spectral width *and* averaging over γ-angles for TPPM in order to meet restrictions on periodicity of rf Hamiltonian). Proper synchronization is still left

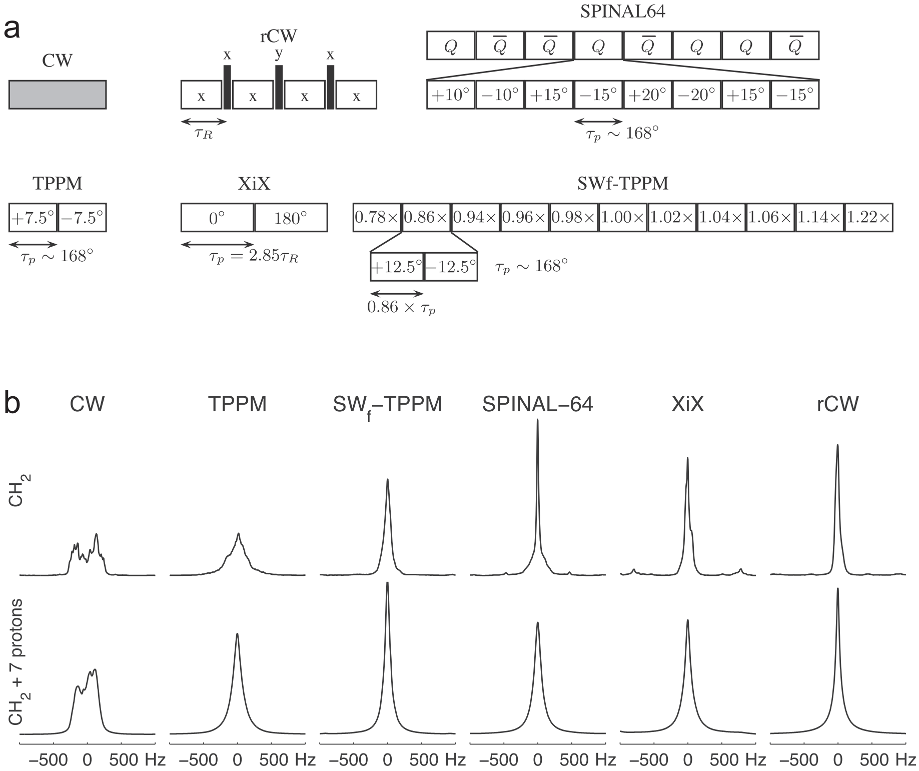

Fig. 3. Simulation of different heteronuclear decoupling methods in SIMPSON. (a) Schematic representation of pulse sequences for continuous wave (CW), refocused continuous wave (rCW), two-pulse phase modulation (TPPM), x-inverse x pulse (XiX), SPINAL-64, and frequency swept TPPM (SWf-TPPM). In all cases rf field strength of 70 kHz on the ^{1}H rf channel were used together with pulse durations corresponding to 168° flip angle and phases as indicated inside the boxes. For rCW the rotor synchronized 180° pulses used an rf amplitude of 150 kHz. For SWf-TPPM, pulse durations were modified according to a scheme suggested in the original publication, for XiX the phase is switched after 2.85 rotor period. (b) ^{1}H-decoupled ^{13}C MAS spectra calculated for CH$_2$ group of tyrosine-59 Cβ carbon taken from ubiquitin 3D structure (top), and CH$_2$ group including 7 additional protons surrounding the Cβ carbon atom (bottom). Simulation details: 15 kHz MAS, 100 REPULSION orientations of (α, β) and 12 γ angles.

to the user's discretion, but SIMPSON can help with reporting how many propagators need be calculated in one period of the total Hamiltonian. In the present evaluation of heteronuclear decoupling sequences, we used 15 kHz MAS and 70 kHz proton rf channel amplitude in all simulations, with the element pulse duration corresponding to 168° flip angle in the TPPM type of sequences, 150 kHz rf strength of the 180° refocusing pulses in rCW, and a set of 100 REPULSION [α, β] orientations, and 12 γ angles. The ^{13}C MAS spectra are generated directly in frequency domain so we are free to choose any spectral width and set it to 5000 Hz. For TPPM, the basic block has to be repeated 5 times in order to fill 1 rotor period, one rCW scheme fills 4 rotor periods, the SWf-TPPM sequence is repeated 5 times in 11 rotor periods, SPINAL-64 is repeated 5 times in 32 rotor periods, and the most poorly synchronized XiX scheme is requiring 10 block repetitions in 57 rotor periods. It is obvious that the calculation time increases correspondingly with the increasing durations of one total Hamiltonian period. However, all simulations including 10 spins finish within a reasonable time which can be readily reduced by distributing powder averages over several resources (wall clock times per single crystallite were 4 s, 45 s, 90 s, 5 min, 9 min, and 15 min for CW, TPPM, rCW, SWf-TPPM, SPINAL-64, and XiX, respectively, on an Intel Xeon processor).

The resulting spectral lineshapes are presented in Fig. 3b. In the first run, we included only directly bonded protons in a CH$_2$ group corresponding to Cβ carbon of tyrosine-59 residue in ubiquitin (PDB structure 1D3Z). As expected, the simulated ^{13}C MAS NMR spectrum obtained under the use of CW decoupling provides the broadest lineshape at the given limited rf power. TPPM is slightly better, but both methods are outperformed by the other schemes from which the SPINAL-64 and rCW are the best. The importance of inter-proton interactions in decoupling is nicely documented in the other series of simulations with 7 additional nearby protons included in the spin system (7 new protons closest to the tyrosine-59 Cβ atom in ubiquitin, 10 spins in total). Attributed to proton self-decoupling phenomena, both CW and TPPM demonstrate here improved lineshapes, while SPINAL-64 degrades significantly with more protons. The most efficient decoupling sequences in this test are SWf-TPPM and rCW. A detailed analysis of this phenomenon is beyond the scope of this publication.

5.4. Spin diffusion: large scale calculations

Spin diffusion is a very important phenomenon in NMR of solids. Several techniques based on proton spin diffusion are used to establish through-space correlations between carbons, for example, in structural studies of biological molecules. Kinetic models [66] have been established to describe the process phenomenologically, and important attempts to use quantum mechanical approaches [67] can be found in literature. In the latter studies, it proves necessary to include a large number of spins in the simulations, and special algorithms in custom-made software have typically been used. However, it may be difficult to handle the vast number of nuclear spin interaction parameters for many spins, to track their impact on spin diffusion processes, and simultaneously handle potential influence from rf pulses. For such studies simulations involving fewer (but still many) spins become practical.

