

Starting with pNMRsim

pNMRsim is a locally-developed simulation program for solid-state NMR problems which is loosely related to commonly-used SIMPSON program. This document is a quick guide to getting started with pNMRsim simulations, but since the programs are closely related, it is also useful to look through original SIMPSON paper ([Common\Articles](#) on the PC network contains the paper and extracts). You can find more information on the differences between pNMRsim and SIMPSON in the pNMRsim documentation (pNMRsim.pdf).

Running pNMRsim

The latest version of pNMRsim is always on the Linux machine fastnet (a version for the PCs will be updated from time to time)¹. The input to pNMRsim/SIMPSON is a text file `<file>.in` which you can create in a text editor. Notepad will do the job, but it does have a tendency to add a `.txt` extension to plain text files which you'll need to delete before pNMRsim will read it.

`pNMRsim <file>.in` will run pNMRsim with the given file. Depending on the instructions, this will generally produce an output spectrum `<file>.spe` (or `<file>.fid` for an unprocessed FID).

`simplot <file>.spe` (or `<file>.fid`) will display the output spectrum using the SIMPLOT program. This allows you to zoom on regions, print the spectrum etc., but doesn't do any processing per se – this must be done in pNMRsim. One particular useful feature is the ability to read in multiple data sets with `simplot <file1> <file2> ...`. Very handy for comparing data sets. SIMPLOT doesn't cope with 2D data sets, but Vadim has a prototype processing/viewing program for 2D data.

Static lineshapes

You can find a very basic input file at the end (1spinstatic.in). The different sections of the file are:

```
spinsys {
  nuclei 13C
  proton_frequency 500e6
  shift 1 0p 120p 0 0 0 0
}
```

The `spinsys` block defines the “spin system”. Here there is just one ¹³C (`nuclei 13C` 1H would represent one ¹³C nucleus and one ¹H). The simulations become much slower as the number of spins increases i.e. we always choose as small as spin system as possible.

`proton_frequency 500e6` sets the ¹H NMR frequency to 500 MHz. This is necessary if chemical shifts etc. are being expressed in ppm, otherwise NMR interactions are specified in directly Hz (in which case `proton_frequency` is not required).

`shift 1 0p 120p 0 0 0 0` defines the chemical shift on spin 1 (spins are numbered from 1). The isotropic chemical shift is 0 ppm and the CSA is 120 ppm (with an asymmetry of 0). The NMR frequencies (and shifts) are calculated from the proton frequency (see below). The final 3 parameters specify the “orientation” of the chemical shift. As we'll see, the orientation is irrelevant for power samples (and can be omitted); it only becomes significant if we have a single orientation present or there are multiple interactions.

The fundamentals of the NMR spin system are now defined. In the next block we define the “experimental” parameters...

```
par {
  sw          50e3          # set spectral width to 50 kHz
  np          256           # set number of points in FID
  crystal_file zcw615       # set powder sampling (see below)
```

¹ SIMPSON runs on both fastnet and PCs that have the Unix↔Windows interface Cygwin installed. Running Cygwin gives a Unix shell prompt from which you can run SIMPSON (just enter `simpson`) and find your Z: drive files: `ls /cygdrive/z/data` should list the files in your data directory (if everything is set up correctly). Note that `/cygdrive/<drive>` is general; you could store your files locally on say the C: drive.

}

The `crystal_file` line determines how the “powder average” is calculated. Because we normally study powder samples, and the NMR parameters, such as shifts, are dependent on the orientation of the crystallites, simulations need to calculate (and sum) together spectra from a large number of orientations in order to obtain a spectrum that corresponds to the experimental situation. Here we use 615 orientations chosen using the “ZCW” scheme. If you run `pNMRsim` with an invalid `crystal_file` parameter, it will list the available options. The choice is important; if you have too few orientations, the spectrum will degrade (usually obvious), while the calculation will take a long time if you have too many.

If no pulse sequences are being applied, `pNMRsim` simulates a simple NMR spectrum, starting with the magnetisation along x (i.e. after a 90 degree pulse) and detecting the normal NMR signal. See the documentation on `start_operator` and `detect_operator` to see how to change this.

The final section specifies the (optional) processing

```
proc {
  addlb 100 0          # add exponential broadening of 100 Hz
  zerofill 2           # zerofill data set to twice its length
  fsave $(name).fid    # save FID
  ft                  # Fourier transform
  fsave $(name).spe     # save processed spectrum
}
```

Note that `$` is used to refer to “variables”—see the documentation for full details.

