

SPINEVOLUTION 3.4.3 Release Notes

Mikhail Veshtort
August, 2011

1. License/Disclaimer	1
2. New features and functionality	2
2.1 General	2
2.2 New Functionality	2
2.3 New Examples	2
2.4 New Main Input File Formats	2
2.5 New Options	3
2.6 New Internal Variables	3
2.7 New Functions	4
2.8 Obsolete Features	4

1. License/Disclaimer

This license grants a non-exclusive, limited right to install and use SPINEVOLUTION software (version 3.4.3) for both academic and commercial purposes, until it expires on September 1, 2012 (by that date, another version with a later or no expiration date will be available).

In order to fund the development and support of this software, the program may have to be commercialized in the future. In that case, we plan on making a special educational version (with a limited functionality) available free of charge or at a low cost.

The software comes "as is" with no warranty of any kind.

The software cannot be redistributed.

2. New features and functionality

2.1 General

The Reference Manual has been significantly extended and updated (see Summary of the new Features, Extended Molecule, Construction of the Molecule, Variables Section Functionality, Output).

2.2 New Functionality

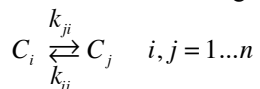
Variable angle spinning (VAS) experiments can be simulated now (see the variable **var_beta_RL** for more details).

Fast exchange between different molecular configurations was implemented (see the description of variables **group_k_phi**, **group_k_R** etc. for more details)

Intermediate regime chemical exchange

Intermediate regime chemical exchange between different molecular configurations can now be simulated. *This functionality was implemented as separate, non-free software. Inquiries regarding purchasing a license should be directed to mveshtort@gmail.com*

The chemical exchange is assumed to take place between n discrete configurations of the spin system as



where k_{ij} and k_{ji} are the rate constants of exchange between configurations C_i and C_j defined by their Hamiltonians H_i and H_j . The state of the system is described by the set of density matrices $\rho_i(t)$ that evolve in time according to equations

$$\frac{d\rho_i}{dt} = -i[H_i(t), \rho_i] + \sum_j k_{ij}\rho_j$$

The initial states of all molecular configurations are identical but may be differently populated, i.e.

$$\rho_i(0) = P_i^0 Q$$

The computation time for systems with chemical exchange will be approximately proportional to the number of exchanging configurations and to the total length of the time evolution sampled. Dependence on the spin system size will be similar to the regular computations (without chemical exchange). However, the chemical exchange computations will be a lot slower.

The full description of the interface is provided with that software.

2.3 New Examples

Two new examples were added. One of them (pdsd) is a simulation of proton-driven spin diffusion (PDS) using the direct method described in our paper accepted for publication in Journal of Chemical Physics this year. In addition to illustrating the method, this example also demonstrates the use of many advanced features of the program designed to facilitate simulation of parts of very large molecules as spin systems. The other example (press) demonstrates the simulation of localization effects in Magnetic Resonance Spectroscopy. It also contains (at the end of the main input file) a very detailed explanation of how this simulation works, which was written for a complete novice SPINEVOLUTION user. With the explanations given, this example may possibly be used as a jump-start even before reading the JMR paper or the Reference Manual (provided that one understands the PRESS sequence from the NMR viewpoint).

2.4 New Main Input File Formats

Nuclei specified at the **exchange_nuclei** line are now interpreted as nuclear indices into the extended molecule rather than into the spin system, as before (see the Extended Molecule section).

Nuclear groups for the **exchange_nuclei**, **bond_len_nuclei**, **bond_ang_nuclei**, **tors_ang_nuclei**, and **groups_nuclei** lines can now be loaded from files.

Comments at the end of the line can now be used in Variables section. The comment must start with a double slash (//) as in C++.

2.5 New Options

-dumpf	Causes the matrix dumping options (-dumprho , -dumpprops , -dumph , -dumpah/e , and -sm) to save the matrices into files, as opposed printing them to the terminal.
-dumpb	Causes the matrix dumping options to output the matrices in the binary format (4-byte floating point numbers packed by columns), which can be loaded into Matlab or other software.
-fft1 -ffti -fftd	By default, before the Fourier transform is applied to the signal, the value of the first point of the signal is corrected by setting it to the average of the first and the last point of the signal. This is a standard procedure, and it is necessary if the discrete FT is intended as an approximation of the continuous FT. These options prevent this correction and should be used whenever the discrete FT is not intended to approximate the continuous FT. -ffti/d prevent the correction in the indirect (direct) dimension only; -fft1 prevents it in both dimensions.
-bc	Apply baseline correction in the directly observed dimension. The correction is computed from the last third of the signal before line broadening.

2.6 New Internal Variables

group_k_O group_k_P group_k_phi group_k_R group_k_T	<p>If any of these variables for group k is specified as a 3 by n matrix (with $n > 1$), then these variables will be interpreted as specifying the set of conformations participating in the (infinitely) fast chemical exchange. All of the group_k_* variables defined for this k (except for group_k_O and group_k_P) are then expected to have the same size (3 by n). These variables will be applied to rotate and translate group k in the same order and manner as described above. The variable group_k_w can be used to specify the weights for this averaging (the weights do not need to be normalized; equal weight are used by default). Variables group_k_O and group_k_P (if used) should be given as 3 by 1 matrices as they are assumed to be common for all the conformations.</p> <p><i>Otherwise, these variables are interpreted as in the previous versions of the program:</i></p> <p>The rotation(s) and the translation to apply to the kth nuclear group specified at the groups_nuclei line. The group is first rotated by the angle group_k_phi, degrees, about the vector \vec{OP}. Then it is rotated about the point group_k_O through the Euler angles group_k_R, degrees, with respect to the Crystal Coordinate System, and then translated by vector group_k_T, Å. These variables must be specified as 3 by 1 (i.e. column) matrices, except for group_k_phi, which is a scalar.</p>
group_k_w	

var_beta_RL	A vector that may be used to specify the values of β_{RL} (spinning axis tilt angle, radians) for the duration of every elementary pulse sequence in Variable Angle Spinning (VAS) experiments; the number of components of the vector is equal to the number of pulse sequences. If the vector is used, all of its components must be specified.
n_d	Number of sampling points in dimension d . This variable is read-only.

2.7 New Functions

tlabel (M, dw)	Generates the “time labeling” column-matrix for M observations performed with dwell time dw .
flabel (M, sw)	Generates the “frequency labeling” column-matrix for a spectrum with M sampling points and spectral width sw .
rand (M, N)	Generates an M by N matrix of zeros of random numbers with a uniform distribution in the interval from 0 to 1.
zeros (M, N)	Generates an M by N matrix of zeros
cross (X, Y)	Cross product of vectors X and Y
diag (X)	If X is a square matrix, this function returns a column-matrix formed by the diagonal of X . If X is a column- or row-matrix, this function returns a diagonal square matrix with the diagonal formed by X .

2.8 Obsolete Features

The `-fastssh` option is now obsolete. In the new version of the program, all communications between the master and the slave processes are carried out in the “fast” mode, and aborting the master *spinev* process should automatically lead to aborting of all slave processes on the remote hosts if they are still accessible.