[redacted]

[redacted]

1 Term Portfolio Project

1.1 Question 01

How did you get up to speed in basic linear algebra (e.g., Euler formula, linear transformations or operators, eigenvectors and eigenvalues) including which resources you consulted in the process?

I got up to speed in basic linear algebra by using the following three resources:

- Reviewing my notes from UVic MATH 211 [1].
- Watching 3Blue1Brown - Essence of Linear Algebra YouTube Playlist [2].
  - Taking notes from Chapters 3-4, 6-9, and 13-15 of the YouTube Playlist.
- Reading Linear Algebra by OpenIntro et al. [3].

Complex Conjugate Transpose, Tensor Product ($\otimes$), and Matrix/Vector Multiplication were three concepts of linear algebra that I had previously not encountered and/or needed to review.

1.2 Question 02

How did you get started documenting linear algebra formulas (e.g., Euler formula) using LaTeX Markdown in Jupyter Notebooks? Develop your own cheat sheet of the quantum computing formulas and Dirac notation to ease assignment typesetting.

I have previous experience using LaTeX and LaTeX Markdown (Obsidian, GitHub, Overleaf).

1.2.1 Euler Identity and Formulas

We have the following Euler Identity: $e^{i\pi} + 1 = 0$. And the following Euler Formulas:

$$e^{ix} = \cos x + i \sin x \quad e^{-ix} = \cos x - i \sin x$$

$$\cos x = \frac{e^{ix} + e^{-ix}}{2} \qquad \sin x = \frac{e^{ix} + e^{-ix}}{2i}$$

1.2.2 Dirac Notation or Bra-Ket Notation

A notation introduced by Paul Dirac in 1930 to represent **quantum states** concisely.

$$\textbf{Ket } |a\rangle = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}, \text{ Column Vector} \qquad \textbf{Bra } \langle b| = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}, \text{ Row Vector}$$

1.2.3 Vector Notation

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} |01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} |11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

1.2.4 Dirac Notation vs Vector Notation

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

1.2.5 Superposition

$$\psi = \alpha|0\rangle + \beta|1\rangle, \quad \alpha, \beta \in \mathbb{C} \text{ and } |\alpha|^2 + |\beta|^2 = 1$$

1.2.6 Quantum Gates

Quantum Gates are a basic quantum circuit operating on a small number of qubits. They are represented by unitary matrices.

The general form of a quantum gate is: $U_f|x\rangle = |f(x)\rangle$ and $B_f|xy\rangle = |xf(x) \otimes y\rangle$.

Notable Quantum Gates:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \quad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}$$

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

1.3 Question 03

How did you get started and experienced running your first quantum circuits using IBM Qiskit and Microsoft QDK Q# Jupyter Notebook platforms?

I got started with quantum circuits by using IBM Qiskit and IBM Quantum Composer. Qiskit is well integrated with Jupyter Notebooks, the setup was simple and only required importing python packages locally and installing the Jupyter Notebook VS Code Extension. The IBM Quantum Composer provided a visual interface to create quantum circuits and provided the Qiskit code to run the quantum circuits. Qiskit also has a textbook [4] that provides examples of quantum circuits and quantum computing concepts.

I had tried to use Microsoft QDK Q# but it did not work with my MacBook Air M1 (issues with Anaconda and .Net). I then tried using the Azure Portal and Notebooks but the interface was unreliable and slow. In the end I had to use my Lenovo ThinkPad E14 Gen 4 AMD to run Microsoft QDK Q# with Anaconda.

1.3.1 Azure Quantum Provider

An interface to submit jobs to Azure Quantum with Qiskit.

```
# Example Code to Connect to Azure Quantum
from azure.quantum.qiskit import AzureQuantumProvider
provider = AzureQuantumProvider (
    resource_id = "resource_id",
    location = "westus"
)
```

1.3.2 QSharp Language

```
<!-- Example Code to create a Quantum Circuit -->
open Microsoft.Quantum.Diagnostics;
open Microsoft.Quantum.Measurement;

operation methodName() : Result {
    use target = Qubit();
    X(target);
    H(target);
    return M(target);
}
%simulate methodName
```

