Electronics and Computer Science
Faculty of Engineering and Physical Sciences
University of Southampton

Eleanor Heywood
December 2023

Educational Quantum Computing Interface to aid Teaching Quantum Chemistry

Project Supervisor: Klaus-Peter Zauner
Second Examiner: TBC

A progress report submitted for the award of
Bachelors of Science

Abstract

Quantum Computing is a field that has existed since Richard Feynman became frustrated at the large amounts of computational power required to simulate the quantum mechanics he was researching. Thanks to Feynman, it has been theorised that quantum computing could lead to exponentially faster algorithms to simulate quantum systems. Therefore, it is important that quantum chemists become aware of the impact that quantum computing is going to have on the complexity and speed of simulations in the near future. The difficulty lies in the complexity and unreliability of quantum computing due to the noise interference that exists within current quantum hardware. The learning curve to understand quantum computing is steep for a computer scientist, but almost inaccessible to many chemists with little understanding within the computational field. This report describes the basics of quantum computing, some example problems in quantum chemistry, and the plans to create an educational tool providing a visual aid for quantum chemists, which abstracts away some of the difficulties of quantum computing.

Statement of Originality

- I have read and understood the ECS Academic Integrity information and the University's Academic Integrity Guidance for Students.
- I am aware that failure to act in accordance with the Regulations Governing Academic Integrity may lead to the imposition of penalties which, for the most serious cases, may include termination of programme.
- I consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to verify whether my work contains plagiarised material, and for quality assurance purposes.

You must change the statements in the boxes if you do not agree with them.

We expect you to acknowledge all sources of information (e.g. ideas, algorithms, data) using citations. You must also put quotation marks around any sections of text that you have copied without paraphrasing. If any figures or tables have been taken or modified from another source, you must explain this in the caption and cite the original source.

I have acknowledged all sources, and identified any content taken from elsewhere.

If you have used any code (e.g. open-source code), reference designs, or similar resources that have been produced by anyone else, you must list them in the box below. In the report, you must explain what was used and how it relates to the work you have done.

Resources produced by other people include: Python, Cirq (including qsim), matplotlib

You can consult with module teaching staff/demonstrators, but you should not show anyone else your work (this includes uploading your work to publicly-accessible repositories e.g. Github, unless expressly permitted by the module leader), or help them to do theirs. For individual assignments, we expect you to work on your own. For group assignments, we expect that you work only with your allocated group. You must get permission in writing from the module teaching staff before you seek outside assistance, e.g. a proofreading service, and declare it here.

I did all the work myself, or with my allocated group, and have not helped anyone else.

We expect that you have not fabricated, modified or distorted any data, evidence, references, experimental results, or other material used or presented in the report. You must clearly describe your experiments and how the results were obtained, and include all data, source code and/or designs (either in the report, or submitted as a separate file) so that your results could be reproduced.

The material in the report is genuine, and I have included all my data/code/designs.

We expect that you have not previously submitted any part of this work for another assessment. You must get permission in writing from the module teaching staff before re-using any of you previously submitted work for this assessment.

I have not submitted any part of this work for another assessment.

If your work involved research/studies (including surveys) on human participants, their cells or data, or on animals, you must have been granted ethical approval before the work was carried out, and any experiments must have followed these requirements. You must give details of this in the report, and list the ethical approval reference number(s) in the box below.

My work did not involve human participants, their cells or data, or animals.

ECS Statement of Originality Template, updated August 2018, Alex Weddell aiofficer@ecs.soton.ac.uk

Contents

1. Project Description

Quantum computing is a field of research that has existed for the past 30 years, ever since Richard Feynman suggested the possibility of simulating quantum mechanics on a machine made up of ‘quantum computer elements’ [15]. By so doing, there is a possibility for quantum algorithms to be exponentially faster than their classical counterparts. However this is only theoretical, as measuring accurate results from such a system is difficult [22]. Recently large strides have been made in research, and thanks to large corporations such as Google and IBM, the public now have access, albeit limited, to some quantum machines. Understandably, quantum computing is unlike classical computing, relying on probabilities and concepts that do not come naturally to an amateur programmer, and herein lies the problem. A steep learning curve must be undertaken to program a quantum machine, and if chemists wish to run simulations, a great deal of time must be invested in understanding and coding before being able to produce any meaningful results. Hence an educational tool that can abstract some of the difficulty away could be extremely useful.

1.1 Goals

The aim of this project is to produce an open-source framework to aid in the teaching of key principles of quantum chemistry, with the option to customise what machine the calculations are running on, whether that be a built in Quantum Virtual Machine (QVM), or a real quantum machine. The tool will be designed to teach quantum chemists the advantages that quantum computing will have in their field and some of the knowledge required to understand it. The goals of the project are broken down as follows:

- Customisable backend.
- Transportable between different cloud infrastructure that supply access to quantum machines.
- Built in customisable QVM.
- Easily extensible due to modular code.
- Sufficient documentation to allow ease of use.
- Tutorials on how to use the tool and visualize basic quantum chemistry concepts.

