Journal of Magnetic Resonance **147,** 296–330 (2000)
doi:10.1006/jmre.2000.2179, available online at http://www.idealibrary.com on IDEAL®

SIMPSON: A General Simulation Program for Solid-State NMR Spectroscopy

Mads Bak, Jimmy T. Rasmussen,[1] and Niels Chr. Nielsen[2]

Laboratory for Biomolecular NMR Spectroscopy, Department of Molecular and Structural Biology, University of Aarhus, DK-8000 Aarhus C, Denmark

Received May 10, 2000; revised July 26, 2000

A computer program for fast and accurate numerical simulation of solid-state NMR experiments is described. The program is designed to emulate a NMR spectrometer by letting the user specify high-level NMR concepts such as spin systems, nuclear spin interactions, RF irradiation, free precession, phase cycling, coherence-order filtering, and implicit/explicit acquisition. These elements are implemented using the Tcl scripting language to ensure a minimum of programming overhead and direct interpretation without the need for compilation, while maintaining the flexibility of a full-featured programming language. Basicly, there are no intrinsic limitations to the number of spins, types of interactions, sample conditions (static or spinning, powders, uniaxially oriented molecules, single crystals, or solutions), and the complexity or number of spectral dimensions for the pulse sequence. The applicability ranges from simple 1D experiments to advanced multiple-pulse and multiple-dimensional experiments, series of simulations, parameter scans, complex data manipulation/visualization, and iterative fitting of simulated to experimental spectra. A major effort has been devoted to optimizing the computation speed using state-of-the-art algorithms for the time-consuming parts of the calculations implemented in the core of the program using the C programming language. Modification and maintenance of the program are facilitated by releasing the program as open source software (General Public License) currently at http://nmr.imsb.au.dk. The general features of the program are demonstrated by numerical simulations of various aspects for REDOR, rotational resonance, DRAMA, DRAWS, HORROR, C7, TEDOR, POST-C7, CW decoupling, TPPM, F-SLG, SLF, SEMA-CP, PISEMA, RFDR, QCPMG-MAS, and MQ-MAS experiments.

INTRODUCTION

During the past decade solid-state NMR spectroscopy has undergone a tremendous evolution from being based on relatively simple one-dimensional pulse sequences to now involving a large repertoire of advanced multiple-pulse and multiple-dimensional experiments designed to extract specific information about the structure and dynamics of molecules in the solid phase (*1–8*). In many respects this evolution resembles the earlier and still strongly ongoing evolution of multidimensional liquid-state NMR spectroscopy (*9–11*). In both cases state-of-the-art experiments are constructed in a modular fashion using pulse sequence building blocks accomplishing certain coherence transfers or evolution under specific parts of the internal Hamiltonian. One major difference, however, is that solid-state NMR is influenced directly by anistropic nuclear spin interactions which on one hand complicate the achievement of high-resolution spectra and on the other hand may provide important information about structure and dynamics. This dual aspect has motivated the design of advanced pulse sequence elements which through decoupling and recoupling tailor the Hamiltonian to cause evolution under the specific interaction(s) probing the desired structural information while efficiently suppressing undesired interactions. Based on analytical evaluation of the perturbed Hamiltonian (*1, 2, 4, 6, 12–15*) and numerical simulations, a large number of experiments have been constructed which, via dipolar coupling, anisotropic chemical shielding, and quadrupolar coupling interactions, provide information about local molecular structure and dynamics in terms of the electronic/nuclear coordination environment, internuclear distances, bonding angles, and models for motional processes.

Often, the internal Hamiltonian in solid-state NMR contains several orientation-dependent terms with amplitudes comparable to or larger than the amplitude of the external manipulation by RF irradiation and sample spinning. This may be the case for desired as well as undesired terms of the Hamiltonian, implying that accurate determination of structural parameters from the desired terms as well as evaluation of the multiple-pulse building blocks providing suppression of undesired terms very often depend on the ability to numerically simulate the spin dynamics of the actual NMR experiment. This applies, for example, to the solid-state NMR experiments for which dipolar recoupling (e.g., rotational resonance (*16, 17*), REDOR (*18*), DRAMA (*19*), DRAWS (*20*), RFDR (*21*), RIL (*22*), HORROR (*23*), BABA (*24*), C7 (*25, 26*), RFDRCP (*27*)), multiple-pulse homo- or heteronuclear decoupling (e.g., BR-24 (*28*), F-SLG (*29*), MSHOT-3 (*30*), TPPM (*31*)), cross-polarization (*32, 33*), QCPMG-MAS (*34*), or MQ-MAS (*35*) pulse sequences are indispensable building blocks. Thus, considering the very large

[1] Present address: Medtronic Functional Diagnostics, Skovlunde, Denmark.
[2] To whom correspondence should be addressed. E-mail: ncn@imsb.au.dk.

1090-7807/00 $35.00

number of advanced experiments already available, the large number of possible combinations between these, and the rapidly increasing number of new experimental procedures presented every year, there is a substantial need for a general and consistent simulation tool to support experiment design, user-specific method implementation, and evaluation of spectral data. This need is reinforced by the fact that most state-of-the-art experiments are simulated using custom-made programs tailored to the specific pulse sequences and typically not accessible to or applicable for the general user. The shortcomings of this currently prevailing approach are apparent. It requires redundant work not only for the involved group but also for other groups implementing the new techniques and does not encourage one to create programs usable or understandable by others. Obviously, a far better solution would be to have a general-purpose program available for simulation of solid-state NMR experiments. General programs of this sort, for example, ANTIOPE (*36*) and the more general GAMMA simulation environment (*37*), are available to the NMR community, although to the best of our knowledge so far none of these programs has specialized in time-efficient simulations within modern solid-state NMR spectroscopy. We should note that highly specialized programs such as STARS (*38, 39*) and QUASAR (*40*), allowing simulation and iterative fitting of single-pulse solid-state NMR spectra for spin-1/2 or half-integer quadrupolar nuclei, are available as integral parts in commercial NMR software.

In this paper we present a general simulation program for solid-state NMR spectroscopy (SIMPSON) which is designed to work as a "computer spectrometer." The primary aim has been to design a program which is relatively easy to use, transparent, and still maintains the flexibility to allow simulation of virtually all types of NMR experiments. With the major focus being solid-state NMR, the program has been optimized for fast calculation of multiple-pulse experiments for rotating powder samples, which generally is considered quite demanding. We note that the program obviously may be used equally well for static powders, single crystals, oriented samples, and liquid-state NMR experiments. The user interface to the program is the Tcl scripting language (*41, 42*), being well-suited to provide the necessary high-level NMR functionality in a transparent form. This covers definition and operation of the basic elements of a NMR experiment (e.g., the spin system, nuclear spin interactions, RF irradiation, frequency switching, coherence-order filtering, free precession, acquisition, etc.) as well as controlling experimental parameters, processing of the experiment, and functions for the data processing. Encapsulating all mathematical and spin-quantum-mechanical calculations at this level of abstraction serves to minimize the content of the input file without sacrificing the functionality of the simulation. Within the proposed simulation environment, it is straightforward to scale the functionality from the most simple simulation of one-dimensional spectra specified by only a few lines of code to coherence transfer functions for advanced pulse sequence elements, scans over parameters describing the internal Hamiltonian, or the experimental manipulations, multidimensional simulations, and iterative fitting of experimental spectra. The program structure encourages the analysis of important, albeit typically disregarded, effects from small couplings to nearby spins, finite RF pulse irradiation, RF inhomogeneity, hardware-induced "hidden" delays, and phase cycling. Thus, in combination with an extensive function library the program is geared to be a tool to systematic experiment design, examination of pulse sequences proposed in the literature, testing pulse sequences on relevant spin systems prior to spectrometer implementation, and checking the consequences of experimental imperfections during pulse sequence implementation and as a tool to extract structural parameters from experimental spectra through least-squares iterative fitting. A major effort has been to devoted to produce a user-friendly tool which serves all elements of multiple-pulse solid-state NMR simulations from the initial testing calculations, pulse sequence implementation, iterative fitting of experimental spectra, and advanced data processing to interactive viewing and manipulation of data.

THEORY

In this section the theory relevant for simulation of solid-state NMR spectra is briefly reviewed. This provides the reader with the basic symbols, definitions, and conventions used for the description of the Hamiltonian as well as the transformations employed in spin and real space to define the actual NMR experiment. To ensure general applicability the theory is described in relation to solid-state NMR on rotating powders. Other cases, including static powders, single crystals, uniaxially oriented molecules, and liquids, are easily handled as special cases to this. To avoid an unacceptably long description, we refrain from going into details with respect to the numerically important aspects of powder averaging, time and spatial symmetry relations, numerical integration of the spin dynamics, etc., but rather make extensive reference to already published material on these aspects.

The simulation of a NMR experiment essentially amounts of a numerical evaluation of the Liouville–von Neumann equation of motion

$$\frac{d}{dt}\rho(t) = -i[H(t), \rho(t)], \qquad [1]$$

where $\rho(t)$ is the reduced density matrix representing the state of the spin system and $H(t)$ the time-dependent Hamiltonian describing the relevant nuclear spin interactions and the external operations. For simplicity we have presently disregarded effects from relaxation and other dissipative processes in the theory as well as in the simulation software described in this paper. Thus, the formal solution to Eq. [1] may be written

$$\rho(t) = U(t, 0)\rho(0)U^{\dagger}(t, 0), \quad [2]$$

where $\rho(0)$ is the density operator at thermal equilibrium (or a density operator resulting from a given preparation sequence) and $U(t, 0)$ is the unitary propagator (i.e., the exponential operator) responsible for the spin dynamics in the period from 0 to t. $U(t, 0)$ is related to the Hamiltonian according to

$$U(t, 0) = \hat{T} \exp\left\{ -i \int_0^t H(t')dt' \right\}, \quad [3]$$

with $\hat{T}$ being the Dyson time-ordering operator relevant for Hamiltonians containing noncommuting components. Although a large number of advanced numerical integration methods (*43*) in principle may be applied to derive $U(t, 0)$, it typically proves most efficient numerically to approximate the integral by a simple time-ordered product

$$U(t, 0) = \prod_{j=0}^{n-1} \exp\{-iH(j\Delta t)\Delta t\}, \quad [4]$$

where n is the number of infinitesimal time intervals Δt over each of which the Hamiltonian may be considered time-independent and which overall span the full period from 0 to $t = n\Delta t$. For each time interval the exponentiation is accomplished by diagonalization of the matrix representation for the Hamiltonian. To ensure fast convergence and to focus on the interactions of specific interest these operations are usually performed in an appropriate interaction frame.

In the most typical cases, the Hamiltonian is described by the high-field truncated components in the Zeeman interaction frame. For a spin system consisting of n spins I, being of the same or different spin species, the Hamiltonian takes the form

$$H = H_{\mathrm{RF}} + H_{\mathrm{CS}} + H_J + H_{\mathrm{D}} + H_{\mathrm{Q}}, \quad [5]$$

where

$$H_{\mathrm{RF}} = \sum_i |\omega^i_{\mathrm{RF}}(t)|(I_{ix}\cos\phi_i + I_{iy}\sin\phi_i) \quad [6]$$

$$H_{\mathrm{CS}} = \sum_i \omega^i_{\mathrm{CS},0}(t) I_{iz} \quad [7]$$

$$H_J = \sum_{i,j} -\omega^{ij}_{J_{\mathrm{iso}},0}(t) \frac{1}{\sqrt{3}} \mathbf{I}_i \cdot \mathbf{I}_j + \omega^{ij}_{J_{\mathrm{aniso}},0}(t) \frac{1}{\sqrt{6}} (3I_{iz}I_{jz} - \mathbf{I}_i \cdot \mathbf{I}_j) \quad [8]$$

$$H_{\mathrm{D}} = \sum_{i,j} \omega^{ij}_{\mathrm{D},0}(t) \frac{1}{\sqrt{6}} (3I_{iz}I_{jz} - \mathbf{I}_i \cdot \mathbf{I}_j) \quad [9]$$

$$H_{\mathrm{Q}} = \sum_i \omega^i_{\mathrm{Q},0}(t) \frac{1}{\sqrt{6}} (3I^2_{iz} - \mathbf{I}^2_i) + \frac{1}{2\omega^i_0} \{\omega^i_{\mathrm{Q},-2}(t)\omega^i_{\mathrm{Q},2}(t)(2\mathbf{I}^2_i - 2I^2_{iz} - \mathbf{1})I_{iz} + \omega^i_{\mathrm{Q},-1}(t)\omega^i_{\mathrm{Q},1}(t)(4\mathbf{I}^2_i - 8I^2_{iz} - \mathbf{1})I_{iz}\}, \quad [10]$$

with i, j specifying the involved spins. The various terms represent contributions from ϕ_i-phase RF irradiation with an angular nutation frequency of $\omega^i_{\mathrm{RF}} = -\gamma_i B^i_{\mathrm{RF}}$ (RF), chemical shift (CS), indirect spin–spin coupling (J), dipole–dipole coupling (D), and quadrupolar coupling (Q). We note that the RF Hamiltonian in Eq. [6] in accord with common practice employs the magnitude of the RF nutation frequency with the pulse phases ϕ_i adopting potential dependence on the sign of the gyromagnetic ratio γ_i (*9*, *44*). The first and second terms in Eq. [8] describe scalar (J_{iso}) and anisotropic (J_{aniso}) J coupling, respectively. Likewise, the first term in Eq. [10] represents first-order quadrupolar coupling while the last term includes the secular components for the second-order quadrupolar coupling. We note that for a coupling between nuclei of different spin species the operator product $\mathbf{I}_i \cdot \mathbf{I}_j$ is truncated to $I_{iz}I_{jz}$. Finally, it should be clearly stated that the Hamiltonian by no means is restricted to the elements in Eqs. [6]–[10]. Using the same formalism, it is straightforward to formulate, for example, second-order cross-terms between the dipolar and quadrupolar couplings.

For the various internal Hamiltonians H_λ with λ = CS, J_{iso}, J_{aniso}, D, and Q, the frequency coefficients depend on some fundamental constants as well as time and spatial (i.e., orientation dependent) functions which in the present formulation are of rank 0 and 2 for isotropic and anisotropic parts of the interactions, respectively. Overall these dependencies may conveniently be expressed in terms of a Fourier expansion,

$$\omega_{\lambda,m'}(t) = \sum_{m=-2}^{2} \omega^{(m)}_{\lambda,m'} e^{im\omega_r t}, \quad [11]$$

where $\omega_r/2\pi$ is the spin rate and the Fourier coefficients are

$$\omega^{(m)}_{\lambda,m'} = \omega^{\lambda}_{\mathrm{iso}}\delta_{m,0} + \omega^{\lambda}_{\mathrm{aniso}}\left\{ D^{(2)}_{0,-m}(\Omega^{\lambda}_{PR}) - \frac{\eta^{\lambda}}{\sqrt{6}} \times [D^{(2)}_{-2,-m}(\Omega^{\lambda}_{PR}) + D^{(2)}_{2,-m}(\Omega^{\lambda}_{PR})] \right\} d^{(2)}_{-m,m'}(\beta_{RL}), \quad [12]$$

where $\delta_{m,0}$ is a standard Kronecker delta and the constants

TABLE 1
Constants Relevant for the Fourier Series Parametrization of the Internal Part of the Nuclear Spin Hamiltonian (see Text)[a]

λ	Spins	$\omega^{\lambda}_{\text{iso}}$	$\omega^{\lambda}_{\text{aniso}}$	η^{λ}
CS	i	$\omega^i_0\delta^i_{\text{iso}} - \omega_{ref}$	$\omega^i_0\delta^i_{\text{aniso}}$	η^i_{CS}
D	i, j	0	$\sqrt{6}\ b_{ij}$	0
J_{iso}	i, j	$-2\pi\sqrt{3}\ J^{ij}_{\text{iso}}$	0	0
J_{aniso}	i, j	0	$2\pi\sqrt{6}\ J^{ij}_{\text{aniso}}$	η^{ij}_J
Q	i	0	$2\pi\sqrt{6}\ C^i_{\text{Q}}/(4I_i(2I_i - 1))$	η^i_{Q}

[a] Given in angular frequency units. The isotropic value, anisotropy, and asymmetry parameter for the chemical shift interaction are related to the principal elements of the shift tensor according to $\delta^i_{\text{iso}} = \frac{1}{3}(\delta^i_{xx} + \delta^i_{yy} + \delta^i_{zz})$, $\delta^i_{\text{aniso}} = \delta^i_{zz} - \delta^i_{\text{iso}}$, and $\eta^i_{\text{CS}} = (\delta^i_{yy} - \delta^i_{xx})/\delta^i_{\text{aniso}}$, respectively, with the principal elements $\delta^i_{11} \geq \delta^i_{22} \geq \delta^i_{33}$ labeled and ordered according to $|\delta^i_{zz} - \delta^i_{\text{iso}}| \geq |\delta^i_{xx} - \delta^i_{\text{iso}}| \geq |\delta^i_{yy} - \delta^i_{\text{iso}}|$. We note that conversion to the chemical shielding convention simply amounts to replacing all δ's by σ's (using the ordering $\sigma^i_{11} \leq \sigma^i_{22} \leq \sigma^i_{33}$) and reverting the sign of σ^i_{iso} and σ^i_{aniso}. The i-spin Larmor frequency is defined as $\omega^i_0 = -\gamma_i B_0$, where γ_i is the gyromagnetic ratio and B_0 the flux density of the static magnetic field. ω_{ref} is an optional rotating frame reference frequency. The dipolar coupling constant is defined as $b_{ij} = -\gamma_i\gamma_j\mu_0\hbar/(r^3_{ij}4\pi)$, where r_{ij} is the internuclear distance (SI units). The quadrupolar coupling constant is defined as $C^i_{\text{Q}} = (e^2Qq)/h$. I_i denotes the spin-quantum number for spin i.

specifying the isotropic ($\omega^{\lambda}_{\text{iso}}$) and anisotropic ($\omega^{\lambda}_{\text{aniso}}$, η^{λ}) contributions to the Fourier coefficients are listed in Table 1 for the various interactions.

The orientation dependence for the anisotropic interactions is expressed in terms of second-rank Wigner ($D^{(2)}$) and reduced Wigner ($d^{(2)}$) rotation matrices (*2, 3*). For a given interaction λ these matrices describe coordinate transformations from the principal-axis frame (P^{λ}) to the laboratory-fixed frame (L) where the experiment is performed. The transformations relevant for rotating powder experiments additionally involve a crystal-fixed frame (C), representing a common frame of reference in the presence of several interaction tensors, as well as a rotor-fixed frame (R). The various frames are illustrated in Fig. 1 with the axes of the ORTEP-type representation designating the three principal elements for an anisotropic interaction tensor as described by Mehring (*2*). The Euler angles relating two frames X and Y are denoted $\Omega^{\lambda}_{XY} = \{\alpha^{\lambda}_{XY}, \beta^{\lambda}_{XY}, \gamma^{\lambda}_{XY}\}$. Thus, the frames P and R are related by

$$D^{(2)}_{m',m}(\Omega^{\lambda}_{PR}) = \sum_{m''=-2}^{2} D^{(2)}_{m',m''}(\Omega^{\lambda}_{PC})\, D^{(2)}_{m'',m}(\Omega_{CR}), \qquad [13]$$

while R is related to L by a Wigner rotation using the Euler angles $\alpha_{RL} = \omega_r t$ (included in Eq. [11]) and β_{RL} (often set to the magic angle, $\beta_{RL} = \tan^{-1}\sqrt{2}$, while γ_{RL} may arbitrarily be set to zero within the high-field approximation. The angles Ω_{CR} describe the orientation of the individual crystallite relative to R.

The spin parts of the interactions are numerically manipulated by operation on their matrix representations. For a single spin I the matrix representation is readily established using the well-known relations

$$\langle m'|I_{\pm}|m\rangle = \sqrt{I(I+1) - m(m \pm 1)}\ \delta_{m',m\pm1} \qquad [14]$$

$$\langle m'|I_z|m\rangle = m\delta_{m',m}, \qquad [15]$$

with the step operators related to the Cartesian operators as $I_{\pm} = I_x \pm iI_y$, while I_z is related to the polarization operators as $I_{\alpha/\beta} = \frac{1}{2}(\mathbf{1} +/- 2I_z)$ with $\mathbf{1}$ being the unity operator. The matrix representation for a given operator Q contains $(2I + 1)^2$ elements $Q_{kl} = \langle I - k + 1|Q|I - l + 1\rangle$, where k and l denote the row and column positions, respectively. For a spin system consisting of n nuclei, the matrix representation for a given I_{iq} spin operator for the nucleus i is described by the direct products of unit operators and the relevant single-spin operator I_q, i.e.,

$$I_{iq} = \mathbf{1}_1 \otimes \ldots \mathbf{1}_{i-1} \otimes I_q \otimes \mathbf{1}_{i+1} \otimes \ldots \mathbf{1}_n. \qquad [16]$$

We note that Eq. [16] implies matrix representations for which the Zeeman product basis functions for, e.g., an ISR three-spin-1/2 system are ordered as $|\alpha\alpha\alpha\rangle$, $|\alpha\alpha\beta\rangle$, $|\alpha\beta\alpha\rangle$, $|\alpha\beta\beta\rangle$, $|\beta\alpha\alpha\rangle$, ... etc., with $|\alpha\rangle = |1/2\rangle$, $|\beta\rangle = |-1/2\rangle$, and the spins ordered I, S, R.

