ACADEMIC PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Magnetic Resonance 161 (2003) 154–167

JMR Journal of Magnetic Resonance

www.elsevier.com/locate/jmr

PJNMR: a platform-independent graphical simulation tool for NMR spectroscopy

Paul-Jean Letourneau,[a,b] Robert Boyko,[c] and Brian D. Sykes[c,*]

[a] *Protein Engineering Network of Centres of Excellence, University of Alberta, Edmonton, Alta., Canada T6G 2H7*
[b] *Department of Biochemistry, University of Alberta, 419 Medical Sciences Bldg., Edmonton, Alta., Canada T6G 2H7*
[c] *CIHR Group in Protein Structure and Function, University of Alberta, Edmonton, Alta., Canada T6G 2H7*

Received 17 October 2002; revised 23 December 2002

Abstract

A new simulation program for multinuclear NMR is introduced. PJNMR (*P*ure *J*ava *NMR*) 2.0, written entirely in the Java programming language, simulates pulse sequences on systems of up to three weakly coupled spins-1/2 with a command-driven, spectrometer-like interface. Users may simulate the effects of pulses, precessions, and pulsed field gradients on the spin system, with a graphical display showing the state of the density matrix (in a novel polar-coordinate representation) as well as the magnetization vectors for each nucleus. Relevant computations and optimizations as implemented in the code are detailed, along with the object-oriented structures used. A description of the simulation environment is given, illustrated with a series of example pulse sequences highlighting the insights gained in the graphical presentation.

1. Introduction

Due to the intrinsic quantum mechanical nature of NMR, and the increasingly complex pulse sequences available to the NMR spectroscopist, it is becoming more and more difficult to reconcile experimental observations with one's intuition. Therefore simulation programs can be very useful, providing the NMR practitioner with the ability to easily, quickly and visually probe the dynamics of spin systems in response to a variety of complex pulse sequences. While approaches like the product-operator [1] calculations are simple and straightforward, they often replace physical insight and intuition with long algebraic expressions. Simulation programs give the scientist the ability to remain focused on physical principles, while a computation engine swiftly handles details like operator algebras and relaxation matrices.

Several NMR simulation tools are currently available. These programs such as 'The Virtual NMR Spectrometer' [2], GAMMA [3], and SIMPLTN [4] focus on high-accuracy data output for external processing, including a large amount of spin physics in their simulations. Due to their ambitious nature, these programs often require programming expertise or computational back-ends (e.g., MATLAB) to run them. Other programs target accurate simulations of a small number of specialized effects in the spin system, such as relaxation matrices (e.g., FIRM [5]), NOE (e.g., MORASS [6]), and CSA (e.g., SIMMOL [7]). In all cases, generality is balanced with ease of use. More complex calculations require more sophisticated computational engines, and more general simulations require the user to learn a more generalized set of commands. To date, there are no existing tools for graphical pulse sequence simulation offering true platform-independence (without the use of a back-end engine) while still presenting the two most important quantities in the spin system: *spin density* and *observed magnetization*.

This paper documents a newly developed program that addresses these needs. PJNMR (*P*ure *J*ava *NMR*) simulates the effect of pulses, precessions, and gradient pulses on systems of up to three weakly-coupled spins-1/2. The program uses an approximate multiple position approach [9] to simulate the effects of gradient pulses. The density matrix, magnetization vectors, and

* Corresponding author.
E-mail address: brian.sykes@ualberta.ca (B.D. Sykes).

1090-7807/03/$ - see front matter © 2003 Elsevier Science (USA). All rights reserved.
doi:10.1016/S1090-7807(03)00008-9

pulse sequence are shown graphically as commands are entered. This allows users to view directly the relationships between quantum mechanical coherences developed during the pulse sequence and the observed magnetization. Written entirely in the Java programming language, PJNMR users benefit from total platform independence and a rich graphical interface. The program is written using the *Model-View-Controller* architecture, a well-known graphical interface design principal [10]. The utility of the program is demonstrated by showing a series of screen shots for several common pulse sequences, illustrating the highly educational and practical nature of the graphical output.

PJNMR is designed to both illustrate difficult quantum mechanical concepts in NMR, as well as allow the NMR spectroscopist accurate and fast insight into the behavior of spin systems in complex pulse sequences. Because it is platform independent, the program appeals to users in all computational environments.

2. Algorithms and methods

The purpose of this section is to provide the theoretical background for the computations performed in PJNMR. The formalism presented in [1] is used throughout. The theory is developed to the extent needed for someone familiar with NMR to understand the mathematical methods used in the program.

2.1. Description of spin system

In Dirac notation, a wavefunction, Ψ, can be represented as a superposition of orthonormal basis functions $|n\rangle$,

$$\Psi = \sum_{n=1}^{N} c_n |n\rangle, \tag{1}$$

where N is the dimensionality of the vector space and c_n are complex coefficients. In a system of M spin-1/2 nuclei, $N = 2^M$, since there are 2 spin states for each nucleus. For example, in a system of 2 spin-1/2 nuclei, the wavefunction has the explicit form

$$\Psi = c_1|\alpha\alpha\rangle + c_2|\alpha\beta\rangle + c_3|\beta\alpha\rangle + c_4|\beta\beta\rangle. \tag{2}$$

We define the *density matrix* (or *spin density*) for the spin system, σ, such that

$$\langle n|\sigma|m\rangle = \overline{c_n c_m^*}, \tag{3}$$

where the overbar indicates the ensemble average. The density matrix has the property that the expectation value $\overline{\langle A\rangle}$ for an observable with associated operator A is given by

$$\overline{\langle A\rangle} = \mathrm{Tr}\{\sigma A\}. \tag{4}$$

Keeping with the example of a 2-spin system, the density matrix for the system has the form

$$\sigma = \begin{pmatrix} c_1c_1^* & c_1c_2^* & c_1c_3^* & c_1c_4^* \\ c_2c_1^* & c_2c_2^* & c_2c_3^* & c_2c_4^* \\ c_3c_1^* & c_3c_2^* & c_3c_3^* & c_3c_4^* \\ c_4c_1^* & c_4c_2^* & c_4c_3^* & c_4c_4^* \end{pmatrix}. \tag{5}$$

Formally, diagonal elements of the density matrix represent *state populations*, and off-diagonal elements represent *state transition probabilities*. Element (1,1) in Eq. (5) thus represents the *population* of the $|\alpha\alpha\rangle$ state, and element (1,3) represents the *probability for transition* from the $|\alpha\alpha\rangle$ state to the $|\beta\alpha\rangle$ state. Representing the complex coefficients c_n in polar coordinates, each has the form

$$c_n = |c_n|\mathrm{e}^{\mathrm{i}\varphi_n}. \tag{6}$$

Each element of the density matrix then has the form

$$c_n c_m^* = |c_n c_m|\mathrm{e}^{\mathrm{i}(\varphi_n - \varphi_m)}, \tag{7}$$

where an ensemble average is implied. At thermal equilibrium, state transitions have zero probability, so the density matrix σ^{eq} is diagonal. Diagonal elements at equilibrium are given by the Boltzman distribution:

$$\sigma_{mn}^{\mathrm{eq}} = \delta_{m,n} - \frac{\mathrm{e}^{-E_n/k_{\mathrm{B}}T}}{\sum_{i=1}^{N} \mathrm{e}^{-E_i/k_{\mathrm{B}}T}}, \tag{8}$$

where $\delta_{m,n}$ is the Kronecker delta, E_i is the energy of state i, k_{B} is Boltzman's constant, and T is the temperature of the ensemble. A convenient computational representation of the density matrix may be obtained by invoking the high-temperature approximation ($E_n \ll k_{\mathrm{B}}T$), keeping only the first linear term in the expansion of the exponentials in Eq. (8), and throwing away the first constant term in the expansions (since the constant term is not relevant to the NMR experiment):

$$\sigma^{\mathrm{eq}} = -\frac{B_0}{Nk_{\mathrm{B}}T} \sum_{m=1}^{M} \gamma_m I_{mz}, \tag{9}$$

where B_0 is the magnetic field strength, γ_m is the *gyromagnetic ratio* of nucleus m, I_{mz} is the z-angular momentum operator for nucleus m (the precise matrix forms of these operators are discussed below), and M is the number of spins in each unit of the ensemble (i.e., chemical group, molecule, etc.).[1] Dividing through by the constant $-B_0/Nk_{\mathrm{B}}T$ in front along with the sum of the (absolute values of) gyromagnetic ratios for all nuclei, we obtain

$$\sigma_{\mathrm{comp}}^{\mathrm{eq}} = \frac{\sigma^{\mathrm{eq}}}{-\frac{B_0}{Nk_{\mathrm{B}}T}\sum_{m=1}^{N}|\gamma_m|} = \frac{\sum_{m=1}^{N}\gamma_m I_{mz}}{\sum_{m=1}^{N}|\gamma_m|}. \tag{10}$$

[1] Note that the density matrix in Eq. (9) has units of $1/\hbar$, where $\hbar$ is Planck's constant divided by 2π, namely $\mathrm{s\,kg^{-1}\,m^{-2}}$. In the form in Eq. (10) below, $\sigma_{\mathrm{comp}}^{\mathrm{eq}} = \sigma^{\mathrm{eq}}/\frac{-B_0}{Nk_{\mathrm{B}}T}\sum_{m=1}^{M}|\gamma_m|$ is nondimensionalized.