1.2 Scope

This project will focus on producing a tool that is modular, versatile, and easy to expand. Different libraries and tools such as Google's Cirq [13], and IBM's Qiskit [7] will be assessed and chosen with the former goals in mind. A mixture of quantum and classical computing will be used for maximum efficiency. A customisable QVM will be supplied as a part of the tool as a minimum. In terms of quantum chemistry visualisation, this project will focus on implementing the Schrödinger wave equation to produce probability densities for subatomic particles. A number of quantum circuits for the Schrödinger equation will be assessed, with the most efficient being implemented as a minimum.

2. Background Research

2.1 Quantum Computing

The current state of quantum devices can be categorised as NISQ devices: Noisy Intermediate-Scale Quantum devices [4], where noisy refers to the interference caused by quantum hardware, and intermediate-scale refers to the number of qubits in a device [2]. Most NISQ devices will also use a mixture of quantum and classical computing for utmost efficiency. The main issue with the current state of quantum machines is the noise that leads to inconsistencies and errors, thus making them unreliable. Many do not believe viable quantum computing solutions will be built for another 10 years minimum [2], with a few people even suggesting they may never exist [14].

2.1.1 Qubits

While classical computing relies on bits that produce deterministic answers, a quantum computer uses qubits whose probability amplitudes define how likely they are to produce an answer. Classical bits can have a state of 0 or 1, however the states of a qubit are represented by the mathematical notation $|0\rangle$ or $|1\rangle$, but can also be in any linear combination of these two states [20]:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2.1}$$

This is what is called a *superposition*. In order to get a result from a qubit, a *measurement* needs to be made, and this *measurement* will cause the qubit to collapse into one of the two states: $|0\rangle$ or $|1\rangle$, the one with the higher coefficient having a higher probability [17].

Introducing more qubits, just like introducing more bits, increases the number of states that the system can be in. For example, a 2 qubit system can be in the states: $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$, as well as any linear combination of these 4 states[1] [20]:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \tag{2.2}$$

[1] Equations 2.1 and 2.1 are both from Nielson and Chuang [20]

Therefore from this it can be seen how with N qubits, 2^N states can be represented in parallel by a quantum system [22]. This is where and why quantum computing could potentially create exponentially faster algorithms than classical computers. However, this is not as easy as it sounds, as measurements and problems with noise can cause difficulties when achieving reliable results [22].

2.1.2 Quantum Gates

Just like in a classical system, quantum machines have logic gates, called 'quantum gates' that allow you to manipulate the state of a qubit. All operations on a qubit, or on qubits, can be represented as a linear transformation, therefore each gate will have a matrix representation of the operation. Following Kasirajan [17], table 2.1.2 shows this, with images for the symbols sourced from [9]. It is worth noting that the states $|0\rangle$ or $|1\rangle$ will not always be the inputs to such gates, as a liner combination of these two states could be inputs. However, for ease of understanding, the truth tables in table 2.1.2 will only contain the results for these simple states. In addition, only the most commonly used gates have been displayed.

2.1.3 Quantum Circuits

Quantum circuits, as expected, are made up of quantum gates. These circuits can be drawn just like logic gate circuits in a classical system. See the coin flip game below for a detailed example.

2.1.3.1 Coin Flip Game

The coin flip game was originally created by David Mayer in 1998, and is an excellent example of a simple game that can be simulated using a quantum circuit [19]. The game plays as follows [27, 6]:

1. A coin is placed Heads up into a box so that both players cannot see the coin within.
2. Player 1 chooses to flip the coin or not.
3. Player 2 chooses to flip the coin or not.
4. Player 1 then chooses for a final time to flip the coin or not.
5. Player 1 will win if the coin lands on Heads, otherwise Player 2 will win.

If using classical probability it would be clear that both Player 1 and 2 would have a 50% chance of winning. However, unbeknownst to Player 2, Player 1 is "utilizing a quantum strategy" [19] which they use to manipulate the state of the coin so it would land on Heads each time. Player 1 puts the coin into a superposition of states, so no matter if Player 2 decides to flip the coin or not, it will stay in a superposition, then finally Player 1 undoes the superposition so that the coin results in Heads again [6].