Allowing for the detection of signals corresponding to hermitian as well as nonhermitian operators (the latter being relevant for quadrature detection and experiments using pulsed field gradients), the NMR response signal for a crystallite characterized by the orientation Ω_{CR} is generally described by the projection or "expectation value" for the transposed and conjugated detection operator ($Q^{\dagger}_{\text{det}}$):

$$s(t; \Omega_{CR}) = \langle Q^{\dagger}_{\text{det}}|\rho(t; \Omega_{CR})\rangle \qquad [17]$$

$$= \text{Tr}\{Q_{\text{det}}\rho(t; \Omega_{CR})\}, \qquad [18]$$

typically sampled equidistantly with respect to time, i.e., $t = m\Delta t$, $m = 0, 1, \ldots, n - 1$, where n is the number of sampling points. In the case of a powder sample, the signal

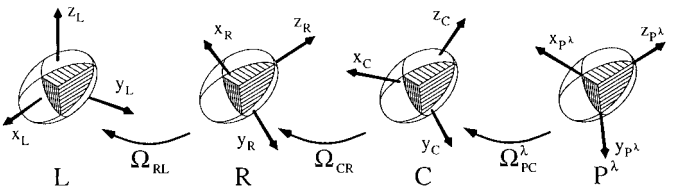

FIG. 1. ORTEP-type representation of a spatial second-rank anisotropic interaction tensor in its principal-axis system (P^{λ}), a crystallite-fixed coordinate system (C), the rotor-fixed coordinate system (R), and the laboratory-fixed coordinate system (L) along with the Euler angles $\Omega_{XY} = \{\alpha_{XY}, \beta_{XY}, \gamma_{XY}\}$ describing transformation between the various frames X and Y.

needs to be averaged over all uniformly distributed powder angles Ω_{CR} according to

$$\bar{s}(t) = \frac{1}{8\pi^2} \int_0^{2\pi} d\alpha_{CR} \int_0^{\pi} d\beta_{CR} \sin(\beta_{CR}) \int_0^{2\pi} d\gamma_{CR} s(t; \Omega_{CR}), \quad [19]$$

where we for the sake of generality assumed averaging over the full sphere. We note that in numerous cases the intrinsic symmetry of the orientation dependence allows reduction of the averaging to one-half or one-quarter of the sphere (*39* ,*45*). For numerical simulations this integral is conveniently approximated by the discrete sum

$$\bar{s}(t) = \sum_{k=1}^{N} \sum_{l=1}^{M} s(t; \alpha_{CR}^k, \beta_{CR}^k, \gamma_{CR}^l) \frac{w_k}{M}, \quad [20]$$

where the averaging is split into M angles γ_{CR}^l with contributions from N pairs of α_{CR}^k and β_{CR}^k powder angles weighted by w_k using the normalization $\sum_{k=1}^{N} w_k = 1$.

Depending on the actual solid-state NMR experiment to be simulated, several options exist for the powder averaging. For efficient α_{CR} and β_{CR} averaging, it is generally recommendable to use averaging schemes providing the most uniform (and thereby equally weighted) distribution of crystallite orientations over the unit sphere. This may be accomplished using angle/weight sets derived using the methods of Zaremba, Conroy, and Cheng *et al.* (*46–48*) or the more efficient REPULSION (*49*) or Lebedev (*45*) powder averaging schemes. In cases of wide powder patterns, such as static or magic-angle-spinning (MAS) powder patterns induced by first- or second-order quadrupolar coupling interactions, it may be recommendable to support this powder averaging by interpolation (*39*) using the recipe of Alderman *et al.* (*50*). For nonspinning samples the signal is invariant to the γ_{CR} crystallite angle which accordingly can be arbitrarily set to zero provided $\Omega_{RL} = \{0, 0, 0\}$. For rotating powders, it is often possible to exploit the symmetric time-dependence of the Hamiltonian to improve the efficiency of the calculations (*51*). In particular for appropriately rotor synchronized pulse sequences, it has proven useful to consider these symmetries in combination with the time–translation relationship between γ_{CR} and the sample-rotation angle $\omega_r t$ as recently described by several authors (*52–54*). We note that, under certain circumstances, it may be even more efficient to systematically reuse other combinations of propagators reflecting certain combinations of γ_{CR} and $\omega_r t$ (*55*). Finally, we should briefly address attention to a number of additional elements used to speed up the simulations: (i) for diagonal Hamiltonians (i.e., Hamiltonians without mutually noncommuting elements) the integration in Eq. [3] is conducted using analytical solutions, (ii) since the internal Hamiltonian commutes with the Zeeman operator(s), evolution under pulses with phase $\phi_i \neq 0$ is most efficiently accomplished by calculating the propagator for a pulse with phase $\phi_i = 0$ (i.e., H is real) followed by the appropriate z rotation (*26*), and (iii) a particularly efficient variant of γ-COMPUTE is applied when the start and detect operators fulfill the relation $\rho(0) = \frac{1}{2}(Q_{\text{det}} + Q_{\text{det}}^{\dagger})$ (*54*).

SIMULATION ENVIRONMENT

While the Hamiltonians and transformations described in the previous section through generality allow for the description of essentially all types of NMR experiments, they do not offer a simple framework for efficient implementation and fast simulation of advanced solid-state NMR experiments. For this purpose, it is essential to establish a user-friendly interface allowing the fundamental definitions and the transformations in spin and real space to be controlled with a minimum of instructions/commands, each requiring as little information as possible. This should be accomplished while maintaining the flexibility as allowed at the level of the Hamiltonians. Thus, with the specific aim of simulating practical solid-state NMR experiments, it is desirable to perform simple operations at the same level of abstraction as on a flexible computer interface to a NMR spectrometer. In this case all spin and spatial dependencies for the internal part of the Hamiltonian are provided by the sample itself, leaving only the external manipulations to be controlled by the experimentator. Obviously, in numerical simulations it is necessary to control both parts of the Hamiltonian, but it appears intuitively that the optimum interface for a simulation program should allow for separate control of these. For example, this would enable fast implementation of pulse sequences and the establishment of pulse sequence libraries.

Considering the practical implementation, the flow of the calculations, and the data processing, we propose a user interface containing four sections. These include a section `spinsys` for definition of the internal Hamiltonian in terms of spin system and nuclear spin interactions, a section `par` for definition of the global experimental parameters (e.g., crystallite orientations, sample spinning, operators for the initial spin state and detection), a section `pulseq` for definition of the pulse sequence, and finally a section `main` to control processing of the pulse sequence, storage of data, and data processing. Obviously, this interface is intimately related to the theory given in the previous section as well as to supplementary software for data manipulation, visualization, and analysis. The overall structure of the simulation environment is illustrated schematically in Fig. 2.

With the aim of specifying the necessary information in a flexible, transparent, and user-friendly manner, the user interface to the program based on the Tcl scripting language (*41, 42*). Tcl is ideally suited to this purpose as it (i) is easier to learn than C and similar high-level programming languages (no type checking, complex data types, or variable declara-

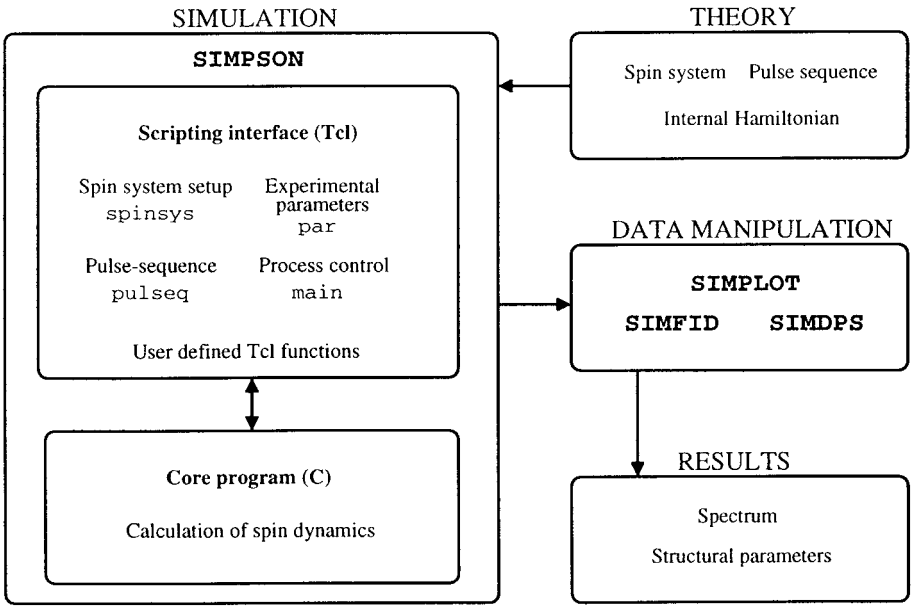

FIG. 2. Flow diagram defining the SIMPSON simulation environment.

tions) and (ii) is an interpreted language as opposed to a compiled language. The latter feature is convenient, for example, in the process of implementing and modifying pulse sequences and experimental conditions for the experiment to be simulated. This flexibility is achieved essentially without cost as the input-file-interpreting overhead amounts to at maximum a few percent of overall computation time. This is ascribed to the fact that the vast majority of the calculations, especially the time-consuming matrix manipulations, are performed by efficient routines implemented in the C language running at native speed. Tcl is an advanced scripting language that implements all standard flow control structures (e.g., `for`, `foreach`, and `if`) and data structures (e.g., lists, normal arrays, and associative arrays) and contains a large set of library routines (e.g., for string manipulation, file handling, and regular expressions). Furthermore, the well-documented behavior and proved correctness of the language implementation give Tcl an advantage over custom-made interpreters. Obviously, these features are important for the present version of the simulation program, but even more so for future versions in the sense that they offer straightforward capability to expand the functionality by writing separate commands within the scripting language. Indeed, this is how several of the commands available in the present version of the program were implemented. If a simulation requires a more specialized feature, an extension to the core program may be necessary. In this case the elements of basic functionality are isolated, implemented in the core program and the associated commands used in the input file to describe and control this specific element of the simulation. This ensures general usability of the functions and minimizes the tendency to collect a lot of functionality in incomprehensible "black boxes." The modular construction of the core program renders it relatively easy to create such user-accessible commands.

The SIMPSON Tcl Input File

All high-level NMR operations required for numerical simulation of a particular solid-state NMR experiment are implemented via one of the four sections of the Tcl scripting interface (i.e., user input file) outlined in Fig. 2. The Hamiltonian as well as the external manipulations/conditions is defined and controlled using a number of general Tcl commands and parameters applicable for the `spinsys`, `par`, and `pulseq` sections of the input file. The most typical commands for these sections are listed in Table 2 along with a description of their function and control parameters.

The spinsys section. In the `spinsys` section, the spin system is defined in terms of the various nuclear spin species in play and the interactions associated with these. The RF channels of the experiment and the nuclei relevant for the spin system are defined via the `channels` and `nuclei` declarations, respectively, using the notation `13C`, `15N`, etc., for the arguments. We note that the `channels` definition, although intuitively relating more directly to the `pulseq` section, is included in `spinsys` to ensure direct relation to the nuclear spin species and unambiguous definition of the number and in particular the assignment of the RF channels. Furthermore, this prevents the pulse sequence from being tied to specific nuclei. The various nuclear spin interactions (`shift`, `dipole`, `jcoupling`, and `quadrupole`) are defined using a notation relating directly to the internal Hamiltonians in Eqs. [7]–[12] with all coefficients in frequency units (hertz) or ppm and all angles in degrees. We should note that `quadrupole`

TABLE 2
Elements of and Scripting Commands for the SIMPSON Input File[a]

Elements of the SIMPSON input file

`spinsys {. . .}`	Spin system and interactions.
`par {. . .}`	Global experiment parameters.
`proc pulseq {} {. . .}`	Pulse sequence.
`proc main {} {. . .}`	Processing control.

Declarations for the `spinsys` section

`channels` N_1 N_2 ... N_n
`nuclei` N_1 N_2 ... N_n
`shift` i $\delta^i_{\mathrm{iso}}|\omega^i_0/2\pi|^b$ $\delta^i_{\mathrm{aniso}}|\omega^i_0/2\pi|^b$ η^{CS} α_{PC} β_{PC} γ_{PC}
`dipole` i j $b_{ij}/2\pi$ α_{PC} β_{PC} γ_{PC}
`jcoupling` i j J^{ij}_{iso} J^{ij}_{aniso} η^J α_{PC} β_{PC} γ_{PC}
`quadrupole` i *order*[c] $C_{\mathrm{Q}}/2\pi$ η^{Q} α_{PC} β_{PC} γ_{PC}

Parameters/commands for the `par` section

`spin_rate`	Sample spinning frequency, $\omega_r/2\pi$
`np`	Number of sampling points.
`ni`	Number of sampling points in the indirect dimension of 2D experiments.
`sw`	Spectral width.
`sw1`	Spectral width for the indirect dimension.
`crystal_file`	Name of the powder averaging file containing the number N of orientations and α^k_{CR}, β^k_{CR}, ω_k values on successive lines.
`gamma_angles`	Number M of γ_{CR} angles. Set to 1 if `spin_rate` is zero. Defines the number of sampling points per rotor period when `method` equals `gcompute`.
`gamma_zero`	Constant value added to all γ_{CR} values in the powder averaging. Specifies the γ_{CR} angle if only a single crystallite is used.
`rotor_angle`	Angle between the rotor axis and the B_0 direction. If `spin_rate` is zero the default angle is zero, otherwise the magic angle.
`method`	Chooses among `direct`, `gammarep`, and `gcompute` methods for the simulation.
`start_operator`	$\rho(0)$ defined as an expression using operators $I_i\alpha$ where α is x, y, z, p (+) or m (−), and i is the nucleus number or n to denote the sum over all nuclei.[d]
`detect_operator`	Detection operator Q_{det}.[d]
`pulse_sequence`	Sets another name than `pulseq` for the pulse sequence.
`proton_frequency`	Absolute ^{1}H Larmor frequency $|\omega^{\mathrm{H}}_0/2\pi|$ in hertz. Used for ppm to hertz convertion and for the second-order quadrupolar coupling.
`verbose`	A row of flags that sets the level of information printed when running the simulation.
`variable` *name*	Sets a user specific variable with a value that can be retrieved throughout the input file by declaring: `$par(`*name*`)`.

Commands for the `pulseq` section.

`pulse` δt $|\omega^1_{\mathrm{RF}}/2\pi|$ ϕ_1 $|\omega^2_{\mathrm{RF}}/2\pi|$ ϕ_2 ...
Extends the current propagator to include a pulse of duration δt, RF-field amplitude of $\omega^i_{\mathrm{RF}}/2\pi$, and phase ϕ_i on the channels numbered successively. Alternatively the phase can be specified as x, y, $-x$, or $-y$ corresponding to phases of 0, 90, 180, or 270 degrees, respectively.

`pulseid` δt $|\omega^1_{\mathrm{RF}}/2\pi|$ ϕ_1 $|\omega^2_{\mathrm{RF}}/2\pi|$ ϕ_2 ...
Same as `pulse` but performs an ideal (i.e., infinitely strong and infinitely short) pulse. δt and $\omega^i_{\mathrm{RF}}/2\pi$ have no physical meaning other than to specify the flip-angle of the pulse. The internal time remains unchanged.

`delay` δt
Extends the current propagator to include a free precession period of duration δt. If the Hamiltonian is diagonal (i.e., no homonuclear spin–spin couplings) the delay is calculated by analytical integration of the Hamiltonian.

`offset` $\omega^1_{\mathrm{off}}/2\pi$ $\omega^2_{\mathrm{off}}/2\pi$ ...
Invokes an offset of $\omega^i_{\mathrm{off}}/2\pi$ Hz to the channels numbered successively. The offset for a channel is defined by the Hamiltonian $\omega^i_{\mathrm{off}} \sum_j I_{jz}$, where j is summed over the nuclei in the spin-system affected by pulses on the channel. The offset applies until reset using offset with zero-value arguments or the pulse sequence is called again.

`acq` [*n prop*] [ϕ]
Propagates $\rho(t)$ using the current propagator, collects a data point corresponding to Q_{det}, and resets the propagator to unity. The optional arguments n and *prop* specifies the number of data points n to collect while evolving with propagator number *prop*. The optional argument ϕ specifies the receiver phase (syntax as for `pulse`).

`maxdt` Δt
Maximum time step (Δt in Eq. [4]) over which the Hamiltonian may be considered time independent. The computation time/accuracy of the simulation is significantly affected by the choice of value for this parameter. Defaults to 1 μs in case of sample spinning and infinity in the static case.

TABLE 2—*Continued*

`store` *n*
Stores the current propagator in memory slot number *n*. The current propagator is not reset.

`reset` [δt]
Resets $\rho(t)$ to the initial operator $\rho(0)$ and the current propagator to the unity operator. Resets the current time and adds δt μs if specified.

`prop` *n* [*times*]
Propagates $\rho(t)$ with the propagator saved in memory slot number *n*. Repeated multiple times if *times* is specified, skipped if *times* is zero, or propagated once if *times* is omitted.

`filter` *n*
Propagates $\rho(t)$ with the current propagator, after which the propagator is reset and elements in the density matrix are set to zero if the corresponding element in matrix number *n* (defined using `matrix set`) is zero.

`select` *n* ...
Renders the next pulse (`pulse` or `pulseid`) selective toward the spins which numbers are given as argument.

`turnoff` *int* ...
Disables the effect of the specified interactions until the end of the pulse sequence or until they are reactivated by `turnon`. Interactions are named *int_n* or *int_n_m* using names and numbers given in the `spinsys` section, or `all` if all interactions should be disabled.

`turnon` *int* ...
Enables the specified interactions or all interactions if `all` is specified.

int `getinteractions`
Returns a list of lists each containing an interaction name (see `turnoff`) and 1 or 0 depending on whether it is enabled or disabled.

`putmatrix` *matrix* ?*format*?
Prints out a matrix returned by `matrix get` optionally in a format different from the standard format "`%9.3g`".

`matrix set` *to from*
Sets a matrix *to* to the contents of a matrix created using the argument *from* described below. *to* can either be an index in the internal array of matrices, the start (`start`) or the detect (`detect`) operator.

matrix `matrix get` *from*
Returns a matrix (printed with `putmatrix`) based on the argument *from* which can be either the same as *to* described above or the Hamiltonian (`hamiltonian`), current propagator (`propagator`), current density operator (`density`), an operator expression (`operator` *expr*) or for the purpose of filtering (undesired elements set to zero) the specific total coherence orders (`totalcoherence {. . .}` with the list containing coherences), coherence orders (`coherence {{. . .} . . .}` with each sublist containing coherence orders for each nuclei), the full matrix (`list {row row . . .}` where each row is a list of elements being either *re* or {*re im*}), specific matrix elements (`elements` {{*i j*} ...}), or all elements excluding specific matrix elements (`notelements` {{*i j*} ...}).

Commands for the `main` section

d `fsimpson` [{{*int_n_nam v*} ...}]
Starts a simulation and returns a data set `d`, optionally overriding specific values of the interactions given in the `spinsys` section. A value is named *int_n_nam* or *int_n_m_nam*, where *int* is the interaction name from `spinsys`, *n* and/or *m* the numbers of the involved nuclei, and *nam* is `iso`, `aniso`, `eta`, `alpha`, `beta`, or `gamma`.

`fsave` *d file* [*-format* `-binary -double`]
Saves the (possibly 2D) data set *d* to a file *file* using the SIMPSON data format in text, binary (`-binary`) single or double (`-double`) precision format, or optionally in another format (*-format*) being (1) `-xreim` with rows of frequency/time, real and imaginary part of data, (2) `-xyreim` (2D data) with rows of frequency/time (indirect dimension), frequency/time (direct dimension), real and imaginary part of data, with an empty line separating succeeding fids, or (3) `-gnu2d -binary` in the binary 2D Gnuplot format (*56*).

`fft` *d* [`-inv`]
`fft` *d rp lp rp1 lp1* [`-phsens`]
The first form performs a direct or inverse fast Fourier transformation of the data set *d*, while the second form performs a 2D transformation using constant (*rp*) and linear (*lp*) phase correction in the direct and indirect (*rp1* and *lp1*) dimensions. The optional argument `-phsens` assumes phase-sensitive 2D data with succeeding pairs of fids corresponding to equal t_1 and 90° different phase.

`fzerofill` *d npz* [*niz*]
Zerofills the data set *d* up to a total of *npz* points, optionally zerofills the 2D data set up to a total of *niz* points in the indirect dimension.

`fphase` *d* [`-rp` *v* `-lp` *v* `-scale` *v* `-offset` *v*]
Performs one or more of first- and second-order phasing vertical scaling, and offset on the data set *d*.

`faddlb` *d lb r* [*lb1 r1* [`-phsens`]]
Apodizes the data set *d* with a Gaussian/Lorentzian (ratio *r*) weighting function causing an extra linebroadening of *lb* Hz, or optionally the 2D data set with *lb1* and *r1* specifying the values for the indirect dimension. The optional argument `-phsens` assumes phase-sensitive 2D data.

`fbc` *d order* {{*from to*} ...} [*skip*]
Baseline corrects the data set *d* by fitting every `skip` (default 1) data point of the baseline in the defined frequency ranges to a polynomial of order *order*.

`fnewnp` *d points*
Changes the number of data points in the data set *d*. Intermediate points are interpolated using a cubic spline.

`fsmooth` *d points order*
Smoothing of the data set *d* to a given order *order*.

TABLE 2—*Continued*

peaks `ffindpeaks` *d th sens* [*from to*]
Finds all peaks in the data set *d* that are higher than *th* and spans at least *sens* data points (optionally restricted to searching inside a specific frequency range) and returns a list of frequencies and peak heights.

areas `fint` *d* {{*from1 to1*} {*from2 to2*}...}
Returns a list of integrated intensities (by summation) for the specified spectral regions in the data set *d*.

areas `fssbint` *d dny shift width*
Returns a list of integrated intensities (by summation) for equidistant spectral regions (separated by *dny* Hz, centered around *shift* Hz, and each having a width of *width*) in the data set *d*.

d `fdup` *s*
Copies the data set *s* to a new dataset *d*.

`fcopy` *d s*
Copies the data set *s* into an existing data set *d*.

`fadd` *d s*
Adds the data sets *s* and *d* and saves the result in *d*.

`fsub` *d s*
Subtracts the data set *s* from *d* and saves the result in *d*.

`frev` *d*
Reverses the order of all data points in the data set *d*.

rms `frms` *d1 d2* [`-re` | `-im`] [{{*from to*}...}]
Returns the normalized root-means-square deviation between the complex, real, or imaginary part of two data sets *d1* and *d2*, optionally within specific frequency ranges.

`fextract` *d from to*
Shrinks the data set *d* to the specified frequency range.

d `fzero` [{{*from to*} ...}]
Attributes zero intensity to frequency regions (or the full region) of the data set *d*.

areas `faddpeaks` *d cutoff* {{*frq int lb r*} ... }
Adds a series of peaks to the data set *d*, each of which is specified by a frequency, intensity, additional linebroadening, and Gauss/Lorentz ratio. The areas of the peaks are returned as a list. The *cutoff* parameter defines the minimum intensity calculated before it is truncated to zero.

