

Computer practical: solid state NMR simulations in *Spinach*

This practical is about getting started with solid state NMR simulations in *Spinach* – you will learn basic spin system specification syntax and run a number of simulations (static powder, magic angle spinning, *etc.*). You will find a large number of other examples in *examples/nmr_solids* directory, and a number of pulse sequences in *experiments* directory.

In the examples below, feel free to modify the various parameters and settings to bring the simulation closer to your actual usage scenarios.

1. Installation

If *Spinach* is not installed already, download the latest version from <http://spindynamics.org>, unpack the zip file and follow the installation instructions. It is important to set *Matlab* path correctly.

2. Static dipolar patterns

- (a) Start Matlab and type 'edit hello_world.m'. This will open the code editor and create an empty file with 'hello_world.m' as the name. At the top of the file type 'function hello_world()', make a few empty lines and put 'end' somewhere underneath.

Never run *Spinach* simulations without this 'function – end' pair. Its purpose is to make sure that variables are cleared when the simulation finishes. Having variables linger in the workspace is not a good idea because of the inevitable confusion it causes when forgotten variables have some effect.

- (b) Specify the typical NMR magnet (600 MHz, 14.1 Tesla) and a pair of protons:

```
% System specification
sys.magnet=14.1;
sys.isotopes={'1H','1H'};
```

- (c) Pick random chemical shifts in ppm and some reasonable coordinates in Angstrom:

```
inter.zeeman.scalar={5.0,-2.0};
inter.coordinates=[0 0 0]; [0 3.9 0.1];
```

Spinach supports a variety of spin system specification methods. In this case, it will work out the dipolar interaction tensors from the coordinates you have provided. It is also possible to specify interaction tensors directly as matrices, or as eigenvalues + Euler angles, or as quadrupolar parameters with Euler angles, *etc.* – see the online manual for further information.

- (d) Specify formalism and basis set. *Spinach* runs most efficiently in Liouville space when the basis operators are irreducible spherical tensors (particular combinations of Pauli matrices with neat rotation properties):

```
% Basis set
bas.formalism='sphten-liouv';
bas.approximation='none';
```

- (e) Call the housekeeping functions that perform input data analysis and pre-processing:

```
% Spinach housekeeping
spin_system=create(sys,inter);
spin_system=basis(spin_system,bas);
```

Run the file. Both housekeeping functions will print copious output to *Matlab* console. Read it carefully, it contains a detailed summary of what *Spinach* thinks the user has supplied. If the input data is found to be inconsistent, an informative error message will be printed to the console in red.

At this point, we have supplied the spin system information to the program. It now needs to know the experiment parameters. We will use the *acquire* experiment (which simply records the FID) within the *powder* context (which handles powder averages and rotating frames).

- (f) Take a good look at the online manual pages for *acquire.m* and *powder.m* functions. They support a great number of parameters, of which we will only need a few:**

```
% Sequence parameters
parameters.spins={'1H'};
parameters.rho0=state(spin_system,'L+', '1H');
parameters.coil=state(spin_system,'L+', '1H');
parameters.decouple={};
parameters.offset=1000;
parameters.grid='rep_2ang_400pts_sph';
parameters.sweep=12000;
parameters.npoints=128;
parameters.zerofill=512;
parameters.axis_units='ppm';
parameters.invert_axis=1;

% Working spins
% Initial state
% Detection state
% Spins to decouple
% Transmitter offset (Hz)
% Powder grid
% Sweep width, Hz
% Number of time steps
% FID zerofilling
% Axis units for plotting
% Invert X axis
```

Parameter names are all self-explanatory. Note that any signals that end up outside the sweep width will get reflected back – a simulation cannot impose a frequency bandpass filter of the same kind that the spectrometer does. The offset is with respect to the chemical shift of 0 ppm. Note that initial state is transverse magnetisation – our current experiment (*acquire.m*) only performs signal detection.

- (g) Call the context function and the pulse sequence using 'nmr' as the assumption set:**

```
% Simulation
fid=powder(spin_system,@acquire,parameters, 'nmr');
```

This takes the *spin_system* object that the housekeeping functions made, the pulse sequence function handle (the file itself is in the experiments directory), the parameters we have supplied, and the name of the assumption set, and feeds all of that to the *powder.m* context function. If you look inside it (put the cursor on it and press Ctrl-D), you will see a lot of scene setting code that sets rotating frames, loads the powder grid, etc. The context function returns the powder averaged free induction decay.

