# pyDNMR Documentation

***Release 0.5.0***

**Geoffrey Sametz**

**Apr 17, 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 will simulate spectra for the two-site exchange of two spin-1/2 nuclei, either uncoupled (two singlets at the slow-exchange limit), or coupled (two doublets, or AB quartet, at the slow-exchange limit).

The short-term goal of this project is 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 EXECUTION

For Windows and Mac OS X, the application is frozen as a single-file, double-click-to-run application (see link above). Nothing besides the executable file needs to be installed. The description below is only for users that want to download and run the Python source code itself.

The essential package components required to run the application are main.py plus the pydnmr subfolder and its contents. The application can be launched by running main.py:

```
$ python main.py
```

The code should work for Python version 3.5 and up. The dependencies listed in requirements.txt are also required. If pip is installed, the following command should automatically install the required dependencies:

```
>>>pip install -r requirements.txt
```

If you are familiar with virtual environments (e.g. using virtualenv, venv, or conda), you may wish to create one specifically for running this code, and install requirements there. If you use an Anaconda installation of Python, it is quite easy to set up and switch between different environments. See the conda documentation for details.

CHAPTER

TWO

# THEORY

The current version of pyDNMR simulates the case of two-site exchange for uncoupled and coupled 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.

## 2.1 Two-Site Exchange of Two Uncoupled Nuclei

Sandström[1] shows how a formula for calculating the intensity (v) at frequency $\nu$ 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 $\tau$, $\delta\nu$, and $\Delta\nu$, as well as determining the transverse relaxation times $T_2$.

Calculating $\tau$ 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) $\nu_A$ and $\nu_B$, and the frequency $\nu$ 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 $\nu$ . The simulated lineshape will be a plot of v vs. $\nu$ !

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 ($W_{0A/B}$).

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

The experimental parameters required to simulate a lineshape are therefore:

- $\nu_A$ and $\nu_B$ 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

## 2.2 Two-Site Exchange of Two Coupled Nuclei

The simulation uses the formulas from Weil et al.[3] This simulation assumes that the population of the two states is equal ($p_A = p_B = 0.5$). The function $F(\nu)$ for intensity at frequency $\nu$ is given as:

$$F = C\left(\frac{r_+ b_+ - s a_+}{a_+^2 + b_+^2} + \frac{r_- b_- - s a_-}{a_-^2 + b_-^2}\right)$$

where:

$$\begin{aligned}
a_\pm &= 4\pi^2(\nu_o - \nu \pm J/2)^2 - (\tau^{-1} + \tau_2^{-1})^2 - \pi^2(\nu_1 - \nu_2)^2 - \pi^2 J^2 + \tau^{-2} \\
b_\pm &= 4\pi(\nu_o - \nu \pm J/2)(\tau^{-1} + \tau_2^{-1}) \mp 2\pi J \tau^{-1} \\
r_\pm &= 2\pi(\nu_o - \nu \pm J) \\
s &= 2\tau^{-1} + \tau_2^{-1}
\end{aligned}$$

Note that in the original publication, there was an error in the formula for $r_\pm$ – the final term erroneously used " $\pm$ " instead of the correct " $\mp$ " shown here.

These functions in turn require the following definitions:

$$\begin{aligned}
\nu_o &= \frac{\nu_1 + \nu_2}{2} \\
\tau &= k^{-1} \\
\tau_2 &= \frac{1}{\pi W}
\end{aligned}$$

The experimental parameters required to simulate a lineshape are therefore:

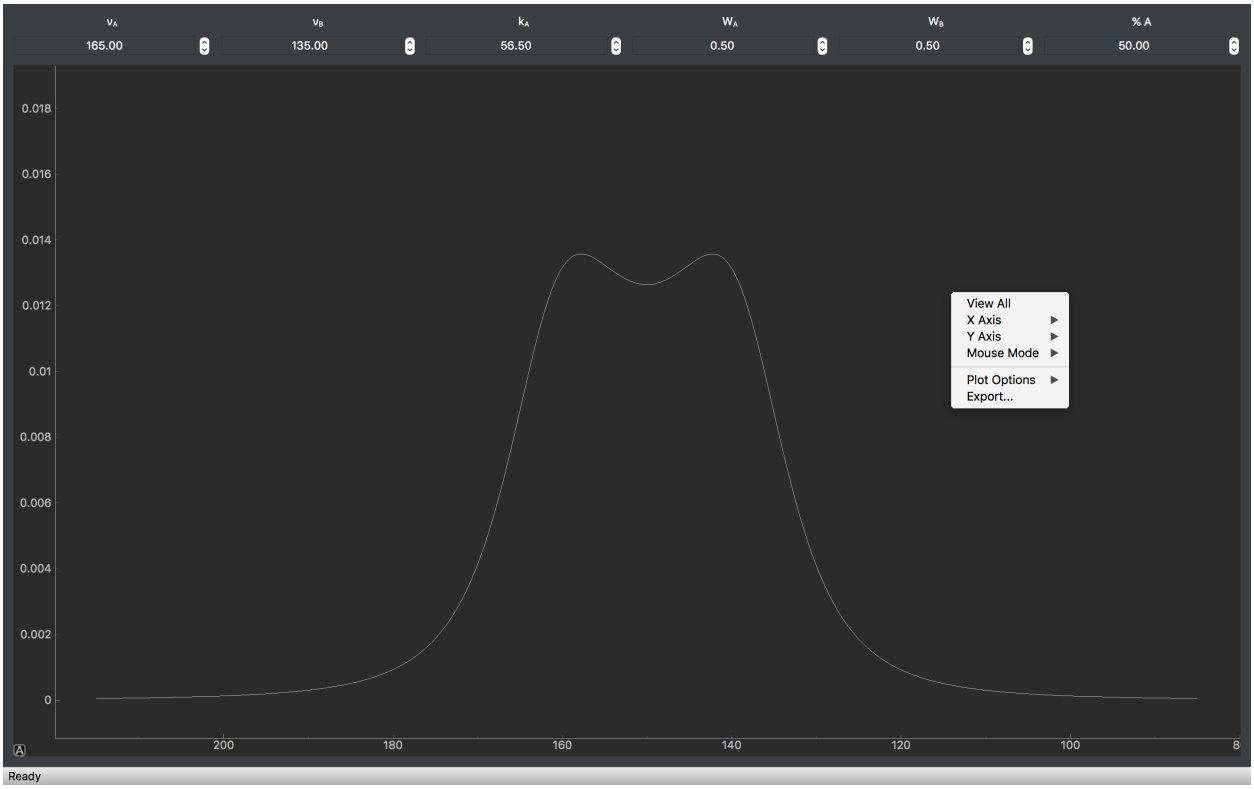
- $\nu_A$ and $\nu_B$ at the slow-exchange limit
- The coupling constant $J$
- a single linewidth $W_{0A}$ and $W_{0B}$ at the slow-exchange limit, and
- the rate constant $k$ for the rate of exchange

[3] Brown, K.C.; Tyson, R.L.; Weil, J.A. *J. Chem. Educ.* **1998**, *75*, 1632.

CHAPTER

# THREE

# USING PYDNMR




## 3.1 Entering Parameters

Numerical values can be entered into the fields at the top of the application window. The calculations assume units of Hz for chemical shifts, or $s^{-1}$ for rate constants. Relative populations of sites are entered as percentages.

## 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. 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.