`fexpr` *d reexpr imexpr*
Applies Tcl expressions to the real and imaginary part of each data point in the data set *d*. Valid variables are the real part of the complex data point `$re`, the imaginary part `$im`, and the point index `$i` starting from one. These variables must be preceded with a backslash if a local variable (`fac`) is used, e.g., `fexpr $f [list \$re*$fac] {$im+$i*1.23}`.

v `findex` *d i* [`-re` | `-im`]
Returns the real and/or the imaginary part of the *i*th complex data point in a data set *d*.

`fsetindex` *d i re im*
Sets the real and imaginary part of the *i*th complex data point in the data set *d*.

v `fx` *d i*
Returns the frequency or time of a data point *i* depending on the type of the data set *d*.

d `fload` *file*
Loads the data set *d* from a *file* and returns a data descriptor.

`funload` [*d*]
Removes all or a specific data set *d* from the memory.

d `fcreate -np` *v* `-sw` *v* [`-ref` *v* `-ni` *v* `-sw1` *v* `-ref1` *v* `-type` *v*]
Creates and returns a descriptor to the new data set *d* with zero points and with specifications corresponding the arguments of which `-np` and `-sw` are required and *v* for `-type` is either `fid` or `spe`.

v `fget` *d* [`-ref` | `-ref1` | `-sw` | `-sw1` | `-np` | `-ni` | `-type`]
Returns either the reference line, spectral width, number of complex data points, or type of data (`fid` or `spe`) from the data set *d* depending on the argument.

`fset` *d* [`-ref` *v* `-ref1` *v* `-sw` *v* `-sw1` *v* `-type` *v*]
Changes the specifications for the data set *d* following the syntax from `fcreate`.

`flt` *array*
Performs iterative fitting using parameters given in the array *array* as described in the text.

`fplot2d` *d name* (`-ppm` | `-ps`) [*scale*]
Creates a Postscript (`-ps`) or portable pixmap (`-ppm`) bitmap plot of a 2D data set *d* using conventional 2D plotting conventions, i.e., shift increasing left and down, optionally vertically scaled with *scale*.

r_{12} `dip2dist` N_1 N_2 $b_{12}/2\pi$
Calculates the distance (in Å) between the nuclei N_1 and N_2 based on the dipolar coupling constant $b_{12}/2\pi$.

$b_{12}/2\pi$ `dist2dip` N_1 N_2 r_{12}
Analogues to `dip2dist` but calculates the dipolar coupling constant in hertz from the distance r_{12} (in Å).

list `csapar` δ_{11} δ_{22} δ_{33}
Returns the isotropic shift, chemical shift anisotropy, and the asymmetry parameter assuming unordered principal elements (in hertz or ppm) as arguments.

TABLE 2—*Continued*

list `csaprinc` δ_{iso} δ_{aniso} η
Returns the ordered principal elements δ_{xx}, δ_{yy}, and δ_{zz} given the isotropic shift, chemical shift anisotropy, and the asymmetry parameter (in hertz or ppm) as arguments.

list `isotopes`
Returns a list with data for the spin isotopes available.

value `gamma` *N*
Returns the magnetogyric ratio of a nucleus *N* given in the unit 10^7rad/(*Ts*).

value `resfreq` *N* [$|\omega_0/2\pi|$]
Returns the value of the absolute resonance frequency in hertz for at nucleus *N* assuming an absolute proton resonance frequency of 10^6 Hz (default) or optionally $|\omega_0/2\pi|$ Hz.

Tcl language constructs

name arg arg ...	Function call with arguments.
`set` *var value*	Sets a variable to a value.
array(*var*)	A variable in an associative array.
{ ... }	Begin and end of a command block.
\	Continuation of a line.
$*var*	Gets a value of a variable.
[`expr` *expression*]	Evaluates a mathematical expression.
`proc` *name* {*args*} {*body*}	Definition of a user procedure.
`global` *var var ...*	Make variables visible outside the current block.
`for` {*start*} {*test*} {*incr*} {*body*}	A `for` loop.
`if` {*test*} {*body*} `elseif` {*test*} {*body*} `else` {*test*} {*body*}	An `if` construct.
[`list` e_1 e_2 ...]	Creates a list with elements e_i.
[`lindex` $*list* *i*]	Returns element *i* from a list (counting from zero).

[a] All values are in hertz, microseconds, and degrees if not specified otherwise. Arguments in square brackets are optional.
[b] The parameter can be given in ppm (δ value) by appending a `p` to the value.
[c] The *order* parameter for the quadrupolar interaction can be 1 or 2 corresponding to first and second order, respectively, according to Eq. [10].
[d] For half-integer quadrupolar nuclei I*ic* may be applied to excite and detect only the {1/2, −1/2} central transition.

includes the quadrupolar coupling Hamiltonian up to order one and two as specified by the argument *order*.

The par section. In the `par` section, the spinning (`spin_rate`) and sampling (`np`, `ni`, `sw`, `sw1`) conditions are defined along with conditions for the powder angles/averaging (`crystal_file`, `gamma_angles`, `rotor_angle`, `method`), the initial and final operators (`start_operator`, `detect_operator`), the pulse sequence (`pulse_sequence`), the ^{1}H Larmor frequency (`proton_frequency`, relevant for second-order quadrupole coupling and shifts expressed in ppm), the information flow from the program (`verbose`), and variables (`variable`) to describe, e.g., the applied RF field strength and rotor synchronization conditions. It should be noted that parameters in the `par` section are set independently of the pulse sequence. This facilitates comparison of the performance for different pulse sequences and supports the creation of pulse sequence libraries.

The pulseq section. A large number of commands are available for the `pulseq` section to provide flexibility to simulate essentially all types of solid-state NMR experiments. In addition to commands such as `pulse`, `pulseid`, `delay`, `offset`, and `acq` describing finite RF pulses, ideal RF pulses, free precession periods, carrier frequency offsets, and acqusition of data, this section may contain a number of commands that have no direct counterpart on the spectrometer but serve to optimize the simulations by reusing propagators, emulating phase cycles, and simulating the effect of coherence-order filtering pulse sequence elements. These include the `maxdt` to adjust the integration intervals, the `store` command for saving propagators, `reset` for resetting, `prop` for applying a previously saved propagator, the `matrix set` and `filter` commands for coherence-order filtration, `select` for restriction of the subsequent pulses to certain spins, and the `turnon` and `turnoff` commands to activate and deactivate parts of the Hamiltonian, respectively. In addition to this come several commands to create and retrieve information about matrices and interactions throughout the calculations. These and similar commands are described in more detail in Table 2. To offer the highest degree of flexibility, it is relevant to mention that all commands may be entered chronologically as they appear in the pulse sequence or may be controlled by loops to allow for efficient implementation of repeating events or scanning through various parameters. This is conveniently accomplished using standard Tcl constructs among which the most relevant are included at the bottom of Table 2. For a more complete description we refer to textbooks on the Tcl language (*41*, *42*).

The main section. With the internal Hamiltonian and the external manipulations defined, the remaining part of the simulation concerns the experiment processing to conduct the calculations and obtain the result in terms of 1D free-induction decays (FIDs) or spectra for single or phase-cycled pulse sequences, parameter scans, coherence-transfer efficiency curves, simultaneous multiple simulations, 2D FIDs or spectra, etc. This is accomplished in the `main` section of the input file. For example, the simulation is started using the command `fsimpson` which returns the data set resulting from the SIMPSON calculation. The data set may be saved using the command `fsave` typically being combined with `fsimpson` as

```
fsave [fsimpson] $par(name).fid
```
[21]

where `$par(name)` per default contains the name of the input file. More illustrative examples of this type are given in the next section. In addition to control of the pulse sequence processing, it is desirable to have built-in options for data processing, e.g., fast Fourier transformation (`fft`), zero-filling (`fzerofill`), phasing and scaling (`fphase`), apodization (`faddlb`), baseline correction (`fbc`), inter/extrapolation (`fnewnp`), smoothing (`fsmooth`), peakfinding (`ffindpeaks`), and integration of spectral regions (`fint`) or sideband patterns (`fssbint`), as well as the ability to duplicate (`fdup, fcopy`), add (`fadd`), subtract (`fsub`), reverse (`frev`), evaluate the root-mean-square deviation between two data set (`frms`), extract regions of a spectrum to a new dataset (`fextract`), zero regions of a spectrum (`fzero`), add peaks to a spectrum (`faddpeaks`), or otherwise manipulate (`fexpr`) the output of one or more simulations. All of this and several other things may be accomplished in the `main` section of the Tcl input file using a variety of commands with the most typical listed in Table 2 along with a short specification of their arguments. For a more complete description of the many available commands, the reader is referred to the examples in the next sections and Ref. (*55*). We note that most of the data manipulation alternatively may be performed after the simulation using some of the supplementary tools described below.

The SIMPSON Data Format and Data Exchange

Before proceeding to procedures for postprocessing of data, it appears relevant to address the data format used in the SIMPSON package (including the productivity tools described below). For example, this is relevant for the import of experimental spectra for simulation and iterative fitting using SIMPSON or for export of FIDs or spectra (one- and higher dimensional) to other software packages for postprocessing or plotting.

The SIMPSON data format has the structure

```
SIMP
NP=n
SW=sw
REF=ref
[ NI=ni ]
[ SW1=sw1 ]
[ REF1=ref1 ]
[ FORMAT=format ]
[ PREC=prec ]
TYPE=type
DATA
re1 im1
re2 im2
. . .
ren imn
END
```

where *n*, *sw*, and *ref* represent the number of complex data points, the spectral width ($sw/2$ is the Nyquist frequency for the sampling), and a reference frequency in the directly sampled dimension. For 2D spectra the optional parameters *ni*, *sw*1, and *ref*1 (default values 0) take finite values describing the number of points, the spectral width, and a reference frequency for the indirect dimension. In general, multidimensional data sets are constructed by concatenating a series of 1D data sets successively after each other in the data file. `FORMAT` is an optional parameter specifying the data format to normal ASCII *TEXT* (default) or *BINARY* format among which the latter is convenient for large 2D data sets. Likewise, `PREC` is an optional specification of the precision of the data being either *SINGLE* (default) or *DOUBLE* for single or double precision floating point representation, respectively. The binary formatted numbers are located between the `DATA` and `END` key words. Finally, `TYPE` specifies whether the data is represented in the time (*FID*) or the frequency (*SPE*) domain.

The output obtained by `acq` is typically given in the time domain and the spectra are obtained through Fourier transformation. Using this notation, the complex intensity for the *i*th time-domain data point corresponds to the time

$$t_i = \frac{i-1}{sw}, \quad i = \{1, 2, \ldots, N\}, \tag{22}$$

while the similar data point in a spectrum corresponds to a frequency of

$$frq_i = sw\left(\frac{i-1}{N} - 0.5\right) + ref. \tag{23}$$

For a given frequency the corresponding index of a data point in a spectrum is found by

$$i = \mathrm{Floor}\left[N\left(\frac{frq_i - ref}{sw} + 0.5 \right) + 1.5 \right], \qquad [24]$$

where Floor is a function that rounds its argument down to the nearest integer value.

To maintain consistency with common practice on commercial NMR spectrometers the chemical shift or deshielding (δ) convention is employed consistently all way from the theory via the SIMPSON input files to the data or spectra resulting from the simulation. In the spectra this implies an axis with the chemical shifts increasing from right to left in opposite direction to the chemical shielding. In this representation the least shielded and thereby most deshielded tensor element δ_{11} is located to the left in the spectrum using the ordering $\delta_{11} \geq \delta_{22} \geq \delta_{33}$. Similarly following common practice, the same direction applies to the frequency scale obtained upon multiplication of the ppm value shifts by the *absolute value* of the Larmor frequency (i.e., $|\omega_0/2\pi|$). This choice (instead of the more correct scaling by $\omega_0/2\pi$) facilitates comparison of experimental and simulated spectra although it inevitably causes an unfortunate confusion with respect to the signs of nuclear spin interactions and their spectral representation as discussed in detail by Levitt (*44*).

To avoid unnecessary contributions to this confusion and maintain clarity of the inner working of SIMPSON in this respect, the following conventions apply: (i) Both δ and frequency scales increases from right to left, (ii) chemical shift parameters should be entered as ppm values or as frequencies obtained by multiplication of these by $|\omega_0/2\pi|$, (iii) dipolar and *J* couplings as well as frequency offsets should be entered with correct sign under consideration of potential influence for γ's, (iv) upon knowledge of the absolute proton (i.e., spectrometer) frequency and the nuclei in play SIMPSON produces the correctly signed Hamiltonian (according to Eqs. [5]–[10]), and (v) SIMPSON per default complex conjugates the acquired data when $\gamma > 0$ for the detect nucleus to obtain correct representation on the chosen frequency/ppm scale. To demonstrate the consequences of this default procedure on the appearance of the simulated spectra, Fig. 3 contains representative static powder spectra influenced by anisotropic chemical shift and second-order quadrupolar coupling. It is emphasized at this point that the automatic conjugation, which is practical to avoid confusing reversion of the spectra for $\gamma < 0$ nuclei but may cause phase confusion for parameters scans, may be overruled using the parameter `conjugate_fid` in the `par` section of the input file. Furthermore, we note that any kind of axis or data-ordering reversal alternatively may be invoked using the `frev` command prior to plotting with SIMPLOT or using the `Reverse axis` and `Reverse data` options in SIMPLOT (*vide infra*). For example, this may be used to reproduce spectra on the chemical shielding (σ) scale, which apart from an appropriate reference point is related to the deshielding scale by a sign reversal, i.e., $\delta_{\mathrm{iso}} = \sigma_{ref} - \sigma_{\mathrm{iso}}$,

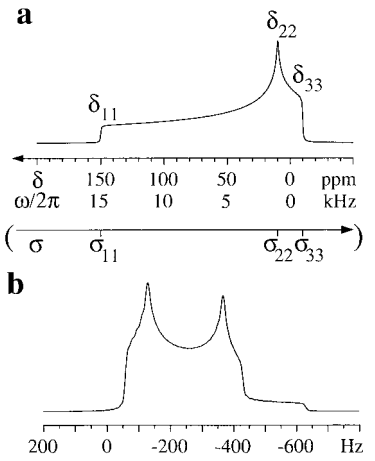

FIG. 3. Typical static-powder solid-state NMR spectra ($|\omega_0/2\pi| = 100$ MHz) for (a) anisotropic chemical shift using $\delta_{\mathrm{iso}} = 50$ ppm ($\delta_{\mathrm{iso}}|\omega_0/2\pi| = 5$ kHz), $\delta_{\mathrm{aniso}} = 100$ ppm ($\delta_{\mathrm{aniso}}|\omega_0/2\pi| = 10$ kHz), and $\eta_{\mathrm{CS}} = 0.2$ corresponding to chemical shift principal elements $\delta_{11} = 150$ ppm, $\delta_{22} = 10$ ppm, and $\delta_{33} = -10$ ppm, (b) and second-order quadrupolar coupling characterized by $I = 3/2$, $C_{\mathrm{Q}} = 1.0$ MHz, $\eta_{\mathrm{Q}} = 0.2$. We note that using the conventions described in the text, the orientation of the spectra (in contrast to the signs of the chemical shift and second-order quadrupolar coupling terms in the Hamiltonian) remain independent on the sign of the gyromagnetic ratio γ_i.

where σ_{ref} is a reference value and $\sigma_{\mathrm{iso}} = \frac{1}{3}(\sigma_{11} + \sigma_{22} + \sigma_{33})$. This reversal leads to a spectrum with the chemical shielding increasing from left to right, i.e., with the most shielded tensor element σ_{33} to the right using the conventional ordering $\sigma_{11} \leq \sigma_{22} \leq \sigma_{33}$.

The SIMPLOT, SIMFID, and SIMDPS Productivity Tools

In order to form a self-standing simulation environment, the SIMPSON simulation package contains a collection of productivity tools SIMPLOT, SIMFID, and SIMDPS, as indicated in the flow diagram in Fig. 2. SIMPLOT is a graphical viewer for display and manipulation of one or multiple 1D spectra, acquisition data, or output from parameter scans. This viewer allows for interactive (mouse controlled) data manipulations such as zooming, phasing, scaling, and postscript plotting. We note that the present version of the SIMPSON package does not include an interactive viewer for display/manipulation of two- or higher dimensional data. However, using an optional program package, VnmrTools, provided along with the SIMPSON program, 2D data can be retrieved directly into the Varian VNMR software.[3] Alternatively, multidimensional data can be accessed and converted to other formats (e.g., for GNUPLOT (*56*)) using the `fsave` and `findex` commands. SIMFID is a program which gives access to most of the SIMPSON `main` section data manipulation commands through arguments on the command line. This tool is useful for postprocessing of data resulting from a SIMPSON simulation. SIMDPS is a pulse

[3] The present version contains only conversion tools for the VNMR data format.

sequence viewer allowing graphical visualization of the pulse sequence implemented via the scripting language. This tool proves convenient for testing of pulse sequence timings, phases, and amplitudes. The various tools will be described in more detail in the following sections addressing specific simulation examples.

Availability and Portability of the SIMPSON Package

Finally, it is relevant to address the availability and portability of the SIMPSON simulation package in terms of computer hardware and distribution. The SIMPSON, SIMPLOT, SIMFID, and SIMDPS programs are released on the Internet as open source software (*55*) under the terms of the GNU General Public License (*57*). Among other things, this implies that any user can freely modify the source code as long the full code is made available under the GNU General Public License. For example, this enables the user to create extensions, make ports to new platforms, find information about the conventions and algorithms used, and correct potential errors. The overall aim is to make the program owned and maintained by the users.

Furthermore, precompiled and self-contained (i.e., no dependencies on special external libraries) binary executables are freely available for the most common operating systems (including Linux/i386, Windows/i386, and the major Unix platforms) and are easy to compile on other platforms due to the portability of the C language and the Tcl language interpreter (open source software). We note that the SIMPLOT program uses the open source GTK widget set (*58*) and is currently available only for Linux and Windows.

ELEMENTS OF THE INPUT FILE: A TUTORIAL EXAMPLE

It appears from the previous section that SIMPSON is based on a relatively large number of commands required to offer the desired compromise among ease of use, transparency, and flexibility to simulate all types of NMR experiments. In order to clarify the use of these commands and to systematically illustrate the simple construction of the Tcl input file, this section demonstrates and explains the input file for a typical solid-state NMR experiment. To extend the perspective beyond the specific example, the discussion additionally addresses alternative typical options to the various commands. For the present purpose, we have chosen the rotational-echo double resonance (REDOR) pulse sequence (*18*) shown in Fig. 4a, which on one hand is a very important solid-state NMR experiment and on the other hand contains many typical pulse sequence elements without being excessively complicated. To maintain appropriate reference to the literature and to the known behavior of the experiment, we reconstruct the ^{13}C-detected REDOR experiment as originally presented by Gullion and Schaefer (*18*)

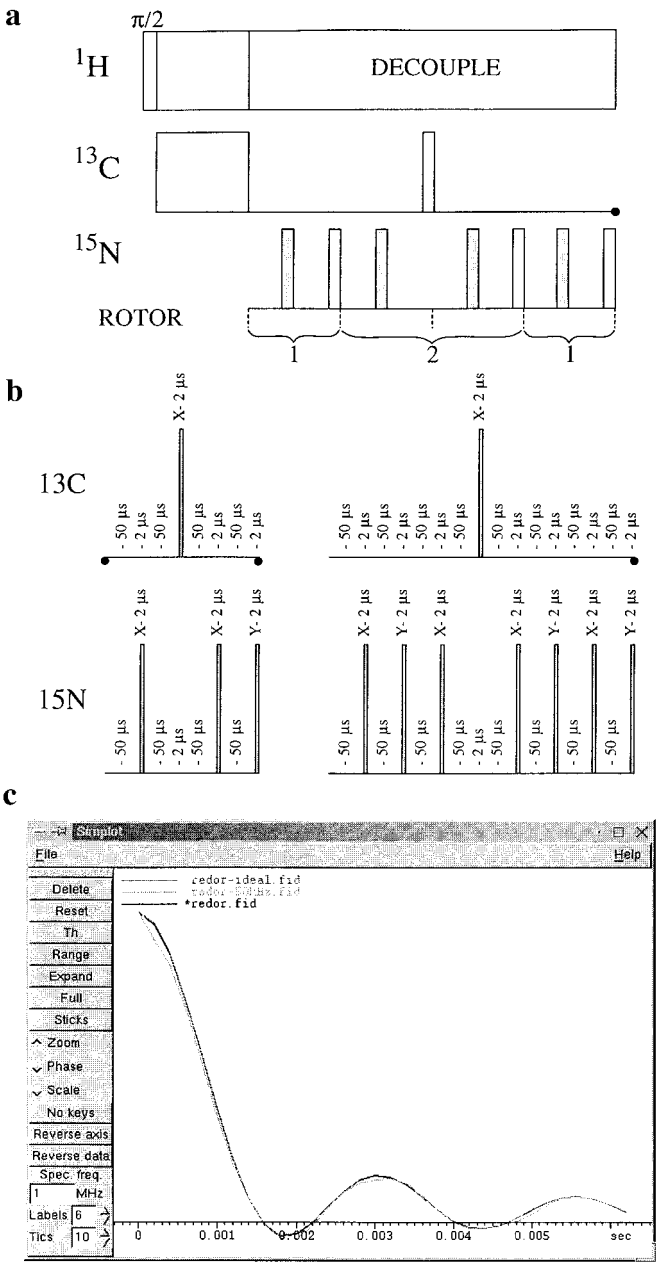

FIG. 4. (a) Timing scheme for the ^{13}C–^{15}N REDOR pulse sequence as typically implemented on the spectrometer. Shaded and open rectangles on the ^{13}C and ^{15}N channels denote π pulses of phase x and y, respectively. (b) Pulse sequences corresponding to the three first sampling points (black dots) as visualized using SIMDPS with the REDOR input file given in the text with `np` changed to 3. We note that all simulations ignore the ^{1}H channel under the assumption of ideal cross polarization and perfect ^{1}H decoupling. (c) SIMPLOT view of dipolar dephasing curves calculated for a powder of ^{13}C–^{15}N spin pairs with $b_{CN}/2\pi = 895$ Hz ($r_{CN} = 1.51$ Å), $\delta^{C}_{iso} = 10$ ppm, $\delta^{C}_{aniso} = 100$ ppm, and $\eta^{C}_{CS} = 0.5$ using REDOR with $\omega_r/2\pi = 10$ kHz and ideal RF pulses (upper curve), as well as finite RF pulse irradiation with amplitudes of $\omega_{RF}/2\pi = 150$ kHz (middle curve) and 50 kHz (lower curve) on both RF channels.

for measurement of ^{13}C–^{15}N dipolar couplings (and thereby internuclear distances) under MAS conditions. In REDOR, coherent averaging of the dipolar coupling interaction by

MAS is interrupted by inserting a π-pulse on the ^{15}N RF channel for every half rotor period with the exception that the pulse exactly in the middle of the evolution period is replaced by a corresponding π-pulse on the ^{13}C RF channel in order to refocus undesired effects from ^{13}C chemicalshielding. The difference between this experiment and a corresponding experiment not using the ^{15}N refocusing pulses provides a direct measure for the dipolar coupling.

