

## QSim, a program for NMR simulations

Magnus Helgstrand<sup>a</sup> & Peter Allard<sup>b</sup>

<sup>a</sup>Lund University, Box 124, Department of Biophysical Chemistry, SE-22100 Lund, Sweden; <sup>b</sup>The Royal Institute of Technology, AlbaNova University Center, Department of Biotechnology, SE-106 91 Stockholm, Sweden

Received 1 March 2004; Accepted 17 May 2004

**Key words:** chemical exchange NMR, effective Liouvillian, hermitian operator basis, homogeneous master equation, NMR simulations, stochastic Liouvillian, superoperator algebra

### Abstract

We present QSim, a program for simulation of NMR experiments. Pulse sequences are implemented and analyzed in QSim using a mouse driven interface. QSim can handle almost any modern NMR experiment, using multiple channels, shaped pulses, mixing, decoupling, phase-cycling and pulsed field gradients. Any number of spins with any spin quantum number can, in theory, be used in simulations. Relaxation is accounted for during all steps of pulse sequences and relaxation interference effects are supported. Chemical kinetics between any numbers of states can be simulated. Both classical and quantum mechanical calculations can be performed. The result of a simulation can be presented either as magnetization as a function of time or as a processed spectrum.

### Introduction

It is becoming increasingly difficult to write and understand modern NMR pulse sequences due to the utilization of complicated physical effects, examples include the TROSY (Pervushin et al., 1997) effect and the study of chemical dynamics using spin-lock experiments (Deverell et al., 1970; Akke and Palmer, 1996). Helpful tools such as the product operator formalism (Sørensen et al., 1983; van de Ven and Hilbers, 1983; Güntert et al., 1993) fails to completely describe these modern experiments. It is thus becoming increasingly evident that new tools are required to calculate the results of the most complicated pulse sequences without unnecessary simplifications and approximations.

Quite a few software packages for simulation of NMR experiments have previously been presented, such as SPINX (Widmer and Wüthrich, 1986), SMART (Studer, 1988), GAMMA (Smith et al., 1994a), SIMPLTN (Allman et al., 1996), 'The Virtual NMR Spectrometer' (Nicholas et al., 2000), PJNMR (Letourneau et al., 2003) and BlochLib (Blanton, 2003). We present QSim, which has the ambition to both be easy to use and to accurately simulate almost all kinds of modern liquid state NMR experiments.

QSim is written for PC's running Microsoft Windows and has a graphical user interface (GUI) similar to other Windows programs. Simulations in QSim are set up by graphically constructing a pulse sequence and after that assigning variables and simulation parameters. Spin systems can either be imported from a standard set or constructed directly in QSim. The result of the simulation can either be represented directly as a spectrum within QSim or exported in a text format suitable for further analysis in Matlab (Moler, 2002).

The computational part of QSim is named HME and available as source code for more advanced users. HME is a set of C++ classes and functions for simulating NMR experiments. HME performs simulations on any number of spins with any spin quantum numbers. It performs calculations using quantum mechanics or optionally, for spin quantum numbers  $S \leq 1$ , classical mechanics. It handles first order equilibrium chemical kinetics with any number of states. It can calculate relaxation rates using chemical shift anisotropy, dipole-dipole and quadrupolar relaxation mechanisms in any combination, i.e., including all possible interference effects. The motions of the spins in the spin system are described by individual model free spectral density functions (Lipari and Szabo, 1982a,

1982b). The spectral density function is implemented as a complex function and the software is thus able to calculate the dynamic frequency shifts of quadrupolar nuclei (Werbelow and London, 1996). Relaxation and chemical exchange are always included in all parts of the simulation since the homogenous master equation is used for quantum mechanical simulations (Jeener, 1982; Levitt and Di Bari, 1992, 1994; Smith et al., 1994b; Levante and Ernst, 1995; Allard et al., 1998) and homogenous versions of the Bloch (1946), Solomon (1955) and McConnell (1958) equations are used for classical mechanical simulations (Allard et al., 1997; Helgstrand et al., 2000). HME contains support for quantum mechanical simulations of partially oriented systems using residual dipolar and quadrupolar coupling constants. A method for calculating the effect of pulsed field gradients in the z-direction is implemented (Allard et al., 1998). Decoupling sequences are simulated either as pulses or, more efficiently, as effective Liouvillians (Levitt and Di Bari, 1992, 1994; Allard et al., 1998).

## Theory

All simulations in QSim/HME are performed in the superspace (Liouville space) (Ernst et al., 1987; Jeener, 1982). Operators, such as the Hamiltonian, are described by vectors and superoperators, such as the relaxation superoperator, are described by matrices in the superspace. It is necessary to work in superspace to be able to account for relaxation during calculations. In the absence of relaxation the simulations can be performed in the Hilbert space, with smaller matrices and faster calculations. Simulations in Hilbert space are not used in QSim/HME.

Calculations in superspace are performed with a complete set of basis operators. We have chosen the Cartesian product operators extended for spin quantum numbers larger than  $S = 1/2$  (Allard and Härd, 2001). This basis set has several advantages over competing sets of basis operators. One advantage is that most NMR users are familiar with the Cartesian product operators. The flow of magnetization/coherence through a pulse sequence can be followed as signed quantities of Cartesian product operators in QSim/HME. The other main advantage with this particular set of basis operators is that all physically relevant spinoperators and superoperators are described by real vectors and matrices (Allard and Härd, 2001). Using any other set of basis operators

the matrices become complex and calculations will be slower.

The Liouville-von Neuman system of equations, i.e. the master equation, must be solved in order to calculate the dynamics of a quantum mechanical system (Abragam, 1961; Ernst et al., 1987; Goldman, 1988). In the presence of relaxation the master equation is inhomogeneous, complicating its solution. It has been shown that the master equation can be rewritten in a homogenous form without any approximations (Jeener, 1982; Levitt and Di Bari, 1992, 1994; Smith et al., 1994b; Levante and Ernst, 1995). QSim/HME uses the homogenous form of the master equation for all quantum mechanical calculations and the corresponding homogenous equations for classic mechanical calculations (Allard et al., 1997; Helgstrand et al., 2000).

