What is Spinach and what does it do?

Ilya Kuprov
University of Southampton

Spinach package

- Magnetic resonance theory library for large-scale time-domain simulation work

Ubiquitin: 76 amino acids, 1060 magnetic nuclei, about 60,000 spin-spin couplings

NOESY: needs full Redfield superoperator, including cross-correlations

Downloads, documentation, tutorials, lectures - http://spindynamics.org
Spinach package

- Magnetic resonance theory library for **large-scale** time-domain simulation work
- All types of magnetic resonance (NMR, EPR, MRI, DNP, PHIP, SQUID, etc.)
- + kinetics, diffusion, hydrodynamics, spatial encoding, off-resonance soft pulses, etc.

A

B

frequency, MHz

Downloads, documentation, tutorials, lectures - http://spindynamics.org

Spinach package

- Magnetic resonance theory library for **large-scale** time-domain simulation work
- All types of magnetic resonance (NMR, EPR, MRI, DNP, PHIP, SQUID, etc.)
- Over 600 pages of docs and tutorials, over 100 real-life simulation examples
- +PhD level spin dynamics lecture course (50 hours of video, 200 pages of handouts)

Downloads, documentation, tutorials, lectures - http://spindynamics.org
Spinach package

- Magnetic resonance theory library for large-scale time-domain simulation work
- All types of magnetic resonance (NMR, EPR, MRI, DNP, PHIP, SQUID, etc.)
- Over 600 pages of docs and tutorials, over 100 real-life simulation examples
- Well-annotated open-source code, clear variable names, informative error messages

Code quality and readability enforcement is militant – the validation block in most functions is longer than the science block.

function fid=hyscore(spin_system,parameters,H,R,K)
% Check consistency
grumble(spin_system,parameters,H,R,K);
% Compose Liouvillian
L=H+1i*R+1i*K;
% Get the pulse operators
Lp=operator(spin_system,'L+','E');
Lm=operator(spin_system,'L-','E');
Lx=(Lp+Lm)/2;
% Calculate timestep and number of steps for tau evolution
[tau_dt,tau_np]=stepsize(L,parameters.tau);
% Apply the first pulse
rho=step(spin_system,Lx,parameters.rho0,pi/2);
% Run the tau evolution
rho=evolution(spin_system,L,[],rho,tau_dt,tau_np,'final');
% Apply the second pulse
rho=step(spin_system,Lx,rho,pi/2);
% Apply coherence filter
rho=coherence(spin_system,rho,{{'E',0}});
% Run the indirect dimension evolution
rho_stack=evolution(spin_system,L,[],rho,1/parameters.sweep,...
parameters.nsteps(1)-1,'trajectory');

We do not actually need to open any of the Kronecker products in spin physics.

\[ [A \otimes B]v = \text{vec}(BVA') \]
\[ \text{exp}(A)v = \sum_{n=0}^{\infty} \frac{1}{n!} (A(A...(A(Av)))) \]
Spinach package

- Magnetic resonance theory library for **large-scale** time-domain simulation work
- All types of magnetic resonance (NMR, EPR, MRI, DNP, PHIP, SQUID, etc.)
- Over 600 pages of docs and tutorials, over 100 real-life simulation examples
- Well-annotated open-source code, clear variable names, informative error messages
- Parallel processing, GPU support, tensor structured object support
- Over 50 developers and contributors, 12 years of full-time programming

Downloads, documentation, tutorials, lectures - http://spindynamics.org

---

Last 6 months: classical degrees of freedom

The evolution happens in the direct product of spin and lab spaces:

\[ \text{dim} \approx 1000 \times 100 \times 100 \times 100 \]

However... all terms in the evolution generator have a kron structure:

\[ [\text{space dynamics}] \otimes [\text{reaction kinetics}] \otimes [\text{spin dynamics}] \]

...and the components are krons themselves, e.g.

\[
D \left( \frac{\partial^2}{\partial t^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) = D \left( \frac{\partial^2}{\partial x^2} \right) \otimes \mathbf{1}_y \otimes \mathbf{1}_z + \frac{\partial^2}{\partial y^2} \otimes \mathbf{1}_x \otimes \mathbf{1}_z + \mathbf{1}_y \otimes \mathbf{1}_x \otimes \frac{\partial^2}{\partial z^2} \right)
\]

It is only when you multiply the krons out that you run out of memory.
Last 6 months: classical degrees of freedom