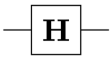
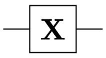
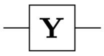
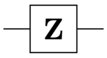
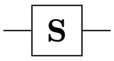
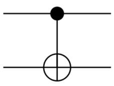
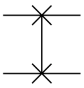
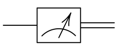
<table>
<tr><th>Name</th><th>Symbol</th><th>Matrix representation</th><th>Truth Table</th><th>Notes</th></tr>
<tr><td>Hadamard</td><td></td><td>$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$</td><td><table><tr><td>Input</td><td>Output</td></tr><tr><td>$|0\rangle$</td><td>$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$</td></tr><tr><td>$|1\rangle$</td><td>$\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$</td></tr></table></td><td>Could also be thought as a bit flip or 'NOT' gate</td></tr>
<tr><td>Pauli-X</td><td></td><td>$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$</td><td><table><tr><td>Input</td><td>Output</td></tr><tr><td>$|0\rangle$</td><td>$|1\rangle$</td></tr><tr><td>$|1\rangle$</td><td>$|0\rangle$</td></tr></table></td><td>Could also be thought as a bit flip or 'NOT' gate</td></tr>
<tr><td>Pauli-Y</td><td></td><td>$\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}$</td><td><table><tr><td>Input</td><td>Output</td></tr><tr><td>$|0\rangle$</td><td>$i|1\rangle$</td></tr><tr><td>$|1\rangle$</td><td>$-i|0\rangle$</td></tr></table></td><td></td></tr>
<tr><td>Pauli-Z</td><td></td><td>$\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$</td><td><table><tr><td>Input</td><td>Output</td></tr><tr><td>$|0\rangle$</td><td>$|0\rangle$</td></tr><tr><td>$|1\rangle$</td><td>$-|1\rangle$</td></tr></table></td><td>Also called the 'phase flip' gate</td></tr>
<tr><td>Phase</td><td></td><td>$\begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix}$</td><td><table><tr><td>Input</td><td>Output</td></tr><tr><td>$|0\rangle$</td><td>$|0\rangle$</td></tr><tr><td>$|1\rangle$</td><td>$i|1\rangle$</td></tr></table></td><td>Also called the S gate</td></tr>
<tr><td>CNOT</td><td></td><td>$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$</td><td><table><tr><td>Input</td><td>Output</td></tr><tr><td>$|00\rangle$</td><td>$|00\rangle$</td></tr><tr><td>$|01\rangle$</td><td>$i|10\rangle$</td></tr><tr><td>$|10\rangle$</td><td>$i|11\rangle$</td></tr><tr><td>$|11\rangle$</td><td>$i|11\rangle$</td></tr></table></td><td>Also known as the controlled-not or controlled x gate. A two-Qubit gate</td></tr>
<tr><td>swap</td><td></td><td>$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$</td><td><table><tr><td>Input</td><td>Output</td></tr><tr><td>$|00\rangle$</td><td>$|00\rangle$</td></tr><tr><td>$|01\rangle$</td><td>$i|10\rangle$</td></tr><tr><td>$|10\rangle$</td><td>$i|01\rangle$</td></tr><tr><td>$|11\rangle$</td><td>$i|11\rangle$</td></tr></table></td><td>Swaps the status of the two qubits. Is implemented using 3 CNOT gates</td></tr>
<tr><td>Measure-ment</td><td></td><td>N/A</td><td>N/A</td><td>Not technically a quantum gate, but represents measuring a Qubit, resulting in one of two states.</td></tr>
</table>

TABLE 2.1: Table displaying some of the most commonly used Quantum gates.

This game can be simulated using a 2-qubit quantum circuit, seen in figure 2.1. Here $|0\rangle$ represents Tails, and $|1\rangle$ represents Heads. The first qubit (top line) is used to represent Player 2's choice to flip the coin, and as all qubits start in a state of $|0\rangle$, we can put it into a superposition using the H-gate, so when measured there is a 50% chance to get either heads or tails. The second qubit (bottom line) is put into state $|1\rangle$ by applying the X-gate (NOT-gate) and then put into superposition [6]. To apply Player 2's decision we use a CNOT-gate, and finally to undo the previous superposition we can use another H-gate before measuring the final outcome [27]. The result of this circuit only relies upon the initial state of the second qubit, thus producing the same outcome as the coin-flip game would suggest.

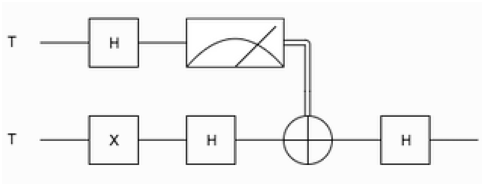


FIGURE 2.1: Circuit that simulates the Coin-flip game. Image sourced from PyQuill Contributors [6]

2.2 Educating Chemists

According to the Journal of Chemical Education, chemistry education needs to be moving towards teaching a "systems thinking" approach. Students need to begin thinking of systems as "dynamic, interconnected components" rather than "isolated, static components" [29]. This is especially important for teaching quantum chemistry, as it can describe very large and complex interconnected systems at a molecular and even subatomic level [21]. To achieve this "systems thinking" within students, different teaching methods can be applied, one of which is the use of interactive tools. Having students explore complex quantum systems themself, can lead to enhancing learning at their own pace, allowing them to gain further understanding [11]. Cruzeiro et al. [11] suggests using Jupyter notebooks to aid in the understanding of a key concept of quantum chemistry: The particle in a box problem (which is explained in detail in section Particle In a Box Example). Before understanding this problem however, students also need to understand what the Schrödinger equation is.

