## Spinach package – input preparation manual

This file goes through the Spinach input options in the order that most people would prepare their inputs. Seeing as the code base keeps evolving, we would encourage the users to look into the actual code for any extra options that might not have made it here yet – *the code was written to be very readable and is extensively commented*.

Please refer to the examples directory – it is very likely that an example calculation of your type is already there and may be used as a template. With very non-standard calculations, it may be a good idea to send an email to Ilya Kuprov (ilya.kuprov@oerc.ox.ac.uk), we are always interested in expanding the example set and would write a template for you.

1. **Spin system and general algorithm options**
2. Specify the spin system composition (*sys.isotopes*) by giving a list of isotope names, for example:

sys.isotopes={'1H','1H','19F','235U'};

An electron may be selected by specifying 'E'. Electron shells with multiplicity higher than 2 may be selected by specifying a multiplicity after the electron label, e.g. ‘E3’ requests a spin-1 electron shell. All known isotopes are supported, you can specify your own particles by editing the *spin.m* file in the kernel. Spinach supports particles of any spin.

Spin system information may also be loaded from a Gaussian03 calculation log – see Section 2.c for further information.

1. If necessary, specify the labels for each spin (*sys.labels*) by giving a list of strings, for example:

sys.labels={'My Proton','Nitrogen','The Pink Electron'};

Labels are printed next to the spin interaction summaries – this makes reading them somewhat easier for larger spin systems. If this field is omitted, all labels are set to empty.

1. If necessary (this is an expert option), alter the tolerance parameters for the internal algorithms (*sys.tols.\**). The full list is given in the *internal\_tolerances.docx* file. The defaults are carefully chosen to provide accurate results in every possible case. Performance may often be improved by relaxing some tolerances, this is particularly true for the spherical averaging grids, where the default is a paranoid accuracy Lebedev grid of rank 131. Several internal algorithms may also be manually switched to alternative pathways (*e.g.* Taylor → Chebyshev sparse exponentiation or Hausdorff → finite-difference propagator differentiation). Examples:

sys.tols.exponentiation=’chebyshev’;  
sys.tols.prop\_zero=1e-16;

Simulations can vary greatly in their tolerance requirements. Several functions (notably, propagator calculation and optimal control modules) run internal checks and would warn you if they detect any accuracy issues.

1. If necessary (another expert option), set the disable switches on the major internal algorithms. The *sys.disable* variable is a cell array of strings that may be set to disable zero track elimination (*‘zte’*), graph-theoretical analysis (*‘pt’*), symmetry factorization (*‘symmetry’*), Krylov subspace propagation (*‘krylov’*), sparse matrix clean-up (*‘clean-up’*) and destination state screening (*‘dss’*). Disabling any of this functionality will lead to a dramatic slowdown and is only provided for debugging purposes.
2. All Matlab console output with the exception of fatal error messages may be redirected to an ASCII text file by setting *sys.output=’file’* or suppressed altogether (useful for large sets of simulations) by setting *sys.output=’hush’*. The default is to print all output to Matlab console. If ASCII file output is selected, the file name (the path defaults to the current directory) should be supplied by setting *sys.logfile* variable:

sys.logfile=’C:\Temp\my\_calculation.log’;

The default is to print all output to Matlab console.

1. A job identifier *sys.identifier* can optionally be supplied to enable restart functionality. Running the job for the second time with the same identifier will cause Spinach to read the operators from the scratch directory instead of re-computing them – with very large spin systems this is often much faster:

sys.identifier=’spectrum\_a’;

If the identifier is not supplied, the restart functionality remains switched off. The cached operators are not transferable between different basis sets.

1. **Interaction specification**
2. Specify the magnet field (*sys.magnet*) in units of Tesla. If no field is specified, zero is assumed.

sys.magnet=14.1;

Indicate whether the calculation is for a liquid, a crystal or a powder (*sys.regime*).

sys.regime=’liquid’;

This has an influence on:

1. how the reduced basis set is built: *'liquid'* builds the reduced basis using the isotropic interaction and distance information, *'crystal'* and *‘powder’* use full interaction and distance information. Because the dipolar coupling network in most systems is very dense and scalar coupling network very sparse, the liquid-state basis sets generated by Spinach are smaller than the solid-state basis sets. See the Basis Selection section of this manual for more information.
2. the behaviour of several pre-programmed pulse sequences: where available, *‘liquid’* produces the isotropic spectrum, *‘crystal’* causes a specific orientation of the spin system to be used in the calculations (the default is the input orientation) and *‘powder’* causes orientation averaging to be applied.

If this field is omitted, the default is used, which is *‘liquid’*. Redfield relaxation theory module would only work with a liquid state system.