Chemical degrees of freedom are another kron:

\[
\frac{d}{dt} \rho_{\text{a}} = \left[ -i \left( \begin{array}{cc} H_A & 0 \\ 0 & H_B \end{array} \right) + \left( \begin{array}{cc} R_A & 0 \\ 0 & R_B \end{array} \right) + \left( \begin{array}{cc} -k_{\text{A}} & 1 \\ 1 & -k_{\text{B}} \end{array} \right) \right] \rho_{\text{a}}
\]

The result is a sum of krons repeatedly acting on a vector:

\[
\rho(t + dt) = \exp \left\{ -i \left[ \text{a sum of krons} \right] dt \right\} \rho(t)
\]

\[
\exp[A]v = \sum_{n=0}^{\infty} \frac{1}{n!} A \left( A(Av) \right)
\]

A short sum of krons of small matrices! Need a product with a vector...

\[
\dim[A] = 1000 \\
\dim[B] = 1000 \\
\dim[A(x)B] = 10^4 \\
\text{numel}(v) = 10^6
\]

\[
[AB]v = \text{vec} \left[ BVA^T \right]
\]

\[
\text{dim}[A] = 1000 \\
\text{dim}[B] = 1000 \\
\text{dim}[V] = 1000
\]

We do not actually need to open any Kronecker products in spin dynamics...

---

Last 6 months: classical degrees of freedom

Synthetic benchmark (random matrices with typical NMR density):

<table>
<thead>
<tr>
<th>Matrix-vector multiplication task</th>
<th>Wall clock time, polyadic rep</th>
<th>Wall clock time, explicit rep</th>
</tr>
</thead>
<tbody>
<tr>
<td>([AB]v) with ([A,B] \leq 64), full</td>
<td>0.37 ± 0.01 ms</td>
<td>0.68 ± 0.12 ms</td>
</tr>
<tr>
<td>([ABC]v) with ([A,C] \leq 64), full</td>
<td>1.5 ± 0.3 ms</td>
<td>Out of RAM</td>
</tr>
<tr>
<td>([ABC]v) with ([A,C] \leq 64), full</td>
<td>17 ± 14 ms</td>
<td>Out of RAM</td>
</tr>
<tr>
<td>([AB]v) with ([A,B] \leq 64), sparse</td>
<td>0.21 ± 0.01 ms</td>
<td>0.05 ± 0.01 ms</td>
</tr>
<tr>
<td>([ABC]v) with ([A,C] \leq 64), sparse</td>
<td>2.1 ± 0.3 ms</td>
<td>11.4 ± 1.6 ms</td>
</tr>
<tr>
<td>([AB]v) with ([A,B] \leq 64), sparse</td>
<td>105 ± 14 ms</td>
<td>Out of RAM</td>
</tr>
</tbody>
</table>

Computer:
32 Xeon cores
256 GB of RAM

Not faster for small systems, but scales much better.

2D and 3D localised NMR excitation with an explicit shaped pulse under a field gradient, a typical metabolite (6 spins).

with Ahmed Allami and Maria Grazia Concilio
Last 6 months: classical degrees of freedom

Result: arbitrary spatial dynamics with quantum description of spin.

All other packages (MRI, DOSY, etc.) use Bloch equations in the spin subspace.

with Ahmed Allami and Pavan Lally

Last 6 months: classical degrees of freedom

Rotenone (22 spins), three-dimensional diffusion, complicated chirps, etc.

First dimension: UF SPEN
Second dimension: DOSY
Third dimension: crushers

Simulation time: hours – we are done!

with Jean-Nicolas Dumez and Ludmilla Guduff
Spinach package

- Latest version of Matlab is strongly recommended.

**Spinach capabilities**

- Long-lived state detection
- Tensor structured formats
- Spin system trajectory analysis
- Coupling tensor visualization
- General rotation matrix
- Isotropic, axial and tensorial diffusion correlation functions
- Hilbert space, Liouville space, Fock/Purcell space
- Accurate thermal equilibria and thermalization
- Polynomial complexity scaling in liquid state NMR
- Parallelization and GPU support
- Arbitrary user-defined pulse sequences
- Common 1D and 2D experiments
- NMR, INNIR, and EPR
- SLE and FP relaxation theories
- High-field and multi-echo systems
- Spin Chemistry experiments

- Spin Hamiltonian
- Thermal equilibrium state
- Relaxation superoperator
- Exponential propagator
- System trajectory
- Project out the observables