Here we present an example of calculation of proton driven spin diffusion (PDSD) [68] and Dipolar Assisted Rotational Resonance (DARR) [70] transfers of magnetization between two ^{13}C spins surrounded by 13 ^{1}H spins in a full Hilbert space representation using SIMPSON. The 15-spin simulations were performed for a rotating powder (10 kHz sample spinning) with focus on the ^{13}Cα–^{13}Cβ pair of Ile-30 in ubiquitin. The spin system parameters for the two ^{13}C spins and the thirteen ^{1}H spins were established from the 1D3Z.pdb ubiquitin structure using SIMMOL [8,9]. A prerequisite of such simulations is to use the formalism of sparse matrices, since storage of the full matrices would exceed the memory capacity of most computers. Using sparse matrices, the memory requirements are greatly reduced. The sparsity of propagators is largely destroyed when proton rf irradiation is used but it is possible to block-diagonalize them according to the spin number of the total ^{13}C I_z operator, and the problem still fits into an enlarged memory computer. Our simulations consumed about 7 and 60 GB of memory for the PDSD and DARR calculations, respectively. The wall clock time per one crystallite (single thread) was 4.5 and 36 h for PDSD and DARR, respectively, on an Intel Westmere 2.93 GHz processor with 96 GB RAM. The buildup curves in Fig. 4 were calculated using the SIMPSON's direct method that enables to avoid matrix diagonalization at all stages, and were generated and summed over 100 crystallites (5 equidistant γ angles for every set of 20 α, β couples generated by REPULSION) distributed over 10 processors. The coherence transfer period without (PDSD) and with (DARR) rf irradiation was digitized with 100 points sampling every 0.1 ms. The recoupling of homonuclear dipolar interactions in the DARR setup clearly speeds up the magnetization transfer.

5.5. Powder interpolation

The simulation of broad lines, e.g. those observed for the second-order quadrupolar lineshape of the central transition for quadrupolar nuclei with half-integer nuclear spin, may be very time consuming, especially when considering magic-angle spinning and effects of finite pulses, since a large number (typically several thousands) of crystallites is required to generate converged lineshape. With the new core implementations of interpolation using the Alderman–Solum–Grant (ASG), fast Wigner transform (FWT), and the combined algorithm (FWTASG), SIMPSON offers substantial reductions of the number of crystallites for such simulations as illustrated in Fig. 5. This figure shows simulations of the central transition lineshape for ^{87}Rb in RbClO$_4$ in a multiple-quantum MAS (MQ-MAS) experiment [102,103], i.e., with 3Q excitation and 3Q → 1Q mixing to illustrate the effects of finite pulses in such experiments [9]. In addition to a substantial reduction in the number of crystallites needed to calculate the converged lineshapes, the interpolation is very simple to use and may be invoked using the `acq_block` command. The simulations above are invoked by the following SIMPSON input file

```
spinsys {
  channels 87Rb
  nuclei 87Rb
  quadrupole 1 2 3.3e6 0.21 0 0 0
}
par {
  method          gcompute freq FWTASGinterpolation
  spin_rate       9000
  crystal_file    LEBh295 ROSELEBh6145
  gamma_angles    16
  sw              spin_rate*gamma_angles
  np              2048
  start_operator  I1z
  detect_operator I1c
  variable tsw    1.0e6/sw
}
proc pulseq {} {
  global par
  matrix set 3 coherence {-3}
  pulse 3 150000 y
  filter 3
  pulse 1 150000 y
  acq_block {
    delay $par(tsw)
  }
}
proc main {} {
  global par
  set f [fsimpson]
  fsave $f $par(name).spe
}
```

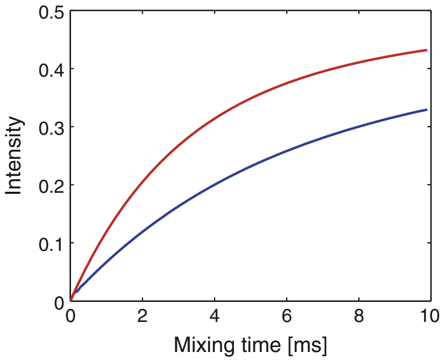

Fig. 4. Polarization transfer between two carbons mediated by spin diffusion among 13 ^{1}H spins included in the simulation (15 spins in total). The transfer is carried out without rf irradiation of protons, corresponding to the so-called proton driven spin diffusion (PDSD) experiment (blue) and with homonuclear rotary resonance (HORROR, [65]) ($\omega_{rf} = \omega_r/2$) rf irradiation, corresponding to the so-called DARR experiment (red), respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The line `method gcompute freq FWTASGinterpolation` specifies the interpolation scheme and that the simulation is performed directly in the frequency domain. The line `crystal_file LEBh295 ROSELEBh6145` specifies the two crystallite files required for FWT and FWTASG interpolation, here corresponding to $N_S = 295$ and $N_T = 6145$ hemispheric GSQ orientations, where N_S and N_T denote the respective numbers of orientations in the source and target sets that were selected according to the LEBh [54,82,83] and $^{\mathrm{ROSE}}$LEBh [84] schemes, respectively. Only one crystallite file is required for ASG interpolation. Note also that the `fsimpson` command generates a spectrum and not an FID when frequency domain calculations are triggered.

A thorough comparison of the different interpolation schemes is beyond the scope of this paper, but we observe the general trend Alderman–Solum–Grant (ASG) method requires evaluation of spin dynamics in more crystallite orientations to achieve converged lineshape than the fast Wigner transform (FWT) approach which interpolates spin dynamics to larger target set of orientations. Most efficient is the combination of the two interpolation methods, FWT followed by ASG (FWTASG).