2.2.1 Schrödinger Equation

In 1926 Erwin Schrödinger submitted a paper that first derives the equation we know today as the Schrödinger equation [24]. The solution to this equation will describe a wavefunction ($\psi(x)$) for a particle of mass m. A version of the time independent Schrödinger equation can be seen in equation 2.3 where $V(x)$ is the potential energy of a particle at point x, with energy E and $\hbar$ is a modified version of Planks constant h such that $\hbar = \frac{h}{2\pi}$ [1]. Note that this equation assumes the particle only moves in one dimension.

$$-\frac{\hbar^2}{2m}\frac{d^2\psi}{dx^2} + V(x)\psi(x) = E\psi(x) \tag{2.3}$$

The Born interpretation of the wavefunction states that integrating the wavefunction squared between a and b, gives the probability of finding that particle between a and b [3]. This can be seen in equation 2.4 [18].

$$\text{Prob}(a \leq x \leq b) = \int_b^a \psi^*(x)\psi(x)dx \tag{2.4}$$

Born's interpretation, can also derive a normalisation constant for the wave function, allowing the probability densities for finding particles to be calculated [3]. An example of how this can be used will be described in the section below.

2.2.1.1 Particle In a Box Example

The particle in a box is a simple example of a particle moving within a 1-dimensional space. We define the potential energy of anything within this box to be 0, and anything outside of it to be ∞. This guarantees that the particle must be somewhere within the box [23]. A visualisation of this can be seen in figure 2.2.

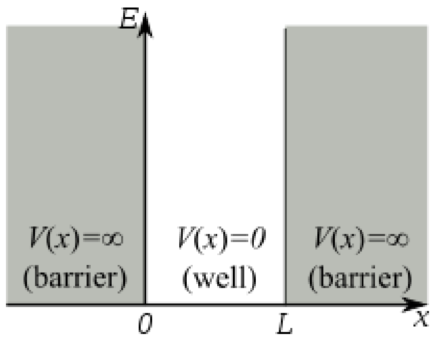
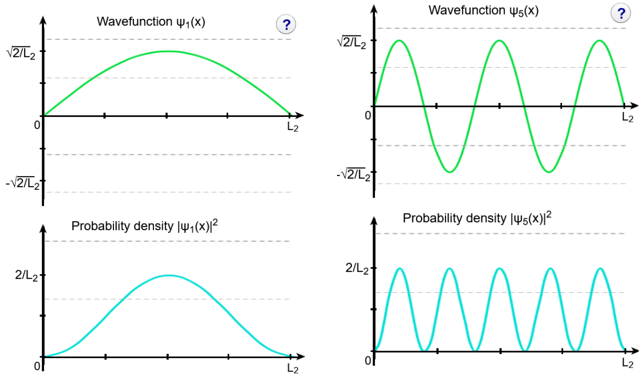


FIGURE 2.2: Visualisation of the particle in a box problem. Image sourced from [8]

From this, it is then possible to use the Born interpretation of the wavefunction [3] to be able to predict the wavefunction and the probability density of finding a particle at different energy states n called "energy quantum numbers" [23]. The Quantum Mechanics Visualisation Project(University of St Andrews) [28] have produced an online simulation tool that aids in visualising the results from performing this calculation. An example of these outputs can be seen in figure 2.3

2.3 Quantum Computing Libraries

There are a number of different languages that provide the functionality to program quantum circuits and computers, all having their own advantages and disadvantages. This assessment will focus on Cirq [13] and Qiskit [7] as their creators provide extensive documentation and cloud-services to access quantum machines. These two open-source pieces of software are 'full stack' applications, as they allow for defining a quantum algorithm and circuit, contains a built in quantum compiler and can run the produced circuit on either a simulation (written in the same language) or a real quantum processor [16]. They are both Python based libraries with



(a) n = 1 (b) n = 5

FIGURE 2.3: Examples of how the wavefunction and probability density changes when the energy quantum numbers change. Images sourced from The Quantum Mechanics Visualisation Project(University of St Andrews) [28]

imperative and functional features, that will allow modularity and easy extensibility of code [25].

An advantage that Cirq has that Qiskit does not, is that Cirq can import and export circuits from JSON files [13]. This could be a useful attribute to aid in extensibility of the tool. In addition, the library has functions to translate circuits to and from PyQuill [26], the Rigetti Computing developed quantum language. The Cirq simulator also allows different noise models to be used, which means that customisability of a QVM is built into the library.