- (h) Run the apodisation, the Fourier transform, and plot the spectrum:**

```
% Apodization
fid=apodization(fid, 'exp-1d', 6);

% Fourier transform
spectrum=fftshift(fft(fid,parameters.zerofill));

% Plotting
figure(); plot_1d(spin_system,real(spectrum),parameters);
```

The details of the three commands may be found in the online manual, or in the headers of the corresponding functions (put the cursor on the function and press Ctrl-D). You will see a picture similar to Figure 1 below.

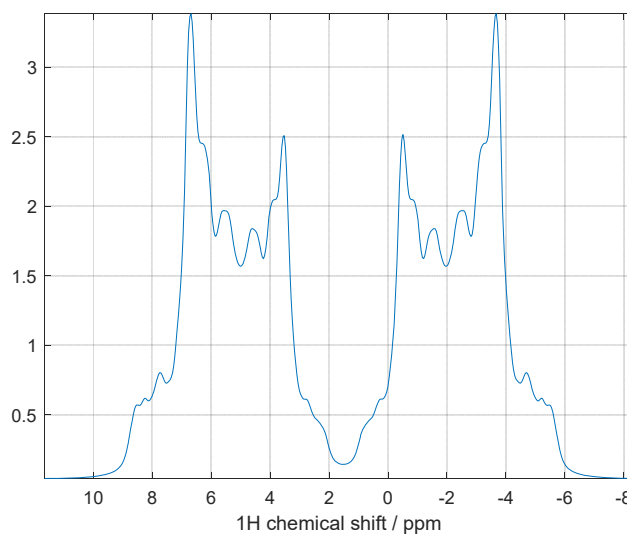


Figure 1. Static powder NMR spectrum of a dipole-coupled two spin system in a situation when the powder averaging grid is too small.

Although Figure 1 does resemble a dipolar spectrum, something is obviously wrong – there are jagged edges and bumps that are not supposed to be there. This is the result of choosing a small powder averaging grid – the spherical integral is not approximated well.

- (i) **Increase the point count in the powder averaging grid until the spectrum converges. Go back to the parameters paragraph and edit the following line:**

```
parameters.grid='rep_2ang_400pts_sph';           % Powder grid
```

Replace 400 with 800, 1600, 3200, 6400, and observe the effect on the resulting spectrum and on the simulation time.

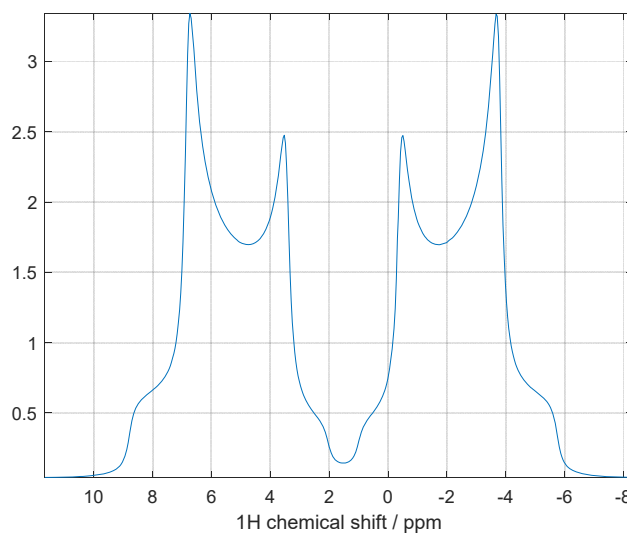


Figure 2. Static powder NMR spectrum of a dipole-coupled two spin system in a situation when the powder averaging grid is sufficient.

- (j) **Vary the parameters we had specified above (chemical shift separation, coordinates, magnet field, offset, sweep, etc.) and observe their effect on the resulting spectrum.**

3. Spinning dipolar patterns

Very little needs to change to convert the static powder simulation we have performed above into a spinning powder simulation. We need to specify spinning axis and rate under *parameters*:

```
parameters.axis=[1 1 1];    % Spinning axis direction
parameters.rate=1000;       % Spinning rate, Hz
parameters.max_rank=3;      % Rotor phase grid rank
```

Here the spinning axis is along the main diagonal of a cube (i.e. at the magic angle to the Z direction), the spinning rate is 1 kHz and there are $2r + 1 = 7$ points in the rotor phase grid. We also need to replace the call to the *powder.m* context function with a call to the *singlerot.m* context function:

```
% Simulation
fid=singlerot(spin_system,@acquire,parameters,'nmr');
```

Note that literally nothing else needs to change, not even the pulse sequence – Spinach context functions are designed to switch smoothly between experimental contexts.