1. Zeeman interactions may be specified as scalars, 3x3 matrices or eigenvalues + Euler angles. If multiple specifications are supplied, they are added together.

|  |  |  |
| --- | --- | --- |
| inter.zeeman.eigs  inter.zeeman.euler | 1 × *nspins* cell arrays of 1 × 3 row vectors | Eigenvalues of the chemical shift tensors (in ppm for nuclei) and *g*-tensors (in the Bohr magneton units for electrons) with Euler angles (in radians). Individual cells in the array may be left empty, in which case zeros are assumed. |
| inter.zeeman.matrix | 1 × *nspins* cell array of 3 × 3 matrices | Full chemical shift tensors (in ppm for nuclei) and *g*-tensors (in the Bohr magneton units for electrons) as matrices. Individual cells in the array may be left empty, in which case zeros are assumed. |
| inter.zeeman.scalar | 1 × *nspins*  cell array of reals | Isotropic chemical shifts (in ppm for nuclei) and *g*-factors (in the Bohr magneton units, for electrons). Individual cells in the array may be left empty, in which case zeros are assumed. |

Examples:

inter.zeeman.eigs={[7 15 -22]; [11 18 -29]};

inter.zeeman.euler={[pi/5 pi/3 pi/11]; [pi/6 pi/7 pi/15]};

inter.zeeman.matrix={[5 0 0; 0 5 0; 0 0 5]

[5 0 0; 0 5 0; 0 0 5]

[2.0023 0 0; 0 2.0025 0; 0 0 2.0027]};

inter.zeeman.scalar={2.0023 1.0 2.0 3.0};

1. Spin-spin couplings may be specified as scalars, 3x3 matrices or eigenvalues + Euler angles. If multiple specifications are supplied, they are added together. If spin coordinates are supplied, they would be used to infer the point dipolar couplings. Internally, all interactions are always stored as 3x3 matrices or irreducible spherical tensor coefficients, depending on the context.

|  |  |  |
| --- | --- | --- |
| inter.coupling.eigs  inter.coupling.euler | *nspins* × *nspins*  cell array of 1 × 3 matrices | Eigenvalues of coupling tensors (in Hz) with Euler angles (in radians). Bilinear coupling is introduced by specifying a coupling between two different spins. Quadratic coupling (*e.g.* quadrupolar) is introduced by specifying a coupling between a spin and itself. Individual cells in the array may be left empty, in which case zeros are assumed. |
| inter.coupling.matrix | *nspins* × *nspins*  cell array of 3 × 3 matrices | Full coupling tensors as matrices (in Hz). Each element of the cell array is accounted for, so the couplings must be divided by two if a symmetric cell array is supplied. Individual cells in the array may be left empty, in which case zeros are assumed. |
| inter.coupling.scalar | *nspins* × *nspins*  cell array of reals | Isotropic couplings (in Hz). Individual cells in the array may be left empty, in which case zeros are assumed. |
| inter.coordinates | *nspins* × 1 cell array of 1 x 3 row vectors | Cartesian coordinates of every spin (in Angstroms). If coordinates are provided, they are used to determine point dipolar interaction tensors. If a cell corresponding to a particular spin is left empty, that spin is assumed to not have any dipolar interactions with the rest of the system. |

Examples:

inter.coupling.scalar={0 50; 50 0};

inter.coupling.eigs{2,2}=[1e4 1e4 -2e4];

inter.coupling.euler{2,2}=[0 0 0];

inter.coordinates={[0.0 0.0 0.0]

[0.0 0.0 1.5]};

1. Interaction and spin system information may also be loaded from a Gaussian03 magnetic properties calculation log. The following properties are supported (if found in the log): coordinates, *g*-tensor, hyperfine coupling tensors, chemical shielding tensors, scalar couplings. There are two functions responsible for the Gaussian input processing. The first one (*g03\_parse*) reads every useful piece of information it can find in the log and returns that information as a Matlab data structure:

props=g03\_parse('..\molecules\strychnine.log')

Below is the list of parameters that the parser returns (all tensors are returned in the coordinate system corresponding to the “standard orientation” as returned by Gaussian03:

|  |  |
| --- | --- |
| props.inp\_geom | Input orientation, XYZ, in Angstrom. |
| props.std\_geom | Standard orientation, XYZ, in Angstrom. |
| props.natoms | Number of atoms in the molecule. |
| props.energy | SCF energy, in Hartrees. |
| props.hfc.iso | Isotropic hyperfine couplings, in Gauss. |
| props.hfc.full.eigvals | Eigenvalues of the full hyperfine coupling tensor, in Gauss. |
| props.hfc.full.eigvecs | Eigenvectors of the full hyperfine coupling tensor. |
| props.hfc.full.matrix | Full hyperfine coupling tensors, in Gauss. |
| props.g\_tensor.eigvals | Eigenvalues of the *g*-tensor. |
| props.g\_tensor.eigvecs | Eigenvectors of the *g*-tensor. |
| props.g\_tensor.matrix | Full *g*-tensor. |
| props.cst | Chemical shielding tensors for each spin. |
| props.j\_coupling | Matrix of scalar couplings between each pair of spins. |
| props.symbols | Atomic symbols. |
| props.atomic\_numbers | Atomic numbers. |
| props.filename | Name of the log file. |