The use of the homogenous master equations has a very nice property (Smith et al., 1994b), all steps in a pulse sequence, such as pulses or delays, can be described by a matrix. A single matrix can thus describe the combined effect of a sequence of pulses and delays, simply by multiplying the time ordered matrices describing the individual elements. QSim finds, calculates and stores these matrices once and uses them repeatedly in order to make simulations faster.

The size of the matrices used in the calculations is dependent on the number of spins, their quantum numbers, the size of the matrix describing kinetics and the choice of classical or quantum mechanical calculations. The number of quantum mechanical basis operators for a single spin with spin quantum number  $S$  is  $(2S + 1)^2$ . For larger spin systems these numbers are multiplied together. For 4 spin  $S = 1/2$  nuclei, the size of the matrices is  $((2 \cdot 1/2 + 1)^2)^4$  by  $((2 \cdot 1/2 + 1)^2)^4$  (= 256 by 256) elements. When performing classical mechanical simulation on  $k$  spins the size of the matrix is  $3 \cdot k + 1$  by  $3 \cdot k + 1$  elements. A classical mechanical simulation on 85 spins is thus similar in execution time to a quantum mechanical simulation on 4 spins with spin quantum numbers  $S = 1/2$ . If chemical exchange is simulated the final matrix size is even larger. If the number of kinetic states is  $m$  and the size of the original matrix is  $n$  by  $n$  elements, the size of the final matrix is  $(n - 1) \cdot m + 1$  by  $(n - 1) \cdot m + 1$  elements. It is thus, in practice, impossible to simulate spin systems larger than 4 to 5 spins, with quantum numbers  $S = 1/2$ , using quantum mechanics due to the exponentially increasing size of the matrices used during simulations.

QSim simulates relaxation during all parts of a pulse sequence, i.e., both during delays and pulses. The relaxation mechanisms implemented in QSim/HME are dipole-dipole relaxation, relaxation due to anisotropy of shielding and quadrupolar relaxation (Abragam, 1961). All interaction tensors are assumed axially symmetric. QSim/HME can calculate all possible interference effects between the different relaxation mechanisms.

The input required by QSim/HME are pair-wise distances, the shielding constants for the parallel and perpendicular direction of the shielding tensor and the nuclear quadrupolar coupling constant for relaxation due to dipole-dipole interactions, anisotropy of shielding and quadrupolar relaxation, respectively. The angle between the unique axes of the interaction tensors must also be supplied in case of interference relaxation.

The dynamics in the spin system is described by the Lipari–Szabo approach (Lipari and Szabo, 1982a, 1982b). All interactions causing relaxation are described by an individual rotational correlation time, a correlation time of internal motions, and an order parameter squared.

The homogeneous master equation makes it possible to formulate a numerical effective Liouvillian (Allard et al., 1998), similar to the average Liouvillian (Levitt and Di Bari, 1992, 1994; Ghose, 2000). Each segment of a pulse sequence is described by a matrix which is the exponent of  $-L*\Delta t$ , where  $L$  is the Liouvillian and  $\Delta t$  the time of the segment. These matrices can be multiplied together in order to describe the effect of a larger pulse sequence with a single matrix. If the logarithm of this single matrix is taken, and then divided with  $-1$  times the total time of the pulse sequence, the effective Liouvillian is obtained. If, for instance, the effective Liouvillian of a decoupling sequence is calculated, it would be expected that the matrix elements in the effective Liouvillian corresponding to scalar couplings would be close to zero, while all other matrix elements would be close to the Liouvillian for a delay.

It should be noted that the numerical effective Liouvillian approach has certain limitations. Resonance lines that evolve more than  $\pi$  radians during the total time of the pulse sequence will be folded and will appear at unexpected positions. The total time over which an effective Liouvillian is calculated should be kept short compared to the inverse of the largest interaction of interest, or more accurately eigenvalue, in order to avoid such problems.

We have implemented pulsed field gradients according to a previously discussed method (Allard et al., 1998). The effect of a pulsed field gradient is a rotation around the  $z$ -axis, identical to the effect of a chemical shift. When the PFG is applied in a pulse sequence a term proportional to the PFG field strength and the magnetogyric ratio is added to all chemical shift terms. Only a single value of  $z$  can be used at a time in the numerical calculations, corresponding to a single slice in the sample tube. The calculations are therefore repeated with linearly spaced values of  $z$  and the normalized results are added together. In this way integration over the height of the sample is performed. The number of slices required is discussed by Meresi et al. (1999). Successful implementation of pulsed field gradients requires that desired magnetizations have the same phase at the beginning of the acquisition and are added constructively while unwanted magnetizations have different phases and are added destructively.

### Spin Hamiltonian

The type of interaction Hamiltonians used in the simulations are important when distinction between homonuclear and heteronuclear scalar couplings or residual dipolar couplings must be made.

The multiple rotating frame spin Hamiltonian is assumed and used in all calculations. The Zeeman Hamiltonian,  $H$ , is given by (Ernst et al., 1987)

$$H = \sum_i \Omega_i I_{iz}, \quad (1)$$

with

$$2\pi\nu_i = \Omega_i = \omega_0 - \omega_{RF}, \quad (2)$$

where  $\nu$  is the chemical shift offset frequency in  $s^{-1}$ ,  $\Omega$  is the chemical shift offset frequency in rad/s,  $\omega_0$  is the resonance frequency in rad/s and  $\omega_{RF}$  is the frequency of the RF field in rad/s. The rotating frame spin Hamiltonian is

$$H = \sum \pi J_{kl} 2 (I_{kx} I_{lx} + I_{ky} I_{ly} + I_{kz} I_{lz}), \quad (3)$$

for homonuclear scalar interactions and

$$H = \sum \pi J_{kl} 2 I_{kz} S_{lz}, \quad (4)$$

for heteronuclear scalar interactions, where  $J_{kl}$  is the scalar coupling constant between spin  $k$  and  $l$  in Hz. The spin Hamiltonian in the rotating frame for the applied RF field is

$$H = \sum (\omega_x I_x + \omega_y I_y), \quad (5)$$

where  $\omega_x$  and  $\omega_y$  are the RF magnetic field components along the  $x$  and  $y$  axes in rad/s with

$$\begin{aligned}\omega_x &= -\gamma B_1 \cos(\phi), \\ \omega_y &= -\gamma B_1 \sin(\phi),\end{aligned}\quad (6)$$

where  $\gamma B_1$  is the magnetogyric ratio times the strength of the RF field and  $\phi$  is the phase of the RF field.