The `redor.in` *Input File*

With this primer, the first step to a SIMPSON simulation is to implement the spectrometer RF channels, the spin system, and the NMR interactions in the `spinsys` section of the input file. This amounts to

```
spinsys {
  channels 13C 15N
  nuclei   13C 15N
  dipole   1 2 895 10 20 30
  shift    1 10p 100p 0.5 50 20 10
}
```

where we for simplicity have disregarded the ^{1}H to ^{13}C cross-polarization sequence which for sensitivity reasons is part of the experimental pulse sequence in Fig. 4a. The `channels` command establishes the ^{13}C and ^{15}N RF channels, while the `nuclei` command line defines the ^{13}C–^{15}N two-spin system. The affected nuclei are ordered according to their appearance on the `nuclei` line. We note that a ^{1}H channel and one or more ^{1}H nuclei (including associated interactions) may easily be implemented in the `spinsys` section to allow for simulation of the effect of cross polarization. The relevant nuclear spin interactions are specified using the `dipole` and `shift` command lines with the arguments referring to the internal Hamiltonians in Eqs. [7]–[9] according to Table 2. This implies that the dipolar coupling should be entered under appropriate consideration of the signs of the gyromagnetic ratios in play (e.g., the dipolar coupling between spins with positive (^{13}C) and negative (^{15}N) gyromagnetic ratios should be positive). In the present example the isotropic chemical shift and the chemical shift anisotropy are entered in ppm at the δ scale by appending the character `p` immediately after the value. Using this information along with knowledge as to γ (via `nuclei`) and the `proton_frequency` (entered in the `par` section (*vide infra*)) SIMPSON automatically calculates the correct chemical shift frequencies for the Hamiltonian. We note that the shift parameters alternatively may be entered as hertz values generated by scaling of the ppm values by the absolute value of the relevant Larmor frequency. In this context it is relevant to note that SIMPSON provides a number of simple tools to convert between, e.g., internuclear distances and dipolar couplings, principal shielding elements and isotropic/anisotropic/asymmetry parameters, lists of available isotopes. These commands, being helpful in setting up the `spinsys` section of the input file, may be invoked directly by writing a simple SIMPSON input file containing exclusively the `main` section with one or more of the lines

```
proc main {} {
  puts [dip2dist 15N 13C 970]
  puts [dist2dip 15N 13C 1.5]
  puts [csapar 30 60 200]
  puts [csaprinc 50 100 0.2]
  puts [join [isotopes] \n]
}
```

where distances are in Angstroms (Å), dipolar couplings are in hertz, and chemical shift principal elements are in ppm. Note that this `main` example is not part of the proposed REDOR input file.

Parameters defining general (global) physical conditions such as sample rotation, crystallite orientations, and sampling conditions are implemented in the `par` section of the input file. In the present case this may take the form

```
par {
  proton_frequency 400e6
  spin_rate        10000
  sw               spin_rate/2.0
  np               32
  crystal_file     rep320
  gamma_angles     18
  start_operator   I1x
  detect_operator  I1p
  verbose          1101
  variable rf      150000
}
```

which, using the self-explanatory names, defines experimental conditions using a 400-MHz spectrometer ($\omega_0^{\mathrm{H}}/2\pi = -400$ MHz), 10-kHz sample spinning at the magic angle (the default value for `rotor_angle` is $\tan^{-1}(\sqrt{2})$ in degrees), the spectral width set to a half the rotor frequency corresponding to sampling every second rotor period, 32 sampling points, powder averaging using 320 pairs of α_{CR}, β_{CR} crystallite angles distributed according to the REPULSION scheme (*49*), and 18 equally spaced γ_{CR} angles. Since the requirements to the number of angles in the powder average may vary significantly for different experiments (and typically need to be tested for convergence), SIMPSON contains a large number of powder files that may be straightforwardly invoked as alternatives to `rep320` (*55*). Obviously, these includes options for liquid-state, single-crystal, and uniaxially oriented molecule conditions. User-defined sets of crystallite angles can be used by setting the `crystal_file` entry to the path of a text file containing the number of angle pairs N, followed by N successive lines each containing α_{CR}^k, β_{CR}^k, and ω_k as given in Eq. [20].

Three options, controlled by the `method` command in the `par` file, can be chosen for the γ_{CR} averaging including `direct`, `gammarep`, and `gcompute` corresponding to direct calculation by chronological time integration, reuse (replication) of propagators for different γ_{CR} angles, or γ-COM-

PUTE (*51–55*). The default method, used here by omitting the `method` parameter in the `par` section, is the direct method. The present simulation assumes the initial (`start_operator`) and final (`detect_operator`) operators to be `I1x` and `I1p`, respectively. Using the specifications in `spinsys` these correspond to I_x and I_+ for ^{13}C. We note that the `start_operator` often is set to the equilibrium polarization which, when applying to all spins, may be implemented in shorthand notation as `Inz`, corresponding to $\sum_{i=1}^{n} I_{iz}$. Specifying the `detect_operator` as `I1p` ensures that the ^{13}C magnetization is sampled by quadrature detection. We should mention that in the case of second-order quadrupolar coupling and chemical shift values entered in ppm, it is necessary to specify that the absolute ^{1}H Larmor frequency (in hertz) in the `par` section using the `proton_frequency` parameter. Finally, a user-declared `variable` denoted `rf` is used to specify the absolute value of the RF field strength to 150 kHz. This (and potential other) variable entered in the `par` section may be accessed in the user procedures `pulseq` and `main` as elements in the `par` array made visible by the `global` key word. A parameter can be set using an expression containing previously defined parameters, as is the case with calculation of `sw`. For a more complete description we refer to Table 2. Finally, we should mention the `verbose` function, which is a set of bits specifying the output returned from the SIMPSON simulation. In the present case the output represents the spin system, progress during the calculation, and various information concerning the simulation (*55*).

In general, the pulse sequence is considered the most crucial part of the simulation and indeed represents the most flexible part of the simulation environment (together with `main` and other Tcl procedures). The pulse sequence is defined through the user-defined Tcl function `pulseq` being called for each crystallite orientation. For the REDOR pulse sequence with finite RF pulses this function may conveniently be written as follows.

```
proc pulseq {} {
  global par

  maxdt 1.0

  set t180 [expr 0.5e6/$par(rf)]
  set tr2 [expr 0.5e6/$par(spin_rate)-$t180]

  reset
  delay $tr2
  pulse $t180 0 x $par(rf) x
  delay $tr2
  pulse $t180 0 x $par(rf) y
  store 1

  reset
  acq
  delay $tr2
  pulse $t180 0 x $par(rf) x
  delay $tr2
  pulse $t180 $par(rf) x 0 x
  prop 1
  store 2
  acq

  for {set i 2} {$i < $par(np)} {incr i} {
    reset
    prop 1
    prop 2
    prop 1
    store 2
    acq
  }
}
```

We note that a version corresponding to an ideal RF pulse may be constructed simply by replacing `pulse` with `pulseid` and avoiding substraction of `t180` from `tr2`.

As an extremely important parameter, any pulse sequence should contain a definition of the maximum time step over which the Hamiltonian may be considered time independent. This parameter, being relevant when the Hamiltonian contains noncommuting elements, is controlled `maxdt` (microseconds) corresponding to Δt in Eq. [4]. At the same level as the number of crystallites used for powder averaging, the value of `maxdt` may act as a tradeoff between accuracy and speed of the simulation and therefore must be considered carefully by running several simulations with different values. In most cases a value of 1 μs is adequate. Setting of typical timings also belongs to the initialization, which in the present case is the durations of a half rotor period `tr2` and a 180° pulse `t180` (both microseconds) being calculated by accessing parameters from the `par` section. We note that the duration of a π pulse is subtracted from the half rotor period to ensure rotor synchronization of the refocusing periods.

In the implementation of any pulse sequence, it is relevant to consider the flow of operations and identify repeating events. The former topic concerns the inner working of the SIMPSON calculations, while the latter addresses more specifically the actual pulse sequence. Starting out with the generalities, it applies to any simulation that evaluation of the spin dynamics requires three internal variables to be changed throughout the pulse sequence: the density matrix, the time, and the propagator. Thus, to any time up to the end of the experiment, the current propagator may (under consideration of Dyson time ordering) be updated by multiplication with the propagator for the following time event and the time incremented appropriately. This is accomplished using the `pulse`, `pulseid`, and `delay` commands. To any time the current density matrix, resulting from operation with the current propagator on the initial density operator, may be obtained using the acquisition `acq` command, which additionally provides the expectation/projection value with respect to the detection operator (i.e., a data point) and resets the current propagator. The current time and the propagator are reset using the `reset` command. These

statements immediately indicate the hierarchy of operations in SIMPSON: first operations on the propagator level and later on the density matrix level. This construction allows for efficient reuse of propagators that to any time can be saved with the `store` command and reused the desired number of times using the `prop` command. For typical solid-state experiments, systematic reuse of propagators may speed up the calculations by several orders of magnitude. For nonspinning samples, the time-independent Hamiltonian allows all propagators to be reused without limitations and `maxdt` is irrelevant. In the case of sample spinning, the Hamiltonian is periodic with the rotor period, implying that propagators calculated to a specific time can be reused an integral number of rotor periods later provided that the pulse sequence fulfills the same periodicity. To allow for sufficient flexibility, the propagator to an arbitrary time within the pulse sequence can be calculated using the time increment as argument to the `reset` command. The same is possible during the data acquisition period provided that the sampling is synchronized to an integral number or integer fraction ($1/R$) of the rotor period. In the latter case R propagators starting at different times need to be precalculated. Note that the `acq` and `filter` commands cannot be stored as they modify the density matrix. The program automatically checks that the propagators are reused at the correct time.

The REDOR pulse sequence in Fig. 4a may conveniently be described in terms of two repeating pulse sequence elements. Before these are activated we reset the propagator and calculate the first point corresponding to time $t = 0$ using the `acq` command. The first pulse sequence element is represented by a rotor period with ^{15}N π pulses in the middle and at the end (marked 1 in Fig. 4a). The two pulses alternate with x and y phases, which is relevant for the spectrometer implementation since it prevents accumulation of pulse rotation errors (e.g., induced by RF inhomogeneity) throughout the train of echo pulses (*59*, *60*). In fact, the utility of this modification may easily be tested by slight misadjustment of the pulse flip angles in the SIMPSON simulation. A more advanced and practically even more relevant approach would be the XY-8 phase scheme described by Gullion *et al.* (*60*). The propagator for the first element is calculated and saved using `store 1` for later activation using `prop 1`. The second element (marked 2 in Fig. 4a) is initially formed by a rotor period with a ^{15}N π pulse in the middle and a ^{13}C π pulse in the end, followed by the `prop 1` pulse sequence element. Thus, `prop 2` corresponds to the central part of the REDOR pulse sequence while `prop 1` corresponds to the bracketing rotor periods with ^{15}N π pulses alone. Using this setup, the second data point from the REDOR experiment may be calculated using the `acq` command upon generation of the propagator `prop 1; prop 2; prop 1`. Now it is evident that all points of the REDOR experiment may be calculated systematically using sequence elements of the type `prop 1; prop 2; prop 1`, which upon calculation of a data point is stored as the new `prop 2` using `store 2`. Upon subsequent bracketing by `prop 1` propagators, we obtain the propagator relevant for calculation of the next data point and so forth. This enables simple calculation of the REDOR dephasing curve using a `for` loop construction as implemented in the `pulseq` code of the present example. We note that the reference spectrum for the REDOR experiment may straightforwardly be generated by zeroing the RF amplitude for the ^{15}N refocusing pulses in `prop 1` and `prop 2` (not shown).

This simple example illustrates two important points concerning simulations within the SIMPSON environment. First, in the setup of multiple-pulse NMR simulations it is important to disentangle the pulse sequence in repeating events to ensure a simple and short program structure as well as the fastest possible calculations in terms of CPU time. In fact, programming in this manner hardly differs from the way pulse sequences should be implemented on the spectrometer. Second, it demonstrates that SIMPSON is sufficiently flexible that essentially all programming structures using and reusing pulse sequence building blocks are feasible.

The `main` section of the input file, which controls the progress of the simulation, may take the following form.

```
proc main {} {
  global par

  set f  [fsimpson]
  fsave $f $par(name).fid
}
```

The `fsimpson` command (evaluated by the brackets) performs the simulation based on the information given in the `spinsys`, `par`, and `pulseq` sections and returns a pointer `f` to the resulting acquisition data. Using the `fsave` command the data is saved to the file `$par(name).fid` with the extension `.fid` added to the basename of the input file. The resulting time-domain signal may be plotted using the SIMPLOT program either directly or upon Fourier transformation using SIMFID. Alternatively, if desired, the Fourier transformation may be accomplished within the `main` section of the input file by appending

```
  fzerofill $f 16384
  faddlb $f 100 0
  fft $f
  fsave $f $par(name).spe -binary
```

to the `main` code. Specifically these commands invoke zero filling of the FID to 16384 points, apodization using 100-Hz Lorentzian line broadening, Fourier transformation of the resulting FID, and saving the spectrum in a binary file with the extension `.spe`.

The SIMPSON REDOR Simulation

The four Tcl code elements presented above form the input file to SIMPSON which conveniently may be called `redor.in`. Upon appropriate installation of the SIMPSON

package (which merely involves copying the stand-alone programs to the desired directory), it is now straightforward to conduct the simulation including pulse sequence testing, data manipulation, and plotting. For convenience the processing is illustrated as commands as one would type them into a typical UNIX shell or DOS prompt environment. The first thing to do is to test the pulse sequence using the SIMDPS command by typing

```
simdps redor.in
```

which by a priori setting the number of sampling points (`np`) in the input file to 3 leads to the postscript output shown in Fig. 4b. The output illustrates the first three sampling points by solid dots on the ^{13}C channel along with the preceding pulse sequences. The detailed output in terms of delay and pulse timings as well as pulse phases provides a valuable test that the pulse sequence is correctly implemented. SIMDPS has a number of optional settings described when the program is run without arguments.

Upon testing the pulse sequence in the input file, the next step is typically the SIMPSON calculation itself being invoked as

```
simpson redor.in.
```

Unless specified otherwise in the `main` section of the input file (using the data manipulation commands) the calculation results in the time-domain output file `redor.fid`. The content of this file may be viewed, manipulated, and plotted using the SIMPLOT program

```
simplot redor.fid redor-ideal.fid redor-50kHz.fid
```

where we for the sake of illustration included dephasing curves corresponding to REDOR pulse sequences with ideal RF pulses as well as finite RF pulse irradiation with $|\omega_{RF}/2\pi| = 50$ kHz on both channels. The screen view from SIMPLOT (Fig. 4c) may be exported (printed) to a postscript file (`redor.ps`) which can be printed or further modified. We should mention that, if required, it is obviously straightforward to simulate the REDOR reference spectrum using SIMPSON and produce the difference between this and the REDOR FID using the SIMFID program, i.e.,

```
simfid redorref.fid redordiff.fid -sub redor.fid.
```

SIMPSON Iterative Fitting

Largely the need for numerical simulations in solid-state NMR spectroscopy may be divided into two classes, the first concerning experiment design, evaluation, and implementation. The other concerns extraction of structural parameters from experimental spectra. The latter not only requires the capability of numerical simulation but also calls for efficient procedures for least-squares iterative fitting of experimental spectra to numerical spectra depending on the relevant structural parameters.

Iterative fitting may be performed in the `main` section of the input file and requires definition of a function (e.g., `fitfunction`) which returns the value to be minimized, typically the root-mean-square (rms) deviation. The `fit` routine performs the minimization by passing the iteratively changed function parameters given in the `$par(values)` list to the fitting function. Each element in the list contains a name, a starting value, the step size, and 1 or 0 depending on whether the variable is iteratively changed during the minimization.

Specifically addressing the REDOR example and assuming that an experimental REDOR decay exists in a file `redorexp.fid`, fitting of the data to the dipolar coupling, the scaling, and the line broadening (exponential decay) may be accomplished by replacing the `main` procedure in the REDOR simulation given above with the fitting function (`fitfunction`) and `main` procedure shown below. The fitting function extracts the parameters from the list `val` (containing elements with the variable name, e.g., `dipole_1_2_aniso` and value) and feeds them to the `fsimpson`, `fphase`, `faddlb`, and `frms` functions to accomplish the simulation, scaling and apodization of the result, and calculation of the rms deviation between the experimental and simulated data.

```
proc fitfunction {val} {
  global par stop

  set scale  [lindex [lindex $val 0] 1]
  set lb     [lindex [lindex $val 1] 1]
  set dipole         [lindex $val 2]

  set sim [fsimpson [list $dipole]]
  fphase $sim -scale $scale
  faddlb $sim $lb 0
  set rms [frms $sim $par(exp)]
  if {$rms < $par(bestrms)} {
    set par(bestrms) $rms
    fsave $sim $par(name).fid
    puts -nonewline "*"
  }
  funload $sim
  puts "$par(iter) $rms $par(bestrms) $val"
  if {$stop || $rms <  $par(maxrms) ||
      $par(iter) > $par(maxiter)} {
    exit
  }
  return $rms
}

proc main {} {
  global par

  set par(fitmethod)  simplex
  set par(function)  fitfunction
  set par(exp)       [fload redorexp.fid]
  set par(bestrms)   1e6
  set par(maxrms)    0.5
  set par(maxiter)   1000
```

```
  set par(values) {
    {scale                1   0.1 1}
    {lb                  40  10   1}
    {dipole_1_2_aniso 1200  50   1}
  }
  fit par
}
```

The `main` procedure sets the fitting method to Simplex (*43*), sets the fitting function to be `fitfunction`, loads the experimental data and saves the data descriptor in the variable `$par(exp)`, initializes the best root-means-square value to an unrealistic large number, and sets the initial function parameters for the minimization. For each parameter the latter includes the name, the initial value, the initial step size, and a number indicating whether the parameter is considered for fitting (`1`) or not (`0`). Finally, the iterative fitting is performed by calling the `fit` procedure with the array that holds the variables guiding the minimization as argument. For each iteration the `fitfunction` function is called, the function parameters are extracted, and the simulation and comparison with the experimental data are performed. If the current rms value is less than the best rms value, the latter is updated, the acquisition data are saved, and a star is printed on the output to indicate that this is a better fit. After use, the simulated spectrum is removed from the memory (`funload`), the parameters for the current fit are printed, and the program exits provided that `maxiter` is reached, the rms value is below `maxrms`, or the special variable `$stop` is one which happens when the keyboard keys Ctrl and C are pressed simultaneously.

TYPICAL EXAMPLES OF SIMPSON SIMULATIONS

In this section we provide a series of examples demonstrating the capability of the SIMPSON environment for essentially simulating all types of solid-state NMR experiments. The examples, for which the SIMPSON input files are included in the Appendix, are chosen to illustrate different typical aspects of numerical simulations in state-of-the-art solid-state NMR. Furthermore, by the selection of current methods which are well-documented by experimental spectra and numerical simulations in the original literature, some of the examples given below additionally serve to document the validity of the simulation procedures implemented in SIMPSON and the robustness of these by applications in different contexts. The examples include rotational resonance, homonuclear dipolar recoupling using DRAMA/DRAWS/HORROR/C7, heteronuclear dipolar recoupling using TEDOR, dipolar decoupling using CW or TPPM irradiation, separated-local-field (SLF) experiments without or with FSLG/MSHOT-3 homonuclear decoupling, and QCPMG-MAS experiments for sensitivity-enhanced quadrupolar-echo NMR of half-integer quadrupolar nuclei. The capability of parameter scans for experiment optimization is demonstrated for heteronuclear coherence transfer based on POST-C7 ^{13}C–^{15}N dipolar recoupling as well as triple-quantum excitation in MQ-MAS experiments of quadrupolar nuclei. The potential of using SIMPSON for generation of advanced "waveforms" (i.e., simulation of RF irradiation with complicated amplitude or phase modulation) is illustrated for a phase-sweeped variant to the FSLG experiment. Finally, we exemplify simulation of 2D spectra by calculation of 2D PISEMA and RFDR homonuclear dipolar correlation spectra for two- and five-spin systems, respectively. For more examples we refer to Ref. (*55*).

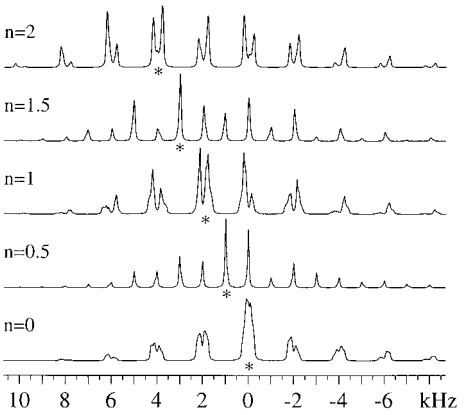

FIG. 5. Rotational resonance type ^{13}C MAS NMR spectra for a ^{13}C–^{13}C two-spin system (powder sample) simulated using $b_{CC}/2\pi = -1.5$ kHz, $\omega_r/2\pi = 2$ kHz, and different ratios $n = |\omega_{iso}^{CS,1} - \omega_{iso}^{CS,2}|/\omega_r$ between the isotropic chemical shift difference and the spinning frequency. The star (*) indicates the position of $\omega_{iso}^{CS,1}/2\pi$, while $\omega_{iso}^{CS,2}/2\pi$ is constant zero.

