pyDNMR Documentation

Release 0.2.0

Geoffrey Sametz

April 02, 2017

CONTENTS

pyDNMR simulates dynamic nuclear magnetic resonance (DNMR) spectra. A graphical user interface provides inputs for simulation parameters (frequencies, rate constants, line widths, and the population of various states), and displays the resulting spectrum.

The current version of pyDNMR only includes the simulation for two uncoupled spin-1/2 nuclei undergoing exchange. The short-term goal of this project is to include the simulation for two coupled nuclei as well, and to create platform-specific executable files ("apps") for educational use. For example, the rate constant for nuclear exchange can be adjusted up and down to demonstrate coalescence of signals.

The longer-term goal of this project is to add simulations for more complex systems, and to provide additional tools (e.g. importation of NMR spectra, and matching experimental to simulated lineshapes) that would result in an application suitable for researchers as well as educators.

A secondary purpose of this project is to provide a test case for the author to learn how to properly test, package, and distribute software.

CHAPTER

ONE

INSTALLATION AND USE

This is a work in progress. The main application (main.py) and its dependencies in the pydnmr subfolder should provide a basic, functional application, if the user is running Python 3.3+ and has the requirements listed in requirements.txt.

Warning: Do not trust setup.py for installation – it has yet to be tested.

The brave and curious can copy main.py plus the pydnmr/pydnmr subfolder and its contents, install the dependencies, and run the program from the command line

```
$ python main.py
```

If you are using Python 2 and/or PyQt5 in your default Python environment, it is recommended that you use a virtualenv, set it up with Python 3 and PyQt5, and install into that.

CHAPTER

TWO

THEORETICAL BACKGROUND

The current version of pyDNMR simulates the case of two-site exchange for uncoupled spin-1/2 nuclei. Other simple systems that do not require a quantum-mechanical treatment will be introduced after a stable Version 1 distribution is finished.

Sandström [1] shows how a formula for calculating the intensity (v) at frequency ν can be derived from the Bloch equations. [2] The formula:

$$\mathrm{v} = -C_0 \frac{\left\{ P\left[1 + \tau\left(\frac{p_B}{T_{2A}} + \frac{p_B}{T_{2B}}\right)\right] + QR \right\}}{P^2 + R^2}$$

requires the calculation of paramaters P, Q, and R:

$$P = \tau\left[\frac{1}{T_{2A} \cdot T_{2B}} - 4\pi^2\Delta\nu^2 + \pi^2(\delta\nu)^2\right] + \frac{p_A}{T_{2A}} + \frac{p_B}{T_{2B}}$$
$$Q = \tau[2\pi\Delta\nu - \pi\delta\nu(p_A - p_B)]$$
$$R = 2\pi\Delta\nu\left[1 + \tau\left(\frac{1}{T_{2A}} + \frac{1}{T_{2B}}\right)\right] + \pi\delta\nu\tau\left(\frac{1}{T_{2B}} - \frac{1}{T_{2A}}\right) + \pi\delta\nu(p_A - p_B)$$

These in turn require calculations for τ, $\delta\nu$, and $\Delta\nu$, as well as determining the transverse relaxation times T_2.

Calculating τ requires knowing the fractional populations of nuclei in each state ($p_{A/B}$), and one of the rate constants k for an exchange process.

$$\tau = \frac{p_A}{k_B} = \frac{p_B}{k_A}$$

where $p_A + p_B = 1$, and $\frac{d[A]}{dt} = -k_A[A]$.

$\delta\nu$ is the difference in frequencies for nuclei at site A versus site B, at the slow exchange limit:

$$\delta\nu = \nu_A - \nu_B$$

and $\Delta\nu$ is the difference between the average of (i.e. midpoint between) ν_A and ν_B, and the frequency ν along the spectrum that the signal intensity v is being calculated at.

$$\Delta\nu = 0.5(\nu_A + \nu_B) - \nu$$

[1] Sandström, J. *Dynamic NMR Spectroscopy;* Academic Press: New York, 1982.

[2] The use of v for intensity is retained from Sandström, despite its resemblance to ν . The simulated lineshape will be a plot of v vs. ν !

