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

theory placeholder

$$
\begin{aligned}
&\mathrm{v} = -C_0 \frac{\Big\{P\Big[1+\tau\Big(\frac{p_B}{T_{2A}}+\frac{p_B}{T_{2B}}\Big)\Big]+QR\Big\}}{P^2+R^2}\\
&\delta\nu = \nu_A-\nu_B;\, \Delta\nu = 0.5(\nu_A+\nu_B)-\nu\\
&P = \tau\left[\frac{1}{T_{2A}\cdot T_{2B}}-4\pi^2\Delta\nu^2+\pi^2(\delta\nu)^2\right]+\frac{p_A}{T_{2A}}+\frac{p_B}{T_{2B}}\\
&Q = \tau[2\pi\Delta\nu-\pi\delta\nu(p_A-p_B)]\\
&R = 2\pi\Delta\nu\left[1+\tau\Big(\frac{1}{T_{2A}}+\frac{1}{T_{2B}}\Big)\right]+\pi\delta\nu\tau\Big(\frac{1}{T_{2B}}-\frac{1}{T_{2A}}\Big)+\pi\delta\nu(p_A-p_B)\\
&\tau = \frac{p_A}{k_B} = \frac{p_B}{k_A}\\
&T_{2A} = \frac{1}{\pi W_{0A}};\, T_{2B} = \frac{1}{\pi W_{0B}}
\end{aligned}
$$

CHAPTER

THREE

USING PYDNMR

operation placeholder

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.