1.3.3 Qiskit Language

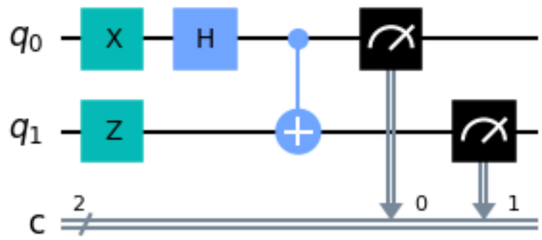
```
# Example Code to create a Quantum Circuit
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit

qreg_q = QuantumRegister(2, 'q')
creg_c = ClassicalRegister(2, 'c')
circuit = QuantumCircuit(qreg_q, creg_c)

circuit.x(qreg_q[0])
circuit.z(qreg_q[1])
circuit.h(qreg_q[0])
circuit.cx(qreg_q[0], qreg_q[1])
circuit.measure(qreg_q[0], creg_c[0])
circuit.measure(qreg_q[1], creg_c[1])

circuit.draw(output='mpl', scale=0.75)
```

1.3.4 Quantum Circuit Visualisation



1.3.5 Quantikz Package

A LaTeX Package [5] that allows quantum circuits to be drawn using LaTeX.

```
% Example Code to Draw Quantum Circuits
\begin{quantikz}
    \lstick{$\ket{1}$} & \qw & \hGate & \qw & \xGate & \qw
\end{quantikz}
```

1.4 Question 04

How would you motivate other students to join the journey into quantum computing given the motivational materials presented in class and found in the references? Your answers to this question will likely evolve during this course. Revisit regularly.

I would motivate other students to join the journey into quantum computing by…

- Introducing them to the Qiskit Textbook. The Qiskit Textbook is a good starting point for students to learn about Quantum Computing without overwhelming them with the mathematics.
- Encouraging them to study SENG 480D by showing them what I learned in the course. I would specifically target students who have expressed interest in algorithms and computer science theory.
- Showing them research opportunities in quantum computing (i.e., papers, articles, grants, funding).

1.4.1 Links

- CQIQC Undergraduate Summer Research Studentships
- A Fast Quantum Mechanical Algorithm for Database Search
- Isabelle Marries Dirac: A Library for Quantum Computation and Quantum Information

1.5 Question 05

What are your personal insights, aha moments, and epiphanies you experience in the first part of your quantum learning journey?

I notable moment was when I was studying and researching the logic behind the $U_{ROT_k} = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{bmatrix}$ gate. I was able to connect the U_{ROT_k} gate with the phase shift gate. The connection allowed me to better understand the Bloch Sphere.

My personal insights are usually questions that I have about the course content after each lecture. A few of these questions:

- Can the target qubit be the top and the control qubit be the bottom?
- Is quantum entanglement limited to pairs of qubits?
- Is quantum teleportation limited to 3 qubits? Quantum Teleportation Proof.
- Does quantum teleportation collapse the state? and is partial teleportation possible?
- Observation: can phase kickback be both constant and balanced?

The majority of the questions that I have after each lecture are usually answered in the next lecture. Thus, a personal insight would be to write down questions as they come up and then to research the questions after the lecture.

1.6 Question 06

Which algorithm is your favourite quantum algorithm so far?

My favourite quantum algorithm so far would be the Quantum Teleportation Algorithm. The concept is simple and a great addition to the No Cloning Theorem. It uses the Bell States to teleport the state of one qubit to another qubit. However, I'm not particularly fond of the use of Alice, Bob, and Eve in the algorithm setup. I would be interested in a different setup that does not use the names, but still effectively conveys the same idea.

In addition, the notion that no matter or energy is transferred between them is interesting. It also leads into the algorithm for swapping Bell pairs (an algorithm used in distributed quantum networks).

1.7 Question 07

What was the most challenging part of understanding Grover's algorithm?

The most challenging part of understanding Grover's Algorithm was understanding the setup and the concept of an oracle. Truthfully, I still do not understand Grover's Algorithm. I need to further consult the IBM Quantum Documentation [6] to learn more about the algorithm.

1.7.1 Further Reading

I need to review the concept of an oracle and the concept of phase kickback.

- Quantum Hoare Logic
- A Fast Quantum Mechanical Algorithm for Database Search
- Complete 3-Qubit Grover Search on a Programmable Quantum Computer
- Qiskit: Grover's Algorithm

2 Term Portfolio Project

2.1 Question 01

In your own words, describe the Quantum Fourier Transform (QFT) algorithm. How is QFT used in quantum computing?