In partially oriented systems, i.e., non isotropic liquids, a small fraction of the quadrupolar and dipolar interactions remain active due to incomplete averaging (Emsley and Lindon, 1975; Canet, 1996). The spin Hamiltonian,  $H$ , for spin  $S > 1/2$  with nuclear quadrupolar coupling and an axially symmetric electric field gradient tensor is

$$H = \pi \nu_Q \left( S_z^2 - \frac{1}{3} S^2 \right), \quad (7)$$

where  $\nu_Q$  is the quadrupolar splitting. The spin Hamiltonian for residual dipolar coupling in nonisotropic liquids is

$$H = \sum \pi D_{kl} 2 \left( I_{kz} I_{lz} - \frac{1}{2} (I_{kx} I_{lx} + I_{ky} I_{ly}) \right), \quad (8)$$

for homonuclear interactions, and

$$H = \sum \pi D_{kl} 2 I_{kz} S_{lz}, \quad (9)$$

for heteronuclear interactions, where  $D_{kl}$  is the residual dipolar coupling between spin  $k$  and  $l$  in Hz. Both residual coupling terms,  $\nu_Q$  and  $D_{kl}$ , are the couplings observed in spectra, without the factor 2 often used.

## QSim

QSim is a simulation program with a graphical user interface (GUI) as shown in Figure 1. QSim consists of four integrated parts: pulse sequence construction and compilation, spin system creation, simulation using HME, and processing. QSim is written in C++ and utilizes Microsoft Foundation Classes (MFC) for command routing and GUI. The different parts of QSim will now be described.

### Sequence construction and compilation

The first step in simulating a pulse sequence is to design the pulse sequence. The pulse sequence design in QSim is done graphically. Pulse sequence design includes adding channels, adding pulse sequence elements to channels and assignment of variables to pulse sequence elements.

Each channel in the pulse sequence is assigned a specific nucleus, except for the optional gradient channel. Pulse sequence elements are added to the channel using the mouse in a 'point and click' manner. Available elements are hard pulses, shaped pulses, delays and decoupling or mixing sequences. Only pulses and delays are allowed in the gradient channel. A vertical line, named sim-marker, is used to indicate points simultaneous in the pulse sequence. When a pulse element or a sim-marker is added the appearance of the pulse sequence is adjusted accordingly.

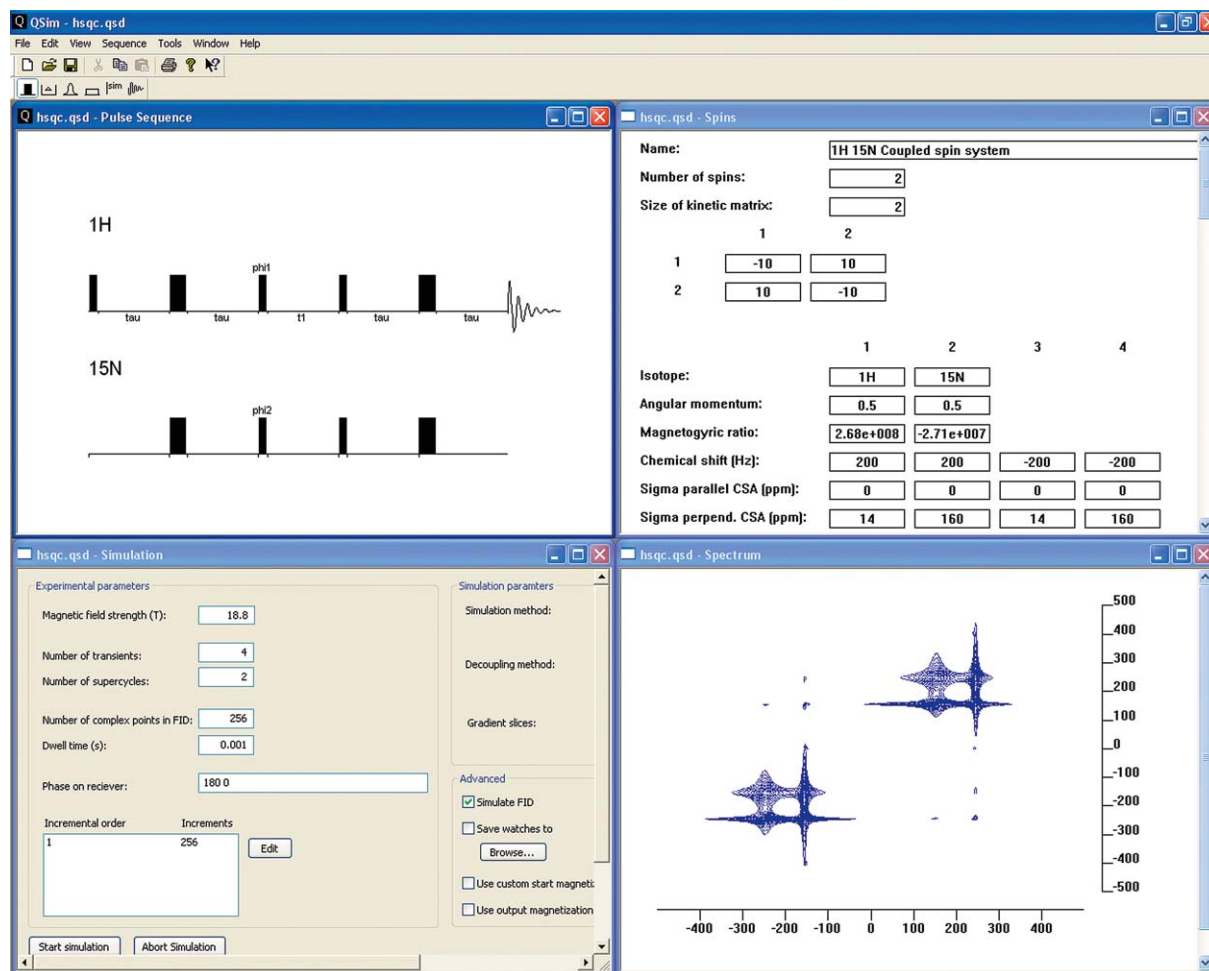
Pulse elements can be assigned a 'watch'. At the watch the magnetization (i.e. the magnitude of all Cartesian product operators) is calculated and stored to a file during the simulation.