5.6. Optimal control: improved convergence

Optimal control tools were introduced in a recent version of SIMPSON [16], and in the present release, the underlying algorithms have been improved by using higher precision

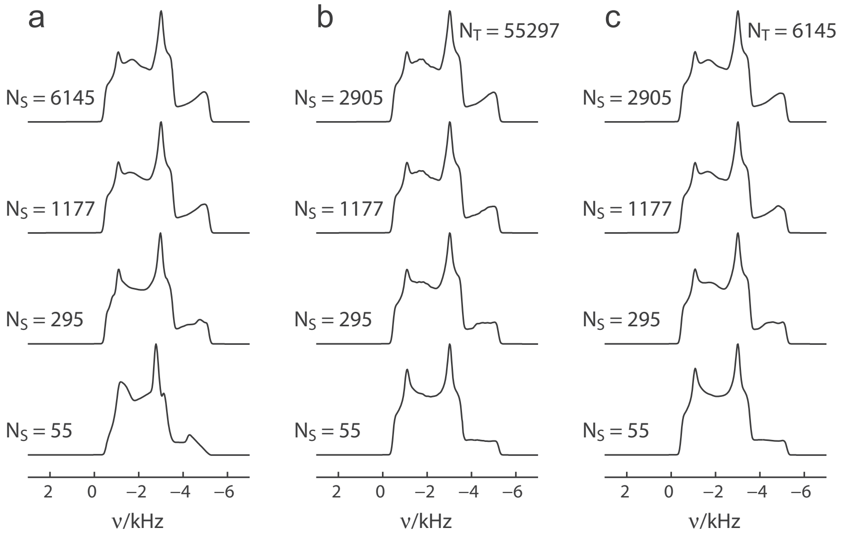

Fig. 5. Demonstration of SIMPSON procedures for powder interpolation used to assess lineshape distortions in MQ-MAS experiment. Short 150 kHz rf pulses were used to excite triple-quantum coherences and to convert it into single-quantum central transition coherence of ^{87}Rb (I = 3/2, C_Q = 3.3 MHz, η = 0.21) [89,101] at 9.4 T magnetic field strength and 9 kHz MAS. The columns correspond to the three interpolation methods: (a) Alderman–Solum–Grant (ASG), (b) Fast Wigner transformation (FWT), (c) The combination of FWT and ASG (FWTASG). Rows correspond to simulations using different numbers of crystallites. The top row corresponds to fully converged spectra.

gradients and quasi-Newton optimization. It was noted in Ref. [61] that the gradient update originally proposed in the original GRAPE algorithm of Khaneja et al. [35] represents only a first-order approximation and that using it in more elaborate optimization algorithms (that make use of second derivatives and are expected to converge faster) does not provide satisfactory results. It is therefore essential to use higher precision gradients of an OC propagator with respect to rf amplitudes in order to speed up convergence. Such calculation is controlled by parameter `par(oc_grad_level)` which corresponds to the order of the gradient approximation. Equipped with this, the user can choose between original conjugated gradients (CG) and L-BFGS algorithms for optimization through definition of `par(oc_method)` to CG and L-BFGS, respectively. Both methods have many additional parameters that are described at our website http://www.nmr.au.dk.

For the purpose of demonstration, we repeat the OC-based ^{2}H–^{13}C cross-polarization published recently [78]. We assume ^{2}H is in spin state corresponding to I_x and want to find a pulse sequence which maximizes the transfer to the carbon I_x state, with efficiency described by a function Φ ($\Phi = 1$ for 100% transfer). We set the duration of the pulse to 800 μs and divide it into 1 μs long elements, resulting altogether to 3200 optimization variables (800 for each x and y rf components on two channels). Further settings include 20 kHz MAS and averaging over 20 (α, β) REPULSION orientations and 8 γ angles. The optimizations were done separately for original CG with first and second order precision gradient (referred to as CG1 and CG2, respectively), and for L-BFGS using second order precision gradient. As can be seen from Fig. 6a, CG2 indeed provides solutions with higher transfer efficiency at the same iteration compared to CG1, and L-BFGS is still even better. So, not only do we gain about 10% in transfer efficiency, the calculation time-savings are also evident. The plot of Φ against CPU time presented in Fig. 6b shows dramatic reduction for the L-BFGS method that not only provides a better solution but also finishes much faster. The final rf sequence depicted in Fig. 6c and d provides 98.7% transfer efficiency at the given conditions and clearly possesses features synchronized with sample rotation (Fourier transform analysis reveals distinct peaks at multiples of MAS frequency, data not shown).

5.7. Spectral fitting: linking with external libraries

To enable spectral fitting in SIMPSON, we have developed a new Tcl optimization package based on the SIMPLEX optimization routines implemented in the package optimization1.0, which is part of the molecular visualization program VMD [104]. This package, called `opt`, replaces the SIMPSON implementation of MINUIT [105]. The package is written in Tcl to ensure smooth installation across all platforms and is loaded through the Tcl interface of SIMPSON as follows

```
lappend::auto_path /path/to/opt/directory
package require opt
```

The first line points to the installation path of the `opt` directory, whereas the second line loads the package into the Tcl interpreter. Since the main time consuming step in the computation is by far the generation of a trial FID/spectrum, there is no significant performance disadvantage of using Tcl scripts.

To illustrate the capabilities of `opt`, Fig. 7a shows the experimental ^{15}N spectrum of alamethicin with ^{15}N label on Aib8 in oriented lipid bilayers, recorded using ^{1}H FSLG decoupling to yield a duplet due to the ^{1}H–^{15}N dipole–dipole coupling between the amide proton and nitrogen. This spectrum has been reported elsewhere [106,107], and represents an obvious case for SIMPSON, since simulation of the spectrum requires inclusion of mosaic spread (non-uniform powder averaging) and effects from finite pulses. To simulate the spectrum, we first generate a new crystallite file containing a Gaussian distribution of crystallites around the North pole to represent the mosaic spread, which may be loaded into SIMPSON using the standard definition of `par(crystal_file)`. We may also use the possibility of using crystallite interpolation, in which case we should also generate a crystallite triangle file describing the vertices of the triangles.