**Time domain simulation flowchart**

1. Gather spin system, instrument and experiment parameters
2. Generate spin Hamiltonian
3. Generate relaxation superoperator
4. Generate exponential propagator
5. Obtain system trajectory
6. Project out the observables

**Spinach is not a black box — it is an open-source Matlab library of infrastructure functions.**

**If you are using Spinach, this is the only thing you would need to do manually.**
What you need to provide

Zeeman interactions

\[ \hat{H} = \hat{H}_Z + \hat{H}_{\text{NN}} + \hat{H}_{\text{EN}} + \hat{H}_{\text{EE}} + \hat{H}_{\text{MW}} \]

electron-nuclear interactions

microwave and radiofrequency terms

inter-nuclear and quadrupolar interactions

inter-electron interactions and zero-field splitting

Zeeman interactions: chemical shielding tensors for nuclei and \( g \)-tensors for electrons.

Where to get: from the literature or from quantum chemistry packages (Gaussian, CASTEP, ORCA, etc.).

\[ \hat{H}_Z = \sum_k \tilde{B}_0 \cdot \mathbf{A}_E^{(k)} \cdot \hat{E}^{(k)} + \sum_k \tilde{B}_0 \cdot \mathbf{A}_N^{(k)} \cdot \hat{N}^{(k)} \]

GIAO DFT B3LYP/cc-pVTZ or similar is generally accurate for small CHNO molecules.

N.B.: Despite the common "scalar coupling" moniker, \( J \)-coupling is actually a tensor too.
What you need to provide

- Zeeman interactions
- Electron-nuclear interactions: isotropic (aka Fermi contact) and anisotropic hyperfine couplings.
- Microwave and radiofrequency terms
- Inter-electron interactions
- Inter-nuclear and quadrupolar interactions
- Inter-electron interactions and zero-field splitting

### Electron-nuclear interactions: isotropic (aka Fermi contact) and anisotropic hyperfine couplings.

Where to get: literature or DFT (requires specialized basis sets). For remote electron-nuclear pairs (10 Angstroms or more), Cartesian coordinates.

\[
\hat{H}_{EN} = \sum_{j,k} \hat{E}_j \cdot \hat{\mathbf{A}}_{EN}^{(j,k)} \cdot \hat{N}_k
\]

Note the strong directionality of some HFC tensors.

\[ N.B.: \text{“anisotropic hyperfine” and “electron-nuclear dipolar” interactions are the same thing.} \]

What you need to provide

- Zeeman interactions
- Electron-nuclear interactions
- Microwave and radiofrequency terms
- Inter-electron interactions
- Inter-nuclear and quadrupolar interactions
- Inter-electron interactions and zero-field splitting

### Inter-electron interactions: exchange interaction, zero field splitting, inter-electron dipolar interactions.

Where to get: literature or DFT for exchange coupling and ZFS. Dipolar couplings are most conveniently extracted from Cartesian coordinates of the spins.

\[
\hat{H}_{EE} = 2 \pi \sum_{j<k} J^{(j,k)}_{EE} \left( \hat{E}_j \cdot \hat{\mathbf{E}}_k \right) + \sum_{j,k} \hat{E}_j \cdot \hat{\mathbf{A}}^{(j,k)}_{ZFS} \cdot \hat{E}_k - \frac{\mu_0}{4\pi} \sum_{j<k} \left( \frac{r_{jk}^{(j,k)} h}{r_{jk}^3} \right) \left( 3(\hat{E}_j \cdot \hat{r}_{jk})(\hat{E}_k \cdot \hat{r}_{jk}) - r_{jk}^2 (\hat{E}_j \cdot \hat{E}_k) \right)
\]

\[ N.B.: \text{the practical accuracy of DFT for exchange coupling and particularly ZFS is very low.} \]
What you need to provide

Zeeman interactions

\[ \hat{H} = \hat{H}_Z + \hat{H}_{NN} + \hat{H}_{EN} + \hat{H}_{EE} + \hat{H}_{MW} \]

electron-nuclear interactions

microwave and radiofrequency terms

inter-nuclear and quadrupolar interactions

inter-electron interactions and zero-field splitting

Microwave and radiofrequency terms: amplitude coefficients in front of the \( L_x \) and \( L_y \) terms in the Hamiltonian.

Where to get: from the pulse calibration curves of the instrument. The RF/MW power (in Hz) is equal to the reciprocal width of the 360-degree pulse.

\[ \hat{H}_{MW} = \cos(\omega_{MW} t) \sum_k a_{MW}^{(k)} \hat{E}^{(k)}_X \]

(N.B.: the direction of the \( B_1 \) field in most MAS experiments is parallel to the spinning axis.)