When the pulse sequence is defined variables must be assigned to the pulse sequence elements. Variables are symbolic and a single variable can be used for multiple elements. It is, for example, convenient to use a variable named  $x$  for phase zero throughout the entire pulse sequence. The only pulse element for which a variable does not need to be assigned is the delay. Delays without an assigned variable are adjusted in length depending on the length of other elements and the location of sim-markers.

After adding all variables they must be set to numerical values. Depending on type of variable they can be static, cyclic, super cyclic or incremental. Static variables are constant throughout the simulation and cyclic variables vary with transient number. Super cyclic variables vary with a super cycle being a set of transients stored separately during acquisition. The 90 degree phase shift for quadrature detection in the indirect dimension using the States scheme (States et al., 1982) is an example of a variable that should be super cyclic. Incremental variables are mostly used for delays involving evolution in indirect dimensions, but could also be used for delays in relaxation experiments. After assigning numerical values to the pulse sequence the pulse sequence can be compiled.

Compilation of pulse sequences is automatically performed upon start of simulations. During compilation the program checks for inconsistencies, and if inconsistencies are found aborts the simulation and indicates positions with errors. If the compilation runs successfully the graphical pulse sequence is converted into a list of events with time marks. These events are later on assigned propagators and used in simulations.

Pulse sequences can be saved and loaded separately from other parameters in QSim.



**Figure 1.** An overview of QSim showing the main window and the windows for pulse sequence construction, spin system setup, simulation and viewing of spectra. The experiment viewed is an HSQC-spectrum without decoupling in either of the two dimensions. The offset chemical shifts were 200 Hz for both proton and nitrogen in one state, which is in exchange with another state with offset chemical shifts of  $-200$  Hz for both proton and nitrogen. The exchange rate constants were  $10 \text{ s}^{-1}$  for both the forward and the backward reaction. 256 complex data points with a dwell time of 0.001 s were simulated in both the direct and indirect dimensions. A cosine apodization function without zero-filling was used in both dimensions. The magnetic field strength was set to 18.8 T (800 MHz for  $^1\text{H}$ ). The scalar coupling between the nuclei was set to 92 Hz and the distance set to 1.02 Å. The dynamics responsible for relaxation was set to  $\tau_m = 25 \text{ ns}$ ,  $\tau_e = 50 \text{ ps}$  and  $S^2 = 0.8$  (Lipari and Szabo, 1982a, 1982b) for all interactions. The proton and nitrogen was relaxed by mutual dipole-dipole interactions and by CSA interactions with the external field using  $(\sigma_{\parallel} - \sigma_{\perp}) = -14 \text{ ppm}$  and  $-160 \text{ ppm}$  respectively. The angle between the unique axis of the CSA tensor and the internuclear vector between proton and nitrogen was set  $0^\circ$  and  $22^\circ$  for the proton and nitrogen respectively. The RF-field strength for both  $^1\text{H}$  and  $^{15}\text{N}$  were set to 50 kHz. The delay  $\tau$  was set to 2.72 ms.

## Spin system creation

Spin systems contain any number of spins with any spin quantum numbers and magnetogyric ratios. The minimum set of parameters for a spin system is the number of spins and the name of their respective isotope. If an isotope name is recognized, the corresponding spin quantum number and magnetogyric ratio is automatically set. Chemical shifts and scalar coupling constants can be modified. Finally, residual

dipolar and residual quadrupolar coupling constants can be set to values different from zero in order to simulate the effect of partially oriented systems.

Parameters that have to do with relaxation include information about dynamics in the form of the model-free parameters  $\tau_m$ ,  $\tau_e$  and  $S^2$  (Lipari and Szabo, 1982a, 1982b). Each interaction has its own separate set of model free parameters of dynamics. The coupling constants in relaxation such as distances,

```

int main(int argc, char* argv[])
{
    super_op L, Ltot, Rtot;
    spin_op H0, sigma0, sigma;

    calcprec sw      = 1000.0;           // Spectral width
    calcprec dwell   = 1.0/sw;          // Dwell time
    calcprec gB1     = 25000.0;         // B1 field strength in Hz
    calcprec pw      = 10.0e-6;         // 90 degrees
    int t2points     = 4096;           // 256 complex datapoints in t2

    Vector<complex<calcprec> > fid(t2points, 0.0), spectrum;

    spin_sys spin(2);                   // Number of spins in spin system
    spin.setfield(14.09);                // Field in Tesla

    spin.setisotope(0, "1H");           // Isotope label, spin 0
    spin.setshift(0,100.0);             // Shift in Hz, spin 0

    spin.setisotope(1, "1H");           // Isotope label, spin 1
    spin.setshift(1, -100.0);           // Shift in Hz, spin 1

    spin.setdistance(0, 1, 1.09e-10);   // Proton to proton distance 1.09 Å
    spin.setJ(0, 1, 50.0);              // Scalar coupling 50 Hz

    spin.setmodelfree(0.85, 5.0e-9, 50.0e-12); // S2, τm and τe

    spin.quantummechanics();            // Setting up quantum mechanics

    Rtot = RDD(spin)+RCC(spin);         // Dipole-dipole and CSA relaxation

    sigma0 = sigma_eq(spin);            // Equilibrium density operator

    H0 = Hiso(spin)+Hresidual(spin);    // Static Hamiltonian
    L = liouville(H0);                 // Static Liouvillian

    homogenizer(Rtot,sigma0);           // Make Rtot homogeneous
    Ltot=L+Rtot;                       // Total homogenous relaxation
                                        // Liouvillian

    sigma = pulse("1H", 90.0, pw, gB1, 0.0, Ltot)*sigma0;
    fid = calcfid(t2points, dwell, "1H", 0.0, Ltot, sigma);

    fid=sinebell(fid, 2);
    fid=zerosfill(fid, 2*t2points);
    spectrum=fft(fid);
    spectrum=phase(spectrum, 0.0, 0.0);
    plotfile("result1d.dat", spectrum, sw);

    return 0;
}