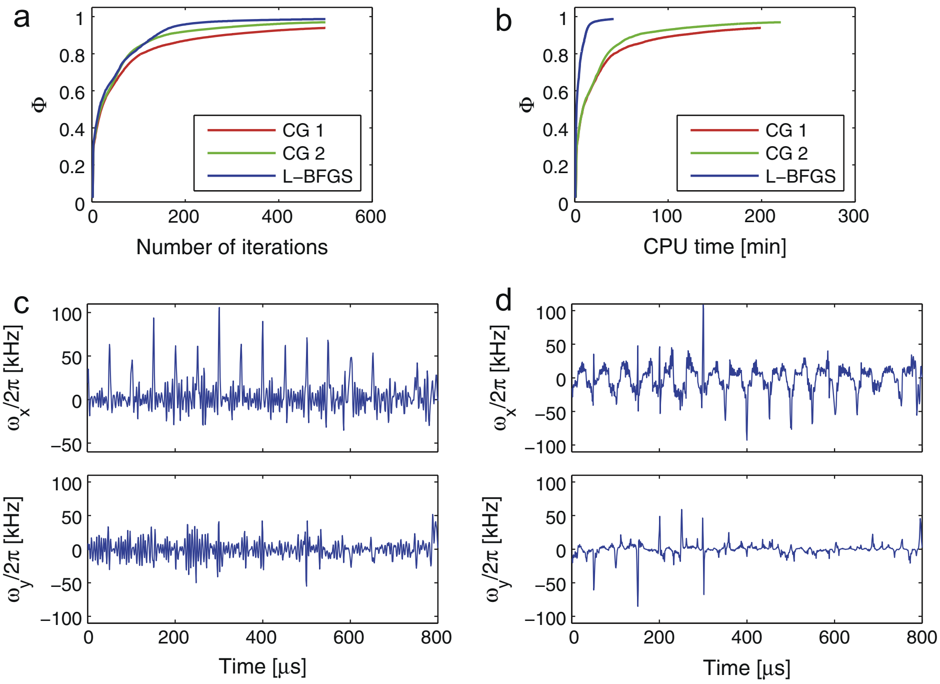

Fig. 6. Demonstration of SIMPSON optimal control tools in pulse sequence design for polarization transfer from ^{2}H to directly attached ^{13}C mediated by dipole–dipole interaction. (a) Transfer efficiency vs. number of iterations for the original conjugated gradient method with first-order approximation of the gradient (CG1, red), second-order approximation (CG2, green), and the limiter memory BFGS algorithm (L-BFGS, blue). (b) Transfer efficiency as a function of CPU time. (c and d) Resulting rf amplitudes (top: *x*-phase, bottom: *y*-phase) for the ^{2}H (c) and ^{13}C (d) channels. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The FSLG-decoupling pulse sequence is easily implemented using the new `acq_block` command

```
proc pulseq {} {
  global par
  acq_block {
    offset 0 $par(offH)
    pulse $par(tsw2) 0 x $par(rfH) x
    offset 0 [expr -$par(offH)]
    pulse $par(tsw2) 0 x $par(rfH) -x
  }
}
```

where the pulse length (`$par(tsw2)`), carrier offset (`$par(offH)`) and rf field strength (`$par(rfH)`) have been calculated previously to match the FSLG condition.

Since the isotropic chemical shift is known from other experiments [108], we only need to optimize the anisotropic chemical shift, the ^{1}H–^{15}N dipole–dipole coupling, and the degree of mosaic spread. To convincingly model the spectrum, we will also include the linebroadening as a parameter in the optimization. These parameters are defined as optimization targets by the following commands

```
opt::newpar csa 6000 50
opt::newpar dip 7000 50
opt::newpar dist 15 1
opt::newpar lb 500 50
opt::function rms
```

For each parameter is specified the name, the initial value, and the initial step size. The current value for these parameters are accessed through the Tcl variables `$opt::csa`, etc. The last line specifies that the optimization routine should optimize these parameters using the return value of the Tcl procedure `rms`. In the present case, this routine contains the elements

```
proc rms {} {
  global par
  generate_mosaic_spread $opt::dist
  set f [fsimpson [list\
    [list shift_1_aniso $opt::csa]\
    [list dipole_1_2_aniso $opt::dip]\
  ]]
  ...
  faddlb $f $opt::lb 1
  ...
  set rms [frms $f $par(f) -re]
  ...
  return $rms
}
```

The package `opt` provides a number of commands for controlling the optimization, fixing, releasing and scanning parameters, and for calculation of 95% confidence intervals, e.g.,

```
opt::scan csa
opt::minimize
opt::fix bdist
opt::release bdist
set c95 [opt::confidence csa]
```

The possibility of calculating 95% confidence intervals is exemplified in Fig. 7b. To calculate the confidence interval, the target parameter is fixed at different values around its optimum value, and for each of these values, all other parameters are optimized.

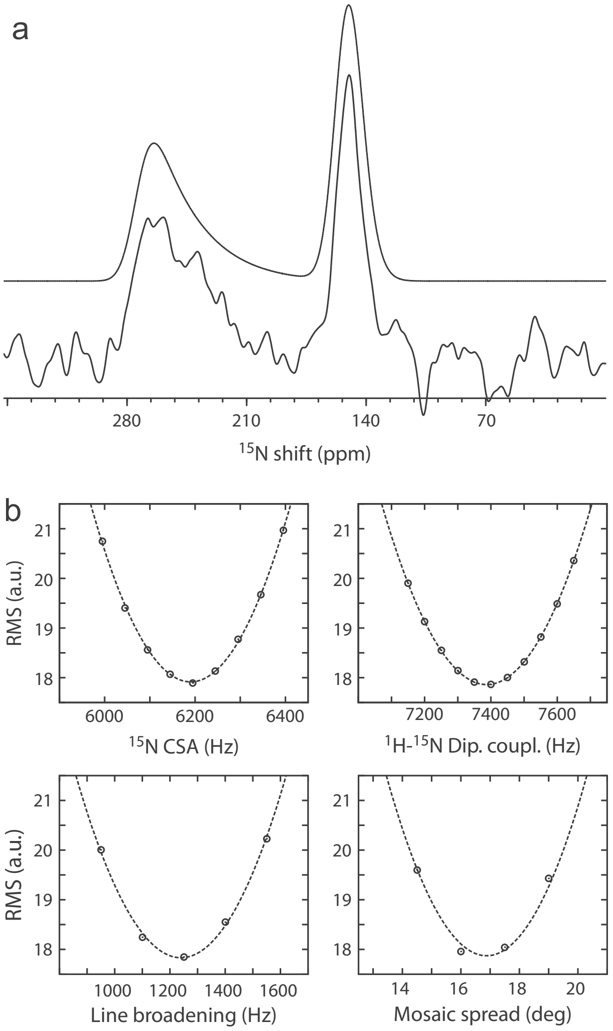

Fig. 7. (a) Experimental (bottom) and simulated (top) ^{15}N spectra of alamethicin with ^{15}N labelling of Aib8 in oriented lipid bilayers. The spectrum is recorded using ^{1}H FSLG decoupling resulting in the presence of a duplet due to the ^{1}H–^{15}N dipole–dipole coupling between the amide proton and nitrogen. The asymmetry of the duplet is due to mosaic spread [106,107]. (b) Calculations of 95% confidence intervals for the four parameters optimized. The solid lines represent the parabolic fit to the points.