The Quantum Fourier Transform (QFT) is a quantum algorithm that transforms a quantum state in the computational basis to its corresponding Fourier basis.

It can be described as the following formula:

$$QFT|x\rangle = |\tilde{x}\rangle$$

where, $|\tilde{x}\rangle$ is the Fourier basis.

Expanding the formula we obtain the following:

$$QFT|x\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi ikx}{N}} |k\rangle$$

where, N is the dimension of the vector space, i is the imaginary unit, and $k = 0, 1, 2, ..., N-1$.

Therefore, the QFT maps the computational basis states to a superposition of all possible Fourier basis states.

In addition, the QFT is a fundamental subroutine (i.e., building block) in quantum computing. It can be used as a subroutine in Shor's algorithm (factoring and computing discrete logarithm) and in Quantum Phase Estimation algorithm (estimating the eigenvalues of unitary operators).

The circuit that implement the QFT uses two gates: a single-qubit Hadamard gate, and a two-qubit controlled rotation gate [Qiskit: Quantum Fourier Transform].

Single-Qubit Hadamard Gate

$$H|x_k\rangle = \frac{1}{\sqrt{2}}(|0\rangle + exp(\frac{2\pi i}{2}x_k)|1\rangle)$$

Two-Qubit Controlled Rotation Gate

$$CROT_k = \begin{bmatrix} I & 0 \\ 0 & UROT_k \end{bmatrix}$$

$$UROT_k = \begin{bmatrix} 1 & 0 \\ 0 & exp(\frac{2\pi i}{2^k}) \end{bmatrix}$$

QFT Properties

- Unitary: Normalization Factor of $1/\sqrt{N}$, where N is usually represented as $N = 2^n$.
- Convolution-Multiplication:
  - Input: Shifted Cyclical Amplitudes

- Output: Static Probability Distribution
- Periodicity: Transformation of Periodic Function

2.1.1 Topics to Research / Investigate

The Hidden Subgroup Problem (HSP)

The HSP are problems related to the field of mathematics and theoretical computer science [Wiki: Hidden Subgroup Problem]. Shor's algorithm relies on quantum computers that solve the HSP for finite Abelian groups. The HSP also encompasses the following subset of related problems: Graph Isomorphism and Shortest Vector Problem.

The Hidden Shift Problem (HSP)

The HSP is defined as the following:

Let $M = N^{\epsilon}$ for any fixed $\epsilon > 0$. Then there is an efficient (i.e., run time $poly(\log N)$) quantum algorithm for the generalized hidden shift problem, using entangled measurements on $k = max\{3, \log \frac{1}{\epsilon}\}$ registers.

2.1.2 Articles to Read

- "A New Quantum Algorithm for the Hidden Shift Problem in $\mathbb{Z}_{2^{\mathfrak{t}}}^{\mathfrak{n}}$" by Gergely Csáji
- "Quantum Algorithm for a Generalized Hidden Shift Problem" by Andrew M. Childs and Wim van Dam
- "Shor's Algorithm and the Quantum Fourier Transform" by Fang Xi Lin

2.2 Question 02

In your own words, describe the Quantum Phase Estimation (QPE) algorithm. How is QPE used in quantum computing?

The Quantum Phase Estimation (QPE) algorithm is a subroutine in quantum computation that uses phase kickback and the inverse of the Quantum Fourier Transform (QFT) to estimate the eigenvalues of a unitary operator [Qiskit: Quantum Phase Estimation], [IBM Quantum: Quantum Phase Estimation]. The algorithm serves as a building block for quantum algorithms such as the Shor's algorithm and the Quantum Amplitude Estimation algorithm.

A Quantum Circuit of the QPE algorithm typically involves two sets of qubits: the counting qubits and the target qubits. The counting qubits are initialized in the state $|0\rangle$, and the target qubits are prepared in the eigenstate of the unitary operator. The algorithms applies a series of controlled unitaries U^{2^j} to the target qubits, where j is the index of the counting qubit.