Example 1: Rotational Resonance Type Spectra for ^{13}C–^{13}C Spin Pairs

Although simulation of standard single-pulse experiments represents the most trivial task discussed in this paper, it is worth noting that up to quite recently it has been considered challenging just to simulate and iteratively fit such spectra for homonuclear two- or three-spin systems. This is ascribed to the presence of "homogeneous" interactions (*61*) which effectively call for a quite time-consuming time-ordered integration of the spin dynamics during sampling of the FID (*17*, *62*). Several breakthroughs, among which range efficient powder averaging (*45*, *49*) and exploitation of time-translational symmetries (*52*–*54*), have greatly improved the conditions for such simulations and their combination with iterative fitting for extraction of accurate structural parameters. To illustrate the straightforward performance of such simulations within the SIMPSON environment, Fig. 5 shows a series of rotational resonance type of spectra for a powder of ^{13}C–^{13}C spin pairs calculated for different combinations of isotropic chemical shifts and sample spinning speeds. The SIMPSON input file required for this sort of arrayed simulation and subsequent data processing only amounts to a few lines of effective code as demonstrated in the Appendix. We note that the acquisition of data under free

precession conditions conveniently is accomplished implicitly using γ-COMPUTE which exploits the time-translational relationship between $\omega_r t$ and γ_{CR} and in the present example automatically samples 4096 points equidistantly using `gamma_angles = 20` points per rotor period. Using 320 pairs of α_{CR}, β_{CR} REPULSION angles each spectrum required 6.1 s of CPU time on a Linux-controlled 450-MHz Pentium III computer, which practically puts no limitations on the combination with powerful iterative fitting procedures. For comparison, we note that a similar simulation for a three-spin system (accomplished by just changing the `spinsys` part of the input file) required 56.5 s of CPU time.

Example 2: Homonuclear Dipolar Recoupling Using DRAMA, DRAWS, HORROR, and C7

The past decade has clearly demonstrated that recoupling of homonuclear dipolar coupling interactions is by no means restricted to the rotational resonance experiment with its attractive and less attractive features. In the latter category belong practical difficulties for small isotropic chemical shift differences, its intrinsic selectivity, and its sensitivity to chemical shift anisotropy which may complicate extraction of information about internuclear distances. These aspects have motivated the design of a large series of experiments which in addition to manipulations of the Hamiltonian in real space rely on quenching of the dipolar averaging by multiple-pulse RF manipulations in spin space. Although most of these experiments are designed for the same purpose, namely selective recoupling of homonuclear dipolar couplings, they are associated with quite different dependencies on "error terms" such as the chemical shift parameters for the spin-pair nuclei. Furthermore they have different demands/limitations with respect to sample spinning speed and RF field performance. To ensure implementation of the recoupling experiment best suited for a specific application, it may be quite useful to evaluate these aspects numerically prior to the experimental work. To demonstrate the feasibility of such comparative analysis, Fig. 6 gives a series of dipolar recoupling spectra using the DRAMA (*19*), DRAWS (*20*), HORROR (*23, 63*), and C7 (*25*) pulse sequences. We note that these pulse sequences, in full analogy to their typical practical use, straightforwardly are implemented in the SIMPSON input file as exchangeable pulse sequence building blocks referring to the general pulse scheme in Fig. 6a. This is accomplished by extensive use of global parameters and the Tcl construct `lindex` in a `foreach` loop. We note that most of these recoupling sequences rely on or are dependent on finite RF pulse effects which accordingly are an integral part in most of the simulations, as specified in the input file in the Appendix. Furthermore, we should note that these simulations were performed using the Zaremba, Conroy, and Cheng *et al.* type of powder averaging with 232 pairs of α_{CR}, β_{CR} angles. This averaging scheme is extremely efficient in cases where the tensors are axially symmetric around the rotor axis, but less efficient than REPULSION and Lebedev for all other orientations.

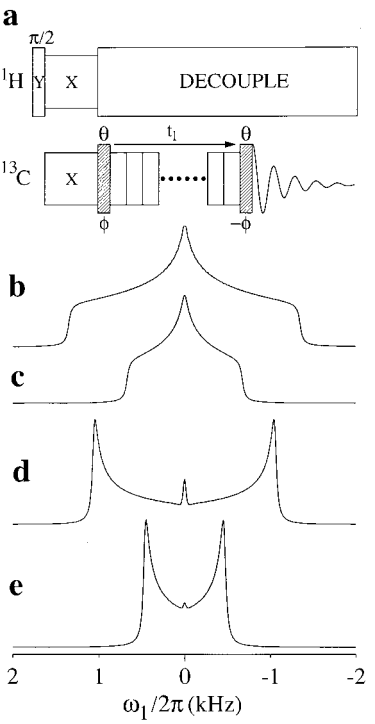

FIG. 6. Numerical simulations of various homonuclear dipolar recoupling experiments operating on a powder of ^{13}C–^{13}C spin pairs characterized by a dipolar coupling of $b_{CC}/2\pi = -2$ kHz under the conditions of MAS with $\omega_r/2\pi = 5$ kHz. (a) Generalized pulse sequence which for the simulations assumes ideal ^{1}H to ^{13}C cross polarization and ^{1}H decoupling as well as ideal ^{13}C $\pi/2$ bracketing pulses (hatched rectangles). The simulations correspond to (b) DRAMA (ideal, 0, 0), (c) DRAWS (8, 0, 0), (d) HORROR (1/2, 0, 0), and (e) C7 (7, $\pi/2$, $-\pi/2$) with the parenthesis giving $\omega_{RF}/2\pi$ in multiples of $\omega_r/2\pi$ (infinitely strong pulses are indicated ideal) as well as the flip-angle θ and phase ϕ for the bracketing pulses.

Example 3: Heteronuclear Dipolar Recoupling Using TEDOR

In the previous section we presented REDOR (*18*) as a typical example of heteronuclear dipolar recoupling under MAS conditions. Obviously several alternatives exist, among which belong the transfered-echo-double-resonance (TEDOR) experiment (*64, 65*) which here serves as another example illustrating the straightforward consideration of finite RF pulse irradiation in complex echo-train experiments in combination with extended reuse of propagators in two dipolar dephasing periods. The attractive feature of TEDOR compared to REDOR is the elimination of background signals due to uncoupled spins. This facilitates the precise measurement of the heteronuclear dipole–dipole coupling and thereby the long-range internuclear distance, e.g., between ^{15}N and ^{13}C spins in peptides or proteins. The TEDOR pulse sequence (Fig. 7a) consists of two periods of

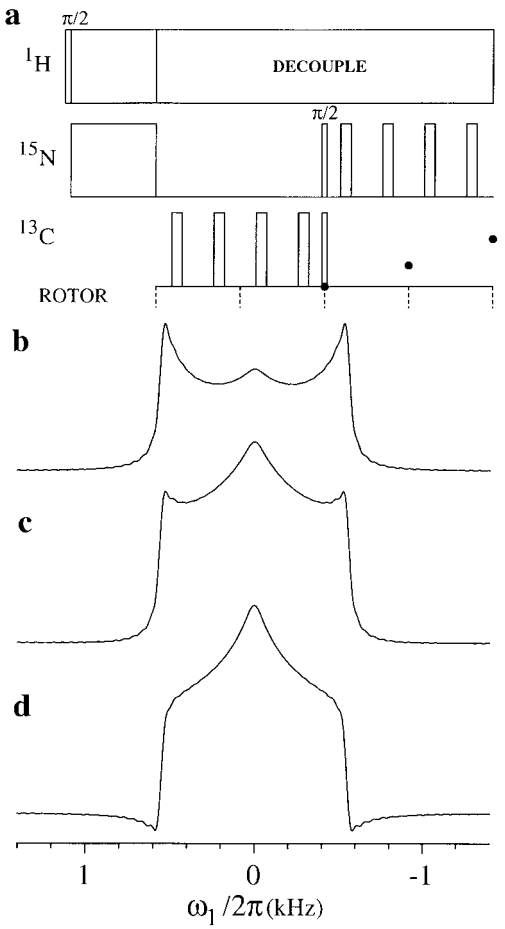

FIG. 7. Simulated TEDOR spectra for the ^{13}C–^{15}N spin pairs in a powder of L-asparagine obtained using the pulse sequence in (a) with rotor synchronized sampling, $\omega_r/2\pi = 3.2$ kHz, $\omega_{RF}/2\pi = 150$ kHz, and $b_{CN}/2\pi = 1230$ Hz ($r_{CN} = 1.36$ Å). The calculations assumed ideal cross polarization and ^{1}H decoupling, a fixed rotor-synchronized duration of the preparation sequence prior to the heteronuclear coherence transfer, and finite RF pulse conditions for all pulses. (b) to (d) TEDOR spectra obtained by Fourier transformation of the dipolar dephasing starting from the point with the maximum intensity with the number of preparational rotor periods `n` being 1, 2, and 4, respectively, as specified in the TEDOR.in input file in the Appendix. All FIDs are apodized using 30-Hz Lorentzian linebroadening.

dipolar dephasing separated by a pair of $\pi/2$ pulses which establish the heteronuclear coherence transfer. The dipolar dephasing periods contain a series of π refocusing pulses applied synchronously with the rotor period to recover the dipolar coupling interaction otherwise averaged over a rotor period. Inspired by Fig. 9 in the paper of Hing *et al.* (*64*), Figs. 7b–7d show a series of Fourier-transformed TEDOR dephasing curves corresponding to the ^{13}C–^{15}N spin-pair in L-asparagine. The corresponding Tcl input file is given in the Appendix. We note that the Fourier transformation of the time-domain data is performed as postprocessing by discarding the data points prior to the echo maximum.

Example 4: POST-C7 Heteronuclear Dipolar Recoupling: Excitation Curve and 2D Parameter Scan

In the design and implementation of new pulse sequences, it is desirable on the level of numerical simulations to be able to scan the dependencies of the pulse sequence toward various parameters of the internal and external Hamiltonian. This is equivalent to the mandatory calibrations of external manipulations always preceding experiments on the spectrometer. For example, in setting up a dipolar recoupling experiment for heteronuclear coherence transfer, it is relevant to know the optimum excitation period for the actual spin system (depending on the internuclear distance, type of nuclei, etc.) and the sensitivity of this transfer toward other parameters such as the chemical shifts of the nuclei. In this manner, it should be possible to avoid disappointments and hours of "blind" spectrometer optimization in cases where pulse techniques designed for one application are transferred to a completely different context with respect to the internal interactions.

As an example of looped parameters scans for pulse sequence optimization and applicability analysis, Fig. 8 investigates ^{13}C to ^{15}N heteronuclear coherence transfer using a heteronuclear variant of the POST-C7 (*26*) pulse sequence as visualized in Fig. 8a and represented by the SIMPSON input file in the Appendix. With respect to the latter we should address attention to the very simple construction of the quite advanced POST-C7 pulse sequence in the `pulseq` section as a super cycle which through looping concatenates the cyclically phase-modified three-pulse building blocks as required to produce the excitation curve given in Fig. 8b. In Fig. 8c we exemplify 2D parameters scans by a 3D plot (2D contour plot), giving the excitation efficiency along the vertical axis and the dependencies on the ^{13}C and ^{15}N isotropic chemical shifts along the horizontal axes. Such curves are highly relevant for deduction of the broadband nature of the recoupling pulse sequence in the given application. For space reasons, the loops required to produce this 2D scan are given in Ref. (*55*). We note that the present simulations restrict to β_{CR} powder averaging (40 angles in the `bcr40` file) being justified by the γ_{CR}-encoding (*23*) properties of C7 and the present ignorating effects from chemical shielding anisotropy. Obviously, at the expense of longer calculation times, it is straightforward to change the powder averaging to cover the full semi-sphere by just entering another file name.

Example 5: Heteronuclear Decoupling in Multiple-Spin Systems Using CW and TPPM

In addition to recoupling techniques recovering parts of the internal Hamiltonian being coherently averaged by MAS in an attempt to obtain high-resolution spectra, decoupling of isotropic or anisotropic *J* and dipolar coupling interactions represents another important element in tailoring the internal Hamiltonian to the desired shape. So far the prevailing method for heteronuclear dipolar decoupling has been "brute-force" high-

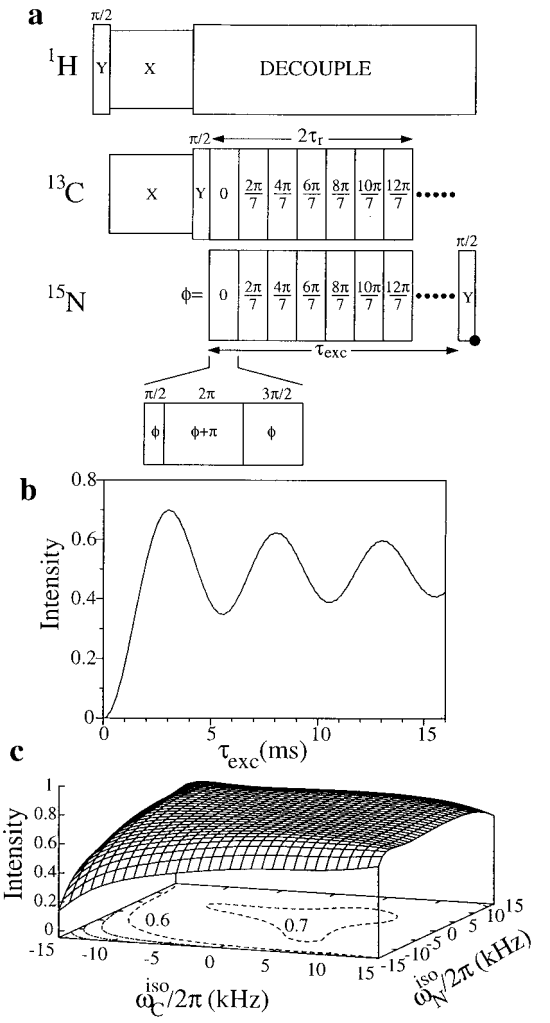

FIG. 8. Simulation of ^{13}C to ^{15}N coherence transfer for a powder of ^{13}C–^{15}N spin pairs characterized by a dipolar coupling constant of $b_{CN}/2\pi$ = 1.3 kHz (r_{CN} = 1.33 Å) using the heteronuclear variant of the POST-C7 pulse sequence (a) with $\omega_r/2\pi = \omega_{RF}/14\pi$ = 8 kHz (both channels). The simulations assumed ideal cross polarization, ^{1}H decoupling, and $\pi/2$ bracketing pulses. (b) Excitation curve with the POST-C7 coherence transfer sampled in steps of two rotor periods ($2\tau_r = \omega_r/\pi$) under conditions of on-resonance RF irradiation on the isotropic chemical shifts for both spins. (c) Transfer efficiency for the POST-C7 pulse sequence using $\tau_{exc} = 12\tau_r$ as function of the isotropic chemical shifts for the two spin species.

power continuous wave (CW) irradiation on the nonobserved spin species (*1–4, 6*), although more efficient schemes such as two-pulse phase modulation (TPPM) (*31*), phase alternated 2π pulse irradiation (*66*), frequency- and phase-modulated decoupling (FMPM) (*67*), off-resonance decoupling (*68*), and 12-fold symmetric C12 decoupling sequences (*69*) have entered the scene recently. The more advanced schemes have in particular proven advantageous in the case of fast sample spinning where dipolar couplings between the protons become less efficient and thereby less helpful in truncating second-order cross-terms involving anisotropic chemical shielding on the directly coupled proton spin (*66, 69*). Exactly this aspect explains why it is very demanding to decouple the influence from the proton in a simple heteronuclear two-spin system unless extremely high decoupling fields are applied. This applies in particular if the proton is influenced by anisotropic shielding (*66*). Furthermore, it explains why combined heteronuclear decoupling and homonuclear proton–proton dipolar recoupling may improve this situation (*69*). To assist the analytical evaluation of such effects and investigate the specific role of the heteronuclear spin systems, it appears to be desirable to have straightforward access to numerical simulations. This may be accomplished by SIMPSON, as illustrated in Fig. 9 by a series of simulations illustrating the effect of CW and TPPM decoupling for a heteronuclear four-spin system. For the purpose of illustration we examine a methylene carbon which, in addition to the directly bonded protons, is influenced by heteronuclear dipolar coupling to a remote spin and indirectly

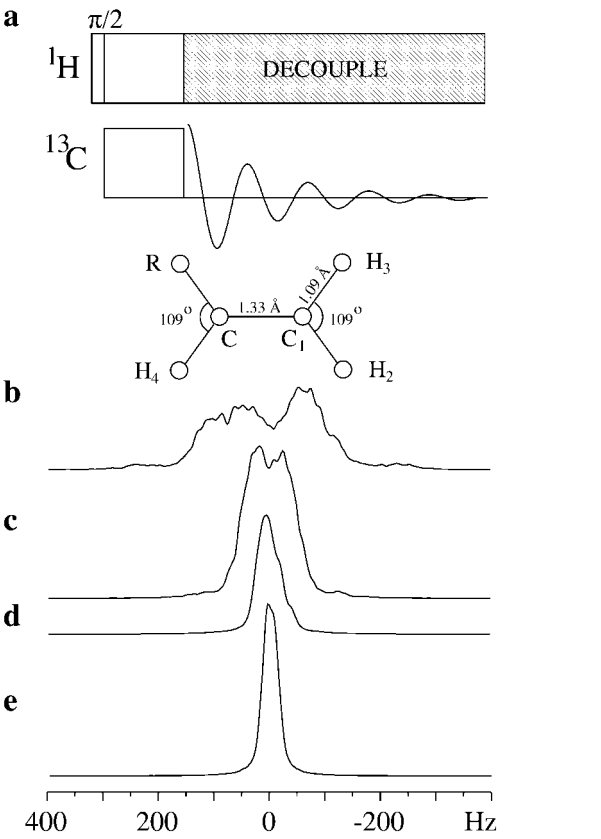

FIG. 9. Simulations comparing CW and TPPM ^{1}H decoupling for a ^{13}C spin dipolar coupled to three mutually coupled protons which additionally are influenced by anisotropic chemical shift. The spin system geometry is visualized above the spectra while the dipolar coupling and chemical shift parameters are given in the SIMPSON input file for this example included in the Appendix. (a) Generalized pulse sequence which for the simulations assume ideal CP and MAS with $\omega_r/2\pi$ = 5 kHz. (b–e) ^{13}C MAS spectra for a powder sample subjected to (b, c) CW and (d, e) TPPM ($180_{25}180_{-25}$ cycles) decoupling using decoupling RF field strengths of (b, d) 80 and (c, e) 160 kHz. We note that the vertical scale for the CW spectra is expanded by a factor 2.5 relative to the TPPM spectra and that all FIDs are apodized using 10-Hz Lorentzian linebroadening.

influenced by a number of homonuclear ^{1}H–^{1}H dipolar couplings and ^{1}H chemical shift anisotropy. The geometry of the spin system is illustrated by an insert above the decoupled ^{13}C MAS spectra in Fig. 9 reflecting experiments using the simple 1D pulse sequence in Fig. 9a under the assumption of ideal cross-polarization. Specifically, the simulations reflect a powder sample subjected to MAS with $\omega_r/2\pi = 5$ kHz along with CW or TPPM decoupling using ^{1}H decoupling field strengths of 80 and 160 kHz. From the spectra it is evident that the CW using the lowest RF field amplitude provides far from sufficient decoupling of the protons. This situation is improved considerably by increasing the CW RF field strength to 160 kHz and even more so by applying TPPM decoupling which in the present case used $\theta_\phi\theta_{-\phi}$ cycles with $\theta = 180°$ and $\phi = 25°$. To our knowledge this represents the first numerical simulations comparing the decoupling performance of CW and TPPM decoupling despite the fact that it has been extensively demonstrated analytically and experimentally during the past couple of years. With specific attention to the input file, we note that the simulations are most efficiently conducted using γ-COMPUTE which requires synchronization of the TPPM irradiation, the sample spinning, and the sampling. To accomplish this the code automatically adjusts the desired spinning frequency `spin_want` to the rotor-synchronized actual frequency `spin_rate`, being informed to the user using the Tcl `puts` command.

Example 6: SLF Experiments without and with MSHOT-3 and FSLG Homonuclear Decoupling

One of the most typical applications for homonuclear dipolar decoupling pulse sequences is as an element in separated-local-field experiments for measurement of heteronuclear dipolar couplings (*70*). For example, this is of interest for determination of the orientation of N–H internuclear axes relative to the external magnetic field in single crystals or uniaxially oriented peptide samples, as described by Opella and co-workers (*33*). In this sort of applications, where the line position is interpreted directly in terms of orientational angles, it is quite important that the applied decoupling scheme provides reasonable decoupling.