The optimum RMS value for the different values of the target parameter should represent a parabola

$$\mathrm{RMS}(x) = a(x-b)^2 + c$$

and the 95% confidence interval, obtained as $C_{95} = 2/\sqrt{a}$, represents the statistical uncertainty of a parameter. In the present optimization, the resulting parameters and 95% confidence intervals are

```
csa = 6161 ± 204 Hz
dip = 7330 ± 332 Hz
bdist = 17 ± 3 deg
lb = 1155 ± 397 Hz
```

The opt package including detailed documentation is available for download from our web site http://www.nmr.au.dk.

6. Conclusion

Since its release in 2000, the open source software package SIMPSON for numerical simulations of solid-state NMR experiments has become very popular within the NMR community. In this paper, we have presented its improved version with many new features, performance matching other comparable software packages, and optimized for parallel applications. Among its major advantages are flexible and yet simple user interface based on the Tcl scripting language, allowing for simulation of essentially any NMR experiment. The user has full control over used numerical algorithms and parallelization procedures, as well as core linear algebra routines. By these means the program can be optimized for a given computer architecture. We have demonstrated its usage for simulating NMR response in relatively large spin systems and under a distribution of experimental conditions. It is also worth to mention that SIMPSON includes optimal control tools enabling automatic computer design of new NMR experiments that strongly benefits from new features introduced in this version.

The SIMPSON software is available for download from our web site, together with input files of all examples presented here. Interested readers are also encouraged to join our effort and contribute to the software through the open project philosophy.

Acknowledgments

We acknowledge support from the Danish National Research Foundation (DNRF59), the Danish Center for Scientific Computing, Lundbeckfonden, the Faculty of Sciences at Stockholm University, the European Commission under the Seventh Framework Programme (FP7) contract Bio-NMR 261863.

References

[1] A.C. Sivertsen, M. Bjerring, C.T. Kehlet, T. Vosegaard, N.C. Nielsen, Numerical simulations in biological solid-state NMR spectroscopy, Annu. Rev. NMR 54 (2005) 243–293.
[2] N.C. Nielsen, Computational aspects of biological solid-state NMR spectroscopy, in: A. Ramamoorthy (Ed.), NMR Spectroscopy of Biological Solids, CRC/Taylor & Francis, 2006, pp. 261–304.
[3] F.S. Debouregas, J.S. Waugh, Antiope, a program for computer experiments on spin dynamics, J. Magn. Reson. 96 (1992) 280–289.
[4] S.A. Smith, T.O. Levante, B.H. Meier, R.R. Ernst, Computer-simulations in magnetic-resonance – an object-oriented programming approach, J. Magn. Reson. A 106 (1994) 75–105.
[5] H. Bildsøe, STARS User's Guide, Spectrum Analysis for Rotating Solids, Publ. No. 87-195233-00, Rev. A0296, Varian Associates Inc., Palo Alto, 1996.
[6] J. Skibsted, N.C. Nielsen, H. Bildsoe, H.J. Jakobsen, Satellite transitions in Mas NMR-spectra of quadrupolar nuclei, J. Magn. Reson. 95 (1991) 88–117.
[7] M. Bak, J.T. Rasmussen, N.C. Nielsen, SIMPSON: a general simulation program for solid-state NMR spectroscopy, J. Magn. Reson. 147 (2000) 296–330.
[8] M. Bak, R. Schultz, T. Vosegaard, N.C. Nielsen, Specification and visualization of anisotropic interaction tensors in polypeptides and numerical simulations in biological solid-state NMR, J. Magn. Reson. 154 (2002) 28–45.
[9] T. Vosegaard, A. Malmendal, N.C. Nielsen, The flexibility of SIMPSON and SIMMOL for numerical simulations in solid- and liquid-state NMR spectroscopy, Monatsh. Chem. 133 (2002) 1555–1574.
[10] P. Hodgkinson, PNMRsim: A General Simulation Program for Large Problems in Solid-state NMR, Durham University, 2000.
[11] D. Massiot, F. Fayon, M. Capron, I. King, S. Le Calvé, B. Alonso, J.-O. Durand, B. Bujoli, Z. Gan, G. Hoatson, Modelling one- and two-dimensional solid-state NMR spectra, Magn. Reson. Chem. 40 (2002) 70–76.
[12] W.B. Blanton, BlochLib: a fast NMR C++ tool kit, J. Magn. Reson. 162 (2003) 269–283.
[13] A. Jerschow, MathNMR: spin and spatial tensor manipulations in Mathematica, J. Magn. Reson. 176 (2005) 7–14.
[14] 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.
[15] J.D. van Beek, MatNMR: a flexible toolbox for processing, analyzing and visualizing magnetic resonance data in Matlab((R)), J. Magn. Reson. 187 (2007) 19–26.
[16] Z. Tosner, 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.
[17] H.J. Hogben, M. Krzystyniak, G.T.P. Charnock, P.J. Hore, I. Kuprov, Spinach – a software library for simulation of spin dynamics in large spin systems, J. Magn. Reson. 208 (2011) 179–194.
[18] M.H. Levitt, SpinDynamica, Southampton, 2011.

[19] M. Edén, Y.K. Lee, M.H. Levitt, Efficient simulation of periodic problems in NMR. Application to decoupling and rotational resonance, J. Magn. Reson. A 120 (1996) 56–71.

[20] T. Charpentier, C. Fermon, J. Virlet, Efficient time propagation technique for MAS NMR simulation: application to quadrupolar nuclei, J. Magn. Reson. 132 (1998) 181–190.