The second function (*g03\_to\_spinach*) picks the interactions that the user would like to import into the Spinach calculation:

[sys,inter]=g03\_to\_spinach(props,nuclei,references,options)

where *props* is the output of *g03\_parse* and *nuclei* is a cell array of the following form:

{{'H','1H'},{'N','15N'}...}

giving the list of elements and isotopes that should be included. All other spins will be ignored. References is a vector of reference chemical shifts in ppm. It is usually necessary to run Gaussian on, say, TMS with the same method, and the resulting chemical shift goes here. Computed absolute isotropic shielding values for TMS in vacuum:

GIAO 13C 1H

B3LYP/6-31G\* 189.6621 32.1833

B3LYP/6-311+G(2d,p) 182.4485 31.8201

HF/6-31G\* 199.9711 32.5957

HF/6-311+G(2d,p) 192.5828 32.0710

CSGT 13C 1H

B3LYP/6-31G\* 188.5603 29.1952

B3LYP/6-311+G(2d,p) 182.1386 31.7788

HF/6-31G\* 196.8670 29.5517

HF/6-311+G(2d,p) 192.5701 31.5989

If you have other isotopes in the system, or a different reference substance, you will have to run Gaussian03 for that substance and use the resulting absolute shielding values. If the isotope list described above involves an electron, for example:

{{'E','E'},{'H','1H'}...}

then ESR mode is assumed -- chemical shielding and scalar couplings are ignored, but *g*-tensor and hyperfine couplings are included. A spin-1/2 electron shell is assumed in this case. A spin-1 (for example) electron may be specified by {'E','3E'}.

The following options may be passed to the Gaussian log parser:

|  |  |  |
| --- | --- | --- |
| **Option** | **Units / Values** | **Comments** |
| options.min\_j | Hz | Scalar coupling threshold. *J*-couplings smaller than this value will be ignored. |
| options.min\_hfc | Hz | Hyperfine coupling threshold. Hyperfine coupling tensors with 2-norm smaller than this value will be ignored. |
| options.purge | logical | If this switch is set, the nuclei with negligible hyperfine couplings are purged from the spin system. |
| options.no\_xyz | logical | If this switch is set, the atomic coordinate information is ignored. |
| options.dilute | cell array of strings | All couplings between spins of the types listed in this array are ignored. |

The following fields are written by *g03\_to\_spinach*: *sys.isotopes, inter.coordinates, inter.zeeman.matrix, inter.coupling.matrix, inter.coupling.scalar*.

1. Spin system information may also be read from the spinsys{} field of a SIMPSON \*.in file using the following syntax:

[sys,inter]=simpson2spinach(filename)

The function absorbs isotope information, all Zeeman tensors and all coupling tensors, and returns the corresponding Spinach data structures.

1. For partially oriented systems, the 3x3 Saupe order matrix may be supplied using *inter.order\_matrix* variable. The interaction tensors should then be transformed into their partially averaged from by calling the *residual.m* function in the *etc* directory.
2. **Relaxation theory**

Spinach includes two popular semi-empirical relaxation models as well as a generalized Bloch-Redfield-Wangsness liquid-state relaxation theory module, which operates using the interaction anisotropies and correlation function parameters provided by the user. BRW theory is implemented in a diagonalization-free formulation and is capable of computing full relaxation superoperators for state space dimensions in excess of 105. Please see IK’s JMR paper on the subject for further details.

The semi-empirical relaxation models are the uniform damping of all magnetization at a user-specified rate and the extended T1/T2 approximation, where the longitudinal and transverse single-spin orders decay at user-specified rates (given separately for each spin) and multi-spin orders decay at the sum of the rates of their constituent spins: R1 is used if a spin is in a longitudinal state, R2 is used otherwise.