Such performance tests may conveniently be conducted using SIMPSON as demonstrated in Fig. 10 by simulated $\omega_1/2\pi$-dimension SLF spectra for a single crystal of a ^{15}N–$(^1\text{H})_2$ three-spin system using the pulse sequence in Fig. 10a without and with FSLG (*29*) and MSHOT-3 (*30*) homonuclear decoupling. Using typical N–H internuclear distances (1.01 Å) and the N–H vectors split by the tetrahedral angle, this spin system is characterized by dipolar constants of $b_{HH}/2\pi = -26.7$ kHz and $b_{NH}/2\pi = 11.8$ kHz which for the given tensor geometry puts some demands on the homonuclear decoupling. The influence from the homonuclear coupling becomes immediately evident from the spectrum without decoupling (Fig. 10b) which contains six resonances, two weak lines at ca. ±11.8 kHz (foldings at ±1455) and four lines belonging to the displayed spectral window. Apart from the resonances at ±2316 Hz this spectrum differs significantly from the spectrum without the homonuclear coupling ideally showing four resonances at ±2316 and ±3160 Hz for the given crystallite orientation. As a first attempt to quench the influence from the homonuclear coupling, Fig. 10c shows a spectrum using FSLG decoupling with an RF field strength of 58 kHz. This spectrum shows a doublet and a quartet, which by its difference to the expected two-doublet spectrum by itself indicates inadequate decoupling. Also, the frequencies of the doublet lines (−1788 and −1315 Hz) and obviously the quartet lines (1260, 1435, 1657, and 1835 Hz) are shifted relative to the theoretical values (±1824 and ±1336 Hz). Clearly, such distortions and frequency shifts are unfortunate provided that information about the orientation of the internuclear axes is extracted from the line positions. The MSHOT-3 spectrum (Fig. 10d) using a decoupling RF field strength of 60 kHz looks more promising, although characterized by a

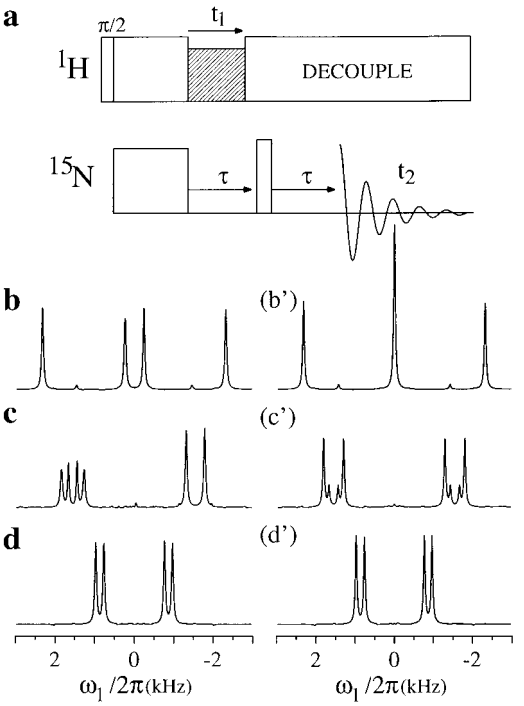

FIG. 10. Simulated $\omega_1/2\pi$ projections from ^{1}H–^{15}N SLF experiments using different homonuclear decoupling sequences for a single crystal (Ω_{CL} = {30°, 65°, 0}) of a ^{15}N$(^1\text{H})_2$ three-spin system. The spin system is characterized by $b_{H_1H_2}/2\pi = -26.7$ kHz, $b_{NH_1}/2\pi = b_{NH_2}/2\pi = 11.8$ kHz, Ω_{PC}^{H1H2} = {0, 35.2°, 0}, Ω_{PC}^{NH1} = {0, 0, 0}, and Ω_{PC}^{NH2} = {0, 70.4°, 0}. (a) General pulse sequence for the constant-time SLF experiment with the homonuclear decoupling sequence indicated by the hatched rectangle. (b–d) Spectra corresponding to SLF (b) without homonuclear decoupling, (c) FSLG decoupling using $\omega_{RF}/2\pi = 58$ kHz, and (d) MSHOT-3 decoupling with $\omega_{RF}/2\pi = 60$ kHz all calculated using $\omega_{iso}^{CS,H1}/2\pi = -\omega_{iso}^{CS,H2}/2\pi = 500$ Hz (no chemical shift anisotropy). (b′–d′) The corresponding spectra calculated under the assumption of identical isotropic chemical shifts for the protons.

lower scaling factor (0.353 as opposed to 0.577 for FSLG), in the sense that two doublets symmetrically disposed around the center of the spectrum are obtained. The positions for the doublet lines (−968 and −765 Hz), however, remain shifted relative to the theoretical values (−1117 and −819 Hz). Obviously, this situation may be improved considerably by increasing the decoupling RF field strength (not shown). Furthermore, it is interesting to note that SLF pulse sequences without decoupling and with FSLG decoupling are quite sensitive to the chemical shifts of the protons. This becomes evident by comparison of the two series of spectra in Figs. 10b–10d and Figs. 10b′–10d′ calculated under the assumption of finite (±500 Hz) and vanishing isotropic ^{1}H chemical shifts, respectively. At the same time these spectra reveal that MSHOT-3 is relatively insensitive to these effects, as discussed previously (*30*). Finally, we should emphasize that the spectra in Fig. 10, the appearance of which highly depends on the selected tensor angles, the decoupling RF field strength, and the orientation of the crystal, by no means give a general evaluation of the different types of decoupling sequences but rather serve to demonstrate the relevance of numerical simulations to investigate and potentially compensate for the effects of insufficient decoupling.

Example 7: Heteronuclear Cross-Polarization Employing FSLG with Frequency Switching or Phase Modulation

In the past few years there has been an increasing interest in exploiting advanced phase- and amplitude-modulated RF pulse sequences as a flexible tool for manipulation of the internal Hamiltonian. For example, such schemes have been proposed for hetero- (*67*) and homonuclear (*71*, *72*) decoupling, cross polarization (*73*), dipolar recoupling (*74*, *75*), and multiple-quantum MAS (MQ-MAS) refocusing of second-order quadrupolar line broadening (*76*, *77*). Obviously, the ability to create continuous phase and amplitude modulation increases the degrees of freedom in experiment design but it also extends the need for numerical simulations to accurately analyze the performance of the pulse sequences under ideal and nonideal conditions. For example, continuous RF modulation is often implemented as discrete steps on the spectrometers which inevitably calls for the question: how fine is the digitalization needed to meet the continuous condition with sufficient accuracy? Often, this question can only be answered by experimental tests or numerical simulations.

To demonstrate that SIMPSON allows handling of advanced time-modulated pulse sequences using simple loop constructs, Fig. 11 shows a series of excitation curves for ^{1}H to ^{15}N coherence transfer using a spin-exchange at the magic angle (SEMA) type of pulse sequences with FSLG irradiation at the ^{1}H channel along with a phase-alternated pulse sequence on the ^{15}N channel (*33*, *79*). Prior to FSLG the ^{1}H magnetization is prepared along an axis parallel to the effective field direction of

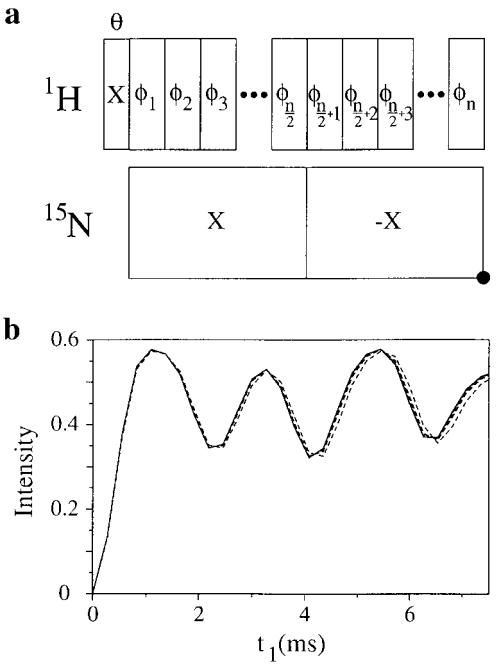

FIG. 11. Simulations of ^{1}H to ^{15}N coherence transfer for a static powder of typical amide ^{1}H–^{15}N spin systems ($b_{NH}/2\pi$ = 11.8 kHz, δ^N_{aniso} = 100 ppm, η^N_{CS} = 0.5, δ^H_{iso} = 100 ppm, η^H_{CS} = 0.5) using a SEMA-type cross-polarization pulse sequence with FSLG irradiation on the ^{1}H channel and phase inversion on the ^{15}N channel according to the pulse sequence in (a). The pulse sequence uses θ = 54.74° along with $\omega^H_{RF}/2\pi = \sqrt{2/3}\ \omega^N_{RF}/2\pi = -\sqrt{2}\ \omega_{off}/2\pi$ = 60 kHz for the simulations in (b). $\omega_{off}/2\pi$ denotes the frequency offsets used for the conventional FSLG sequence. (b) SEMA CP excitation curves for sequences using conventional frequency switching (solid line) as well as phase modulations with different phase steps (208° divided into 8, 13, 16, 26, 52, 104, and 208 steps) (dashed lines).

the Lee–Goldburg (LG) sequence using a ^{1}H preparation pulse with flip angle 54.74° and phase *x*. The basic pulse sequence is shown in Fig. 11a. The FSLG element of this pulse sequence can be executed either directly using frequency (i.e., offset) switching as originally proposed by Bielecki *et al.* (*29*) or using phase modulation as recently discussed by Vinogradov *et al.* (*71*) and Gan (*72*). Both approaches have been implemented in the SIMPSON input file for this example in the Appendix. Thus, in addition to an excitation curve produced using SEMA CP with conventional frequency switching, Fig. 11b shows excitation curves for various phase-modulated realizations of the experiment using 8, 31, 16, 26, 52, 104, and 208 steps in the 208° phase sweeps. By comparison, these curves allow for strict evaluation of the effect of discretization of the continuous phase modulation. It appears that a fairly good reproducibility of the frequency-switched experiments is achieved even using phase steps as large as 26°, which timing-wise definitely are less demanding to implement on the spectrometer than the more real continuous-phase sweep. We note that for the selected ^{1}H–^{15}N two-spin system, characterized by dipolar coupling and chemical shift parameters typical for amide NH spin systems in peptides and experimental conditions corresponding

to a 400 MHz spectrometer, the transfer efficiency for a static powder sample is about 0.57. Finally, we should note that FSLG implemented either in the conventional frequency-switching way or by phase modulation represents a very useful building block in a number of multidimensional solid-state NMR correlation experiments including, e.g., heteronuclear chemical shift correlation (*78*) and heteronuclear chemical shift versus dipolar coupling correlation, among which the latter is addressed in the following example.

Example 8: 2D ^{1}H–^{15}N PISEMA SLF Experiment

Obviously, SIMPSON simulations are not by any means restricted to one-dimensional spectra or parameter scans. They may equally well be applied to simulate multidimensional solid-state NMR spectra, as demonstrated in the next couple of examples. The first example is a simulation of a ^{15}N chemical shift versus ^{1}H–^{15}N dipolar coupling correlated spectrum obtained using the PISEMA (*33, 80, 81*) pulse sequence in Fig. 12a. The parameters for the simulated spectrum shown in Fig. 12b are chosen to match a typical amide spin-pair in a peptide. According to the pulse sequence in Fig. 12a and assuming ideal cross-polarization, the ^{1}H part of the initial $-I_{1y}$ (^{1}H) and I_{2x} (^{15}N) coherence is tilted to a "magic angle" orientation where it is spin locked by a LG sequence while the ^{15}N coherence is spin locked by CW irradiation in full analogy to the pulse sequence described in the previous example. The full 2D scheme is readily implemented in SIMPSON (see Appendix) by sampling of the normal t_2-dimension FID for each of the t_1 values incremented in steps of the FSLG block using a conventional loop construction. We note in passing that the PISEMA experiment may be considered a powerful alternative to the conventional SLF experiment in the sense that it typically provides a significantly better resolution and has a better dipolar scaling factor (0.83) than typical SLF experiments using homonuclear multiple-pulse decoupling (0.30–0.58). These features have rendered PISEMA a popular building block in various 2D and 3D experiments for heteronuclear coherence transfer and chemical shift evolution while simultaneously suppressing the dominant homonuclear dipole–dipole couplings among the abundant proton spins (*72, 79–82*).

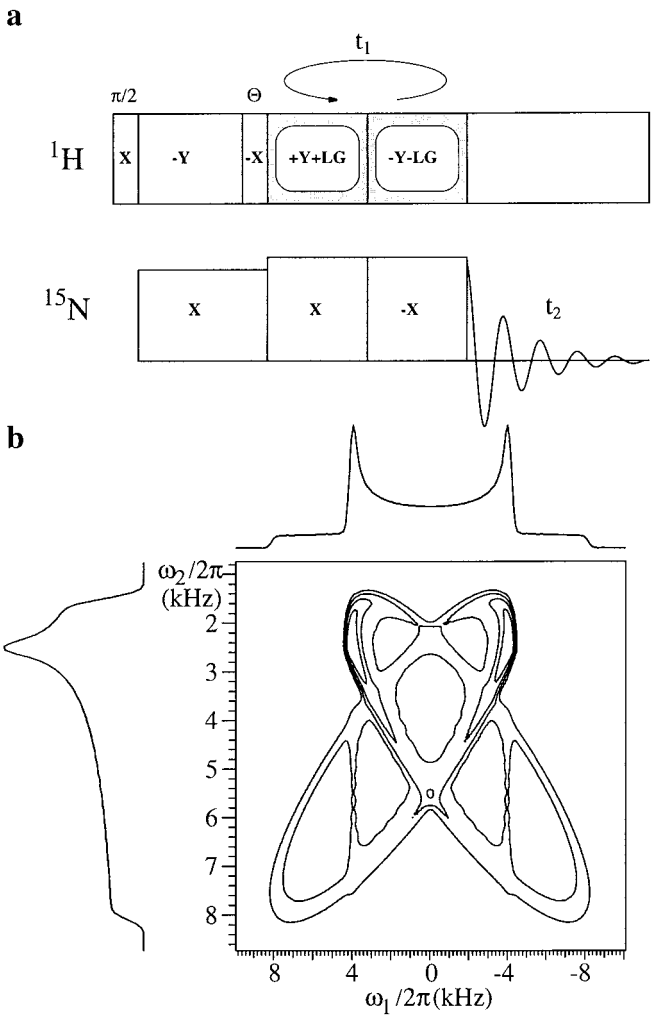

FIG. 12. 2D ^{15}N chemical shift versus ^{1}H–^{15}N dipolar coupling correlated spectrum (b) calculated for a static powder of typical amide ^{1}H–^{15}N spin pairs using the 2D PISEMA separated-local-field pulse sequence in (a) with Larmor frequencies corresponding to a 400-MHz spectrometer. The spin pair is characterized by the parameters (nonzero values only) $b_{NH}/2\pi = 10$ kHz, $\delta^{N}_{iso} = 100$ ppm, $\delta^{N}_{aniso} = 100$ ppm, and $\eta^{N}_{CS} = 0.25$ while the pulse sequence used $\omega^{H}_{RF}/2\pi = \sqrt{2/3}\ \omega^{N}_{RF}/2\pi = -\sqrt{2}\ \omega_{off}/2\pi = 83$ kHz for the SEMA part and $\omega^{H}_{RF}/2\pi = 130$ kHz decoupling. The 2D spectrum was sampled using 128 points in both dimensions.

Example 9: 2D ^{13}C, ^{13}C Chemical-Shift Correlation in a Five-Spin-1/2 System Using RFDR Dipolar Recoupling

The second example of a 2D SIMPSON simulation addresses homonuclear ^{13}C chemical shift correlation for a five-spin system using a radiofrequency-driven dipolar recoupling (RFDR) (*21*) pulse sequence in the mixing period of a 2D MAS experiment. RFDR represents a frequently used dipolar recoupling experiment, which as a disadvantage is not γ-encoded such as the HORROR (*23*) and C7 (*25, 26*) class of recoupling experiments, but benefits from an attractive experimental robustness and forgiveness with respect to isotropic and anisotropic chemical shifts (unless the isotropic shift differences for the involved spin pairs are very small). These features have rendered the 2D RFDR pulse sequence shown in Fig. 13a quite popular for obtaining 2D ^{13}C, ^{13}C chemical shift correlated spectra for biological macromolecules in the solid phase (*83, 84*). In the present example we employ SIMPSON to simulate the 2D correlation spectrum for five-dipolar-coupled ^{13}C nuclei arranged in a "zigzag" coordination as illustrated in Fig. 13b and each exhibiting typical values for the anisotropic chemical shifts. For clarity of the illustration we have arbitrarily

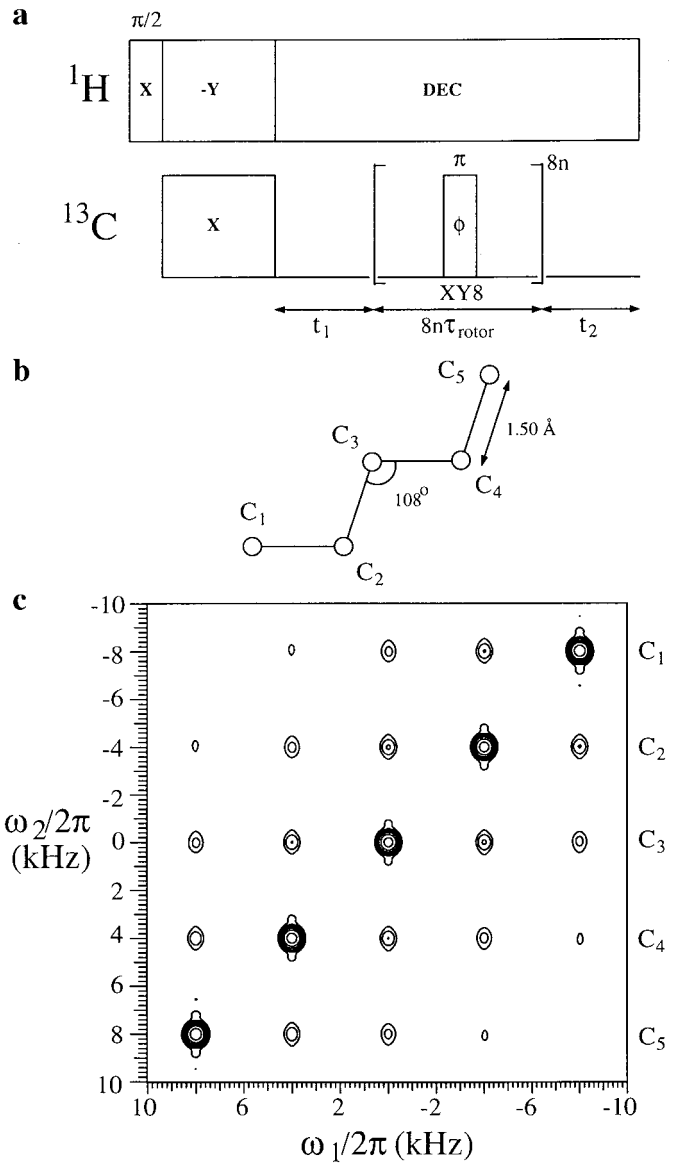

FIG. 13. Simulation of a 2D ^{13}C, ^{13}C chemical shift correlated spectrum (c) for a dipolar coupled five-spin system of ^{13}C nuclei (b) obtained using the 2D pulse sequence in (a) with dipolar mixing based on the RFDR recoupling sequence with XY-8 phase alternation (*60*). The pulse sequence assumed ideal RF pulses (including CP and ^{1}H decoupling) and a sample spinning speed of $\omega_r/2\pi = 20$ kHz. Details on the spin system and pulse sequence parameters are given in the input file in the Appendix.

assumed a 40-ppm isotropic chemical shift difference between neighboring spins. Using this setup (specified in more detail in the SIMPSON input file in the Appendix) and a mixing period of 2.4 ms, allowing magnetization to be transferred over distances corresponding at least to three bonds, we obtain the simulated 2D correlation spectrum shown in Fig. 13c. Simulation of this five-spin 2D MAS powder spectrum, using 30 pairs of α_{CR}, β_{CR} angles, 5 γ_{CR} angles, and 32 t_1 increments, required approximately 6 h of CPU time on a standard PC 450-MHz Pentium III processor.

Example 10: Sensitivity-Enhanced Quadrupolar-Echo NMR: The QCPMG-MAS Experiment

The last two examples serve to demonstrate that SIMPSON, obviously, may also be used for simulation of multiple-pulse solid-state NMR experiments involving quadrupolar nuclei. The first example addresses the application of the quadrupolar Carr–Purcell–Meiboom–Gill (QCPMG) (*85, 86*) pulse sequence in combination with MAS to improve the sensitivity of quadrupolar-echo spectra for half-integer quadrupolar nuclei (*34*). Calculation of QCPMG-MAS NMR experiments based on the pulse sequence in Fig. 14a may be numerically very demanding when taking into account the quite large quadrupolar coupling interaction calling for extensive powder averaging, the potential for large matrix dimensions, and the need for consideration of finite RF pulse effects throughout a train of RF pulses applied under fast sample spinning conditions. Typically, and in particular for quadrupolar nuclei with large I-spin quantum numbers, this invites custom-made software heavily optimized for the specific problem in mind as described previously (*34, 87*). Nonetheless, although at the expense of slightly longer calculation times (less than a factor of 2), such simulations may also be conducted within the flexible simulation environment offered by SIMPSON. This is illustrated in Fig. 14b by simulation of a typical QCPMG-MAS spectrum for a powder of spin $I = 3/2$ nuclei characterized by a large

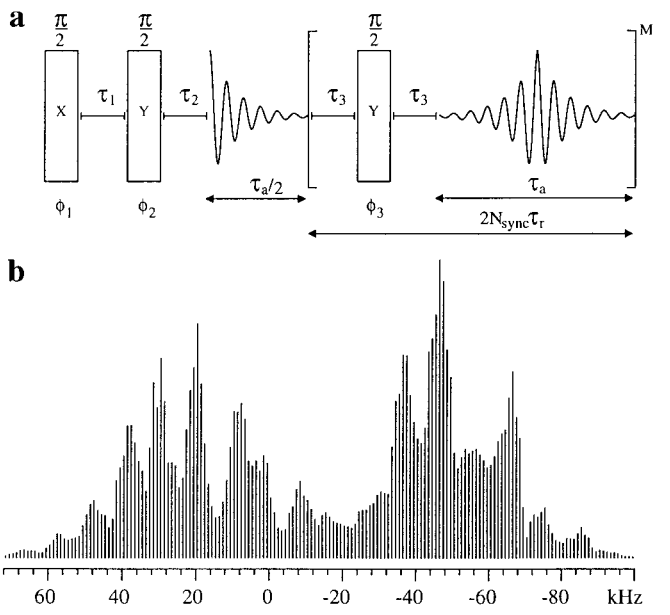

FIG. 14. (a) QCPMG-MAS pulse sequence for sensitivity-enhanced quadrupolar-echo spectroscopy of half-integer quadrupolar nuclei. (b) Simulated QCPMG-MAS spectrum for a powder of $I = 3/2$ nuclei characterized by $C_Q = 10$ MHz, $\eta_Q = 0.12$ $\Omega_{PC}^Q = \{0, 0, 0\}$, $\delta_{aniso} = -150$ ppm, $\eta_{CS} = 0.60$, and $\Omega_{PC}^{CS} = \{90°, 30°, 90°\}$. The pulse sequence used $M = 30$, $\tau_1 = 103.25$ μs, $\tau_2 = 104.45$ μs, $\tau_3 = 22.32$ μs, $\tau_a = 1$ ms, $N_{sync} = 5$, a dwell time of 4 μs, $\omega_r/2\pi = 9.5$ kHz, $\omega_{RF}/2\pi = 64.1$ kHz, and $\omega_0/2\pi = -130$ MHz. For more details we refer to the input file in the Appendix.

quadrupolar coupling interaction tensor (C_Q = 10 MHz) and a differently oriented anisotropic chemical shift tensor along with experimental conditions using $\omega_r/2\pi$ = 9.5 kHz, $\omega_{RF}/2\pi$ = 64.1 kHz, and $\omega_0/2\pi$ = −130 MHz. We note that the finite RF pulses for the echo train (cf. Appendix) were calculated using the approximative replication scheme described in Ref. (*34*) and that the spin-echo sideband intensities were integrated to form the stick-plot shown in Fig. 14b. The latter was accomplished using the `fssbint` integration command in the `main` section of the input file. We note that the "stick-plot" representation is useful for numerical evaluation and iterative fitting toward experimental spin-echo sideband intensities to obtain information about the magnitude and relative orientation of quadrupolar coupling and anisotropic chemical shift tensors.