[21] M. Hohwy, H. Bildsoe, H.J. Jakobsen, N.C. Nielsen, Efficient spectral simulations in NMR of rotating solids. The gamma-COMPUTE algorithm, J. Magn. Reson. 136 (1999) 6–14.

[22] P. Hodgkinson, L. Emsley, Numerical simulation of solid-state NMR experiments, Prog. Nucl. Magn. Reson. 36 (2000) 201–239.

[23] M. Edén, Computer simulations in solid-state NMR. I. Spin dynamics theory, Concepts Magn. Reson. 17A (2003) 117–154.

[24] M. Edén, Computer simulations in solid-state NMR. II. Implementations for static and rotating samples, Concepts Magn. Reson. 18A (2003) 1–23.

[25] M. Edén, Computer simulations in solid-state NMR. III. Powder averaging, Concepts Magn. Reson. 18A (2003) 24–55.

[26] N.C. Nielsen, T. Vosegaard, A. Malmendal, Analytical and numerical tools for experiment design in solid-state NMR spectroscopy, in: G.A. Webb (Ed.), Modern Magnetic Resonance. Applications in Chemistry, Biological and Marine Sciences, Springer, 2006, pp. 679–687.

[27] I. Kuprov, N. Wagner-Rundell, P.J. Hore, Polynomially scaling spin dynamics simulation algorithm based on adaptive state-space restriction, J. Magn. Reson. 189 (2007) 241–250.

[28] 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.

[29] L. Emsley, M.C. Butler, J.N. Dumez, Dynamics of large nuclear-spin systems from low-order correlations in Liouville space, Chem. Phys. Lett. 477 (2009) 377–381.

[30] H.J. Hogben, P.J. Hore, I. Kuprov, Strategies for state space restriction in densely coupled spin systems with applications to spin chemistry, J. Chem. Phys. 132 (2010).

[31] L. Emsley, J.N. Dumez, M.C. Butler, Numerical simulation of free evolution in solid-state nuclear magnetic resonance using low-order correlations in Liouville space, J. Chem. Phys. 133 (2010).

[32] T. Vosegaard, Challenges in numerical simulations of solid-state NMR experiments: spin exchange pulse sequences, Solid State Nucl. Magn. Reson. 38 (2010) 77–83.

[33] L.S. Pontryagin, The Mathematical Theory of Optimal Processes, Interscience Publishers, New York, 1962.

[34] A.E. Bryson, Y.-C. Ho, Applied Optimal Control: Optimization, Estimation, and Control, Rev. printing. ed., Hemisphere Pub. Corp.; distributed by Halsted Press, Washington, New York, 1975.

[35] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbruggen, 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.

[36] C.T. Kehlet, A.C. Sivertsen, M. Bjerring, T.O. Reiss, N. Khaneja, S.J. Glaser, N.C. Nielsen, Improving solid-state NMR dipolar recoupling by optimal control, J. Am. Chem. Soc. 126 (2004) 10202–10203.

[37] I.I. Maximov, Z. Tosner, N.C. Nielsen, Optimal control design of NMR and dynamic nuclear polarization experiments using monotonically convergent algorithms, J. Chem. Phys. 128 (2008). 184505-184501-184505-184514.

[38] I.I. Maximov, J. Salomon, G. Turinici, N.C. Nielsen, A smoothing monotonic convergent optimal control algorithm for nuclear magnetic resonance pulse sequence design, J. Chem. Phys. 132 (2010) 084107.

[39] T. Vosegaard, C. Kehlet, N. Khaneja, S.J. Glaser, N.C. Nielsen, Improved excitation schemes for multiple-quantum magic-angle spinning for quadrupolar nuclei designed using optimal control theory, J. Am. Chem. Soc. 127 (2005) 13768–13769.

[40] Z. Tosner, S.J. Glaser, N. Khaneja, N.C. Nielsen, Effective Hamiltonians by optimal control: solid-state NMR double-quantum planar and isotropic dipolar recoupling, J. Chem. Phys. 125 (2006) 184502.

[41] C. Kehlet, M. Bjerring, A.C. Sivertsen, T. Kristensen, J.J. Enghild, S.J. Glaser, N. Khaneja, N.C. Nielsen, Optimal control based NCO and NCA experiments for spectral assignment in biological solid-state NMR spectroscopy, J. Magn. Reson. 188 (2007) 216–230.

[42] A.B. Nielsen, M. Bjerring, J.T. Nielsen, N.C. Nielsen, Symmetry-based dipolar recoupling by optimal control: band-selective experiments for assignment of solid-state NMR spectra of proteins, J. Chem. Phys. 131 (2009) 025101.

[43] N.C. Nielsen, C. Kehlet, S.J. Glaser, N. Khaneja, Optimal control methods in NMR spectroscopy, in: R.K. Harris, R.E. Wasylishen (Eds.), Encyclopedia of Magnetic Resonance, John Wiley & Sons Ltd., Chichester, 2010.

[44] C. Kehlet, J.T. Nielsen, Z. Tosner, N.C. Nielsen, Resolution-enhanced solid-state NMR 13C–13C correlation spectroscopy by optimal control dipolar-driven spin-state-selective coherence transfer, J. Phys. Chem. Lett. 2 (2011) 543–547.

[45] N.M. Loening, M. Bjerring, N.C. Nielsen, H. Oschkinat, A comparison of NCO and NCA transfer methods for biological solid-state NMR spectroscopy, J. Magn. Reson. 214 (2012) 81–90.

[46] C. Bretschneider, A. Karabanov, N.C. Nielsen, W. Kockenberger, Conversion of parahydrogen induced longitudinal two-spin order to evenly distributed single spin polarisation by optimal control pulse sequences, J. Chem. Phys. 136 (2012) 094201.

[47] M.S. Vinding, Maximov II, Z. Tosner, 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.

[48] C. Laustsen, S. Bowen, M.S. Vinding, N.C. Nielsen, J.H. Ardenkjaer-Larsen, Storage of magnetization as singlet order by optimal control designed pulses, Magn. Reson. Med. 71 (2014) 921–926.

[49] M.S. Vinding, C. Laustsen, Maximov II, L.V. Sogaard, J.H. Ardenkjaer-Larsen, N.C. Nielsen, Dynamic nuclear polarization and optimal control spatial-selective C-13 MRI and MRS, J. Magn. Reson. 227 (2013) 57–61.