```

Figure 2. An overview of a HME C++ program. (A) Variables are defined and simulation parameters are set. (B) The spin system is set up and the function `quantummechanics()` is called. By calling this function a set of basis operators is created for the spin system. (C) The spinoperators and superoperators are constructed. (D) The actual simulation is performed and (E) the FID is processed and exported to a file. The program is further described in the text.

anisotropy of shielding and quadrupolar coupling constants can be modified. Parameters for relaxation interference effects can also be supplied.

The size and elements of a matrix of kinetic rate constants can be set in order to simulate chemical exchange (Helgstrand et al., 2000). It is the responsibility of the user to supply a physically realistic matrix of rates, see Supporting Information.

Spin systems in QSim can be exported and imported. It is thus possible to set up a library of spin systems for later use.

## Simulation

When the pulse sequence is set up and a spin system is defined, the simulation can proceed. However, simulation parameters must be set before the actual simulation can start. It is possible to set magnetic field strength, number of transients, number of points to acquire and the corresponding dwell time. The number of super cycles is set to 1 for 1D spectra and to 2 for States and echo/anti-echo 2D spectra. The number of points in the indirectly detected dimensions is also set. A choice between classical and quantum mechanical simulation must be made. It must also be decided if complete simulations of decoupling and mixing sequences should be performed, or if time efficient

simulations using an effective Liouvillian should be used instead. It is also possible to use a start magnetization. The start magnetization could either be a user defined magnetization or the residual magnetization from the previous transient. If watches are assigned in the pulse sequence, there is an option to calculate these and store the result in a separate file.

The simulation is then started and a basis set constructed, relaxation rates calculated and Liouvillians constructed. Finally propagators are calculated for all events in the compiled pulse sequence. Propagators are then multiplied according to transient number, super cycle and increment number, to create a free induction decay (FID).

## Processing

Free induction decays (FIDs) are available for processing after the simulation has finished. The processing modes supported are 1D, stacked 1D, 2D magnitude, 2D States et al. (1982), 2D TPPI (Drobny et al., 1978) and 2D echo/anti-echo (Kay et al., 1992). Standard window functions such as the exponential, gaussian, sinebell and squared sinebell functions can be multiplied to the free induction decays. The final size of the spectrum can be set, thus obtaining zero-filling. Zero and first order phasing parameters are available.

A flag can be set so that both a complex conjugate of the free induction decay is performed before Fourier transformation and that the spectrum is plotted with the frequency increasing from right to left. This is according to the recommended procedure to present spectra of nuclei with negative magnetogyric ratios (Levitt, 1997).

Finally, parameters for contouring the 2D spectrum such as the number of levels and the level multiplier can be adjusted. Spectra are viewed on screen and it is possible to zoom and scroll. Spectra can also be printed. The calculated spectrum can be exported in a text-format suitable for manipulations using Matlab.

## HME

The NMR simulation software package HME is a stand alone part of QSim and is written in C++ (Stroustrup, 2000). The source code of HME is available for download and it is possible to use HME directly without QSim. HME is more flexible than

QSim, it could for example be used for optimizations, but demands more of computer skills from the user. It is also more complicated to set up large pulse sequences with many pulses and/or shaped pulses in HME than in QSim.

To create a simulation using HME, a C++ program must be written using the classes and functions defined in HME. This program is then compiled and run. Compilers tested with HME are Visual Studio .NET and gcc (3.2). Results from simulations are usually stored to a text file and then analyzed using software such as Microsoft Excel, Matlab or Gnuplot (Williams et al., 1999).

Programs constructed with HME typically have different sections. To show the sections necessary for a simulation using HME a small example program is presented (Figure 2). The example is a simple 90 degree y-pulse followed by acquisition. The simulated spin system is a system of two coupled  $^1\text{H}$  spins.

In the first section of the program (marked section A), variables are defined and simulation parameters set. These parameters are used in other parts of the program.

In the next section (section B) the spin system is set up. In this case a spin system with two spins is defined. An isotope for the respective spin is set, this isotope is then recognized by HME and parameters in the spin system set accordingly. Chemical shifts are set in Hertz for the respective spin. After the spins have been defined interactions between them are set up. In this example the distance between the nuclei is set to 1.09 Å and the scalar coupling set to 50 Hz. Next the parameters for the dynamics are given. After all parameters are set the function `quantummechanics()` is called. By calling this function a basis set for the spin system is created. If classical mechanics should be used for simulation, the `classicalmechanics()` function is called instead.

In section C the relaxation rates are calculated and Hamiltonians and Liouvillians constructed. In this example only dipole-dipole and CSA relaxation are included. Next the equilibrium density operator is calculated and stored. The static Hamiltonian is constructed from the isotropic part and the part including effects due to partial orientation. From the Hamiltonian the Liouvillian is constructed. After the relaxation superoperator has been homogenized it is added to the static Liouvillian and the result is the Liouvillian describing the system in absence of pulses and gradients.