In the form of Eq. (10), the density matrix is conveniently normalized for numerical calculations, and has dimension unity.

All information about the spin system is contained in the density matrix, and changes to the system are represented by successive transformations of the density matrix. The time-dependence of the spin system is obtained by solving the Schrödinger equation for the wavefunction in Eq. (1) in the presence of a particular Hamiltonian H. The equivalent form of the Schrödinger equation in terms of the density matrix is the *Liouville-von Neumann* equation:

$$\frac{\mathrm{d}\sigma(t)}{\mathrm{d}t} = \mathrm{i}[\sigma(t), H]. \tag{11}$$

For time-independent Hamiltonian H, the solution to Eq. (11) is

$$\sigma(t) = \mathrm{e}^{-\mathrm{i}Ht}\sigma(0)\mathrm{e}^{\mathrm{i}Ht}. \tag{12}$$

Therefore all information about the time-evolution of the spin system is contained in the Hamiltonian for each transformation performed on it, and the spin density transforms as in Eq. (12), provided the Hamiltonians can be rendered time-independent with a suitable transformation.

2.2. Hamiltonians

In order to describe the evolution of the spin system during a pulse sequence, it suffices, in light of Eq. (12), to provide a description of the relevant Hamiltonians. There are four Hamiltonians used in PJNMR: the Zeeman, Scalar Coupling, Pulse, and Gradient Hamiltonians.

2.3. Zeeman Hamiltonian

The Zeeman Hamiltonian, H_Z, is given by

$$H_Z = \sum_{i=1}^{M} \omega_i I_{iz}, \tag{13}$$

where ω_i is the angular frequency, and I_{iz} is the z-angular momentum operator, for the ith nucleus in an M-spin system. The Zeeman Hamiltonian is active during all periods of chemical shift evolution, such as during free precessions and gradient pulses.

2.4. Scalar coupling Hamiltonian

In the limit of weak coupling ($2\pi J_{ij} \ll |\omega_i - \omega_j|$), the scalar coupling Hamiltonian is given by

$$H_J = 2\pi \sum_{i=2}^{M} \sum_{j=1}^{i-1} J_{ij} I_{iz} I_{jz}, \tag{14}$$

where J_{ij} is the coupling constant between spins i and j, and $I_{i,jz}$ are the z-angular momentum operators for spins i and j. The scalar coupling Hamiltonian is active during all periods of free precession and gradient pulses for coupled nuclei.

2.5. RF pulse Hamiltonian

With the application of (the quantum-mechanical equivalent of) a rotating-frame transformation, the rf pulse Hamiltonian is rendered time-independent, and is given by

$$H_{\mathrm{e}}^{i} = \gamma_i B_1 (I_x \cos\phi + I_y \sin\phi) \tag{15}$$

for an on-resonance pulse on nuclei with gyromagnetic ratio γ_i, applied at a phase angle of ϕ radians to the x-axis in the xy plane, with rf field strength B_1. Eq. (15) is a special case for pulses in the xy-plane; the term in brackets is in general given by $\mathbf{n} \cdot \mathbf{I}$, where $\mathbf{n}$ is the unit normal vector in the direction of the applied rf field (in the rotating frame), and $\mathbf{I}$ is the angular momentum vector. The rotation angle of the pulse with respect to the z-axis, α, where the rf pulse lasts τ seconds, is given by $\alpha = \gamma_i B_1 \tau$.

2.6. Gradient Hamiltonian

During a linear z-gradient pulse, the magnetic field along the z-axis becomes spatially inhomogeneous through the sample, and therefore the precession frequencies of the nuclei in the sample become position-dependent. All spins during the gradient evolve under the Chemical Shift and Scalar Coupling Hamiltonians, with an additional Gradient Hamiltonian identical in form to the Chemical Shift Hamiltonian with a position-dependent frequency. Using an approximate multiple-position approach [9], the NMR sample is divided up into n segments in the z-direction, with the Gradient Hamiltonian in segment k given by

$$H_{\mathrm{G}}^{k} = \sum_{i=1}^{M} \gamma_i G_z z_k I_{zi}, \tag{16}$$

where z_k is the z-position of segment k, in meters (where the position is measured from the center of the NMR sample to the *center* of the segment), and G_z is the gradient strength, in Tesla per meter.

2.7. Evolution of the spin system

The spin density evolves under the influence of time-independent Hamiltonians as a transformation of the form of Eq. (12). The exponential operators themselves are defined by their Taylor series [8], which can be conveniently brought to closed-form using Pauli matrices and their associated identities. Exponentials of single-quantum operators I_η, where $\eta = x, y, z$, are given by

$$\mathrm{e}^{\pm \mathrm{i}\omega t I_\eta} = E \cos\left(\frac{\omega t}{2}\right) \pm 2\mathrm{i} I_\eta \sin\left(\frac{\omega t}{2}\right), \tag{17}$$

where E is the identity operator. In particular, the exponential of the rf pulse Hamiltonian $R_\phi(\alpha)$ with phase ϕ in the xy-plane and rotation angle α can be expressed as

$$R_\phi(\alpha) = e^{-i\alpha \mathbf{n}\cdot\mathbf{I}} = E\cos\left(\frac{\alpha}{2}\right) - 2i\mathbf{n}\cdot\mathbf{I}\sin\left(\frac{\alpha}{2}\right). \tag{18}$$

Exponentials of the weak scalar coupling operator of the form I_zS_z as in Eq. (14) are given by

$$e^{\pm i2\pi J t I_z S_z} = E\cos\left(\frac{2\pi J t}{4}\right) \pm 4iI_zS_z\sin\left(\frac{2\pi J t}{4}\right). \tag{19}$$

Evolution of the spin density can then be computed by implementing transformations of the form of Eq. (12) with Pauli operator representations as in Eqs. (17)–(19).

2.8. Operator representations for multiple spins

In order to do numerical computations, the matrix representations of the angular momentum operators must be known for arbitrary numbers of spins. All M-spin operators are computed as *direct products* of their single-spin representations. The angular momentum operator for spin k in an M-spin system, $I_{\eta k}^{(M)}$, where $\eta = x, y, z$, is computed from direct products of the single-spin operator $I_{\eta k}^{(1)}$ and the identity operator E as follows:

$$I_{\eta k}^{(M)} = E_1 \otimes E_2 \cdots \otimes E_{k-1} \otimes I_{\eta k}^{(1)} \otimes E_{k+1} \cdots \otimes E_M. \tag{20}$$

The single-spin operators have the following explicit form:

$$I_x = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad I_y = \frac{1}{2}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},$$

$$I_z = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{21}$$

The elements of the angular momentum operators correspond to energy-level transitions in the spin system. As an example, consider the matrix representation of the I_x operator in a two-spin system. Eq. (22) gives the matrix representation and the energy level transitions to which the nonzero elements of the matrix correspond.

$$I_x = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & |\beta\alpha\rangle \to |\alpha\alpha\rangle & 0 \\ 0 & 0 & 0 & |\beta\beta\rangle \to |\alpha\beta\rangle \\ |\alpha\alpha\rangle \to |\beta\alpha\rangle & 0 & 0 & 0 \\ 0 & |\alpha\beta\rangle \to |\beta\beta\rangle & 0 & 0 \end{pmatrix}. \tag{22}$$