Although the T_2 s for the nuclei at sites A and B could be determined experimentally, in practice they are usually estimated from the width of the peaks at half height, at the slow exchange limit ($W_{0A/B}$).

$$T_{2A} = \frac{1}{\pi W_{0A}}; T_{2B} = \frac{1}{\pi W_{0B}}$$

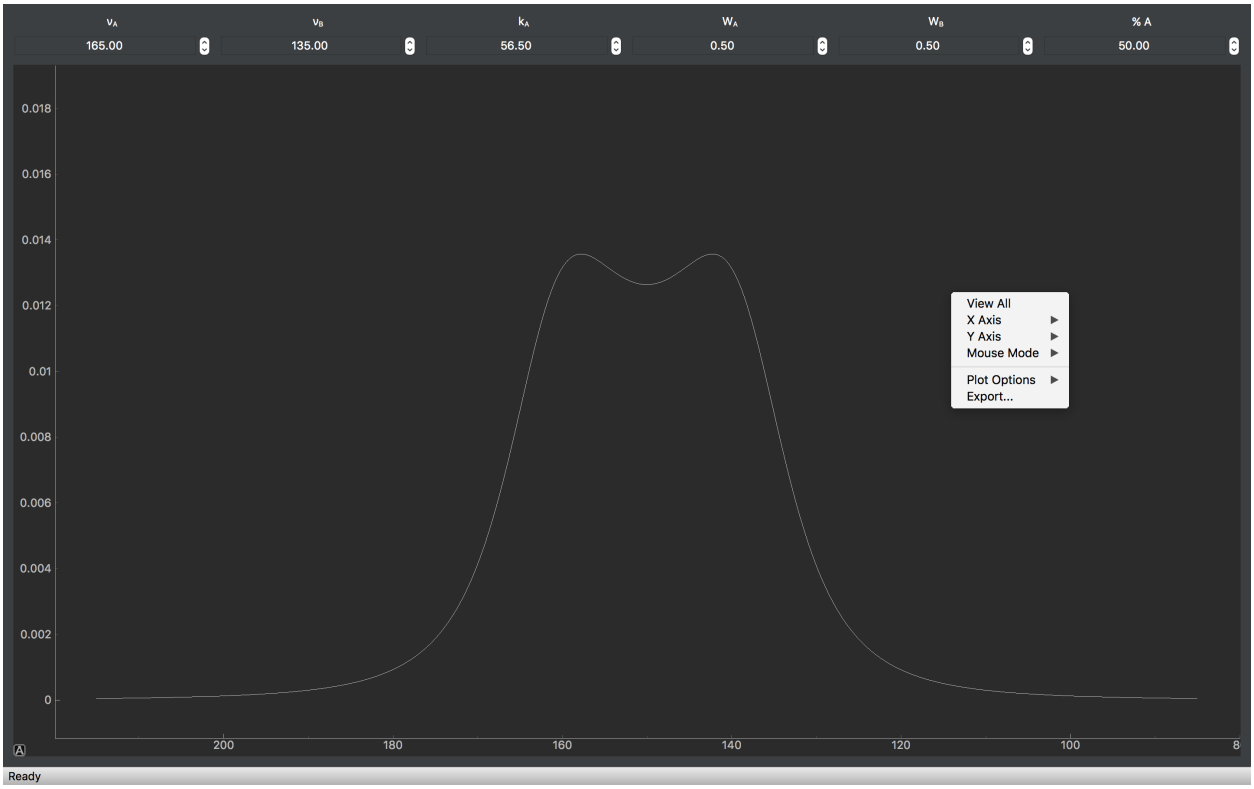
So, the experimental parameters required to simulate a lineshape are:

- ν_A and ν_B at the slow-exchange limit
- linewidths W_{0A} and W_{0B} at the slow-exchange limit
- the population p_A of one of the sites, and
- the rate constant k_A for the rate of exchange from site A to site B

CHAPTER

THREE

USING PYDNMR



3.1 Entering Parameters

Numerical values for $\nu_A, \nu_B, k_A, W_A, W_B$, and % A can be entered into the fields at the top of the application window. The calculations assume units of Hz or s^{-1} for the first five. The final entry is the population of site A expressed as a percentage (rather than as the fraction (p_A) used in the previous section's formulas).

3.2 Mouse Controls

Clicking on the up/down arrows, or scrolling with the mouse wheel while the field's contents are active, raises/lowers the values inside in increments of 1.

Inside the plot of the spectrum:

- Left click and drag moves the plot.
- Scroll up/down zooms in/out.
- Right click and drag expands/contracts the spectrum horizontally/vertically.
- Right click currently opens up menus of options that have not been fully tested yet. Some options (exporting to .csv) appear to work, but others (exporting as a matplotlib plot or an image file) crash the program. These options are built into the third-party pyqtgraph library that pyDNMR imports.

CHAPTER

FOUR

FEEDBACK

Disclaimer: the author is neither an NMR spectroscopist, nor a software engineer. I'm figuring this out as I go along. I welcome feedback on the project. If you think the simulation or the code could be improved upon, feel free to leave an issue on GitHub, or contact me by email (see setup.py for my current address).

CHAPTER

FIVE

ACKNOWLEDGEMENTS

This project is inspired by Hans Reich's WINDNMR application. I thank him for our conversations, and his sharing of WINDNMR's Visual Basic 6 code.