[50] S.K. Zaremba, Good lattice points, discrepancy, and numerical integration, Ann. Mat. Pura Appl. 73 (1966) 293.

[51] H. Conroy, Molecular Schrodinger equation. 8. A new method for evaluation of multidimensional integrals, J. Chem. Phys. 47 (1967) 5307.

[52] V.B. Cheng, Investigations of a nonrandom numerical-method for multidimensional integration, J. Chem. Phys. 59 (1973) 3992–3999.

[53] M. Bak, N.C. Nielsen, REPULSION, a novel approach to efficient powder averaging in solid-state NMR, J. Magn. Reson. 125 (1997) 132–139.

[54] M. Edén, M.H. Levitt, Computation of orientational averages in solid-state NMR by Gaussian spherical quadrature, J. Magn. Reson. 132 (1998) 220–239.

[55] U. Haeberlen, J.S. Waugh, Coherent averaging effects in magnetic resonance, Phys. Rev. 175 (1968) 453.

[56] M. Hohwy, N.C. Nielsen, Systematic design and evaluation of multiple-pulse experiments in nuclear magnetic resonance spectroscopy using a semi-continuous Baker–Campbell–Hausdorff expansion, J. Chem. Phys. 109 (1998) 3780–3791.

[57] T.S. Untidt, N.C. Nielsen, Closed solution to the Baker–Campbell–Hausdorff problem: exact effective Hamiltonian theory for analysis of nuclear-magnetic-resonance experiments, Phys. Rev. E 65 (2002) 021108.

[58] D. Siminovitch, T. Untidt, N.C. Nielsen, Exact effective Hamiltonian theory. II. Polynomial expansion of matrix functions and entangled unitary exponential operators, J. Chem. Phys. 120 (2004) 51–66.

[59] D.W. Alderman, M.S. Solum, D.M. Grant, Methods for analyzing spectroscopic line shapes. NMR solid powder patterns, J. Phys. Chem. 84 (1986) 3717.

[60] B. Stevensson, M. Edén, Interpolation by fast Wigner transform for rapid calculations of magnetic resonance spectra from powders, J. Chem. Phys. 134 (2011) 124104.

[61] P. de Fouquieres, S.G. Schirmer, S.J. Glaser, I. Kuprov, Second order gradient ascent pulse engineering, J. Magn. Reson. 212 (2011) 412–417.

[62] J. Nocedal, Updating quasi-Newton matrices with limited storage, Math. Comput. 35 (1980) 773–782.

[63] S. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, R.C. Whaley, An updated set of Basic Linear Algebra Subprograms (BLAS), ACM Trans. Math. Softw. 28 (2002) 135–151.

[64] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users' Guide, third ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.

[65] N.C. Nielsen, H. Bildsoe, H.J. Jakobsen, M.H. Levitt, Double-quantum homonuclear rotary resonance – efficient dipolar recovery in magic-angle-spinning nuclear-magnetic-resonance, J. Chem. Phys. 101 (1994) 1805–1812.

[66] A. Abragam, Principles of Nuclear Magnetism, Oxford University Press, Oxford, 1983.

[67] J.N. Dumez, M.C. Butler, E. Salager, B. Elena-Herrmann, L. Emsley, Ab initio simulation of proton spin diffusion, Phys. Chem. Chem. Phys. 12 (2010) 9172–9175.

[68] A. Grommek, B.H. Meier, M. Ernst, Distance information from proton-driven spin diffusion under MAS, Chem. Phys. Lett. 427 (2006) 404–409.

[69] C. Moler, C. Van Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later, SIAM Rev. 45 (2003) 3–49.

[70] K. Takegoshi, S. Nakamura, T. Terao, 13C–1H dipolar-assisted rotational resonance in magic-angle spinning NMR, Chem. Phys. Lett. 344 (2001) 631–637.

[71] N.J. Higham, The scaling and squaring method for the matrix exponential revisited, SIAM Rev. 51 (2009) 747–764.

[72] R.S. Dumont, S. Jain, A. Bain, Simulation of many-spin system dynamics via sparse matrix methodology, J. Chem. Phys. 106 (1997) 5928–5936.

[73] M.M. Veshtort, Numerical Simulations in NMR: Theory and Applications, Massachusetts Institute of Technology, 2003.

[74] V.E. Zorin, M. Ernst, S.P. Brown, P. Hodgkinson, Insights into homonuclear decoupling from efficient numerical simulation: techniques and examples, J. Magn. Reson. 192 (2008) 183–196.

[75] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore and London, 1996.

[76] R. Kosloff, Propagation methods for quantum molecular dynamics, Annu. Rev. Phys. Chem. 45 (1994) 145–178.

[77] M.H. Levitt, M. Edén, Numerical simulation of periodic nuclear magnetic resonance problems: fast calculation of carousel averages, Mol. Phys. 95 (1998) 879–890.

[78] D. Wei, U.m. Akbey, B. Paaske, H. Oschkinat, B. Reif, M. Bjerring, N.C. Nielsen, Optimal 2H rf pulses and 2H–13C cross-polarization methods for solid-state 2H MAS NMR of perdeuterated proteins, J. Phys. Chem. Lett. 2 (2011) 1289–1294.

[79] A.W. MacGregor, L.A. O'Dell, R.W. Schurko, New methods for the acquisition of ultra-wideline solid-state NMR spectra of spin-1/2 nuclides, J. Magn. Reson. 208 (2011) 103–113.

[80] L.A. O'Dell, K.J. Harris, R.W. Schurko, Optimized excitation pulses for the acquisition of static NMR powder patterns from half-integer quadrupolar nuclei, J. Magn. Reson. 203 (2010) 156–166.

[81] N. Okazaki, libLBFGS: A Library of Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS). <http://www.chokkan.org/software/liblbfgs/>.

[82] V.I. Lebedev, Values of the nodes and weights of ninth to seventeenth order Gauss-Markov quadrature formulae invariant under the octahedron group with inversion, USSR Comput. Math. Math. Phys. 15 (1975) 44–51.

[83] V.I. Lebedev, D.N. Laikov, A quadrature formula for the sphere of the 131st algebraic order of accuracy, Doklady Math. 59 (1999) 477–481.