In general, nonzero elements (n, m) in the transverse (x and y) Cartesian angular momentum operators correspond to single-quantum transitions between energy levels n and m. The transverse Cartesian angular-momentum operators may be partitioned into operators specific to a single coherence. Thus the I_x operator for a 2-spin system in Eq. (22) may be represented as a sum of single-coherence operators $I_x^{(n,m)}$:

$$I_x = I_x^{(3,1)} + I_x^{(4,2)} + I_x^{(1,3)} + I_x^{(2,4)}. \tag{23}$$

The gradient, scalar coupling, and chemical shift evolution operators are by nature *non-selective*, meaning they always act on all nuclei in the spin system. However, rf pulses can be confined to a small frequency range where they act only on select nuclei types (*soft* or *selective* pulses). Eq. (18), together with Eqs. (20) and (21) give the matrix representation for a *selective* pulse operator on spin k in an M-spin system. A rf pulse operator acting on a *set* of spins S consisting of multiple nuclei of the same type in an M-spin system[2] is given by

$$R_{S\phi}^{(M)}(\alpha) = \prod_{k\in S} R_{k\phi}^{(M)}(\alpha), \tag{24}$$

where the selective operator $R_{k\phi}^{(M)}(\alpha)$ is a straightforward extension of Eq. (20):

$$R_{k\phi}^{(M)}(\alpha) = E_1 \otimes E_2 \otimes \cdots E_{k-1} \otimes R_\phi^{(1)}(\alpha) \otimes E_{k+1} \otimes \cdots E_M. \tag{25}$$

2.9. Magnetization

The NMR observable, magnetization, is computed as in Eq. (4) with the magnetization operator $N\gamma_m\hbar\mathrm{Tr}\sigma\cdot I_{m\eta}$ for η-magnetization on nucleus m, where $\eta = x, y, z$:

$$M_{m\eta} = N\gamma_m\hbar\,\mathrm{Tr}\{\sigma\cdot I_{m\eta}\}. \tag{26}$$

For computational purposes, the magnetization components in Eq. (26) can be normalized by the total length of the magnetization vector for nucleus m, $|\mathbf{M}_m|$:

$$M_{m\eta}^{\mathrm{comp}} = \frac{M_{m\eta}}{|\mathbf{M}_m|}. \tag{27}$$

As noted above, the nonzero elements in the transverse Cartesian angular momentum operators can be associated with specific single-quantum energy level transitions. The magnetization in Eq. (26) can then be partitioned similarly into a sum of magnetizations produced by individual single-quantum coherences:

$$M_{m\eta} = \sum_{(n,m)\in 1\mathrm{QC}} M_{m\eta}^{(n,m)}, \tag{28}$$

where "1QC" refers to the set of all pairs of energy level indices (n, m) corresponding to single-quantum coherences.

[2] For example, if there were two protons, the first and third nuclei in a three-spin system, the set would consist of the integers 1 and 3: $S = \{1, 3\}$.

3. Program organization

3.1. Pure Java NMR

PJNMR is an acronym for *P* ure *J* ava *NMR*. It is a 100% pure Java application, meaning it is written entirely in the Java programming language. Java was chosen as the sole development language for PJNMR for several reasons. First, Java allows complete platform independence, making PJNMR a truly *portable* tool. Thus, PJNMR may be run on any machine where a valid Java Runtime Environment (JRE) is installed.[3] Second, Java's object-oriented structure provides an intuitive framework for developing the graphical interface (see section Object-Oriented Framework). Third, Java provides a large array of packages for future extensions of the program. The result is a versatile, richly graphical, and highly usable program for NMR education and professional pulse sequence design. See Section 4 for example PJNMR sessions.

3.2. Object-oriented framework

PJNMR follows the model-view-controller object-oriented design pattern [10]. In this design methodology, object s are grouped into three categories: "model", "view", and "control". All objects designated as *model* are solely responsible for performing meaningful operations on the underlying data in the program. *View* objects are then responsible for giving some appropriate representation of the data to the user. *Controller* objects coordinate the manipulation of the underlying model through user interaction. Fig. 1 shows a standard inheritance diagram [11] for the objects used in PJNMR, shown with a number of fundamental Java libraries (or "packages") they depend on. The diagram shows how the objects are grouped into the model, view, and controller categories.

The "model" in PJNMR is completely represented by the MathEngine class, which is responsible for storing the current state of the density matrix, and computing the effects of all transformations initiated by user commands. "Control" is facilitated by subclasses of two main object types: AbstractAction and AbstractCommand. Fig. 2 shows an event flowchart depicting a pulse command. When a user command is registered through interaction with the graphical interface (e.g., a button is clicked or a menu item is selected), an object of type AbstractAction is notified. In the figure, a clicking on a pulse button notifies a PulseAction object (Fig. 1 shows that PulseAction is one of the subclasses of AbstractAction, which resides in the javax.swing Java library). The AbstractAction object then creates an object of type AbstractCommand (in the figure, a PulseCommand object is generated by the PulseAction object). The AbstractCommand is then parsed (or *consumed*) by MathEngine, and the required transformation on the density matrix is performed.

This signaling pathway is intermediated by the Globals class, shown as a member of the "control" group in Fig. 1. The Globals class holds all global variables (i.e., all data accessible to the rest of the class hierarchy) and contains methods for coordinating program initialization, command execution, and modifying global data structures. The most notable data structure is sequence of commands given by the user, which is of type Macro, a subclass of the Vector data structure in Java (see Fig. 1). The Macro object represents the pulse sequence being generated by the user.

The "view" in PJNMR consists of all classes of type JPanel and JFrame, as shown in Fig. 1. The final action in the sequence of events shown in Fig. 2 is an update of the PJNMRPanel object, which displays the resulting density matrix and magnetization vectors.

3.3. Gradients

PJNMR employs an approximate multi-position (MP) approach [9] to simulate the effect of a linear field gradient in the z-direction (along the long axis of the NMR sample). Using the MP approach, the sample is divided up into slices of finite thickness, with the Gradient Hamiltonian in each slice given by Eq. (16). Table 1 lists the gradient parameters used in PJNMR.

According to [9], for an accurate phase sampling of a particular coherence in a general heteronuclear system, the number of sample slices, N, must not have a common factor with the quantity

$$\frac{\sum_a^{\text{in the coherence}} \gamma_a}{\gamma_{\mathrm{H}}}, \tag{29}$$

where the sum extends over all nuclei involved in the coherence in question, and γ_{H} is the gyromagnetic ratio of the proton. Choosing $N = 201$ ensures this condition is satisfied for all possible coherences in the spin systems available in PJNMR, and allows complete averaging in every case.

Fig. 3 shows a sequence of transformations undergone by the density matrix for a pulse sequence that includes gradient pulses. After the application of the first gradient G_1, the program must keep track of the density matrix from each slice, σ_k^i, for the remainder of the pulse sequence. However, in steps in between gradient pulses G_1 and G_2, the average of all the transformed density matrices $\bar{\sigma}_{k+1}$ is simply equal to the transform of the average. Therefore, transformations on the density matrix in between gradient pulses may be accomplished by simply applying transformations to the average over all the slices. However, the density matrix at each slice, σ_k^i,

[3] See the PJNMR release notes for what constitutes a "valid" Java Runtime Environment.

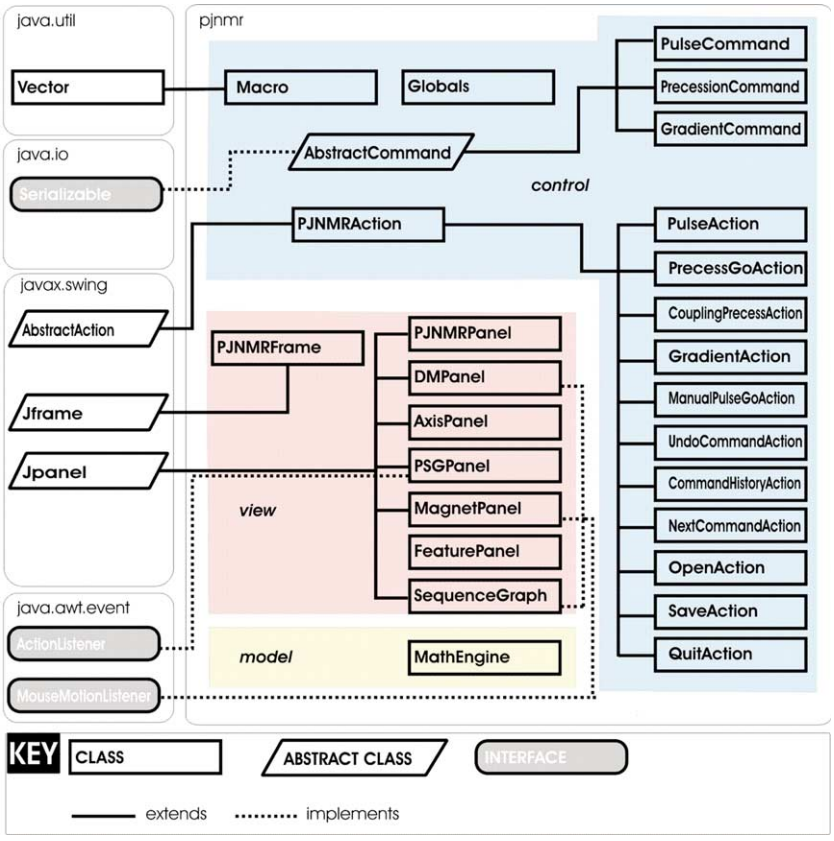

Fig. 1. Standard inheritance diagram for PJNMR, showing the main Java library dependencies (diagram format taken from [7]). See [6] for further reference to Java class inheritance rules.