Thus, the main idea of the QPE algorithm is that the phase of the eigenstate of the unitary operator is encoded in the state of the counting qubits. Using the inverse of the QFT on the counting qubits we can obtain an estimation of the phase. We then obtain an estimation of the eigenvalue.

QPE Properties

- Hadamard Gates

- Controlled Unitaries
- Inverse QFT

2.2.1 Topics to Research / Investigate

Unbiased Quantum Phase Estimation Algorithm (UQPEA)

Xi Lu and Hongwei Lin published a paper in 2022 proposing an Unbiased Quantum Phase Estimation (UQPE) algorithm. The algorithm would implement an additional step to make the Quantum Counting algorithm unbiased. It would also improve the robustness of the Quantum Phase Estimation (QPE) algorithm. The paper makes special consideration about the limitations of a direct substitution of the UPEA and PEA algorithms. A direct substitution alone does not make Quantum Counting unbiased.

2.3 Question 03

What is the role of eigenvalues and eigenstates (eigenvectors) in quantum computing?

Eigenvalues and eigenstates play a crucial role in Quantum Computing. In particular, they are important for Quantum Algorithms that involve quantum state preparation, quantum phase estimation, and quantum algorithms for linear algebra [Qiskit: Quantum Phase Estimation].

We know that eigenvalues are used when determining if an operator is Normal and Hermitian.

Consider an operator A then...

- A is said to be Normal if $AA^\dagger = A^\dagger A$, where A is diagonalisable if and only if it is normal.
- A is said to be Hermitian if $A = A^\dagger$.

Additionally, a Normal operator is Hermitian if, and only if, it has real eigenvalues.

Consequently, all eigenvalues of a unitary operator have modulus 1. For example, the Pauli Matrices all have eigenvalues of ± 1 since they are unitary and Hermitian.

Quantum Algorithms prepare states that are eigenstates of a Hermitian operator. They can also estimated the eigenvalues of a Hermitian operator. Notably, in Shor's algorithm and in the Quantum Phase Estimation (QPE) algorithm.

Furthermore, the eigenstates of Hermitian operators represent the states of a system that always measures the same result and the eigenvalues represent the possible outcomes of the measurements.

Given an unitary operator U, the algorithm estimates θ in $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$, where $|\psi\rangle$ is an eigenstate and $e^{2\pi i\theta}$ is the corresponding eigenvalue (i.e., norm of 1).

$$U^{2^j}|\psi\rangle = \frac{1}{2^{\frac{n}{2}}}\sum_{k=0}^{2^n-1} e^{2\pi i\theta k}|k\rangle \otimes |\psi\rangle$$

where k is an integer representing n-bit binary numbers and $0 \leq j \leq n-1$.

In addition, the Variational Quantum Eigensolver (VQE) a hybrid quantum-classical algorithm that aims to find the minimum value of some cost objective uses eigenvalues and eigenstates.

2.3.1 Topics to Research / Investigate

Changpeng Shao published a paper in 2020 titled "Computing Eigenvalues of Diagonalizable Matrices in a Quantum Computer". Shao proposes an improvement on an algorithm to solve the eigenvalue problem. It closely follows the Quantum Phase Estimation algorithm. The paper explores the possibilities of cases such as diagonalizable matrices that have complex eigenvalues. Shao's main goal is to estimate the eigenvalues of non-unitary (non-Hermitian) matrices.

2.3.2 Articles to Read

- "A Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors" by Daniel S. Abrams and Seth Lloyd

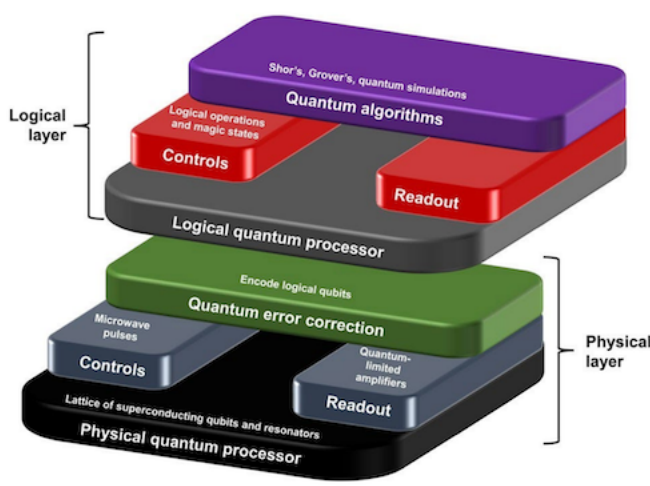
2.4 Question 04

Summarize the decomposition strategies presented in class for developing hybrid quantum-classical systems.