A tool worth noting, is Quirk; an open-source web-based tool that allows the user to build quantum circuits using a drag and drop interface [10]. It also shows the probabilities of the resulting value of each qubit in the circuit, making it a very useful tool to test that a quantum circuit is running as intended. Figure 2.4 shows the Penny Flip game circuit (discussed in section Coin Flip Game) simulated in Quirk.

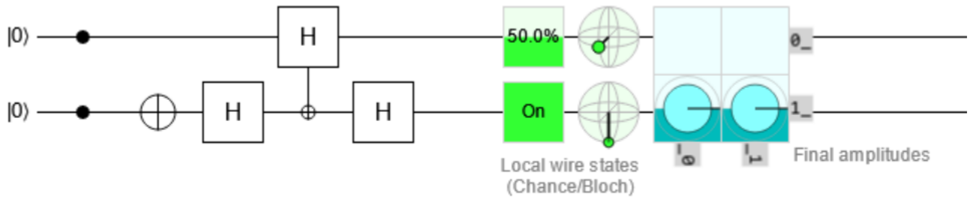
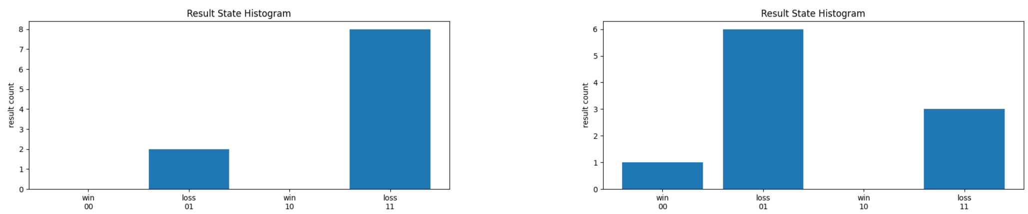


FIGURE 2.4: The penny game circuit created in Quirk. Image sourced from [10]. Note the green box denoting the probability of the two outcomes of measuring a Qubit: On (|1⟩) or Off(|0⟩)

3. Technical Progress

3.1 Quantum Virtual Machine Implementation

Not having easy access to a quantum machine, I set up a QVM, using Cirq's [13] qsim library, as it allows the simulation of some of Google's quantum processors. To ensure it was working as intended, it was tested with the Penny-Flip game circuit described in section Coin Flip Game. Also using this library gave an opportunity to experiment with the noise model that Cirq has built into its simulator; the state histograms from this experiment can be seen in figure 3.1. The first qubit represents the choice of Player 2 to flip the coin or not, and the second quibit represents the final resulting outcome from the coin-flip: Heads or Tails. As can be seen, the simulation with noise does not produce the 100% reliable, mathematically expected results. This is because by introducing noise, the simulation within the QVM becomes more like the current quantum processors that exist today: NISQ devices. By removing the noise, the simulation produces the mathematically expected results.



(a) No noise model

(b) Using a noise model

FIGURE 3.1: Histograms representing the results from simulating the Penny Flip game on a QVM, one with a noise model built into the QVM, one without.

Cirq was used to simulate a quantum machine as it already has some built in functionality to 'translate' different quantum circuits so that they may be run on the Cirq simulator. This functionality will be taken advantage of, as it will help with the ability to use different quantum backends.

3.2 System Design

While learning about the quantum chemistry required, I drafted together a very basic GUI design of the final tool. Inspiration was taken from other educational tools, such as the Virtual Lab from Chem Collective [12] and Phet Interactive Simulations: Quantum Bound States [5]. While the drag and drop nature of some of the Virtual Lab simulations will be difficult to use within the tool produced, the simple colour scheme and intuitive interface is worth using as a reference [12]. The more useful example to look to is the Phet Interactive Simulation [5]. The split between the output panel and input panel is clean and allows ease of use. In addition, the slider approach to increase and decrease the parameters can help guide the user to know what sort of inputs the problem requires.

The image manipulation tool GIMP was also a big inspiration, as it is a complex tool with many parts to it, yet is intuitive to use. The design created can be seen in figure 3.2. The aim is by separating the input and output panels, not only will it be simple to use, but also should allow the tool to be extended for multiple uses easily, alongside having the ability to add other calculations to the tool.