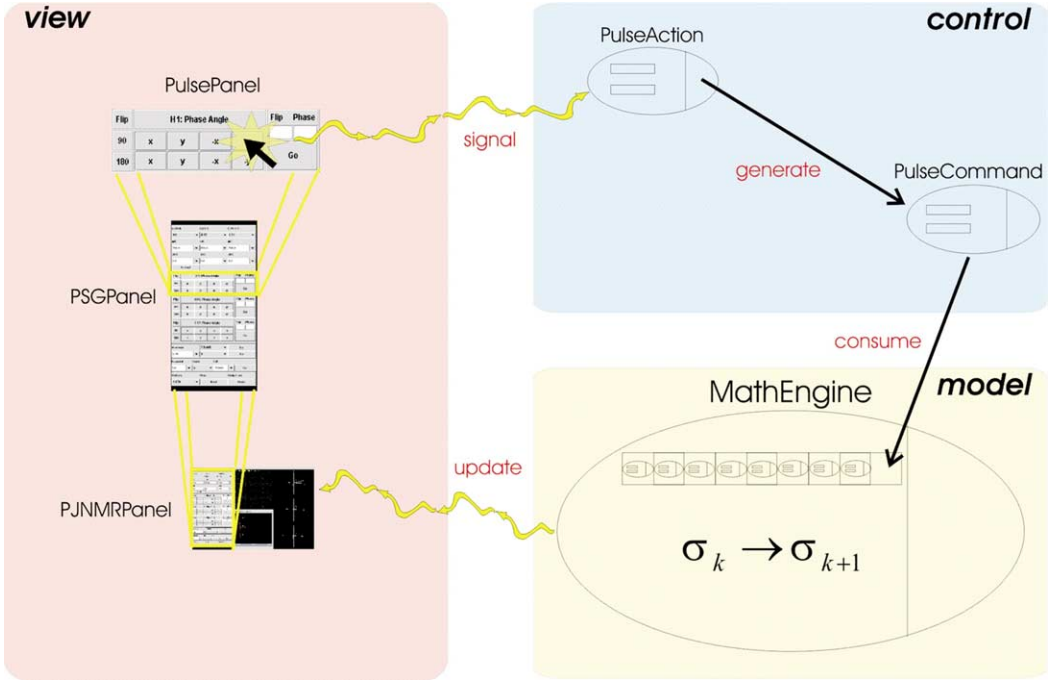

Fig. 2. Action sequence initiated by a pulse command. Objects are grouped into model, view, and control categories.

Table 1
PJNMR gradient parameters

Maximum gradient strength (G_z)	60 G/cm
Number of sample slices (N)	201
Sample length	2.0 cm
Number of gradient level increments (DAC units)	32,000

must be known in order to properly apply the next gradient pulse. This requires that the total (accumulated) transformation applied since the last gradient, U_*, must be stored in memory, as well as the per-slice density matrices immediately after the application of the last gradient, σ_k^i. Upon receiving the second gradient com-

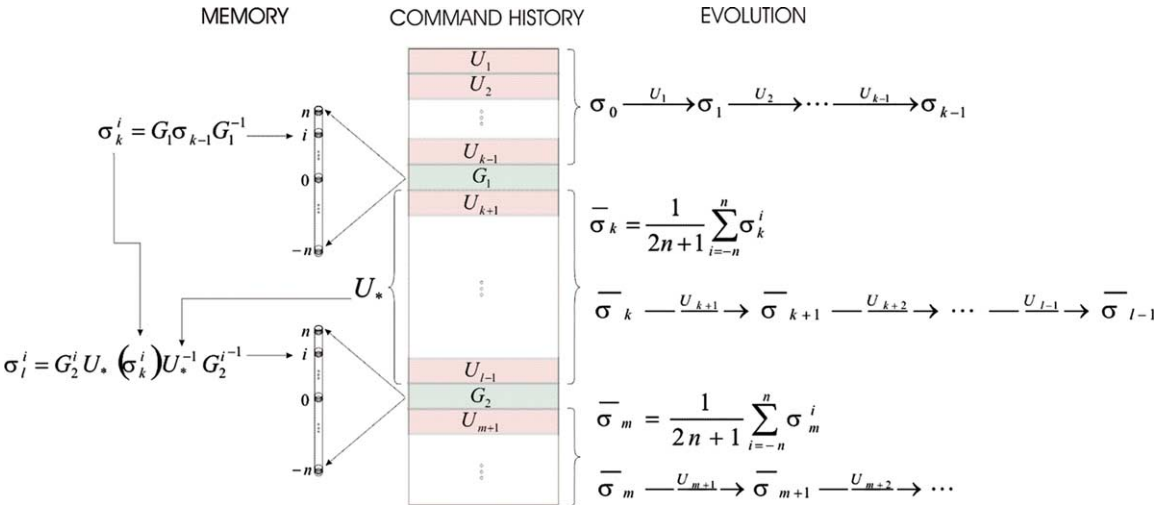

Fig. 3. Overview of transforms in a pulse sequence with gradients.

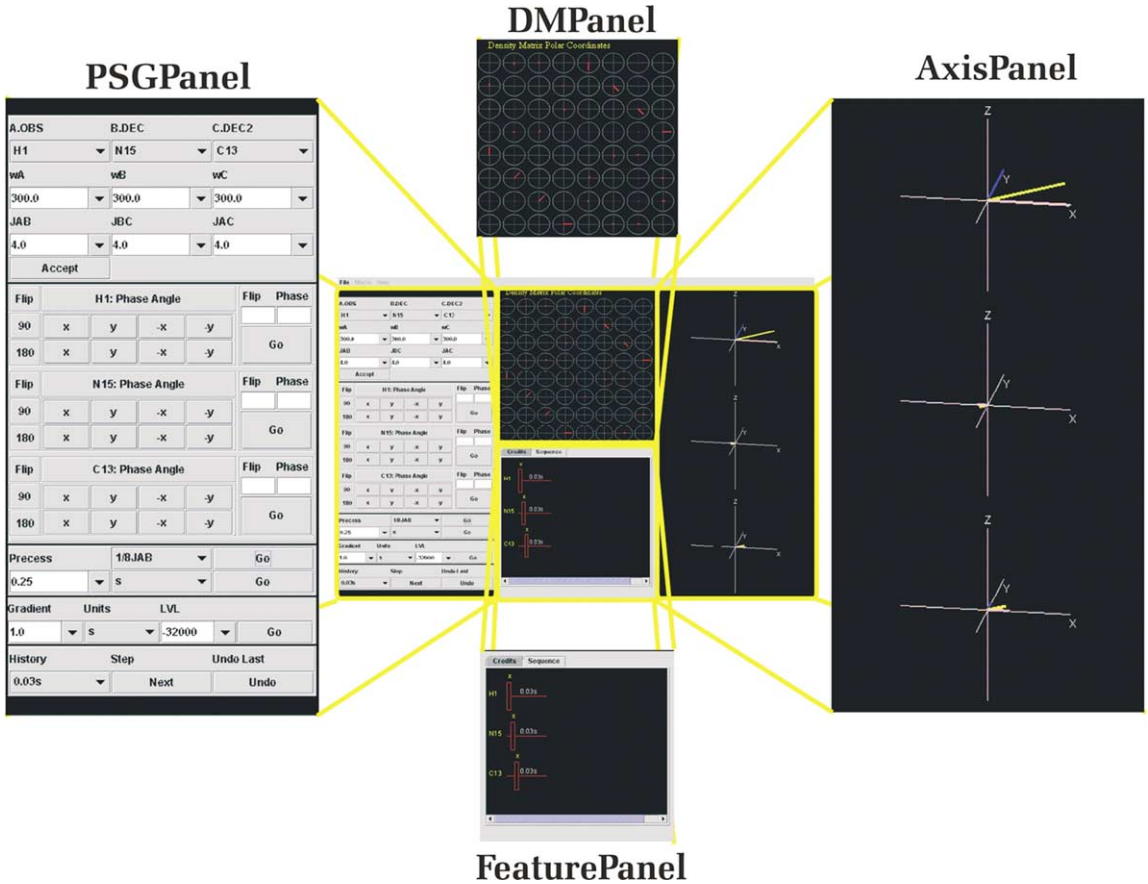

Fig. 4. Graphical interface for PJNMR.