[84] B. Stevensson, M. Edén, Efficient orientational averaging by the extension of Lebedev grids via regularized octahedral symmetry expansion, J. Magn. Reson. 181 (2006) 162–176.

[85] J. Keiner, S. Kunis, D. Potts, Using NFFT 3 – a software library for various nonequispaced fast Fourier transforms, ACM Trans. Math. Softw. 36 (2009) 1–30.

[86] S. Kunis, D. Potts, Fast spherical Fourier algorithms, J. Comput. Appl. Math. 161 (2003) 75–98.

[87] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, V. Sunderam, PVM: Parallel Virtual Machine: A Users Guide and Tutorial for Networked Parallel Computing, MIT Press, Cambridge, MA, 2005.

[88] E. Gabriel, G.E. Fagg, G. Bosilca, T. Angskun, J.J. Dongarra, J.M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R.H. Castain, D.J. Daniel, R.L. Graham, T.S. Woodall, Open MPI: goals, concept, and design of a next generation MPI implementation, Lect. Notes Comput. Sci. 3241 (2004) 97–104.

[89] T. Vosegaard, J. Skibsted, H. Bildsoe, H.J. Jakobsen, Combined effect of 2nd-order quadrupole coupling and chemical shift anisotropy on the central transition in MAS NMR of quadrupolar nuclei – Rb-87 MAS NMR of RbClO4, J. Phys. Chem. 99 (1995) 10731–10735.

[90] A.E. Bennett, C.M. Rienstra, M. Auger, K.V. Lakshmi, R.G. Griffin, Heteronuclear decoupling in rotating solids, J. Chem. Phys. 103 (1995) 6951–6958.

[91] B.M. Fung, A.K. Khitrin, K. Ermolaev, An improved broadband decoupling sequence for liquid crystals and solids, J. Magn. Reson. 142 (2000) 97–101.

[92] A. Detken, E.H. Hardy, M. Ernst, B.H. Meier, Simple and efficient decoupling in magic-angle spinning solid-state NMR: the XiX scheme, Chem. Phys. Lett. 356 (2002) 298–304.

[93] M. Weingarth, P. Tekely, G. Bodenhausen, Efficient heteronuclear decoupling by quenching rotary resonance in solid-state NMR, Chem. Phys. Lett. 466 (2008) 247–251.

[94] M. Edén, M.H. Levitt, Pulse sequence symmetries in the nuclear magnetic resonance of spinning solids: application to heteronuclear decoupling, J. Chem. Phys. 111 (1999) 1511–1519.

[95] M. Carravetta, M. Edén, X. Zhao, A. Brinkmann, M.H. Levitt, Symmetry principles for the design of radiofrequency pulse sequences in the nuclear magnetic resonance of rotating solids, Chem. Phys. Lett. 321 (2000) 205–215.

[96] M. Ernst, Heteronuclear spin decoupling in solid-state NMR under magic-angle sample spinning, J. Magn. Reson. 162 (2003) 1–34.

[97] S. Paul, N.D. Kurur, P.K. Madhu, On the choice of heteronuclear dipolar decoupling scheme in solid-state NMR, J. Magn. Reson. 207 (2010) 140–148.

[98] M. Leskes, R.S. Thakur, P.K. Madhu, N.D. Kurur, S. Vega, Bimodal Floquet description of heteronuclear dipolar decoupling in solid-state nuclear magnetic resonance, J. Chem. Phys. 127 (2007).

[99] R.S. Thakur, N.D. Kurur, P.K. Madhu, Swept-frequency two-pulse phase modulation for heteronuclear dipolar decoupling in solid-state NMR, Chem. Phys. Lett. 426 (2006) 459–463.

[100] J.M. Vinther, A.B. Nielsen, M. Bjerring, E.R.H. van Eck, A.P.M. Kentgens, N. Khaneja, N.C. Nielsen, Refocused continuous-wave decoupling: a new approach to heteronuclear dipolar decoupling in solid-state NMR spectroscopy, J. Chem. Phys. 137 (2012).

[101] T. Vosegaard, J. Skibsted, H. Bildsoe, H.J. Jakobsen, Quadrupole coupling and anisotropic shielding from single-crystal NMR of the central transition for quadrupolar nuclei. Rb-87 NMR of RbClO4 and Rb2SO4, J. Magn. Reson. A 122 (1996) 111–119.

[102] L. Frydman, J.S. Harwood, Isotropic spectra of half-integer quadrupolar spins from bidimensional magic-angle-spinning NMR, J. Am. Chem. Soc. 117 (1995) 5367–5368.

[103] A. Medek, J.S. Harwood, L. Frydman, Multiple-quantum magic-angle spinning NMR: a new method for the study of quadrupolar nuclei in solids, J. Am. Chem. Soc. 117 (1995) 12779–12787.

[104] W. Humphrey, A. Dalke, K. Schulten, VMD: visual molecular dynamics, J. Mol. Graph. 14 (1996) 33–38.

[105] F. James, M. Roos, MINUIT – system for function minimization and analysis of parameter errors and correlations, Comput. Phys. Commun. 10 (1975) 343–367.

[106] T. Vosegaard, K. Bertelsen, J.M. Pedersen, L. Thøgersen, B. Schiøtt, E. Tajkhorshid, T. Skrydstrup, N.C. Nielsen, Resolution enhancement in solid-state NMR of oriented membrane proteins by anisotropic differential linebroadening, J. Am. Chem. Soc. 130 (2008) 5028–5029.

[107] K. Bertelsen, B. Paaske, L. Thøgersen, E. Tajkhorshid, B. Schiøtt, T. Skrydstrup, N.C. Nielsen, T. Vosegaard, Residue-specific information about the dynamics of antimicrobial peptides from (1)H–(15)N and (2)H solid-state NMR spectroscopy, J. Am. Chem. Soc. 131 (2009) 18335–18342.

[108] J. Dittmer, L. Thøgersen, J. Underhaug, K. Bertelsen, T. Vosegaard, J.M. Pedersen, B. Schiøtt, E. Tajkhorshid, T. Skrydstrup, N.C. Nielsen, Incorporation of antimicrobial peptides into membranes: a combined liquid-state NMR and molecular dynamics study of alamethicin in DMPC/DHPC bicelles, J. Phys. Chem. B 113 (2009) 6928–6937.