Example 11: MQ-MAS NMR of Half-Integer Quadrupolar Nuclei

As a final example, also addressing quadrupolar nuclei, we consider the MQ-MAS experiment which recently has found widespread application as a tool to obtain high-resolution spectra for half-integer quadrupolar nuclei (*35*). This experiment relies on combined evolution under triple- (in the case of spin $I = 3/2$ nuclei) and single-quantum coherence, implying that the success of the experiment heavily relies on the ability to perform efficient transformations between these states. The transfer efficiency for a specific application has a complicated dependence on the quadrupolar coupling constant, the available RF field strength, and the sample spinning frequency. Thus, it is of interest to optimize the experiment for given combinations of RF field strengths and quadrupolar coupling constants. This is conveniently accomplished numerically as demonstrated in Fig. 15 by simulated curves for the efficiency of single-pulse excitation of triple-quantum coherence for a powder of spin $I = 3/2$ nuclei as function of the excitation pulse length for the two-pulse sequence (*88*) in Fig. 15a using $\omega_{RF}/2\pi$ = 80 kHz and different quadrupolar coupling constants (Fig. 15b). The curves were calculated using the `matrix set` and `filter` commands to accomplish detection only through the triple-quantum transitions.

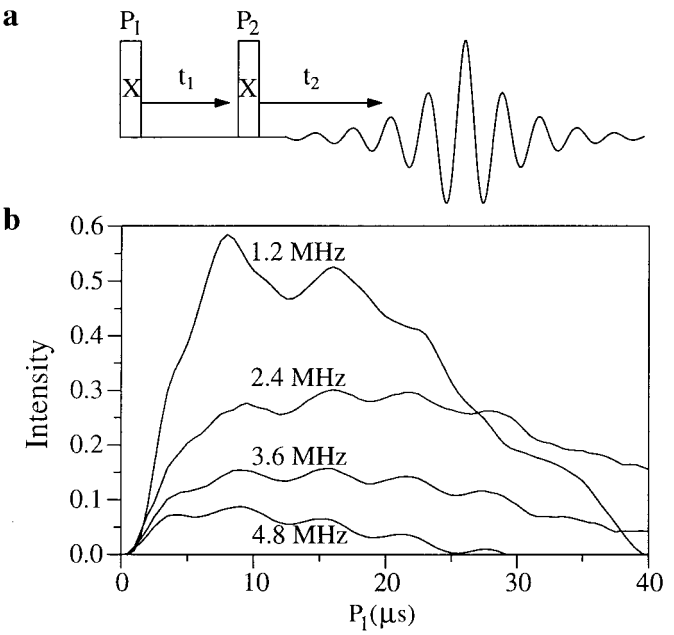

FIG. 15. (a) 2D pulse sequence for MQ-MAS NMR of half-integer quadrupolar nuclei. (b) Triple-quantum excitation curve calculated as a function of the pulse length P_1 using $\omega_{RF}/2\pi$ = 80 kHz for various values of C_Q (η_Q = 0).

CONCLUSION

In conclusion, we have presented a new and powerful software package for fast simulation of essentially all solid-state NMR experiments. The package, consisting of the simulation tool SIMPSON along with supplementary programs for processing and visualization, allows easy and flexible implementation of advanced multiple-pulse experiments at a level of abstraction closely resembling the operation of a modern solid-state NMR spectrometer. Thus, acting as a "computer spectrometer," it is foreseen that SIMPSON will form an important platform for spin engineers systematically constructing and evaluating new pulse techniques as well as for the solid-state NMR spectroscopists using SIMPSON along with its iterative fitting procedures to extract structural parameters from solid-state NMR spectra resulting from more or less advanced experimental methods.

APPENDIX

A: SIMPSON Input Files for the Example Simulations

Example 1: Rotational Resonance Type Spectra for ^{13}C–^{13}C Spin Pairs

```
spinsys {
  channels 13C
  nuclei 13C 13C
  shift 1 0 6000 1 0 0 0
  shift 2 0 6000 0 0 0 0
  dipole 1 2 -1500 0 0 0
}

par {
  method          gcompute
  spin_rate       2000
  gamma_angles    20
  sw              spin_rate*gamma_angles
  np              4096
  crystal_file    rep320
  start_operator  Inx
  detect_operator Inp
}
```

```
proc pulseq {} {
  maxdt 5.0
  delay 1e6
}

proc main {} {
  global par

  for {set iso 0} {$iso <= 4000} {incr iso 1000} {
    set f [fsimpson [list [list \
        shift_2_iso $iso]]]
    faddlb $f 50 0
    fft $f
    fsave $f $par(name)-$iso.spe
    funload $f
  }
}
```

Example 2: Homonuclear Dipolar Recoupling Using DRAMA, DRAWS, HORROR, and C7

```
spinsys {
  channels 13C
  nuclei 13C 13C
  dipole 1 2 -2000 0 0 0
}

par {
  spin_rate   5000
  sw          spin_rate
  np          256
  crystal_file zcw232
}

proc pulseq {} {
  global par

  maxdt 1.0
  set nprop 1

  if {$par(type) == "drama"} {
    set tr4 [expr 0.25e6/$par(spin_rate)]
    set tr2 [expr 0.5e6/$par(spin_rate)]
    reset
    delay $tr4
    pulseid 1 250000 x
    delay $tr2
    pulseid 1 250000 -x
    delay $tr4
    store 1

  } elseif {$par(type) == "draws"} {
    set rf   [expr 8*$par(spin_rate)]
    set t360 [expr 1.0e6/$rf]
    reset
    pulse $t360 $rf y
    pulse $t360 $rf -y
    pulseid 1 250000 x
    pulse $t360 $rf y
    pulse $t360 $rf -y
    pulse $t360 $rf -y
    pulse $t360 $rf y
    pulseid 1 250000 x
    pulse $t360 $rf -y
    pulse $t360 $rf y
    store 1

  } elseif {$par(type) == "horror"} {
    set tsw [expr 1.0e6/$par(sw)]
    set rf  [expr $par(spin_rate)/2.0]
    reset
    pulse $tsw $rf x
    store 1

  } elseif {$par(type) == "c7"} {
    set nprop 2
    set rf    [expr 7*$par(spin_rate)]
    set t360  [expr 1.0e6/$rf]
    reset
    pulse [expr 2*$t360] $rf [expr 360/7.0*0]
    pulse [expr 2*$t360] $rf [expr 360/7.0*1]
    pulse [expr 2*$t360] $rf [expr 360/7.0*2]
    pulse [expr 1*$t360] $rf [expr 360/7.0*3]
    store 1
    reset [expr 7*$t360]
    pulse [expr 1*$t360] $rf [expr 360/7.0*3]
    pulse [expr 2*$t360] $rf [expr 360/7.0*4]
    pulse [expr 2*$t360] $rf [expr 360/7.0*5]
    pulse [expr 2*$t360] $rf [expr 360/7.0*6]
    store 2
  }

  reset
  acq
  for {set i 1} {$i < $par(np)} {incr i} {
    prop [expr (($i-1) % $nprop)+1]
    acq
  }
}

proc main {} {
  global par

  foreach p {{drama 100 x} {draws 100 x}
        {horror 1 x} {c7 1 z}} {

    set par(type)            [lindex $p 0]
    set par(gamma_angles)    [lindex $p 1]
    set par(start_operator)  I1[lindex $p 2]
    set par(detect_operator) $par(start_operator)

    set f [fsimpson]
    faddlb $f 50 0
    fzerofill $f 8192
    fft $f
```

```
    fsave $f $par(name)-$par(type).spe -binary
    funload $f
  }
}
```

Example 3: Heteronuclear Dipolar Recoupling Using TEDOR

```
spinsys {
  channels 15N 13C
  nuclei 15N 13C
  dipole 1 2 1230 0 0 0
}

par {
  spin_rate       3200
  sw              spin_rate
  np              64
  crystal_file    zcw376
  gamma_angles    100
  start_operator  I1x
  detect_operator I2p
  variable rf     150000
  variable n      3
}

proc pulseq {} {
  global par

  maxdt 1.0

  set rf   $par(rf)
  set tr   [expr 1.0e6/$par(spin_rate)]
  set t90  [expr 0.25e6/$rf]
  set t180 [expr 2.0*$t90]

  reset
  delay [expr $tr/4.0-$t180/2.0]
  pulse $t180 0 x $rf x
  delay [expr $tr/2.0-$t180]
  pulse $t180 0 x $rf x
  delay [expr $tr/4.0-$t180/2.0]
  store 1

  reset $t90
  delay expr $tr/4.0-$t180/2.0]
  pulse $t180 $rf x 0 x
  delay [expr $tr/2.0-$t180]
  pulse $t180 $rf x 0 x
  delay [expr $tr/4.0-$t180/2.0]
  store 2

  reset
  prop 1 $par(n)
  pulse $t90 $rf x $rf x
  acq $par(np) 2
}

proc main {} {
  global par

  fsave [fsimpson] $par(name).fid
}
```

Example 4: POST-C7 Heteronuclear Dipolar Recoupling: Excitation Curve

```
spinsys {
  channels 13C 15N
  nuclei 13C 15N
  dipole 1 2 1300 0 0 0
}

par {
  spin_rate       8000
  sw              spin_rate/2.0
  np              64
  crystal_file    bcr40
  gamma_angles    40
  start_operator  I1z
  detect_operator -I2z
}

proc pulseq {} {
  global par

  maxdt 3.0

  set rf  [expr 7.0*$par(spin_rate)]
  set t90 [expr 0.25e6/$rf]

  for {set i 0} {$i < 7} {incr i} {
    set ph [expr $i*360.0/7.0]
    pulse $t90 $rf $ph $rf $ph
    pulse [expr 4.0*$t90] $rf \
          [expr $ph+180] $rf [expr $ph+180]
    pulse [expr 3.0*$t90] $rf $ph $rf $ph
  }
  store 1
  acq $par(np) 1
}

proc main {} {
  global par

  fsave [fsimpson] $par(name).fid
}
```

Example 5: Heteronuclear Decoupling in Multiple-Spin Systems Using CW and TPPM

```
spinsys {
  channels 1H 13C
  nuclei 13C 1H 1H 1H
  shift 2   0 2000 0    0.0 0 0
  shift 3 200 2000 0  109.5 0 0
  shift 4 500 2000 0 -100.9 0 0
  dipole 1 2 -23300 0    0.0 0
  dipole 1 3 -23300 0  109.5 0
  dipole 1 4  -3040 0 -101.0 0
  dipole 2 3 -21300 0  144.7 0
  dipole 2 4  -6900 0 -125.3 0
  dipole 3 4  -3880 0  -91.0 0
}
```

```
par {
  method              gcompute
  np                  8192
  crystal_file        rep168
  gamma_angles        16
  start_operator      I1x
  detect_operator     I1p
  variable rf         160000
  variable flip       180
  variable tp         1.0e6*flip/(rf*360.0)
  variable spin_want  5000
  variable n          round(0.5e6/(tp*spin_want))
  variable cycle      (n-1)/gamma_angles+1
  sw                  0.5e6/(cycle*tp)
  spin_rate           sw/gamma_angles
}

proc pulseq {} {
  global par

  maxdt 1
  for {set i 1} {$i <= $par(cycle)} {incr i} {
    pulse $par(tp) $par(rf)  $par(ph) 0 0
    pulse $par(tp) $par(rf) -$par(ph) 0 0
  }
}

proc main {} {
  global par

  puts "Actual spin_rate = $par(spin_rate) Hz"

  foreach p {{cw 0} {tppm 25}} {
    set type    [lindex $p 0]
    set par(ph) [lindex $p 1]

    set f [fsimpson]
    faddlb $f 10 0
    fzerofill $f 32768
    fft $f
    fsave $f $par(name)-$type.spe
    funload $f
  }
}
```

Example 6: SLF Experiments without and with MSHOT-3 and FSLG Homonuclear Decoupling

```
spinsys {
  channels 1H 15N
  nuclei 1H 1H 15N
  dipole 1 2 -26700 0 35.2 0
  dipole 1 3  11800 0  0   0
  dipole 2 3  11800 0 70.4 0
  shift 1  500 0 0 0 0 0
  shift 2 -500 0 0 0 0 0
}

par {
  spin_rate           0
  np                  1024
  crystal_file        alpha30beta65
  gamma_angles        1
  start_operator      I3x
  detect_operator     I3p
  rotor_angle         0
  variable tau        10500
  variable rf         60000
  variable rfdec      150000
  variable mshot_fac  10.125
  sw                  rf/mshot_fac
}

proc pulseq {} {
  global par

  set tsw [expr 1.0e6/$par(sw)]

  if {$par(type) == "none"} {
    set rf 0
    reset
    delay $tsw
    store 1

  } elseif {$par(type) == "fslg"} {
    set tp  [expr 1.0e6*sqrt(2.0/3.0)/$par(rf)]
    set n   [expr round($tsw/$tp/2.0)]
    if {$n == 0} { set n 1 }
    set tp  [expr $tsw/$n/2.0]
    set rf  [expr 1.0e6*sqrt(2.0/3.0)/$tp]
    set off [expr -$rf*sqrt(0.5)]
    reset
    for {set i 1} {$i <= $n} {incr i} {
      offset $off 0
      pulse $tp $rf y 0 0
      offset [expr -$off] 0
      pulse $tp $rf -y 0 0
      offset 0 0
    }
    store 1

  } elseif {$par(type) == "mshot"} {
    set rf  $par(rf)
    set tp  [expr $tsw/40.5]
    set td  [expr 1.75*$tp]
    set tp4 [expr 4.0*$tp]
    for {set i 1} {$i <= 3} {incr i} {
      set ph [expr 120*($i-1)]
      delay $td
      pulse $tp  $rf [expr ( 90+$ph) % 360] 0 0
      pulse $tp4 $rf [expr (  0+$ph) % 360] 0 0
      pulse $tp4 $rf [expr (180+$ph) % 360] 0 0
      pulse $tp  $rf [expr (270+$ph) % 360] 0 0
      delay $td
    }
    store 1
  }
  puts "rf for $par(type) = $rf Hz"

  for {set t1 1} {$t1 <= $par(np)} {incr t1} {
    reset
    if {$t1 > 1} {
      prop 4
      prop 1
    }
    store 4
    set tdec [expr $par(tau)-$tsw*($t1-1.0)]
    pulse $tdec $par(rfdec) 0 0 0
    pulseid 2 0 0 250e3 0
    pulse $par(tau) $par(rfdec) 0 0 0
    acq
  }
}
```

```
proc main {} {
  global par

  puts "sw = $par(sw) Hz"
  foreach par(type) {none fs1g mshot} {
    set f [fsimpson]
    faddlb $f 50 0
    fzerofill $f 32768
    fft $f
    fsave $f $par(name)-$par(type).spe -binary
    funload $f
  }
}
```

Example 7: Heteronuclear Cross-Polarization Employing FSLG with Frequency Switching or Phase Modulation

```
spinsys {
  channels 1H 15N
  nuclei 1H 15N
  dipole 1 2 11800 0 0 0
  shift 1 0  10p 0.5 0 0 0
  shift 2 0 100p 0.5 0 0 0
}

par {
  spin_rate        0
  np               32
  crystal_file     rep168
  proton_frequency 400e6
  start_operator   I1z
  detect_operator  I2p
  variable rf      60000
  sw               rf*sqrt(3.0/2.0)/2.0
  variable theta   54.7356103172
}

proc pulseq {} {
  global par

  set rf2 [expr sqrt(3.0/2.0)*$par(rf)]
  set off [expr -sqrt(1.0/2.0)*$par(rf)]
  reset
  if {$par(steps) == "offset"} {
    set tp [expr 1.0e6*sqrt(2.0/3.0)/$par(rf)]
    offset $off 0
    pulse $tp $par(rf) y $rf2 x
    offset [expr -$off] 0
    pulse $tp $par(rf) -y $rf2 -x
    offset 0 0

  } else {
    set step [expr 208/$par(steps)]
    set dmf [expr sqrt(3.0/2.0)* \
        $par(steps)*$par(rf)]
    set tp [expr 1.0e6/$dmf]
    for {set ph $step} \
          {$ph <= 208} {incr ph $step} {
      pulse $tp $par(rf) \
          [expr (90+$ph) % 360] $rf2  X
    }
    for {set ph [expr 388-$step]} \
          {$ph >= 180} {incr ph -$step} {
      pulse $tp $par(rf) \
          [expr (90+$ph) % 360] $rf2 -X
    }
  }
  store 1

  reset
  pulse [expr $par(theta)/90.0* \
      0.25e6/$par(rf)] $par(rf) x 0 x
  acq $par(np) 1 -x
}

proc main {} {
  global par

  foreach par(steps) {offset 8 13 16 26 \
        52 104 208} {
    set f [fsimpson]
    fsave $f $par(name)-$par(steps).fid
    funload $f
  }
}
```

Example 8: 2D ^{1}H–^{15}N PISEMA SLF Experiment

```
spinsys {
  channels 1H 15N
  nuclei 1H 15N
  dipole 1 2 10000 0 0 0
  shift 2 100p 100p 0.25 0 17 0
}

par {
  spin_rate        0
  crystal_file     zcw4180
  start_operator   I2x-I1y
  detect_operator  I2p
  proton_frequency 400e6
  verbose          1101
  np               128
  ni               128
  variable rf      83000
  variable dec     130000
  sw               40000
  sw1              rf/2.0/sqrt(2.0/3.0)
  variable theta   90-54.73561032
}

proc pulseq {} {
  global par

  set tsw [expr 1e6/$par(sw)]
  set tp  [expr 1e6*sqrt(2.0/3.0)/$par(rf)]
  set tth [expr 1e6*$par(theta)/360/$par(rf)]
  set off [expr -sqrt(0.5)*$par(rf)]
  set rf2 [expr sqrt(1.5)*$par(rf)]

  reset
  offset $off 0
  pulse $tp $par(rf) y $rf2 x
  offset [expr -$off] 0
  pulse $tp $par(rf) -y $rf2 -x
  offset 0 0
  store 1

  reset
  pulse $tsw $par(dec) x 0 x
  store 2

  for {set i 1} {$i <= $par(ni)} {incr i} {
    reset
```

```
    if {$i == 1} {
      pulse $tth $par(rf) -x 0 x
    } else {
      prop 3
      prop 1
    }
    store 3
    acq $par(np) 2
  }
}

proc main {} {
  global par

  set f [fsimpson]
  fsave $f $par(name).fid -binary
  fzerofill $f 512 512
  faddlb $f 300 1 300 1
  fft $f 0 0 0 0
  fsave $f $par(name).spe -binary
  fplot2d $f $par(name).ppm -ppm
}
```

Example 9: 2D ^{13}C, ^{13}C Chemical-Shift Correlation in a Five-Spin-1/2 System Using RFDR Dipolar Recoupling

```
spinsys {
  channels 13C
  nuclei 13C 13C 13C 13C 13C
  shift 1 -80p 30p 0.1  10 -50  10
  shift 2 -40p 20p 0.2  80  20 -30
  shift 3   0p 40p 0.7 -80 120 -40
  shift 4  40p 30p 0.4  10  50 -30
  shift 5  80p 20p 0.5  60 -40 -90
  dipole 1 2 -2250 0  0 0
  dipole 2 3 -2250 0 72 0
  dipole 3 4 -2250 0  0 0
  dipole 4 5 -2250 0 72 0
  dipole 1 3 -530  0 36 0
  dipole 2 4 -530  0 36 0
  dipole 3 5 -530  0 36 0
  dipole 1 4 -148  0 22 0
  dipole 2 5 -148  0 50 0
}

par {
  spin_rate        20000
  proton_frequency 400e6
  crystal_file     rep30
  gamma_angles     5
  start_operator   Inz
  detect_operator  Inp
  ni               64
  np               32
  sw               spin_rate
  sw1              spin_rate
  variable n       6
  verbose          1101
}

proc pulseq {} {
  global par

  mxdt 1.0

  set tr  [expr 1.0e6/$par(spin_rate)]
  set tr2 [expr $tr/2.0]

  reset
  delay $tr2
  pulseid 2 250000 x
  delay $tr2
  store 1

  reset
  delay $tr2
  pulseid 2 250000 y
  delay $tr2
  store 2

  reset
  foreach i {1 2 1 2 2 1 2 1} {
    prop $i
  }
  store 8

  reset
  pulseid 1 250000 x
  prop 8 $par(n)
  pulseid 1 250000 -y
  store 8

  reset
  delay [expr 1.0e6/$par(sw)]
  store 6

  for {set i 0} {$i < $par(ni)} {incr i 2} {
    reset
    if {$i > 0} {
      prop 1
      prop 6
    }
    store 1

    foreach ph {x -y} {
      reset
      pulseid 1 250000 $ph
      prop 1
      prop 8
      acq $par(np) 6
    }
  }
}

proc main {} {
  global par

  set f [fsimpson]
  fsave $f $par(name).fid -binary
  fzerofill $f 512 512
  faddlb $f 20 0 20 0 -phsens
  fft $f 0 0 0 0 -phsens
  fplot2d $f $par(name).ppm -ppm
  fsave $f $par(name).spe -binary
}
```