The development of Hybrid Quantum-Classical Systems are supported by an integration of three main components: Classical Computers (e.g., Personal Laptop), Quantum Cloud Service (e.g., IBM Quantum Computing), and Quantum Processing Unit (QPU).

- Classical Computer → Quantum Circuit → Quantum Cloud Service
- Quantum Cloud Service → Job → QPU
- QPU → Result → Quantum Cloud Service
- Quantum Cloud Service → Result → Classical Computer

In class the following illustration was presented:



where the stack is sub divided into a logical layer and a physical layer.

The logical Layer: Logical Quantum Processor, Controls (Logical Operators and Magic State), Readout, and Quantum Algorithms (Shor's, Grover's, and Quantum Simulations).

The Physical Layer: Physical Quantum Processor (Lattice of Superconducting Qubits and Resonators), Controls (Microwave Pulses), Readout (Quantum Limited Amplifiers), and Quantum Error Correction (Encode Logical Qubits).

A hybrid quantum processor can take advantage of a hybrid quantum stack. The hybrid stack consists of the following:

- Topological Qubits (Quantum Plane)
- Cryogenic Control (Quantum-Classical Interface)
- Room Temp Control (Quantum-Classical Interface)
- Classical Compute (Applications and Software/Solutions)

2.4.1 Table: Decomposition Strategies

Decomposition	Strategy
Decomposition of Problems	Quantum Simulation - Classical Solution Pipelines with Natural Quantum Parts - Decomposing Molecules - State Prep (Ansatz) and Hamiltonian - Divide and Conquer - Dynamic Programming
Decomposition of Algorithms	Classical and Quantum Parts - Building Blocks - VQE and QAOA
Decomposition of Programs	Libraries and Building Blocks - Intermediate Representations - Quantum-Classical Runtimes - Cloud Platforms - Simulators vs. Quantum Computers - Qiskit and OpenQasm3
Decomposition of Circuits	Circuit Depth Limitations - Qubit Topology

The table lists a few types of decomposition strategies related to hybrid quantum-classical computing. We can see that the majority of the decomposition strategies consist of breaking down complex problems into smaller, more manageable subproblems that can then be tackled classically or quantumly. We can then use the above hybrid stacks to solve problems with these decompositions.

2.5 Question 05

Which one is your favourite decomposition strategy? Describe this strategy in detail.

I don't have any preferred or favourite decomposition strategy. The topic of decomposition for hybrid classical-quantum computing is something I will need to explore more outside of the classroom before I can gain a solid opinion.

Lanzagorta and Uhlmann were some of the first to present hybrid algorithms. The main idea was that algorithms would benefit from quantum computing. The quantum algorithms would serve as subroutines to speed up classical algorithms.

In addition, Frank Phillipson and Niels Neumann and Robert Wezeman have recently proposed two different strategies for hybrid quantum computing. The following are the current proposed categorizations:

- Vertical Hybrid Quantum Computing: All controlling activities required to control and operate a quantum circuit on a quantum computer (Decomposition, Implementation, and Controlling Hybrid).
- Horizontal Hybrid Quantum Computing: All operational activities required to use a quantum computer and a classical computer to perform an algorithm (Processing Hybrid, Micro Hybrid Split, Macro Hybrid Split, Parallel Hybrid, and Breakdown Hybrid).

2.5.1 Articles to Read

- "Decompositions of General Quantum Gates" by Mikko Möttönen and Juha J. Vartiainen
- "OpenQASM 3: A Broader and Deeper Quantum Assembly Language" by IBM Quantum, AWS Center for Quantum Computing, and Dept. Informatics at the University of Sussex
- "Classification of Hybrid Quantum-Classical Computing" by Frank Phillipson and Niels Neumann and Robert Wezeman
- "Hybrid Quantum-Classical Computing Models" by Guen Prawiroatmodjo
- "Hybrid Quantum-Classical Computing with Applications to Computer Graphics" by Marco Lanzagorta and Jeffrey K. Uhlmann
- "No-Go Theorem and a Universal Decomposition Strategy for Quantum Channel Compilation" by Weiyuan Gong, Si Jiang, and Dong-Ling Deng

I'm the most interested in reading about the No-Go Theorem. The No-Go Theorem is a presentation of a sequence of events that may never occur. The paper briefly mentions the theorem as...