The following options are supported:

|  |  |  |
| --- | --- | --- |
| inter.relaxation | string:  *‘none’, ’damp’,  ‘t1\_t2’, ‘redfield’*  (default is *‘none’*) | A switch controlling the selection of the relaxation superoperator. The *‘damp’* option requests non-selective damping of all spin states. The ‘t1\_t2’ option requests the extended T1, T2 approximation. The ‘redfield’ option computes the Redfield superoperator. |
| inter.damp\_rate | real | Damping rate (in Hz) to be used if *‘damp’* is selected in the relaxation theory switch. |
| inter.r1\_rate | nspins x 1 vector of reals | R1 relaxation rates (in Hz) for each spin, to be used if *‘t1\_t2’* is selected in the relaxation theory switch. |
| inter.r2\_rate | nspins x 1 vector of reals | R2 relaxation rates (in Hz) for each spin, to be used if *‘t1\_t2’* is selected in the relaxation theory switch. |
| inter.rlx\_keep | string:  *‘diagonal’, ’kite’, ‘secular’, ‘full’*  (default is *‘kite’*) | A switch controlling the terms to be kept in the relaxation superoperator. The *‘kite’* option includes the diagonal relaxation terms and the longitudinal cross-relaxation terms (corresponding to what is known in the trade as the “Redfield kite”. The *‘full’* option keeps all terms and should only be used in laboratory frame simulations. |
| inter.rlx\_dfs | string:  *‘keep’, ‘ignore’*  (default is *‘ignore’*) | A switch controlling the fate of the dynamic frequency shift components of the relaxation superoperator. |
| inter.tau\_c | 1x1 or 1x2 or 1x3  vector of reals | A three-element vector gives the three principal rotational correlation times, corresponding to rotations around X, Y and Z principal axes (in that order) of the rotational diffusion tensor of an ellipsoid in an isotropic liquid. The molecular reference frame must coincide with the eigenframe of the diffusion tensor.  A two-element vector gives the rotational correlation time around the symmetry axis of an axially symmetric ellipsoid (first element) and the correlation time of rotation around an axis perpendicular to the symmetry axis (second element). The Z axis of the molecular reference frame must coincide with the symmetry axis of the diffusion tensor.  A one-element vector gives the isotropic rotational correlation time of what would be assumed to be a spherical molecule. |
| inter.equilibrium | string:  *‘zero’, ‘thermal’* | The state that the system is driven to by the relaxation superoperator. |
| inter.temperature | positive real number | Absolute temperature to be used for the equilibrium state calculation. If the field is left empty or zero is supplied, the “high-temperature approximation” is used automatically. |

A user-specified correlation function may be supplied to the relaxation theory module by editing the correlation\_function.m file in the kernel/utils directory. This is a very advanced option – please examine the Redfield theory code in r\_superop.m before modifying the correlation function.

1. **Chemical kinetics**

*Spinach* can generate superoperators that carry out state population transport within a topologically fixed spin system. In a typical scenario, a molecule (or a system of molecules) with a given spin in position A gets transformed into another molecule (or a system of molecules) with that spin in a position B. If the two systems are declared within the same calculation, the chemical exchange process amounts to state population flux between the two systems. The exchange rate matrix that Spinach accepts should connect the corresponding spins of the two systems with the non-zero elements being equal to the corresponding flux rates.

The second supported situation is for a spin-selective recombination process in radical pairs, where the two-electron singlet state typically disappears faster than the two-electron triplet. The Haberkorn and the Hore-Jones models are supported. The radical pair recombination theory is only applicable to systems with exactly two unpaired electrons, which should be indicated as described in the table below.

The following parameters are recognized by the kinetics module:

|  |  |  |
| --- | --- | --- |
| inter.chem.exchange | *nspins* × *nspins*  matrix of reals | Magnetization transport rates (in Hz). |
| inter.chem.type | *string:*  *‘intermolecular’, ‘intramolecular’* | Controls the fate of multi-spin orders, which are damped in the ‘intermolecular’ case and transported in the ‘intramolecular’ case. |
| inter.chem.rp\_theory | string:  *‘off’, ‘haberkorn’, ’jones-hore’*  *(default is ‘off’)* | Radical pair recombination theory |
| inter.chem.rp\_spins | 1 x 2 vector of integers | Electron spins involved in the recombination reaction. |
| inter.chem.rp\_rates | 1 x 2 vector of reals | Singlet and triplet recombination rates in a radical pair. |

The policy on chemical transport of multi-spin orders is case-dependent. In principle, the coherence with any observer spins must be preserved during chemical exchange, and, so long as we are treating a single system (or an ensemble of systems, which have chemical exchange happening inside them), this should be taken into account. In a typical solution, however, this would lead to the emergence of inter-molecular coherences between identical molecules – a situation that density matrix formalism is ill-equipped to accommodate. Because such coherences are non-observable and the spin is very unlikely to jump back to the exact molecule that it originally came from, the coherences in question are counted as lost and the corresponding contribution to the kinetics superoperator effectively causes relaxation.