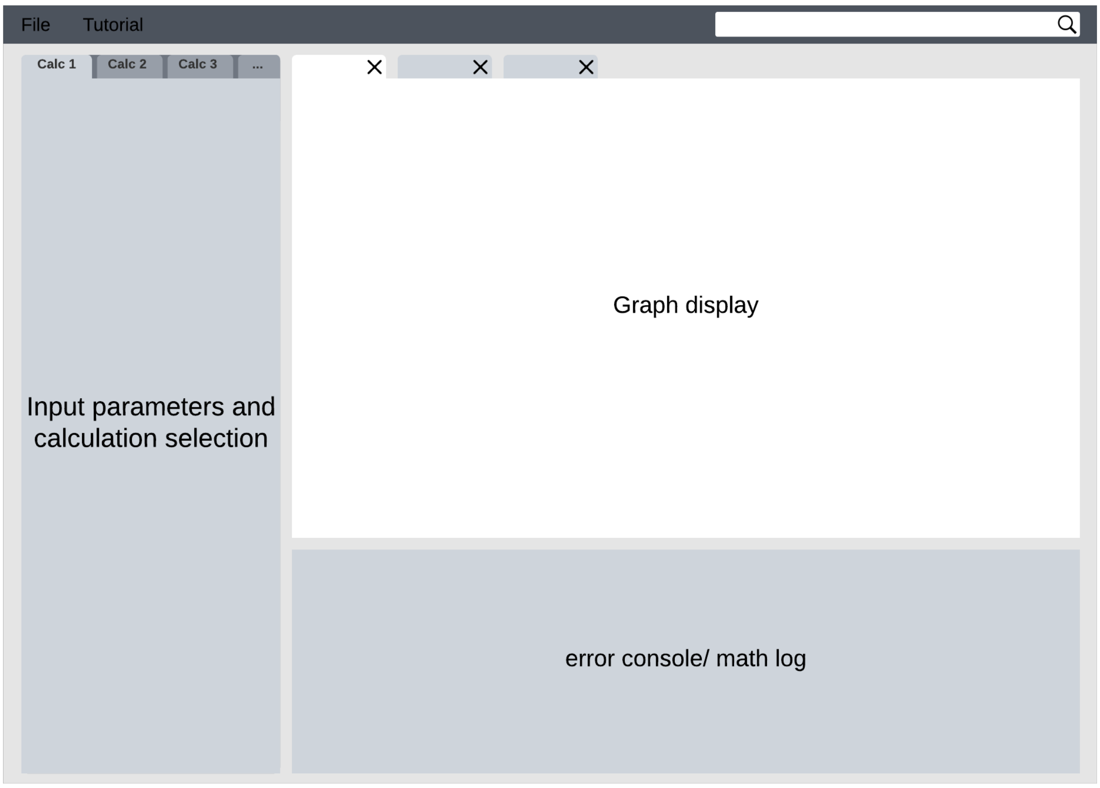


FIGURE 3.2: Basic GUI design of the final tool.

Within the File drop down button, there will be options to customise the backend, whether it be the built in QVM as a part of the tool, or linking it to an actual quantum machine. There will also be tutorials that will teach how to use the tool, as well as some basic quantum chemistry visualisations; the particle in a box model will be one of these tutorials, perhaps as well the Schrödinger equation for a Hydrogen molecule as it is a simple example.

4. Plan on remaining Work

The remaining work in this project can be split into three sections: Remaining research, design and planning, and implementation and testing. See figure 5.1 (bottom) to see a Gantt chart describing the rough time frame for these tasks to be completed within.

4.1 Remaining Research

Whilst most of the research required has already been carried out, I expect that I will need to continue learning more about quantum computing and chemistry throughout this project. More specifically however, research on what quantum algorithm/circuit to implement for simulating Schrödinger's equation is still required. A number of different implementations will be researched and assessed to find which is the best for the desired outcome. More research into how to output the results and display graphs of the results is required.

4.2 Design and Planning

Before moving into the implementation, I intend to plan and design parts of the system so the aim of modularity and ease of extensibility is considered from the start of coding. Using the MoSCoW system will allow prioritisation of tasks, and aid with requirements. A class diagram will also be created, using the Model-View-Controller paradigm as this is an excellent way to ensure modularity, as well as aid with the plan to be able to swap out different backends.

In terms of further GUI designing and planning, I intend to storyboard the planned tutorials before implementation as this will help with testing towards the end of the project.

4.3 Implementation and Testing

For implementation, the backends and frontend will be built separately before linking it together, and testing will be ongoing throughout implementation. Once complete, I will carry out formal testing on the final tool.

5. Project Management

5.1 Risk assessment

Issue	Loss	Prob	Risk	Mitigation
Unexpected event causes large delay in project or miss Gantt chart deadline	2	3	6	Include 3 week contingency plan in Gantt chart so it can be used if necessary. Also be aware that Gantt chart may need to be amended and updated throughout the project.
Unable to simulate a quantum machine on personal computer due to high computing needs	3	5	15	Find alternative ways to simulate one, through cloud infrastructure available. See if it is possible to gain access to the Universitys HPC facilities.
Difficulty learning quantum circuits and the chemistry knowledge required.	3	5	15	Seek help from the Chemistry department for resources to learn the Chemistry required. For the Quantum circuits, ask within ECS department for any help from those with experience. Look up tutorials and classes online to learn.
Google or IBM shut down the cloud services that allow access to Quantum Machines or Unable to gain access to a real quantum machine	4	1	4	Use a QVM to implement the quantum backend and search for an alternate cloud service. If no cloud services available, implement multiple different QVM's to simulate different quantum machines.