Exercise

Save the `1spinstatic.in` file, run it and check the output in `SIMPLOT`. You should have a smooth CSA “powder pattern”. You can now experiment with the parameters of the spin system, the simulation and the processing:

- See what happens when you change the anisotropy and asymmetry (ranges from 0 to 1) of the CSA. Confirm that the powder pattern is independent of the “orientation” of the CSA (the 3 angles are expressed in degrees). Why is this?
- Drop the number of orientations e.g. `zcw54`. What happens?
- Reduce the line-broadening to, say, 20 Hz. You should find that you need to increase the length of the FID to avoid truncation. If you increase it too far, you will find that you need to increase the number of crystal orientations to retain a smooth lineshape, why?
- Modify the program to simulate the dipolar coupled ^{13}C spin pair (remove the CSA which would otherwise obscure the pattern). A dipolar coupling of 2 kHz between spins 1 and 2 is added using

```
dipole 1 2 -2000 0 0 0
```

Again, the orientation of the coupling is not significant. Note the dipolar couplings have no asymmetry parameter.

- Simulate a single deuterium (^2H). Again, ignore the (small) chemical shift, but include a first-order quadrupole coupling of, say, 200 kHz:
- ```
quadrupole 1 1 200e3 0 0 0 0
```

You will need to adjust the spectral width, line-broadening etc.

## MAS simulations

Simulations of 1D MAS spectra are also straightforward. An example program (`1spin.in`) can be found below. These are the sections that have changed.

From the `par` block:

```

spin_rate 10000 # spin rate of 10 kHz
gamma_angles 8 # "number of gamma integration angle" (see below)
sw $gamma_angles*$spin_rate # calculate the spectral width
maxdt 1 # integration time step
crystal_file zcw143 # Note we can generally use fewer orientations

```

MAS simulations require an extra angle to be included in the power averaging. This “gamma angle” corresponds to the degree of rotation (why is the corresponding angle in static simulations, the angle around the  $z$  axis, not important?). As the rule of thumb, the number of integration steps, `gamma_angles`, needs to exceed (but not hugely) the number of spinning sidebands. The efficient algorithm used requires the spectral width to be a multiple of the spin rate, hence it’s convenient to set `sw` to `gamma_angles*spin_rate`, obviously choosing `gamma_angles` to such that the spinning sideband manifold will fit comfortably within the spectral width.

`maxdt` is the integration timestep, expressed in  $\mu\text{s}$ . Because the Hamiltonian is time-dependent in MAS experiments (due to the dependence of the NMR interactions on orientation), it is necessary to “integrate” the NMR response over time. Apart from very simple cases (including this one!), this must be done by breaking up the rotation into small time steps and calculating the evolution over each of these steps. Obviously, the smaller this timestep (`maxdt`), the slower the calculation, while the calculation will become inaccurate if it is too large. A 1  $\mu\text{s}$  timestep is fine for most problems (and is the default if `maxdt` is not set explicitly). To check whether `maxdt` is sufficiently small, simply reduce it and check there are no observable changes in the spectrum. As it happens the problem is sufficiently simple in this case that `pNMRsim` can integrate the time evolution directly and `maxdt` will be ignored.

### Exercise

You can now experiment with the parameters of the spin system, the simulation and the processing:

- Run the `1spin.in` file and check the output in SIMPLOT. You should have a set of spinning sidebands from the  $^{13}\text{C}$  CSA. Check that the sideband manifold changes as you would expect when the anisotropy of the CSA is modified. The overall envelope should tend towards that of the static simulation in the limit of low spinning speeds.
- Create a heteronuclear system by adding a proton with a heteronuclear dipolar coupling of, say, 20 kHz to the  $^{13}\text{C}$  (representing an isolated CH group). You will now need to specify on which nucleus the magnetisation starts and which magnetisation should be detected in `par`:

```

start_operator 13C:x # start with 13C x magnetisation
detect_operator 13C:p # detect 13C signal (" +1 coherence")

```

If you suppress the  $^{13}\text{C}$  CSA (just comment out the `shift` line with an initial `#` character), you should obtain a symmetric set of spinning sidebands. Note that in typical organic molecules, a C–H is never isolated, but it is not practical to include a large number of protons in such simulations.

### Including RF

To add RF irradiation, we must add a `channels` directive to `spinsys`:

```
channels 1H
```

This means that RF channel 1 is  $^1\text{H}$ . We don’t need an explicit  $^{13}\text{C}$  channel.