Example 10: Sensitivity-Enhanced Quadrupolar-Echo NMR: The QCPMG-MAS Experiment

```
spinsys {
  channels 87Rb
  nuclei 87Rb
  shift 1 0 -150p 0.60 90 30 90
  quadrupole 1 2 10e6 0.12 0 0 0
}
```

```
par {
  spin_rate         9506
  sw                26*spin_rate
  crystal_file      zcw4180
  gamma_angles      20
  start_operator    I1z
  detect_operator   I1c
  proton_frequency  400e6
  verbose           1101
  variable rf       64102.6
  variable n        124
  variable r        30
  variable nsync    5
  variable t2add    1.20

  np                2*n*(1+r)+n
  variable n2       2*n
  variable tsw      1.0e6/sw
  variable tr       1.0e6/spin_rate
  variable nprop    round(tr/tsw)
  variable t90      0.25e6/rf/2.0
  variable t180     0.5e6/rf/2.0
  variable t1       tr-t90
  variable t2       t1+t2add
  variable ta       2.0*n*tsw
  variable t3       (2*nsync*tr-ta-t180)/2.0
}

proc pulseq {} {
  global par

  maxdt 0.4

  matrix set 1 coherence {1 -1}

  pulse $par(t90) $par(rf) x
  filter 1
  delay $par(t1)
  pulse $par(t180) $par(rf) y
  filter 1
  delay $par(t2)

  for {set n 1} {$n <= $par(n)} {incr n} {
    acq
    if [expr $n > $par(nprop)] {
      prop [expr (($n-1) % $par(nprop))+1]
    } else {
      delay $par(tsw)
      store $n
    }
  }
  delay $par(t3)
  pulse $par(t180) $par(rf) y
  filter 1
  delay $par(t3)

  for {set n 1} {$n <= $par(n2)} {incr n} {
    acq
    if [expr $n > $par(nprop)] {
      prop [expr (($n-1) % $par(nprop))+1]
    } else {
      delay $par(tsw)
      store $n
    }
  }
}

proc main {} {
  global par

  set f [fsimpson]

  for {set i 1} {$i <= $par(n2)} {incr i} {
    set c  [findex $f [expr $i + $par(n)]]
    set re [lindex $c 0]
    set im [lindex $c 1]
    for {set j 1} {$j <= $par(r)} {incr j} {
      fsetindex $f \
        [expr $i+$j*$par(n2)+$par(n)] $re $im
    }
  }
  fsave $f $par(name).fid
  fzerofill $f 32768
  faddlb $f 20 0
  fft $f
  fphase $f -rp 90
  fsave $f $par(name).spe -binary
  fsave [fssbint $f 1000 0 1000] $par(name)-int.spe
}
```

Example 11: MQ-MAS NMR of Half-Integer Quadrupolar Nuclei

```
spinsys {
  channels 23Na
  nuclei 23Na
  quadrupole 1 2 1.2e6 0.6 0 0 0
}

par {
  spin_rate         8000
  variable tsw      0.5
  sw                1.0e6/tsw
  np                81
  crystal_file      rep320
  gamma_angles      20
  start_operator    I1z
  proton_frequency  400e6
  variable rf       80000
}

proc pulseq {} {
  global par

  maxdt 0.5

  matrix set detect coherence {-3}
  acq
  for {set i 1} {$i < $par(np)} {incr i} {
    pulse $par(tsw) $par(rf) -y
    acq
  }
}

proc main {} {
  global par

  fsave [fsimpson] $par(name).fid
}
```

ACKNOWLEDGMENTS

The authors thank Dr. H. Bildsøe, Dr. F. H. Larsen, and L. Odgaard for constructive comments. The initial parts of the project were conducted using the facilities at the Department of Chemistry, University of Aarhus.

REFERENCES

1. U. Haeberlen, "High-Resolution NMR in Solids. Selective Averaging," Academic Press, New York, 1976.
2. M. Mehring, "Principles of High Resolution NMR in Solids," Springer-Verlag, New York, 1983.
3. H. W. Spiess, "Rotations of Molecular and Nuclear Spin Relaxation, NMR Basic Principles and Progress," Vol. 15, Springer-Verlag, Berlin, 1978.
4. B. C. Gerstein and C. R. Dybowski, "Transient Techniques in NMR of Solids. An Introduction to Theory and Practice," Academic Press, Orlando, 1985.
5. M. Munowitz, "Coherence and NMR," Wiley, New York, 1988.
6. K. Schmidt-Rohr and H. W. Spiess, "Multidimensional Solid-State NMR and Polymers," Academic Press, London, 1996.
7. A. E. Bennett, R. G. Griffin, and S. Vega, Recoupling of homo- and heteronuclear dipolar interactions in rotating solids, *in* "NMR Basic Principles and Progress," Vol. 33, p. 1, Springer-Verlag, Berlin, 1994.
8. S. Dusold and A. Sebald, Dipolar recoupling under magic-angle spinning conditions, *Ann. Rep. NMR Spectrosc.* **41,** 185–264 (2000).
9. R. R. Ernst, G. Bodenhausen, and A. Wokaun, "Principles of Nuclear Magnetic Resonance in One and Two Dimensions," Clarendon, Oxford, 1987.
10. K. Wüthrich, "NMR of Proteins and Nucleic Acids," Wiley, New York, 1986.
11. J. Cavanagh, W. J. Fairbrother, A. G. Palmer, III, and N. J. Shelton, "Protein NMR Spectroscopy: Principles and Practice," Academic Press, San Diego, 1996.
12. U. Haeberlen and J. S. Waugh, Coherent averaging effects in magnetic resonance, *Phys. Rev.* **175,** 453–467 (1968).
13. A. Schmidt and S. Vega, The Floquet theory of nuclear magnetic resonance spectroscopy of single spins and dipolar coupled spin pairs in rotating solids, *J. Chem. Phys.* **96,** 2655–26800 (1992).
14. M. Hohwy and 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,** 3780–3791 (1998).
15. J. H. Kristensen, H. Bildsøe, H. J. Jakobsen, and N. C. Nielsen, Application of Lie algebra to NMR spectroscopy, *Progr. NMR Spectrosc.* **34,** 1–69 (1999).
16. D. P. Raleigh, M. H. Levitt, and R. G. Griffin, Rotational resonance solid state NMR, *Chem. Phys. Lett.* **146,** 71–76 (1988).
17. M. H. Levitt, D. P. Raleigh, F. Creuzet, and R. G. Griffin, Theory and simulations of homonuclear spin pair systems in rotating solids, *J. Chem. Phys.* **92,** 6347–6364 (1990).
18. T. Gullion and J. Schaefer, Rotational-echo double-resonance NMR, *J. Magn. Reson.* **81,** 196–200 (1989).
19. R. Tycko and G. Dabbagh, Measurement of nuclear magnetic dipole–dipole couplings in magic angle spinning NMR, *Chem. Phys. Lett.* **173,** 461–465 (1990).
20. D. M. Gregory, D. J. Mitchell, J. A. Stringer, S. Kiihne, J. C. Shiels, J. Callahan, M. A. Mehta, and G. P. Drobny, Windowless dipolar recoupling: The detection of weak dipolar interactions between spin 1/2 nuclei with large chemical shift anisotropies, *Chem. Phys. Lett.* **246,** 654–663 (1995).
21. A. E. Bennett, J. H. Ok, R. G. Griffin, and S. Vega, Chemical shift correlation spectroscopy in rotating solids: Radio frequency driven dipolar recoupling and longitudinal exchange, *J. Chem. Phys.* **96,** 8624–8627 (1992).
22. M. Baldus, M. Tomasselli, B. H. Meier, and R. R. Ernst, Broadband polarization-transfer experiments for rotating solids, *Chem. Phys. Lett.* **230,** 329–336 (1994).
23. N. C. Nielsen, H. Bildsøe, H. J. Jakobsen, and M. H. Levitt, Double-quantum homonuclear rotary resonance: Efficient dipolar recovery in magic-angle spinning nuclear magnetic resonance, *J. Chem. Phys.* **101,** 1805–1812 (1994).
24. W. Sommer, J. Gottwald, D. E. Demco, and H. W. Spiess, Dipolar heteronuclear multiple-quantum NMR spectroscopy, *J. Magn. Reson. A* **113,** 131–134 (1995).
25. Y. K. Lee, N. D. Kurur, M. Helmle, O. G. Johannessen, N. C. Nielsen, and M. H. Levitt, Efficient dipolar recoupling in the NMR of rotating solids. A sevenfold symmetric radiofrequency pulse sequence, *Chem. Phys. Lett.* **242,** 304–309 (1995).
26. M. Hohwy, H. J. Jakobsen, M. Edén, M. H. Levitt, and N. C. Nielsen, Broadband dipolar recoupling in the nuclear magnetic resonance of rotating solids: A compensated C7 pulse sequence, *J. Chem. Phys.* **108,** 2686–2694 (1998).
27. B. Q. Sun, P. R. Costa, and R. G. Griffin, Heteronuclear polarization transfer by radiofrequency-driven dipolar recoupling under magic angle spinning, *J. Magn. Reson. A* **112,** 191–198 (1995).
28. D. P. Burum and W.-K. Rhim, Analysis of multiple pulse NMR in solids. III, *J. Chem. Phys.* **71,** 944–956 (1979).
29. A. Bielecki, A. C. Kolbert, and M. H. Levitt, Frequency-switched pulse sequences: Homonuclear decoupling and dilute spin NMR in solids, *Chem. Phys. Lett.* **155,** 341–346 (1989).
30. M. Hohwy and N. C. Nielsen, Elimination of high order terms in multiple pulse nuclear magnetic resonance spectroscopy: Application to homonuclear decoupling in solids, *J. Chem. Phys.* **106,** 7571–7586 (1997).
31. A. E. Bennett, C. M. Rienstra, M. Auger, K. V. Lakshmi, and R. G. Griffin, Heteronuclear decoupling in rotating solids, *J. Chem. Phys.* **103,** 6951–6958 (1995).
32. A. Pines, Gibby, and J. S. Waugh, Proton-enhanced NMR of dilute spins in solids, *J. Chem. Phys.* **59,** 569–590 (1973).
33. C. H. Wu, A. Ramamoorthy, and S. J. Opella, High-resolution heteronuclear dipolar solid-state NMR spectroscopy, *J. Magn. Reson. A* **109,** 270–272 (1994).
34. F. H. Larsen, H. J. Jakobsen, P. D. Ellis, and N. C. Nielsen, High-field QCPMG-MAS NMR of half-integer quadrupolar nuclei with large quadrupole couplings, *Mol. Phys.* **95,** 1185–1195 (1998).
35. L. Frydman and J. S. Harwood, Isotropic spectra of half-integer quadrupolar spins from bidimensional magic-angle spinning NMR, *J. Am. Chem. Soc.* **117,** 5367–5368 (1995).
36. F. Stickney de Bouregas and J. S. Waugh, ANTIOPE, A program for computer experiments on spin dynamics, *J. Magn. Reson.* **96,** 280–289 (1992).
37. S. A. Smith, T. O. Levante, B. H. Meier, and R. R. Ernst, Computer simulations in magnetic resonance. An object-oriented programming approach, *J. Magn. Reson.* **106,** 75–105 (1994).
38. H. Bildsøe, "STARS User's Guide. Spectrum Analysis for Rotating Solids," Publ. No. 87-195233-00, Rev. A0296, Varian Associates Inc., Palo Alto, 1996.
39. J. Skibsted, N. C. Nielsen, H. Bildsøe, and H. J. Jakobsen, Satellite transitions in MAS NMR spectra of quadrupolar nuclei, *J. Magn. Reson.* **95,** 88–117 (1991).
40. J.-P. Amoureux, personal communication.
41. B. B. Welch, "Practical Programming in Tcl and Tk," Prentice Hall, Englewood Cliffs, NJ, 1995.

42. J. K. Ousterhout, "Tcl and the Tk Toolkit," Addison-Wesley, Reading, MA, 1994.
43. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Wetterling, "Numerical Recipes," Cambridge Univ. Press, Cambridge, 1986.
44. M. H. Levitt, The signs of frequencies and phases in NMR, *J. Magn. Reson.* **126,** 164–182 (1997).
45. M. Edén and M. H. Levitt, Computation of orientational averages in solid-state NMR by gaussian spherical quadrature, *J. Magn. Reson.* **132,** 220–239 (1998).
46. S. K. Zaremba, Good lattice points, discrepancy, and numerical integration, *Ann. Mat. Pure. Appl.* **4–73,** 293 (1966).
47. H. Conroy, Molecular Schrödinger equation. VIII. A new method for the evaluation of multidimensional integrals, *J. Chem. Phys.* **47,** 5307–5318 (1967).
48. V. B. Cheng, H. H. Suzukawa, Jr., and M. Wolfsberg, Investigations of a nonrandom numerical method for multidimensional integration, *J. Chem. Phys.* **59,** 3992–3999 (1973).
49. M. Bak and N. C. Nielsen, REPULSION, A novel approach to efficient powder averaging in solid-state NMR, *J. Magn. Reson.* **125,** 132–139 (1997).
50. D. W. Alderman, M. S. Solum, and D. M. Grant, Methods for analyzing spectroscopic line shapes. NMR solid powder patterns, *J. Chem. Phys.* **84,** 3717–3725 (1986).
51. M. Edén, Y. K. Lee, and M. H. Levitt, Efficient simulation of periodic problems in NMR. Application to decoupling and rotational resonance, *J. Magn. Reson. A* **120,** 56–71 (1996).
52. T. Charpentier, C. Fermon, and J. Virlet, Efficient time propagation technique for MAS NMR simulation: Application to quadrupolar nuclei, *J. Magn. Reson.* **132,** 181–190 (1998).
53. M. H. Levitt and M. Edén, Numerical simulation of periodic nuclear magnetic resonance problems: Fast calculation of carousel averages, *Mol. Phys.* **95,** 879 (1998).
54. M. Hohwy, H. Bildsøe, H. J. Jakobsen, and N. C. Nielsen, Efficient spectral simulations in NMR of rotating solids. The γ-COMPUTE algorithm, *J. Magn. Reson.* **136,** 6–14 (1999).
55. M. Bak, J. T. Rasmussen, and N. C. Nielsen, Web address and download site for the SIMPSON program: http://nmr.imsb.au.dk.
56. The GNUPLOT plotting program. Web site: http://www.gnuplot.org.
57. The GNU General Public License is available from the Free Software Foundation, Boston, MA, at the web address: http://www.gnu.org.
58. The GTK widget set. Web site: http://www.gtk.org.
59. A. A. Maudsley, Modified Carr–Purcell–Meiboom–Gill sequence for NMR Fourier imaging applications, *J. Magn. Reson.* **69,** 488–491 (1986).
60. T. Gullion, D. B. Baker, and M. S. Conradi, New, compensated Carr–Purcell sequences, *J. Magn. Reson.* **89,** 479–484 (1990).
61. M. Maricq and J. S. Waugh, NMR in rotating solids, *J. Chem. Phys.* **70,** 3300–3316 (1979).
62. N. C. Nielsen, F. Creuzet, R. G. Griffin, and M. H. Levitt, Enhanced double-quantum nuclear magnetic resonance in spinning solids at rotational resonance, *J. Chem. Phys.* **96,** 5668–5677 (1992).
63. M. Bak and N. C. Nielsen, Relative orientation of chemical shielding and dipolar coupling tensors: Mixed single- and double-quantum homonuclear rotary resonance nuclear magnetic resonance of rotating solids, *J. Chem. Phys.* **106,** 7587–7599 (1997).
64. A. W. Hing, S. Vega, and J. Schaefer, Transferred-echo double-resonance NMR, *J. Magn. Reson.* **96,** 205–209 (1992).
65. A. W. Hing, S. Vega, and J. Schaefer, Measurement of heteronuclear dipolar coupling by transferred-echo double resonance NMR, *J. Magn. Reson. A* **103,** 151–162 (1993).
66. M. Ernst, S. Bush, A. C. Kolbert, and A. Pines, Second-order recoupling of chemical-shielding and dipolar-coupling tensors under spin decoupling in solid-state NMR, *J. Chem. Phys.* **105,** 3387–3397 (1996).
67. Z. Gan and R. R. Ernst, Frequency- and phase-modulated heteronuclear decoupling in rotating solids, *Solid State NMR* **8,** 153–159 (1997).
68. D. L. Vanderhart and G. C. Campbell, Off-resonance proton decoupling on-resonance and near-resonance. A close look at ^{13}C CPMAS linewidth in solids for rigid, strongly coupled carbons under cw proton decoupling, *J. Magn. Reson.* **134,** 88–112 (1998).
69. M. Edén and M. H. Levitt, Pulse sequence symmetries in the nuclear magnetic resonance of spinning solids: Application to heteronuclear decoupling, *J. Chem. Phys.* **111,** 1511–1519 (1999).
70. R. K. Hester, J. L. Ackerman, V. R. Cross, and J. S. Waugh, Resolved dipolar coupling spectra of dilute nuclear spins in solids, *Phys. Rev. Lett.* **34,** 993–995 (1975).
71. E. Vinogradov, P. K. Madhu, and S. Vega, High-resolution proton solid-state NMR spectroscopy by phase-modulated Lee–Goldburg experiment, *Chem. Phys. Lett.* **314,** 443 (1999).
72. Z. Gan, Spin dynamics of polarization inversion spin exchange at the magic angle in multiple spin systems, *J. Magn. Reson.* **143,** 136–143 (2000).
73. R. Fu, P. Pelupessy, and G. Bodenhausen, Frequency-modulated cross-polarization for fast magic-angle spinning NMR at high fields, *Chem. Phys. Lett.* **264,** 63–69 (1997).
74. K. Takegoshi, K. Takeda, and T. Terao, Modulatory resonance recoupling of heteronuclear dipolar interactions under magic angle spinning, *Chem. Phys. Lett.* **260,** 331–335 (1996).
75. R. Fu, S. A. Smith, and G. Bodenhausen, Recoupling of heteronuclear dipolar interactions in solid state magic-angle spinning NMR by simultaneous frequency and amplitude modulation, *Chem. Phys. Lett.* **272,** 361–369 (1997).
76. A. P. M. Kentgens and R. Verhagen, Advantages of double frequency sweeps in static, MAS and MQMAS NMR of spin $I = 3/2$ nuclei, *Chem. Phys. Lett.* **300,** 435–443 (1999).
77. P. K. Madhu, A. Goldbourt, L. Frydman, and S. Vega, Sensitivity enhancement of the MQMAS NMR experiment by fast amplitude modulation of the pulses, *Chem. Phys. Lett.* **307,** 41–47 (1999).
78. B.-J. van Rossum, H. Förster, and H. J. M. de Groot, High-field and high-speed CP-MAS ^{13}C NMR heteronuclear dipolar-correlation spectroscopy of solids with frequency-switched Lee–Goldburg homonuclear decoupling, *J. Magn. Reson.* **124,** 516–519 (1997).
79. A. Ramamoorthy, C. H. Wu, and S. J. Opella, Experimental aspects of multidimensional solid-state NMR correlations spectroscopy, *J. Magn. Reson.* **140,** 131–140 (1999).
80. A. Ramamoorthy and S. J. Opella, Two-dimensional chemical shift/heteronuclear dipolar coupling spectra obtained with cross polarization spin exchange at the magic angle and magic-angle sample spinning (PISEMAMAS), *Solid state NMR* **4,** 387–392 (1995).
81. C. H. Wu, A. Ramamoorthy, L. M. Gierasch, and S. J. Opella, Simultaneous characterization of the amide ^{1}H chemical shift, ^{1}H–^{15}N dipolar, and ^{15}N chemical shift interaction tensors in a peptide bond by three-dimensional solid-state NMR spectroscopy, *J. Am. Chem. Soc.* **117,** 6148–6149 (1995).
82. Z. T. Gu and S. J. Opella, Two- and three-dimensional $^{1}H/^{13}C$ PISEMA experiments and their application to backbone and side chain sites of amino acids and peptides, *J. Magn. Reson.* **140,** 340–346 (1999).

83. T. A. Egorova-Zachernyuk, B. van Rossum, G.-J. Boender, E. Franken, J. Ashurst, J. Raap, P. Gast, A. J. Hoff, H. Oschkinat, and H. J. M. de Groot, Characterization of pheophytin ground states in *Rhodobacter sphaeroides* R26 photosynthetic reaction centers from multispin pheophytin enrichment and 2-D ^{13}C MAS dipolar correlation spectroscopy, *Biochemistry* **36,** 7513–7519 (1997).

84. J. M. Griffith, A. E. Bennett, M. Engelhard, F. Siebert, J. Rapp, J. Lugtenburg, J. Herzfeld, and R. G. Griffin, Structural investigation of the active site in bacteriorhodopsin: Geometric constraints on the roles of Asp-85 and Asp-212 in the proton-pumping mechanism from solid state NMR, *Biochemistry* **39,** 362–371 (2000).

85. H. Y. Carr and E. M. Purcell, Effects of diffusion on free precession in nuclear magnetic resonance experiments, *Phys. Rev.* **94,** 630 (1954).

86. S. Meiboom and D. Gill, Modified spin-echo method for measurement of nuclear relaxation times, *Rev. Sci. Instrum.* **29,** 688–691 (1958).

87. F. H. Larsen, H. J. Jakobsen, P. D. Ellis, and N. C. Nielsen, Sensitivity-enhanced quadrupolar-echo NMR of half-integer quadrupolar nuclei. Magnitudes and relative orientation of chemical shielding and quadrupolar coupling tensors. *J. Chem. Phys. A* **101,** 8597–8606 (1997).

88. G. Wu, D. Rovnyank, B. Sun, and R. G. Griffin, High-resolution multiple quantum MAS NMR spectroscopy of half-integer quadrupolar nuclei, *Chem. Phys. Lett.* **249,** 210–217 (1996).