Running the file produces the picture that looks *only vaguely correct* (Figure 2).

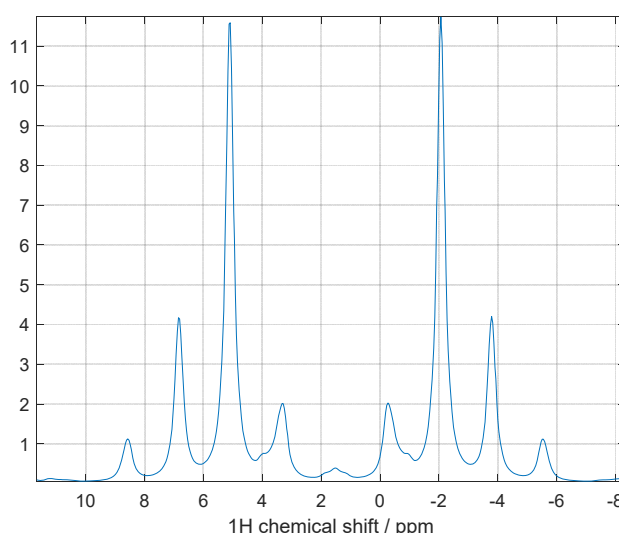


Figure 3. Spinning powder NMR spectrum of a dipole-coupled two spin system in a situation when the rotor phase discretisation grid is too small.

The reason for this is that rotor phase grid must be large enough to reproduce the spin dynamics accurately during the MAS process. A rule of thumb is that the rotor rank must be greater than the expected number of spinning sidebands per signal: about 10 in the example above, but potentially thousands in NQI and DNP systems where interactions are in the MHz range.

(k) Increase the rotor rank until the spectrum converges. Go back to the parameters paragraph and edit the following line:

```
parameters.max_rank=3;      % Rotor phase grid rank
```

Replace 3 with 6, 12, 24, and observe the effect on the resulting spectrum and on the simulation time.

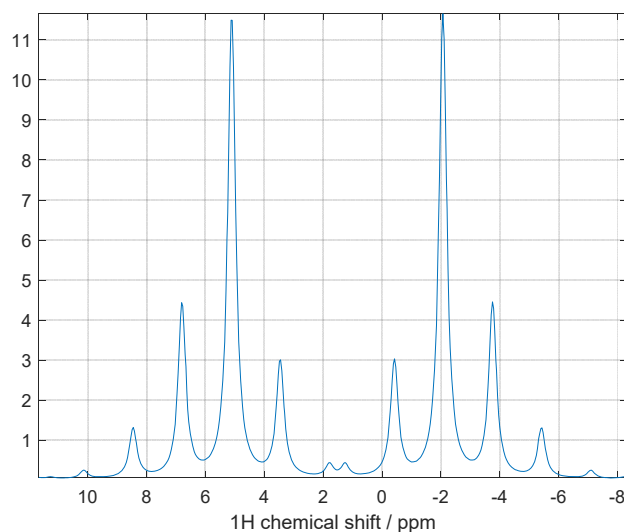


Figure 4. Spinning powder NMR spectrum of a dipole-coupled two spin system in a situation when the rotor phase discretisation grid is sufficient.

Spinning powder simulations have lower spherical powder grid requirements than static powder simulations, and the grid size may be reduced.

- (I) **Reduce the point count in the powder averaging grid until the spectrum converges. Go back to the parameters paragraph and edit the following line:**

```
parameters.grid='rep_2ang_400pts_sph';           % Powder grid
```

Replace 400 with 200, 100, and observe the effect on the spectrum and on the simulation time.

4. Exploration

1. *Spinach* supports spinning at angles other than the magic angle. Move the rotor slightly off magic angle by editing *parameters.axis* and see what happens.
2. Move back to the powder simulation and replace *acquire* with *sp_acquire* in the simulation line. Look up *sp_acquire.m* in the online manual and add the necessary parameters. Experiment with the effect that soft pulses of different amplitudes, frequency offsets, and durations would have on the powder pattern excitation.
3. Replace the spin system with a single deuterium spin, and update the subsequent code accordingly. Use *eeqq2nqi.m* function to specify the quadrupolar coupling (typically hundreds of kHz), and repeat the steps described above.