In section D the actual simulation is performed. The propagator of a  $90^\circ$  pulse is calculated and multiplied to the equilibrium density operator. The result is fed into the function for FID calculations and the result stored.

In the last section (E) the FID is processed and exported to file. The file can then be plotted using for example Gnuplot (Williams et al., 1999).

### Example simulations

To illustrate the capabilities of QSim and HME we present a few example simulations.

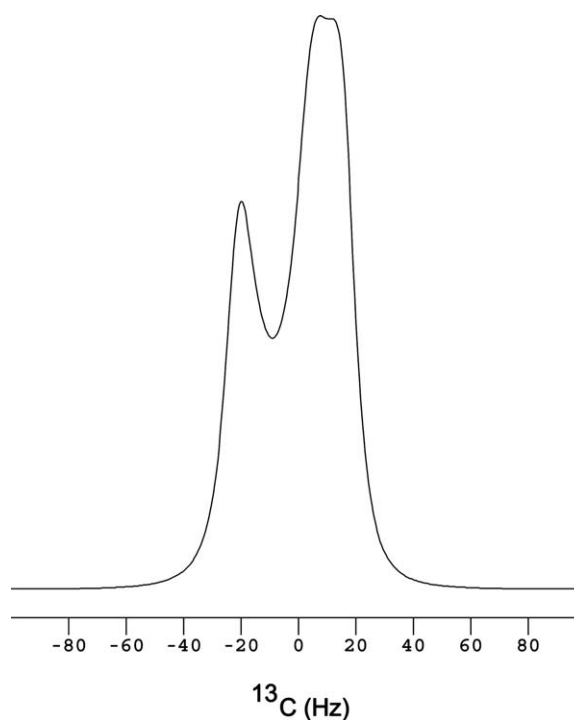
The first example (Figure 1) is a coupled HSQC experiment for a large protein at high field: The spin system is a  $^1\text{H}$ - $^{15}\text{N}$  moiety and it experiences two state chemical exchange. This example illustrates the relaxation interference effect between dipole-dipole relaxation and CSA relaxation (Goldman, 1984), the TROSY effect (Pervushin et al., 1997). It also exemplifies the use of chemical exchange in simulations. The simulation time for the 2D spectrum was 6 s on a 2.4 GHz Intel 4 PC with 512 MB of memory.

The second example (Figure 3) shows a simulation of a  $^2\text{H}$ - $^{13}\text{C}$  spin system and illustrates the dynamic frequency shift obtained due to the relaxation interference between the dipole-dipole relaxation and the quadrupolar relaxation (Grzesiek and Bax, 1994). The simulation time for the 1D spectrum was 4 s on a 2.4 GHz Intel 4 PC with 512 MB of memory.

The third example (Figure 4) shows the result of a pulse sequence for measurement of carbonyl  $^{13}\text{C}$  transverse relaxation rates (Mulder and Akke, 2003). The example illustrates the use of complicated pulse sequences in QSim, including gradients, shaped pulses and decoupling sequences. The spin system used is a  $^1\text{H}$ - $^{15}\text{N}$ - $^{13}\text{C}$  moiety in a protein backbone. The simulation time for the full relaxation decay with 60 points (Figure 4b) was 8 h on a 3 GHz Intel 4 PC with 1 GB of memory.

### Conclusions

Simulations of NMR experiments have several possible uses. It is expensive to operate NMR instruments and simulations can therefore be very important in the learning and teaching of NMR. Experienced NMR users can use computer simulations when they develop and test new pulse sequences. It has been shown



*Figure 3.* An experiment showing the interference between dipolar and quadrupolar interactions (Grzesiek and Bax, 1994). The pulse sequence simulated was a  $90^\circ$  degree y-pulse on  $^{13}\text{C}$  followed by acquisition. The  $^{13}\text{C}$  RF-field strength was set to 25 kHz. The magnetic field was set to 14.09 T. The spin system contains one  $^2\text{H}$  and one  $^{13}\text{C}$  with a scalar coupling of 20 Hz and a distance of 1.09 Å. The chemical shifts were set to 10 Hz and 0 Hz for the deuteron and the carbon, respectively. The quadrupolar coupling constant for  $^2\text{H}$  was set to 170 kHz. The dynamics responsible for relaxation was set to  $\tau_m = 10$  ns and  $S^2 = 1.0$  (Lipari and Szabo, 1982a, 1982b) for all interactions. The deuteron and the carbon were relaxed by mutual dipole-dipole interactions. The interference interaction between quadrupolar relaxation and dipole-dipole relaxation was included. The angle between the unique axis of the quadrupole tensor and the internuclear vector between the deuteron and the carbon was set to  $0^\circ$ . The spectrum was processed without apodization function.

that simulations can be useful in the analysis of experimental results concerning both the structure and the dynamics of molecules. Finally, simulations of NMR experiments are important in the numerical optimization of pulse sequence elements, such as mixing sequences and shaped pulses.