mand G_2, the total accumulated transformation since G_1, U_*, must be applied along with the new gradient operator for slice i as follows

$$\sigma_l^i = G_{2k} U_* \sigma_{l-1}^i U_*^{-1} G_{2k}^{-1}. \tag{30}$$

Here, U_* is given by

$$U_* = U_{m-1} U_{m-2} \cdots U_{k+2} U_{k+1}. \tag{31}$$

3.4. Simulation environment

Fig. 4 shows a screenshot of PJNMR as it appears on the UNIX platform. The separate panels are magnified and labeled with their corresponding object names (shown in Fig. 1). All commands are input on the left-hand panel (''PSGPanel''), and graphical output is given in the three panels to the right (''DMPanel'', ''Axis-Panel'', and ''SequenceGraph''). These three panels display the density matrix, the magnetization vectors for each nucleus, and the pulse sequence as it is generated by user input. A brief description of each panel follows.

3.5. PSG panel

The panel labeled ''PSGPanel'' (*P*ulse *S*equence *G*enerator *Panel*) in Fig. 4 is the 'control center' of the program. All spin system configuration and pulse sequence commands are entered in this panel.

The PSG Panel is divided vertically into five subpanels. The top panel accepts parameters for the spin system, including number and type of nuclei, chemical shifts, and coupling constants. Preset values for chemical shifts and coupling constants can be selected or arbitrary user values entered. The next panel holds buttons for applying selective pulses to the different nuclei in the spin system. A number of preset pulses with flip and phase angles in increments of 90 degrees are available, as well as pulses of arbitrary flip and phase angles. Below the pulse panel is the precession panel, with commands for initiating precessions of arbitrary length, as well as precessions of the form $1/xJ$, where J is a coupling constant and x is one of the integers 2, 4, or 8. The next panel down is for entering gradient commands, where the length of time of the gradient and the power level of the gradient in DAC units is specified. The bottom-most panel on the PSG Panel contains an interactive command history.

Once the spin system configuration is chosen and accepted in the top panel, the remainder of the PSG Panel configures itself to accommodate pulse and precession commands appropriate for the chosen spin system.

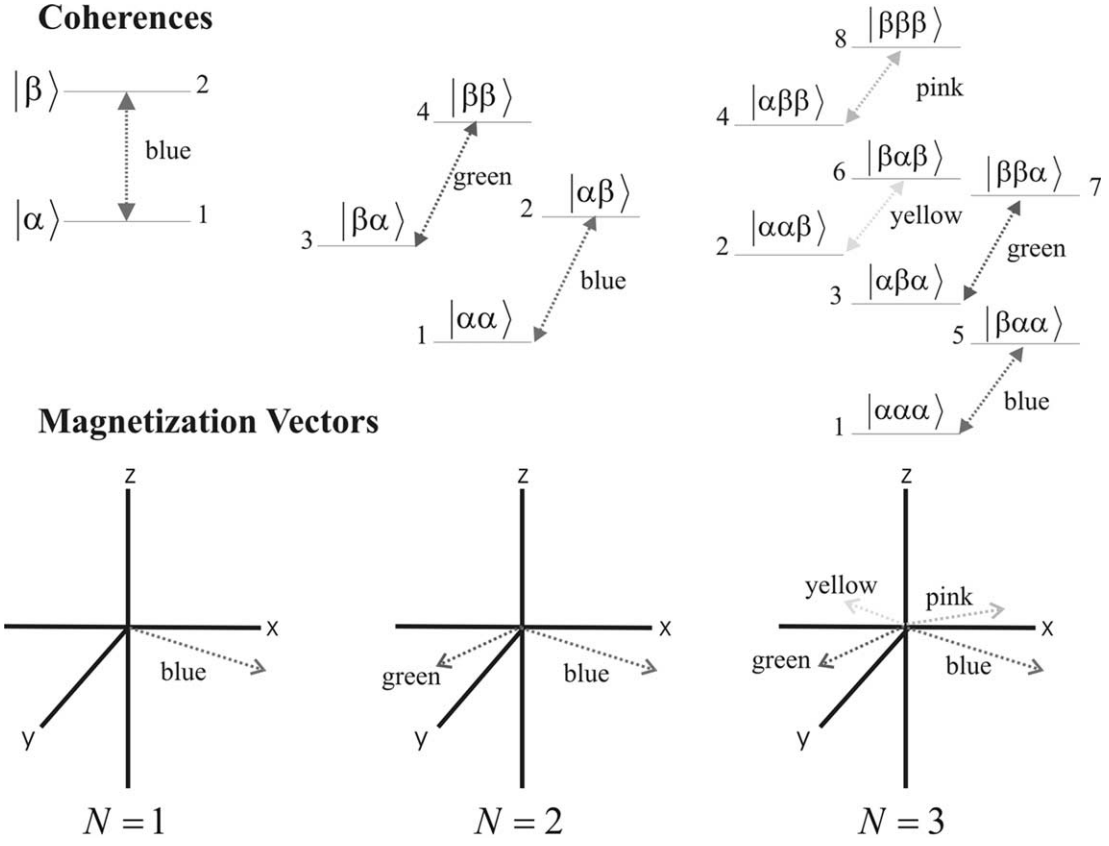

Fig. 5. Correspondence between magnetization vectors and single-quantum coherences in a 1, 2, and 3-spin system. The colors used in PJNMR for the vectors are also shown.

3.6. DM panel

The panel labeled "DMPanel" (*D*ensity *M*atrix *Panel*) in Fig. 4 shows the density matrix for the current state of the spin system. The density matrix elements are imaginary numbers displayed in *polar coordinates*. The numerical value of a given element of the density matrix can be displayed at any time by clicking on the element.

3.7. Axis panel

The panel labeled "AxisPanel" (magnetization *Axis Panel*) in Fig. 4 shows the magnetization vectors for each nucleus in the spin system. The magnetization for each nucleus is displayed as multiple vectors corresponding to the single-quantum coherences it participates in (Eq. (28)). Fig. 5 shows the correspondence between the colored magnetization vectors and their associated coherences for 1, 2, and 3 spins.

3.8. Sequence graph

The panel labeled "SequenceGraph" in Fig. 4 shows the pulse sequence as it is generated by user commands.

4. Results and discussion

Because PJNMR can be run identically on Windows, Macintosh, UNIX, and Linux machines, users have the ability to take the program into almost every area of their work environment. Users may also share PJNMR macro files amongst each other, allowing scientific collaboration across diverse computational environments. These features make PJNMR an ideal educational and professional tool. Early versions of PJNMR are already gaining recognition as a valuable in-class tool for demonstrating NMR concepts, as well as an accurate, fast, and easily understandable simulation program for complex pulse sequence design.

4.1. Example pulse sequences

To illustrate the utility of PJNMR, several pulse sequences will be demonstrated here.