Table 5.1: Risk assessment for the project

5.2 Project Planning

Towards the beginning of this project, a Gantt chart was created as a method aid in the breakdown of what tasks needed to be completed and in what sort of expected timeframe. Figure 5.1 (top) shows this initial Gantt chart. The 3 weeks leading up to the final deadline are left empty to allow for time to fix any issues that may arise during the project, as discussed above in the risk assessment. 3-weeks of exams and revision are highlighted in red, as I am aware that not much progress will be made in this time. The upcoming holidays have also been highlighted for my benefit: Christmas (green stripes and red text), Easter (pink stripes and green text).

However, despite creating a Gantt chart to aid with my project planning, things did not go exactly according to plan. I severely underestimated how much time it would take to learn and understand the fundamentals of quantum computing, let alone the quantum chemistry that was required. Due to this, an extra 4 weeks was spent researching, and I was unable to create a class diagram and requirements plan for the project. Therefore, I have revised the initial Gantt chart to show what actually happened as well as edited the plan for next semester. To aid in remedying the situation, more time has been allowed to continue learning before starting implementation, and the plan has been changed to accommodate testing and the writing my final report in parallel. If this does not work out, then the 3-weeks contingency are still in place, and will be used if required. See figure 5.1 (bottom) for the up to date Gantt chart.

In order to avoid the same mistakes, I am aware that I may need to 'learn as I go' while implementing the tool. This will allow me to make progress programming as well as understand exactly what I need to produce the intended tool.

