MPHYG002 Coursework One Report

David F. Wise

April 11, 2017

1 Introduction

This report details the background to the small C++ library I have created as part of the MPHYG002: Research Computing With C++ course at University College London. This library uses the C++ Eigen library to find the energy levels of the common quantum system of a single electron spin coupled to a nuclear spin.

2 Background

My PhD at UCL is primarily concerned with the manipulation of electron and nuclear spins, with the aim of demonstrating their potential use as qubits in a quantum computer [Kan98]. Although the ultimately this technology will require single spin control, much of the initial work being performed to better understand the systems at hand is conducted via electron spin resonance (ESR) which studies multiple spins at once [Feh59]. Qubits require a quantum two level system, provided in this case by an electron spin in a static magnetic field. The electron has two energy levels, one where its spin is aligned with the magnetic field (lower) and one where it is anti-aligned (higher). It is possible to drive the electron spins between these two states using microwave radiation possessing the same energy as the difference between the two states.

The systems that I study are electrons bound to the nuclei of dopants in silicon [TMB+06, TLAR03, MMEL11]. In these systems the problem is complicated by the presence of the nuclear spin, introducing extra energy levels via the hyperfine coupling between electron and nucleus. Whilst the electron spin is restricted to $\pm 1/2$, the nuclear spin can have a variety of half-integer values, with the exact spin of the nucleus determining the number of energy levels of the system. The energy levels of the system are described by a Hamiltonian which can be represented as a matrix, whose eigenvalues are the system energy levels. In order to calibrate our equipment to the correct magnetic fields and microwave frequency we need to find the eigenvalues of the matrix at a given magnetic field.

A common approach is to use the Matlab software EasySpin [SS06]. Whilst this software is capable, it tends to be slow. Hence a natural exercise is to use a similar approach in C++ to examine whether we can speed up the process of finding results. The simple initial task chosen here is to create a piece of software that is able to find the eigenvalues of an electron-nuclear spin system with arbitrary nuclear spin, hyperfine coupling and magnetic field. This work could then be extended to find the rate of change of energy levels with magnetic field, for example to locate clock transitions where the system is insensitive to magnetic field noise [WTG+13].

3 Implementation

For this software to be useful for all members of my group (and potential collaborators) it is obviously beneficial for it to run on multiple platforms (MacOS, Linux and Windows) but at present it has only been tested on MacOS.

The end user of this package will be experienced with the underlying physics so detailed explanation of this is not required. They will likely not be experienced with C++, meaning that a command line interface would be ideal. Unfortunately I have not yet had time to implement this, meaning that direct editing of the main.cpp file will be required. To offset the potential confusion I have provided instructions in the project readme as well as a working example in the `main.cpp` file. This working example includes comments with instructions on how to modify it to solve a desired problem. The core concepts of this will be familiar to anyone with programming experience and thus I do not anticipate it causing significant issues.

The main library used in this project is the Eigen library. This allows fast manipulation of matrices and calculation of their eigenvalues. For simplicity I have included the library within my package. This avoids the added complication for the end user of installing it themselves and adding it to their C++ path (an exercise that caused me considerable trouble). In addition I have used the Google Test framework for test implementation, again this is included in the package.

The development environment used was CLion with CMake version 3.6 and Eigen version 3.3.3.

3.1 Code Structure

My code is built around one simple `Donor` class, which includes all the present functionality of the program. This Donor class is constructed with a nuclear spin and hyperfine coupling value. On construction a function, `setSpinsMats` is called. This uses the given nuclear spin value to construct the necessary spin matrices which are later used for constructing the system Hamiltonian.

There is additionally an `initialise` function, which effectively constructs a Donor that is not constructed with nuclear spin and hyperfine coupling values. This was necessary for compatibility with Google tests.

After construction the nuclear spin value and hyperfine coupling value can be changed using typical `set` functions.

To calculate the eigenvalues of the system a function called `getEigs` is called. This first constructs the Hamiltonian using the spin matrices and the one function parameter `B0`: the magnetic field. It returns these in the form of a 1-Dimensional Eigen matrix.

The `main.cpp` file contains an example of how this class can be used to calculate how the eigenvalues change with static magnetic field. This is done by looping over a series of magnetic field strengths, calculating the eigenvalues at each one and adding them to an Eigen matrix. This matrix can then be written to a text file for analysis in a more suitable language.

4 Code Demonstration and Commentary

For demonstration purposes the eigenvalues of the simplest spin system (spin 1/2 nucleus) were calculated for magnetic fields between 0 and 0.01 Tesla. The results are shown in figure 4. The results calculated here are in agreement with those calculated by EasySpin and experimental data [SS06, Feh59].