QSim can simulate almost any liquid state NMR experiment. It is easy to design and test pulse sequences using QSim. The program hides the complexity of NMR theory from the inexperienced user. QSim can with advantage be used for teaching and learning NMR as well as in the development and testing of pulse sequences. The QSim executable is available



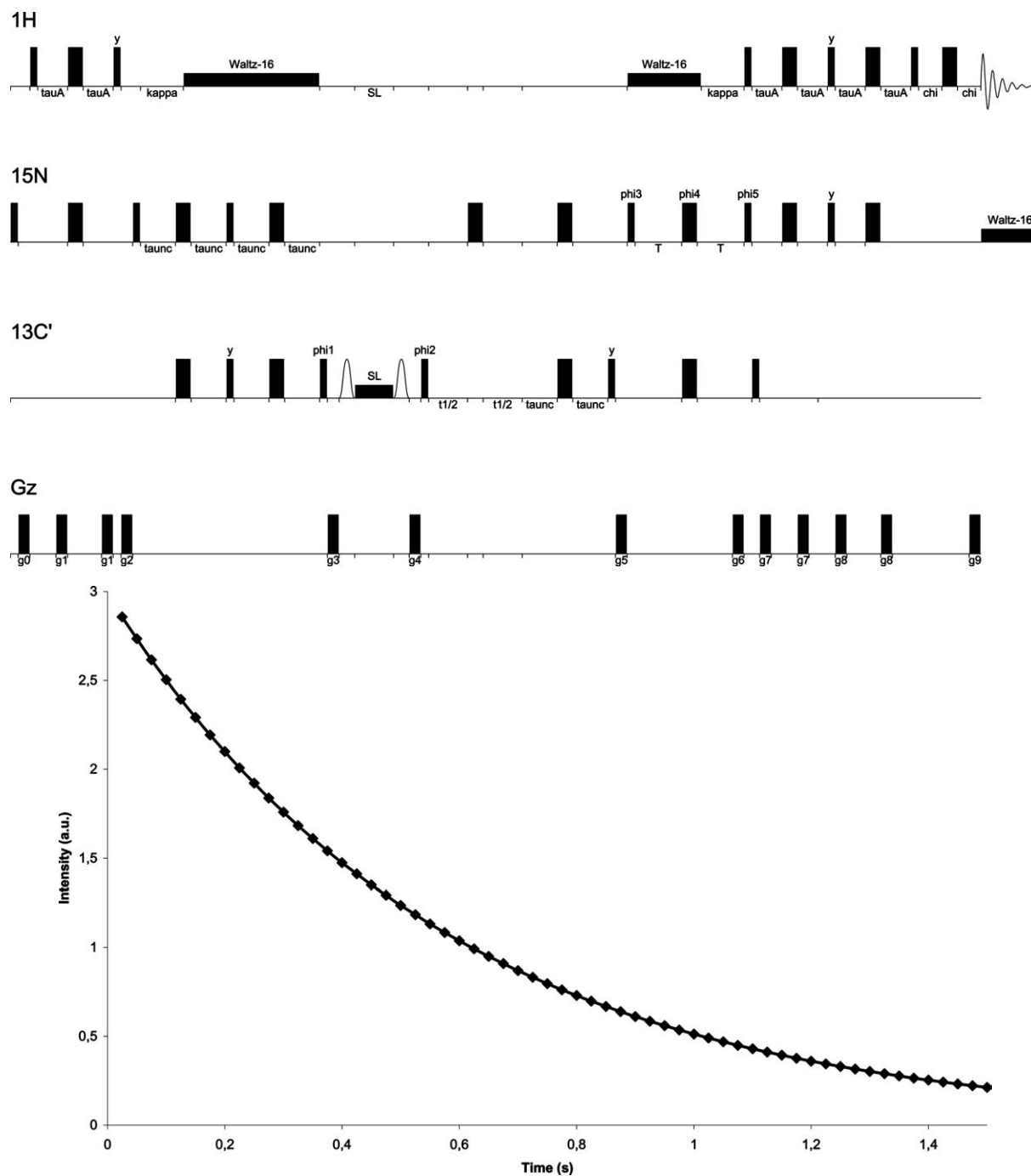


Figure 4. An experiment for measurement of carbonyl  $^{13}\text{C}$  transverse relaxation rate (Mulder and Akke, 2003). The pulse sequence, as printed from QSim, is shown in (a) and a relaxation decay in (b). The spin system simulated was a  $^1\text{H}$ - $^{15}\text{N}$ - $^{13}\text{C}$  moiety in a protein backbone. Chemical shifts were set to 100 Hz, 50 Hz and  $-75$  Hz for the proton, nitrogen and carbon respectively. The distances were set to 0.98 Å between  $^1\text{H}$  and  $^{15}\text{N}$ , 1.3 Å between  $^{15}\text{N}$  and  $^{13}\text{C}$  and 1.98 Å between  $^1\text{H}$  and  $^{13}\text{C}$ . The scalar couplings were set to 92 Hz between  $^1\text{H}$  and  $^{15}\text{N}$ , 15 Hz between  $^{15}\text{N}$  and  $^{13}\text{C}$  and 0 Hz between  $^1\text{H}$  and  $^{13}\text{C}$ . The CSA for the  $^{13}\text{C}$  was set to  $(\sigma_{\parallel} - \sigma_{\perp}) = 90$  ppm. The dynamics responsible for relaxation was set to  $\tau_m = 5$  ns,  $\tau_c = 50$  ps and  $S^2 = 0.8$  (Lipari and Szabo, 1982a, 1982b) for all interactions. Simulations were performed at 14.1 T. Adiabatic ramps were simulated using shaped pulses and were 4 ms long. The pulse sequence is slightly modified from Figure 2b in the paper by Mulder and Akke (2003). Water flip-back pulses and  $C_{\alpha}$  pulses have been removed, since water and  $C_{\alpha}$  were not included in the simulation. Parameters and phase cycles used were otherwise the same as in the original pulse sequence. 60 points in a relaxation decay were simulated using the entire 8 step phase cycle and 100 slices for gradients. The entire simulation included  $60 * 8 * 100 = 4800$  full simulations of the pulse sequence, which took 8 h on a 3 GHz Pentium 4 PC with 1 GB of memory.

for download at [www.bpc.lu.se/QSim](http://www.bpc.lu.se/QSim), together with a manual and example simulations.

HME is a C++ toolkit and its use requires basic knowledge of C++ programming. HME is useful for numerical optimization of pulse sequence elements. Parameters can be optimized if a target function can be formulated. HME can also be used when evaluating experimental data. In this case a target function is used to compare experimental data with simulated data. It is then possible to extract interesting experimental data such as rate constants. The HME source code is available for download at [www.bpc.lu.se/QSim](http://www.bpc.lu.se/QSim), together with a manual and example programs.