4.1.1. ^1H 90-x

Fig. 6 shows a screen shot of PJNMR after applying a simple 90*x* pulse and 0.33 ms precession on a single proton. The density matrix shows that after the 90it x pulse, single-quantum coherence develops be-

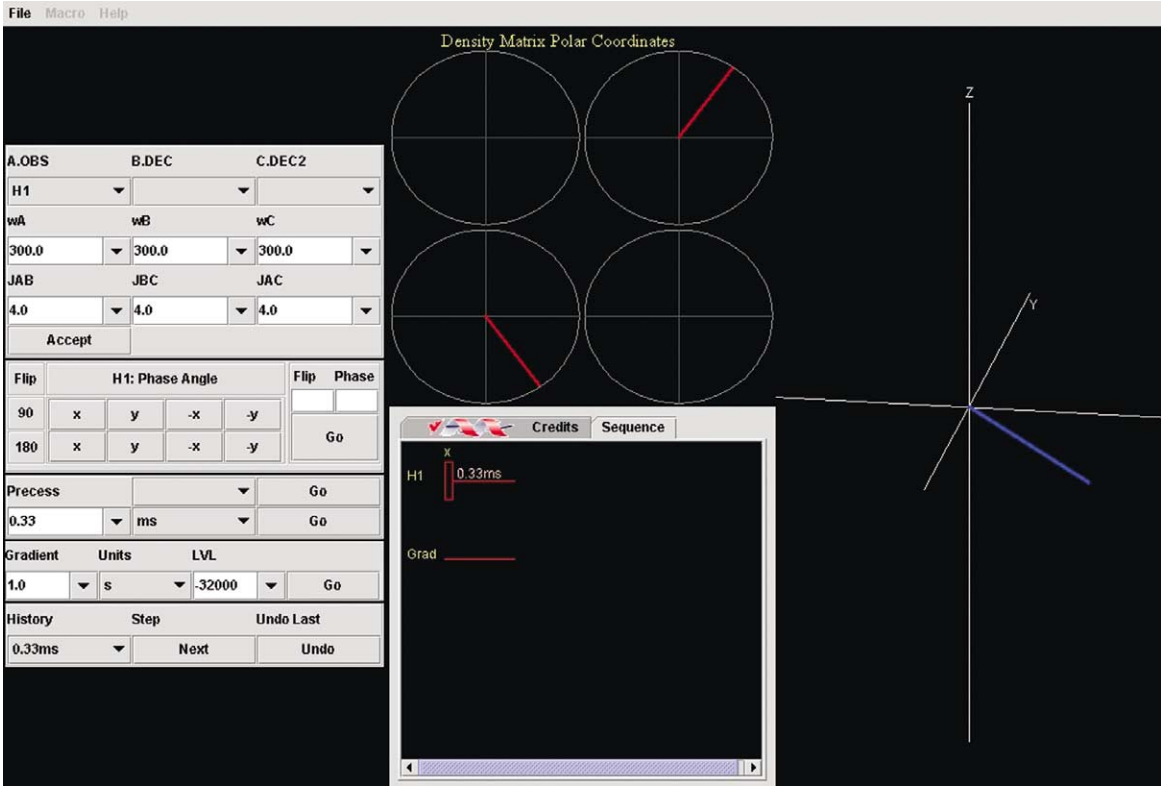

Fig. 6. Screen shot of PJNMR 2.0 showing the state of a single proton after a 90*x* pulse and a 0.33 ms precession.

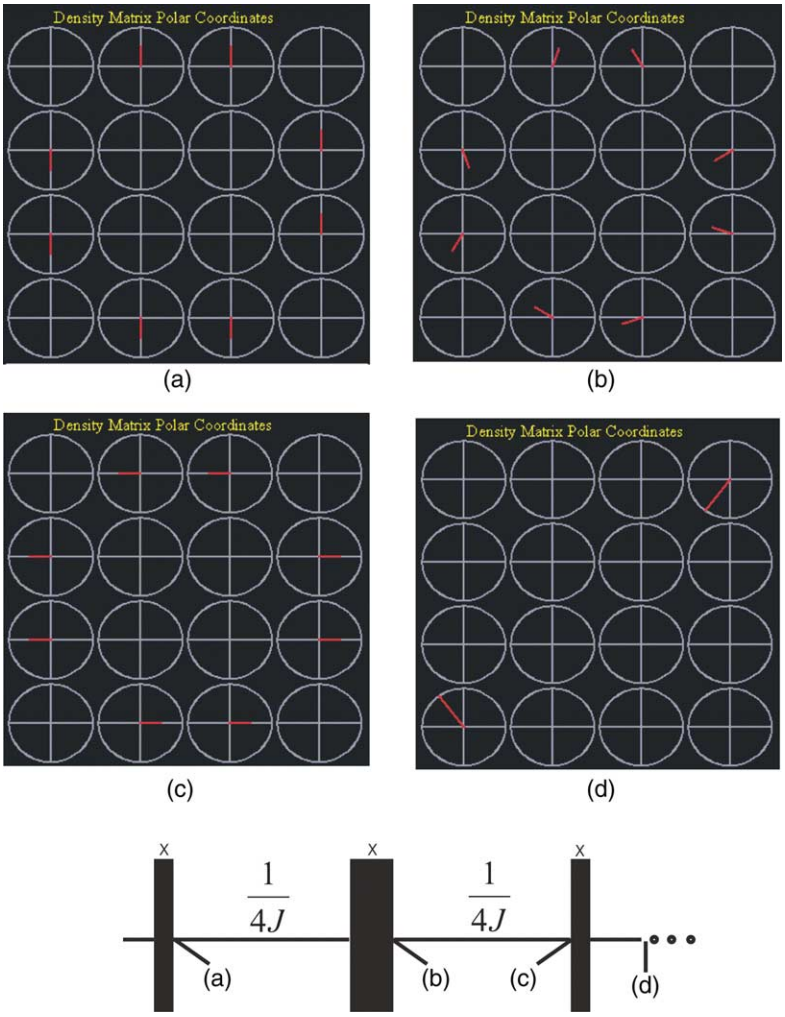

Fig. 7. Screen shots of the density matrix at four stages in a ^{1}H–^{1}H double quantum filter pulse sequence.

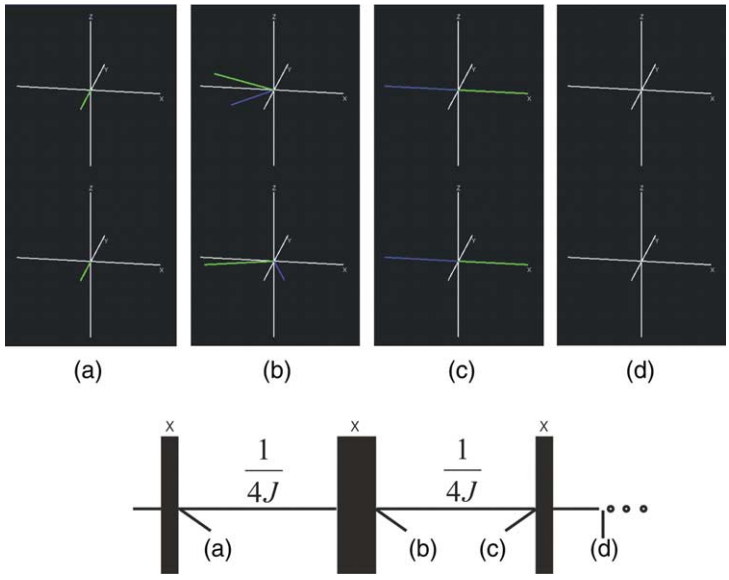

Fig. 8. Screen shots of the magnetization vectors at four stages in a ^{1}H–^{1}H double-quantum filter pulse sequence.

