

Interactive comment on “ssNMRLib: a comprehensive library and tool box for acquisition of solid-state NMR experiments on Bruker spectrometers” by Alicia Vallet et al.

Alicia Vallet et al.

paul.schanda@ibs.fr

Received and published: 12 November 2020

Replying to question (1), about the safety check and duty cycle:

Our reply: There are multiple ways, how one could implement security checks for a given pulse sequence, and all have their pros and cons. In the present version of ssNMRLib, we have opted for checking a range of pulse sequence parameters by comparing them to values listed in a file (see my reply to reviewer 1). In the future, we may also exploit the possibility of implementing security features based on duty cycle calculations.

Replying to question (2), about on-the-fly shape generation:

C1

Our reply: WaveMaker is part of the NMRLib distribution, and it has been used for a number of solution state NMR pulse sequences. Actually, when a new data set is loaded, WaveMaker is automatically executed, which makes it easy to use (or not) wavemaker commands in new pulse sequences.

For ssNMRLib, we have considered using WaveMaker to create shapes. WaveMaker would have one advantage over the current implementation: it would allow to calculate shapes on the fly, including arbitrary shapes, e.g. tangential CP ramps. WaveMaker would allow doing e.g. a poppt optimization of the shape file directly within one poppt run, rather than by comparing different shapes in different experiments.

We ended up deciding against implementing WaveMaker at the current stage. One reason is that we would have to go back and change some 140 pulse sequences; we decided for now not to do it and rather have a functional library in place. The second reason is that we wanted to have a coherent naming convention, where each type of CP (e.g. H-N, H-C, etc) would have the same name across all experiments. This means that we have almost all of the available constant names (cnst) assigned to CPs or other parameters. If we wanted to use WaveMaker, we would need to have additional parameters, such as the adiabaticity. The number of constants in Topspin is limited to 64 (cnst0 to cnst63). We cannot fit all constants that we would need within these 64. This is not a good reason, and we hope that Topspin will alleviate this limitation soon.

Replying to question (3) about a "Troubleshooting" section and discussion group:

Our reply: This is a good idea. We have thought of a forum already, and will set it up soon.

Replying to question (4), related to the use of python/matplotlib/nmrglue:

Our reply: A number of macros in Topspin already use python scripts for data analysis (including nmrglue). For example, there are scripts in NMRLib which fit diffusion or overall-tumbling (TRACT) data. Likewise, scripts are available to export spectra as

C2

matplotlib figures.

These scripts are executed e.g. by clicking the cogwheel symbol right next to the DOSY or TRACT setup buttons in the solution-state library. Those examples indicate how to extract and export data to python/matplotlib. The software is completely open to exporting data using python, in the way that is indicated with these examples.

We thank the reviewer for the positive evaluation.

Interactive comment on Magn. Reson. Discuss., <https://doi.org/10.5194/mr-2020-25>, 2020.