### Acknowledgements

The authors would like to acknowledge Frans Mulder and Patrik Lundström for beta-testing and valuable comments on the software.

**Supporting Information Available:** A description of the theoretical background (PDF). This material is available free of charge via the Internet at <http://kluweronline.com/issn/0925-2738>

### References

- Abragam, A. (1961) *Principles of Nuclear Magnetism*, Oxford University Press, Oxford.
- Akke, M. and Palmer, A.G. (1996) *J. Am. Chem. Soc.*, **118**, 911–912.
- Allard, P. and Härd, T. (2001) *J. Magn. Reson.*, **153**, 15–21.
- Allard, P., Helgstrand, M. and Härd, T. (1997) *J. Magn. Reson.*, **129**, 19–29.
- Allard, P., Helgstrand, M. and Härd, T. (1998) *J. Magn. Reson.*, **134**, 7–16.
- Allman, T., Bain, A.D. and Garbow, J.R. (1996) *J. Magn. Reson. A*, **123**, 26–31.
- Blanton, W.B. (2003) *J. Magn. Reson.*, **162**, 269–283.
- Bloch, F. (1946) *Phys. Rev.*, **70**, 460–474.
- Canet, D. (1996) *Nuclear Magnetic Resonance: Concepts and Methods*, John Wiley & Sons, Chichester.
- Deverell, C., Morgan, R.E. and Strange, J.H. (1970) *Mol. Phys.*, **18**, 553–559.
- Drobny, G., Pines, A., Sinton, S., Weitekamp, D. and Wemmer, D. (1978) *Faraday Symp. Chem. Soc.*, **13**, 49–55.
- Emsley, I.W. and Lindon, J.C. (1975) *NMR Spectroscopy Using Liquid Crystal Solvents*, Pergamon Press, Oxford.
- Ernst, R.R., Bodenhausen, G. and Wokaun, A. (1987) *Principles of Nuclear Magnetic Resonance in One and Two Dimensions*, Oxford University Press, Oxford.
- Ghose, R. (2000) *Concepts Magn. Reson.*, **12**, 152–172.
- Goldman, M. (1984) *J. Magn. Reson.*, **60**, 437–452.
- Goldman, M. (1988) *Quantum Description of High-Resolution NMR in Liquids*, Oxford University Press, Oxford.
- Grzesiek, S. and Bax, A. (1994) *J. Am. Chem. Soc.*, **116**, 10196–10201.
- Güntert, P., Schaefer, N., Otting, G. and Wüthrich, K. (1993) *J. Magn. Reson. A*, **101**, 103–105.
- Helgstrand, M., Härd, T. and Allard, P. (2000) *J. Biomol. NMR*, **18**, 49–63.
- Jeener, J. (1982) *Adv. Magn. Reson.*, **10**, 1–51.
- Kay, L.E., Keifer, P. and Saarinen, T. (1992) *J. Am. Chem. Soc.*, **114**, 10663–10665.
- Letourneau, P.-J., Boyko, R. and Sykes, B.D. (2003) *J. Magn. Reson.*, **161**, 154–167.
- Levante, T.O. and Ernst, R.R. (1995) *Chem. Phys. Lett.*, **241**, 73–78.
- Levitt, M.H. (1997) *J. Magn. Reson.*, **126**, 164–182.
- Levitt, M.H. and Di Bari, L. (1992) *Phys. Rev. Lett.*, **69**, 3124–3127.
- Levitt, M.H. and Di Bari, L. (1994) *Bull. Magn. Reson.*, **16**, 94–114.
- Lipari, G. and Szabo, A. (1982a) *J. Am. Chem. Soc.*, **104**, 4546–4559.
- Lipari, G. and Szabo, A. (1982b) *J. Am. Chem. Soc.*, **104**, 4559–4570.
- McConnell, H.M. (1958) *J. Chem. Phys.*, **28**, 430–431.
- Meresi, G.H., Cuperlovic, M., Palke, W.E. and Gerig, J.T. (1999) *J. Magn. Reson.*, **137**, 186–195.
- Moler, C. (2002) The MathWorks Inc., Natick.
- Mulder, F.A.A. and Akke, M. (2003) *Magn. Reson. Chem.*, **41**, 853–865.
- Nicholas, P., Fushman, D., Ruchinsky, V. and Cowburn, D. (2000) *J. Magn. Reson.*, **145**, 262–275.
- Pervushin, K.V., Riek, R., Wider, G. and Wüthrich, K. (1997) *Proc. Natl. Acad. Sci. USA*, **94**, 12366–12371.
- Smith, S.A., Levante, T.O., Meier, B.H. and Ernst, R.R. (1994a) *J. Magn. Reson. A*, **106**, 75–105.
- Smith, S.A., Palke, W.E. and Gerig, J.T. (1994b) *J. Magn. Reson. A*, **106**, 57–64.
- Solomon, I. (1955) *Phys. Rev.*, **99**, 559–565.
- Sørensen, O.W., Eich, G.W., Levitt, M.H., Bodenhausen, G. and Ernst, R.R. (1983) *Prog. NMR Spectrosc.*, **16**, 163–192.
- States, D.J., Haberkorn, R.A. and Ruben, D.J. (1982) *J. Magn. Reson.*, **48**, 286–292.
- Stroustrup, B. (2000) *The C++ Programming Language*, Addison – Wesley, Reading.
- Studer, W. (1988) *J. Magn. Reson.*, **77**, 424–438.
- van de Ven, F.J.M. and Hilbers, C.W. (1983) *J. Magn. Reson.*, **54**, 512–520.
- Werbelow, L. and London, R.E. (1996) *Concepts Magn. Reson.*, **8**, 325–338.
- Widmer, H. and Wüthrich, K. (1986) *J. Magn. Reson.*, **70**, 270–279.
- Williams, T., Kelley, C., Lang, R., Kotz, D., Campbell, J. and Elber, G. (1999).