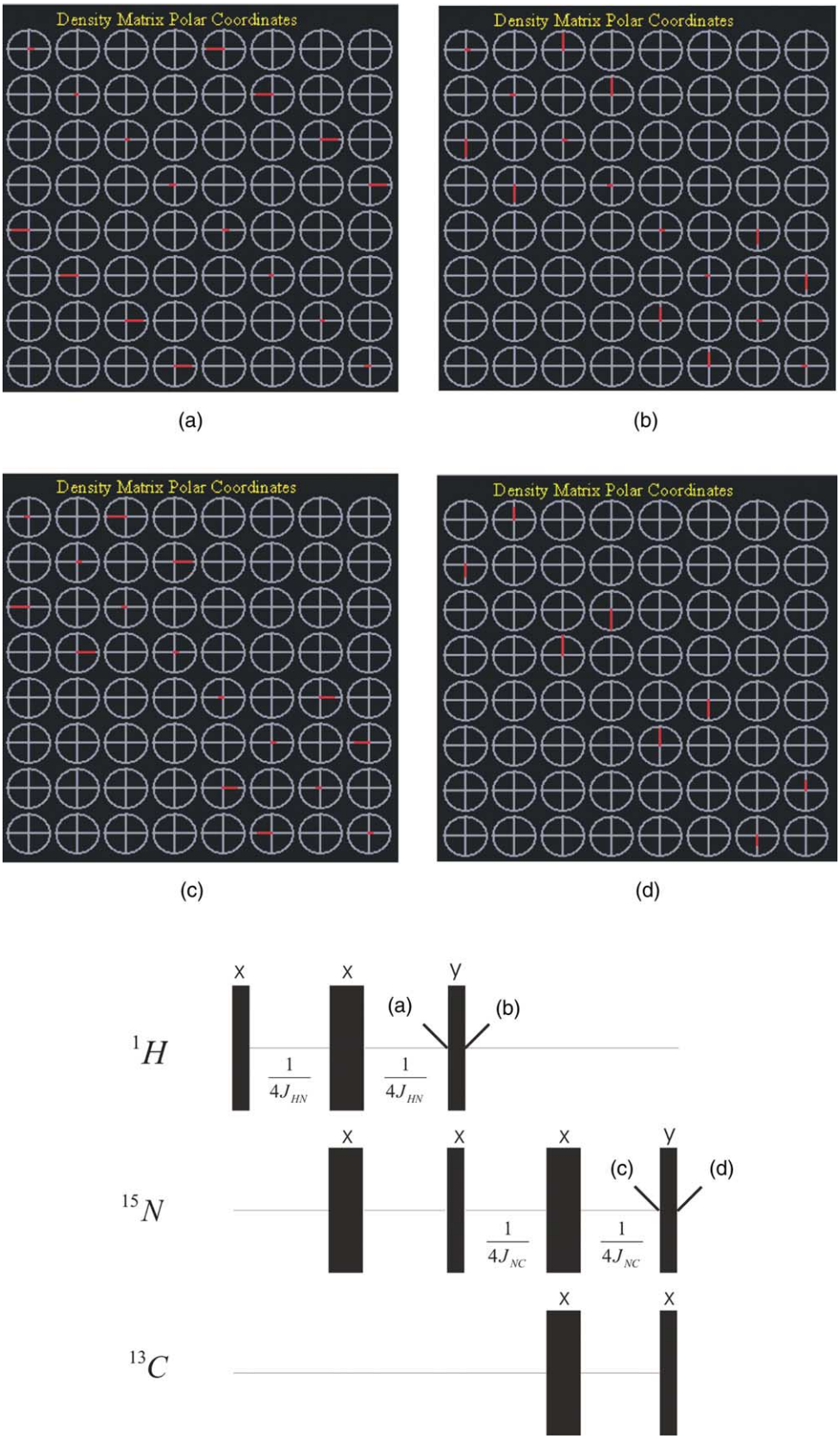

Fig. 9. Screen shots of the density matrix at four stages in a HNCO pulse sequence.

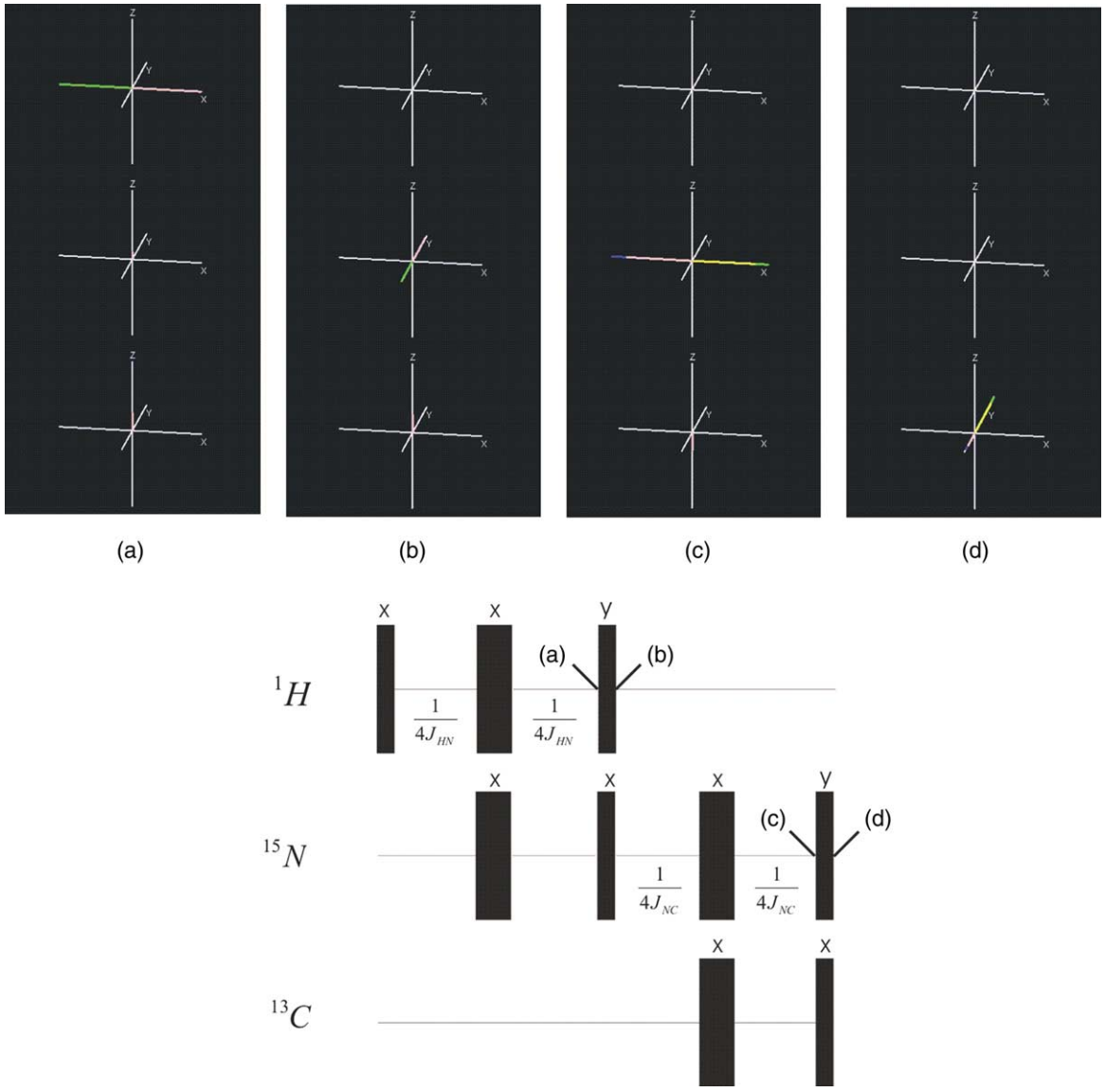

Fig. 10. Screen shots of the magnetization vectors at four stages in a HNCO pulse sequence.

tween the two energy states of the system, indicated by the nonzero values in the off-diagonal elements of the matrix. The single quantum coherence then precesses as the system evolves under the Chemical Shift Hamiltonian, shown by the rotation of the vectors on the matrix and the corresponding rotation of the magnetization vector. The two symmetric off-diagonal elements represent the single coherence present between the up and down states in the spin system. These are complex conjugates of each other and therefore rotate in the opposite sense as the coherence evolves.

4.1.2. $^1H - ^1H$ double-quantum filter (DQF)

Figs. 7 and 8 show a series of screen shots of the density matrix and magnetization vectors during a $^1H - ^1H$ double-quantum filter (DQF) pulse sequence. The indices of the density matrices in Fig. 7 correspond to the numbering of basis states given in Fig. 5(b), as well as the partitions of the I_x operator in Eq. (23).[4] Captions (a)–(c) in Figs. 7 and 8 show that only single-quantum coherences are present in the $^1H - ^1H$ system, characterizing the presence of x-magnetization on both nuclei, until the final $90x$ pulse is given. Caption (d) in both figures shows pure double-quantum coherence precessing after the $90x$ pulse, in the

[4] For example, the density matrix in Fig. 7(a) has a pure imaginary element at index (1,2). Referring to Fig. 5(b), indexes 1 and 2 correspond to the $|\alpha\alpha\rangle$ and $|\alpha\beta\rangle$ basis states, respectively. Therefore, the presence of a nonzero element at index (1,2) in the density matrix in Fig. 7(a) corresponds to a single-quantum coherence between the $|\alpha\alpha\rangle$ and $|\alpha\beta\rangle$ basis states, or x-magnetization for the second proton in the spin system.

absence of any observable magnetization. *This clearly illustrates in graphical form the fundamental quantum-mechanical property that only single-quantum coherences are experimentally observable in NMR.* The polar-coordinate representation of the density matrix also highlights that *all* orders of coherences have phase and magnitude properties, which can be read out either directly or indirectly by suitably chosen pulse sequences.