Time domain simulation mathematics

\[ \hat{H}(t) = \hat{H}_0 + \sum a_k(t) \hat{H}_k \]

\[ \hat{L}(t) = \hat{H}(t) + i\hat{R} + i\hat{K} \]

\[ \hat{\rho}(t + dt) = \exp[-i\hat{L}(t)dt] \hat{\rho}(t) \]

\[ \hat{\rho}_m = \frac{\exp(-\hat{H}_0/kT)}{\text{Tr}[\exp(-\hat{H}_0/kT)]} \]

Spinach kernel provides all operators and states

The mathematics is quite simple: apply this equation with some small step \( dt \) until you are done!

\[ f(t) = \langle \hat{f} | \hat{\rho}(t) \rangle \]

\[ \bar{f} = \frac{1}{|\Omega|} \int f(\alpha, \beta, \gamma) d\Omega \]

Powder averages are slow, but not hard.

N.B.: this mathematics is hidden from casual users, but the code is open source.
**Spinach architecture**

Simple NMR simulations in *Spinach*

```plaintext
% Spin system
sys.magnet=3.4;
sys.isotopes={'1H', '1H'};

% Zeeman interactions
inter.zeeman.scalar=[1.5 2.4];

% J-couplings
inter.coupling.scalar=[0.0 7.4 0.0 0.0];

% Coordinates
inter.coordinates=[[0.0 0.0 0.0] [0.0 2.0 0.0]];

% Simulation formalism
bas.formalism='sphten-liouv';
bas.approximation='none';

% Relaxation theory
inter.relaxation='redfield';
inter.equilibrium='levitt';
inter.rlx_keep='secular';
inter.temperature=298;
inter.tau_c=10^-12;
```

Spinach has a very detailed input checker – if something is amiss, it would tell you.
Simple NMR simulations in Spinach

% System specification
sys.magnet=9.4;
sys.isotopes={'14N'};
inter.coupling.eigs={[-1e6 -2e6 3e6]};
inter.coupling.euler={[0.0 0.0 0.0]};

DOR and MAS quadrupolar NMR

Powder averaging is an expensive operation, but it runs in parallel.

Spinach developer team

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution</th>
<th>Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ahmed Allami</td>
<td>University of Southampton</td>
<td>SPEN module</td>
</tr>
<tr>
<td><a href="mailto:tutor@ucsf.edu">tutor@ucsf.edu</a></td>
<td>University of California, San Francisco</td>
<td>SPEN module</td>
</tr>
<tr>
<td>Hristina Armoni</td>
<td>Harvard University</td>
<td>Protein and RNA modules</td>
</tr>
<tr>
<td>Antonio Biscarini</td>
<td>UCL</td>
<td>SpinsIML and BiNMR</td>
</tr>
<tr>
<td>Anastasia Boston</td>
<td>Harvard University</td>
<td>Protein and RNA modules</td>
</tr>
<tr>
<td>Alice Bowser</td>
<td>University of Oxford</td>
<td>Pulsed digital EPR sequences</td>
</tr>
<tr>
<td>Luca Brinda</td>
<td>University of Cambridge</td>
<td>NMR data reduction</td>
</tr>
<tr>
<td>Marko Cumerotti</td>
<td>University of Southern California</td>
<td>Overtone and solid state NMR module</td>
</tr>
<tr>
<td>Gareth Durnin</td>
<td>University of Oxford</td>
<td>PINET module, NMR experiments</td>
</tr>
<tr>
<td>Tim Gaitzoff</td>
<td>University of Oxford</td>
<td>Subroutine theory examples</td>
</tr>
<tr>
<td>Marketa Cincio</td>
<td>University of Southern California</td>
<td>ESR examples, SPEN NMR module</td>
</tr>
<tr>
<td>Marketa Cincio</td>
<td>University of Southern California</td>
<td>continuous NMR examples</td>
</tr>
<tr>
<td>Ben Cooper</td>
<td>Gothenburg University</td>
<td>SNP examples</td>
</tr>
<tr>
<td>Sergey Dolgikh</td>
<td>University of Bath</td>
<td>Human brain modules</td>
</tr>
<tr>
<td>Jean-Michel Dumez</td>
<td>Paris-Sud University</td>
<td>Ultraslow NMR experiments</td>
</tr>
</tbody>
</table>

SPEN module: Ahmed Allami, Maria Grazia Concilio