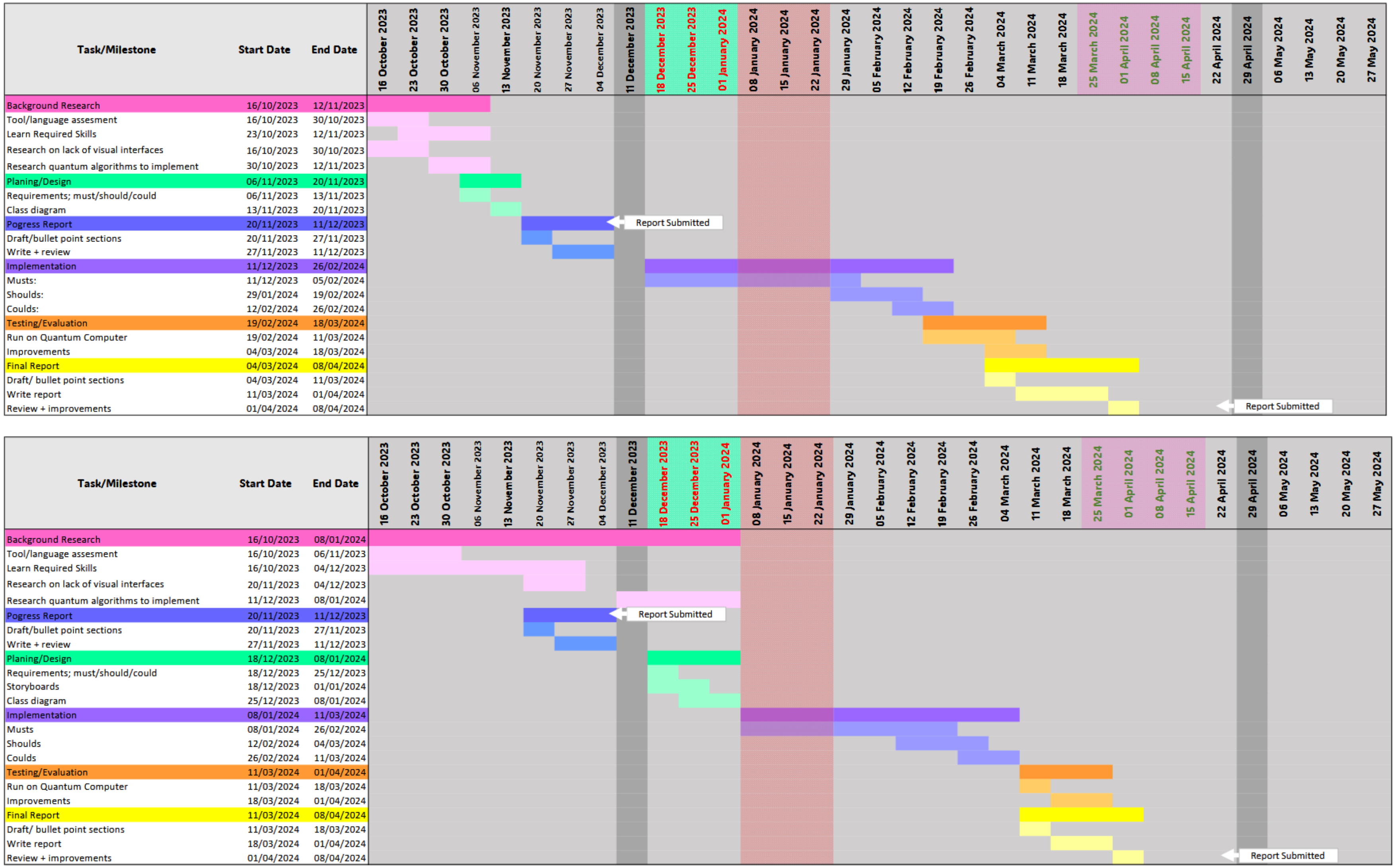


FIGURE 5.1: Gantt charts describing the timeframe of the project.Top: Initial, Bottom: Revised

Bibliography

[1] Peter Atkins and Julio de Paula. *Physical Chemistry*. Oxford University Press, 10th edition, 2014.

[2] Koen Bertels, Aritra Sarkar, and Imran Ashraf. Quantum computing - from nisq to pisq. *IEEE Micro*, 41:24–32, 9 2021. ISSN 19374143.

[3] Max Born. Quantum mechanics of collision processes. *Zeit. Phys*, 38:803–827, 1927. Translated by D. H. Delphenich.

[4] Sitan Chen, Jordan Cotler, Hsin Yuan Huang, and Jerry Li. The complexity of nisq. *Nature Communications*, 14, 12 2023. ISSN 20411723.

[5] Kathy Perkins Chris Malley, Sam McKagan. Phet interactive simulations: Quantum bound states, 2017.

[6] PyQuil Documentation Contributors. The quantum virtual machine: examples of quantum programs, 2018.

[7] Qiskit contributors. Qiskit: An open-source framework for quantum computing, 2023.

[8] Wikipedia Contributors. Particle in a box, 2023.

[9] Wikipedia Contributors. Quantum logic gate, 2023.

[10] Craig Gidney, and other contributors. Quirk, 2019.

[11] Vinícius Wilian D. Cruzeiro, Xiang Gao, and Valeria D. Kleiman. Implementing new educational platforms in the classroom: An interactive approach to the particle in a box problem. *Journal of Chemical Education*, 96:1663–1670, 8 2019. ISSN 19381328.

[12] Sandra Raysor David Yaron, Ryan Dwyer. Chem collective: Virtual lab, 2017.

[13] Cirq Developers. Cirq: A python library for writing, manipulating, and optimizing quantum circuits and running them against quantum computers and simulators., 2023.

[14] Mikhail Dyakonov. The case against quantum computing. *IEEE Spectrum*, 56:24–29, 2019.

[15] Richard P Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21, 1982.

[16] Mark Fingerhuth, Tomáš Babej, and Peter Wittek. Open source software in quantum computing. 12 2018.

[17] Venkateswaran Kasirajan. *Fundamentals of Quantum Computing*, volume 1. Spinger Cham, 1 edition, 6 2021.

[18] Donald A McQuarrie. *Quantum Chemistry*. University Science Books, 2nd edition, 2008.

[19] David A. Meyer. Quantum strategies, February 1999. ISSN 1079-7114.

[20] Michael A. Nielson and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th edition, 2012.

[21] Matthew Otten, Matthew R. Hermes, Riddhish Pandharkar, Yuri Alexeev, Stephen K. Gray, and Laura Gagliardi. Localized quantum chemistry on quantum computers. *Journal of Chemical Theory and Computation*, 18:7205–7217, 12 2022. ISSN 15499626.

[22] Eleanor Rieffel and Polak Wolfgang. An introduction to quantum computing for non-physicists. *ACM Computing Surveys*, 32:300–335, 2000.

[23] William Moebs Samuel J. Ling, Jeff Sanny. *University Physics Volume 3*. OpenStax, 2016.

[24] Erwin Schrodinger. Quantisation as an eigenvalue problem. *Annalen der Physik*, 79, 1926. Translated by Oliver F. Piattella.

[25] Manuel A. Serrano, Ricardo Pérez-Castillo, and Mario Piattini. *Quantum Software Engineering*. Springer International Publishing, 10 2022. ISBN 9783031053245.

[26] Robert S. Smith, Michael J. Curtis, and William J. Zeng. A practical quantum instruction set architecture, 2016.

[27] Presh Talwalkar. Quantum coin flipping game, 2012.

[28] The Quantum Mechanics Visualisation Project(University of St Andrews). The one dimensional particle in a box, 2016.

[29] Sarah York and Mary Kay Orgill. Chemist table: A tool for designing or modifying instruction for a systems thinking approach in chemistry education. *Journal of Chemical Education*, 97:2114–2129, 8 2020. ISSN 19381328.