4.1.3. HNCO

Figs. 9 and 10 show screen shots of the density matrix and magnetization vectors in PJNMR throughout the HNCO pulse sequence. The indices of the density matrices in Fig. 9 correspond to the numbering of the basis states for a 3-spin system shown in Fig. 5(c) (see Footnote 4). The density matrices in Fig. 10 show that as magnetization is transferred from the proton to the nitrogen and finally to the carbon, single-quantum coherences involving each respective nucleus appear. Nonzero elements along the diagonals are present at stages (a) through (c) due to z-magnetization in the spin system, and are not present at stage (d), where only pure anti-phase magnetization on the carbon nucleus remains.

4.1.4. Gradient echo

Figs. 11 and 12 show the density matrices and magnetization vectors throughout a $^1H - {}^1H$ gradient echo pulse sequence. As in the DQF experiment above, single-quantum coherence in this pulse sequence gives way to pure double-quantum coherence. In Fig. 11(b), an initial "crusher" gradient has dephased single-quantum coherences, destroying the x-magnetization present in Fig. 12(a). After a refocussing period, a second identical gradient pulse is applied, rephasing the single-quantum coherences and recovering the x-magnetization, as shown in Figs. 11(c) and 12(c). The final 90x pulse

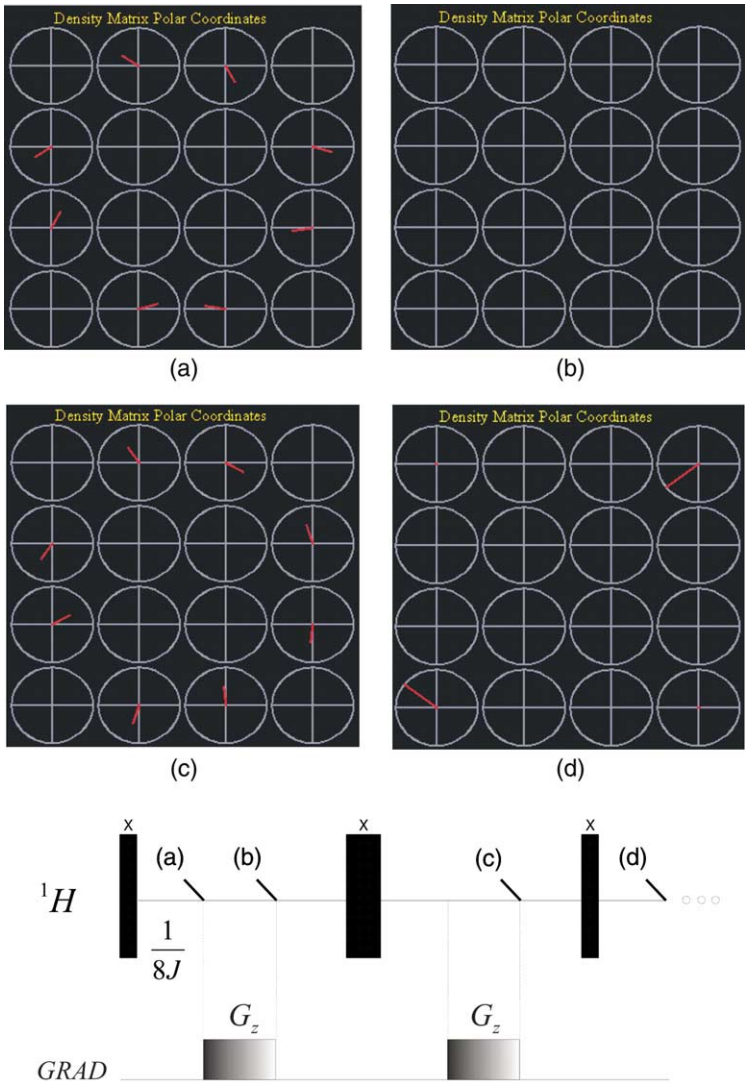

Fig. 11. Screen shots of the density matrix at four stages in a gradient echo pulse sequence.

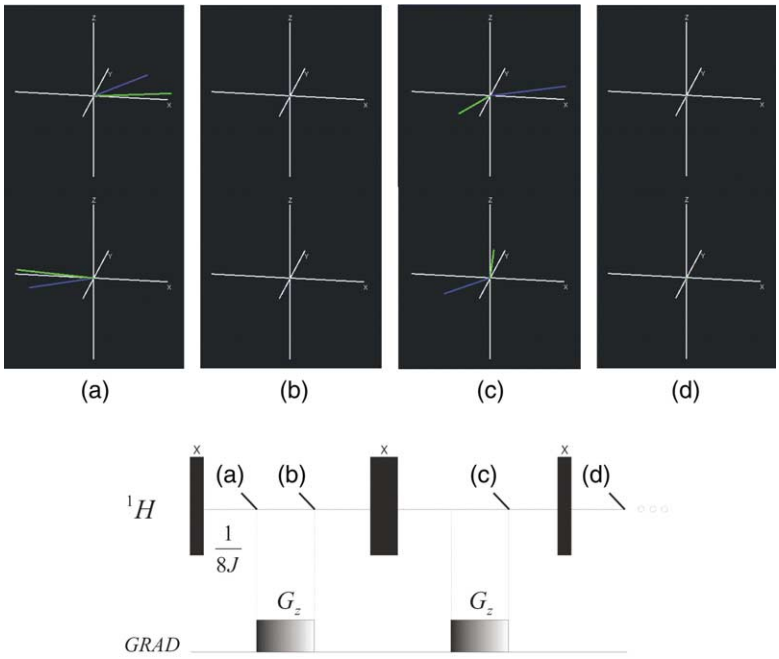

Fig. 12. Screen shots of the magnetization vectors at four stages in a gradient echo pulse sequence.

produces pure double-quantum coherence, which then precesses under chemical shift.

5. Conclusions

PJNMR 2.0 is a new program for pulse sequence simulation on systems of up to 3 weakly-coupled spins-1/2. The program is written in the Java programming language, providing complete platform independence, a rich graphical interface, and an intuitive program design using object-oriented principles. Users learn about the quantum-mechanical and experimentally observable state of the spin system by viewing the density matrix and magnetization vectors at each step in the pulse sequence. Pulses, precessions, and gradients can all be simulated with the program, making it ideally suited for illustration of quantum mechanical concepts in complex pulse sequences. PJNMR 2.0 is available for download at www.nanuc.ca.

Acknowledgments

This work was supported by the Canadian Institutes for Health Research (CIHR), the Protein Engineering Network Centers of Excellence (PENCE), and the National High Field Nuclear Magnetic Resonance Center (NANUC).

References

[1] J. Cavanagh, W.J. Fairbrother, A.G. Palmer III, N.J. Skelton, in: Protein NMR Spectroscopy: Principles and Practice, Academic Press, San Diego, 1996.
[2] P. Nicholas, D. Fushman, V. Ruchinsky, D. Cowburn, Journal of Magnetic Resonance 145 (2000) 262–275.
[3] S.A. Smith, T.O. Levante, B.H. Meier, R.R. Ernst, Journal of Magnetic Resonance Series A 106 (1994) 75–105.
[4] T. Allman, A.D. Bain, J.R. Garbow, Journal of Magnetic Resonance Series A 123 (1996) 26–31.
[5] S.P. Edmondson, Journal of Magnetic Resonance 98 (1992) 283–298.
[6] C.B. Post, R.B. Meadows, D.G. Gorenstein, Journal of the American Chemical Society 112 (1990) 6796–6803.
[7] M. Bak, R. Schultz, T. Vosegaard, N.C. Nielson, Journal of Magnetic Resonance 154 (2002) 28–45.
[8] C.P. Slichter, in: Principles of Magnetic Resonance, Harper & Row, NY, 1963, pp. 23–25.
[9] G.H. Meresi, M. Cuperlovic, W.E. Palke, J.T. Gerig, Journal of Magnetic Resonance 137 (1999) 186–195.
[10] C.S. Horstmann, G. Cornell, Core Java 2, fifth ed., Sun Microsystems Press, Palo Alto, CA, 2001.
[11] D. Flanagan, NetLibrary Inc., O'Reilly, Sebastopol, CA, 1999.