We define a “sequence” for CW decoupling in `par`:

```

pulse 1e6 80e3 x
store CW

```

The syntax of `pulse` is `<duration /  $\mu\text{s}$ > <RF1 / Hz> <phase1 / degrees>`, where `RF1` is the nutation rate on channel 1 [e.g. a 5  $\mu\text{s}$  90° pulse corresponds to a nutation rate of  $1/(4 \times 5 \mu\text{s}) = 50 \text{ kHz}$ ] and `phase1` is the phase of the RF (which can be specified as `x`, `y` etc. or explicitly as an angle in degrees). There should be a pair of RF and phase values for each channel (setting the RF intensity to zero if the

channel is not active) e.g. `pulse 1000 50e3 x 50e3 x` would correspond to matched 1 ms pulses on both channels 1 and 2 i.e. the contact period of a CP experiment. `pulseid` simulates an “ideal” pulse in which the RF is “infinitely strong”. The RF nutation rate is still required, however, in order to calculate the tip angle of the pulse (from the duration and nutation rate). Similarly the pulse duration is only used to calculate the tip angle and pulse has zero duration, simplifying timing issues. `store CW` saves the “sequence” with the name CW.

A `pulseseq` block must be added in order to use the defined sequences:

```
pulseseq {
 acq 0 CW
}
```

This means acquire the NMR signal with a zero receiver phase shift while applying RF sequence CW. If `pulseseq` is omitted (as above), it effectively defaults to `acq 0`.

### Exercise

Verify that `1spin.in` modified to include RF on the protons does indeed produce the expected  $^1\text{H}$ -decoupled  $^{13}\text{C}$  spectrum.

- Restore the  $^{13}\text{C}$  CSA. What happens if you turn off the decoupling i.e. have both a CSA and heteronuclear coupling?

### Going further

Read the pNMRsim documentation for detailed information on using the program. You learn more about simulating solid-state NMR experiments in SIMPSON using the SIMPSON article (in [\tyne\home\Common\Articles](#)) and look at the examples in `Y:/doc/simpson-examples-1.0.1`. Note, however, that RF sequences in particular are defined quite differently in pNMRsim and SIMPSON.

For information on how these simulations work, you can read a review article from Progress in NMR (in the Articles directory) or recent articles by Mattias Eden in Concepts in NMR (volumes 17A and 18A). These are not for the faint-hearted however!

SIMPSON/pNMRsim have limitations which should be born in mind before trying to use it for problems which it can't really handle:

- Relaxation is not included. This requires quite different theory, which is not particularly well understood for solids in any case. We generally treat relaxation parameters as empirical quantities. We can simulate the effects of relaxation with line-broadening, but this is obviously inadequate for modelling real relaxation effects e.g. the NOE.
- The calculations slow down rapidly with increasing number of spins (as  $O(2^{3N})$  for a collection of  $N$  spin-1/2 species). pNMRsim handles this much better than SIMPSON, but cannot work miracles.
- Exchange effects cannot be described. Again, this requires additional theory.
- Be careful with 2<sup>nd</sup> order quadrupoles. SIMPSON's approach to second order effects on quadrupoles is adequate for isolated nuclei, but fails if the nuclei are coupled to other spins. For instance, simulations of  $^{13}\text{C}$ - $^{14}\text{N}$  pairs will not show the broadening effects on  $^{13}\text{C}$  due to “residual couplings” to the  $^{14}\text{N}$  i.e. SIMPSON gets it wrong! pNMRsim's approach should be more general, but has not been properly evaluated.

pH

## 1 static spin-1/2 (1spinstatic.in)

```
one spin 1/2 static (CSA powder pattern)

spinsys {
 nuclei 13C
 proton_frequency 500e6
 shift 1 0p 120p 0 0 0 0
}

par {
 sw 50e3
 np 256
 crystal_file zcw615
}

proc {
 addlb 100 0
 zerofill 2
 save $(name).fid
 ft
 save $(name).spe
}
```

## 1 spin-1/2 under MAS (1spin.in)

```
one spin-1/2 under MAS

spinsys {
 nuclei 13C
 proton_frequency 500e6
 shift 1 0p 120p 0 0 0 0
}

par {
 spin_rate 10000
 gamma_angles 8
 maxdt 1
 sw $gamma_angles*$spin_rate
 np 4096
 crystal_file zcw143
}

proc {
 addlb 50 0
 zerofill 2
 ft
 save $(name).spe
}
```

## Additions and pulseseq to add CW decoupling

```
spinsys {
 ...
 channels 1H
}

par {
 start_operator 13C:x
 detect_operator 13C:p
 ...
 pulse 1e6 80e3 x
 store CW
}

pulseseq {
 acq 0 CW
}
```