4.1 Commentary

Although the results of running the code from `main.cpp` were accurate, there is a problem when running from within the test environment that can cause some tests to fail. This is caused by random floating point errors which result in inaccurate spin matrices and thus eigenvalues. This was originally observed in `main.cpp` as well but was identified as being caused by using a `pow` instead of `sqrt` function. For some reason the problem remained unsolved when running from the test environment and I was unable to discern the reasons for this despite considerable attempts.

On the scalability of the code the chief barrier is the need to calculate eigenvalues at each magnetic field point. This slows things down considerably. Happily my program is significantly quicker than EasySpin at solving the same problem but I think could be made faster if I was able to determine away of using symbolic calculations. A similar program I implemented in Python using the Sympy library allowed a formula for each eigenvalue to be determined, meaning that matrix diagonalisation was only performed once. This allows the python code to run significantly faster than the C++ code. A clear next step would be to see if I could use similar techniques in C++ to narrow the speed gap to Python.

References

[Feh59] G. Feher. Electron spin resonance experiments on donors in silicon. I. Electronic structure of donors by the electron nuclear double resonance technique. *Physical Review*, 114(5):1219–1244, 1959.

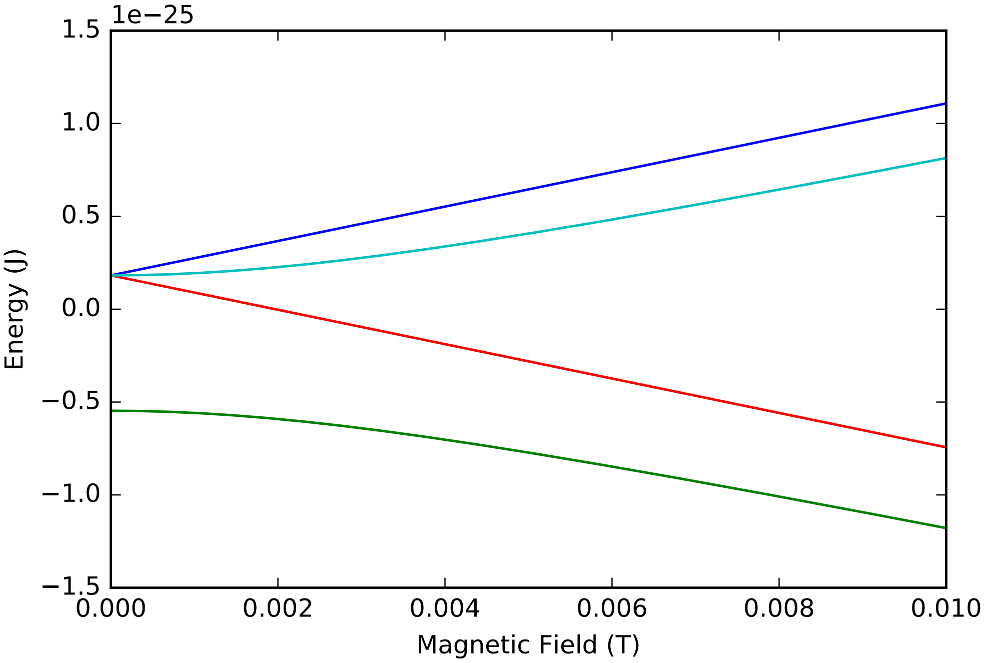

Figure 1: Figure snowing how the eigenvalues of a simple nuclear spin 1/2 donor evolve with magnetic field, as calculated by my program. These are the same as predicted by EasySpin [SS06]

[Kan98] B E Kane. A silicon-based nuclear spin quantum computer. *Nature*, 393(6681):133–137, 1998.

[MMEL11] John J. L. Morton, Dane R. McCamey, Mark a. Eriksson, and Stephen a. Lyon. Embracing the quantum limit in silicon computing. *Nature*, 479(7373):345–353, 2011.

[SS06] Stefan Stoll and Arthur Schweiger. EasySpin, a comprehensive software package for spectral simulation and analysis in EPR. *Journal of Magnetic Resonance*, 178(1):42–55, 2006.

[TLAR03] a. M. Tyryshkin, S. a. Lyon, a. V. Astashkin, and a. M. Raitsimring. Electron Spin-Relaxation Times of Phosphorus Donors in Silicon. *Physical Review B*, pages 12–15, 2003.

[TMB+06] A M Tyryshkin, J J L Morton, S C Benjamin, A Ardavan, G A D Briggs, J W Ager, and S A Lyon. Coherence of Spin Qubits in Silicon. *Journal of Physics: Condensed Matter*, 18:783–794, 2006.

[WTG+13] Gary Wolfowicz, Alexei M Tyryshkin, Richard E George, Helge Riemann, Nikolai V Abrosimov, Peter Becker, Hans-Joachim Pohl, Mike L W Thewalt, Stephen a Lyon, and John J L Morton. Atomic clock transitions in silicon-based spin qubits. *Nature nanotechnology*, 8(8):561–4, 2013.