> Given a finite set of elementary channels with arbitrary unitary gates, it is impossible to compile an arbitrary single-qubit channel to arbitrary accuracy.

Thus,

> It implies that a finite number of elementary channels cannot approximate an arbitrary target channel to arbitrary accuracy, regardless of the specific structure of each elementary channel and the length of the compiling sequence.

2.6 Question 06

Write a summary of the different quantum applications presented by the graduate students.

Quantum Applications

- Reinforcement Learning: Is a subset of machine learning that exploits neural networks. Often time there are two subgroups of unsupervised and supervised learning. It is categorized as a continuous interaction with an agent and a controlled environment.
- Machine Learning: ML protocols can be split into two categories: supervised and unsupervised. Where one is reliant on pre-labelled categories and the other is reliant on it's learning to create categories.
- Cryptography: A technique focused on securing communication and data. It aims to prevent data breaches and secure sensitive data.

- Quantum Annealing: An optimization process for finding the global minimum. It uses quantum fluctuations to solve the problems.
- Portfolio Optimization: A process to select the best portfolios out of a set. The goal is to maximize reward.
- Quantum Networks: The transmission of information using qubits between quantum processors.

The following are brief summary of the presentations delivered by the graduate students.

Quantum Reinforcement Learning (QRL): Consists of the following model of agent, environment, action, reward, and policy. QRL is different from Machine Learning (ML) since the samples required to learn need to be collected from the environment. Adding Quantum Computing to Classical Reinforcement Learning (CRL) helps solve the downfalls such as slow training, narrow generalization, and instability.

Quantum Associative Memory and Pattern Recognition: Associative Memory is the ability to retrieve a stored memory based on a partial/incomplete cue. Machine Learning aids in this aspect by being able to perform parallel computation. The algorithm presented relies on two phases; the training phase and the recall phase.

Post-Quantum Cryptography (Multi-Collision Finding): Consider a k-Collision problem. We want to find a set of k distinct inputs that all map to the same output. The Quantum query complexity for finding k-collisions can be represented as $\Theta(N^{1/3})$ where as the best classical case can be represented as $\Theta(N^{1/2})$. In addition, multi-collisions can be used as a proof-of-work for blockchains.

Solving Phase Unwrapping Problem Using Quantum Annealing: The phase unwrapping problem can be described as a hard computational problem. The goal is to find the absolute change in the phase for each pixel. Quantum Annealing can be used to solve these hard computation problems.

Quantum Computing for Portfolio Optimization: In context, the Modern Portfolio Theory is choosing investments for maximum benefit. It also takes advantage of Quantum Annealing, similar to the Phase Unwrapping Problem. Unfortunately, the algorithms that have been created cannot be fully utilize until quantum computing has matured.

Portfolio Optimization Using Quantum Computing: The portfolio optimization problem can be defined as trying to find the best distribution of assets to maximize or minimize a desired metric. The Quantum Computing solution uses VQE, QAOA, and Quantum Annealing. A solution that was proposed used QPLEX a python library.

Quantum Generative Adversarial Networks (QGCAN): Classical GANS have two neural networks; Discriminator (distinguish real from fake) and Generator (makes fakes close to real). Quantum Computing unloads the taxing cost of Classical GANS and speeds up the training.

Quantum Machine Learning (QML): Trying to find the k-nearest neighbours (KNN) where we define k as the number of nearest neighbours and we define a distance metric to measure the similarity. It also encompasses Quantum Convolutional Neural Network (QCNN). QCNN finds the hidden states until the fully connected circuit is able to predict the classification results. Finally, Quantum Error Correction (QEC) helps detect and correct local quantum errors.

Protein Structure Problems Solved Quantumly: Using Quantum Computing we can use Deep Learning to model weaknesses. It also utilizes Quantum Annealing. The problem also involves Dark Proteome which are proteins with no defined three-dimensional structure.

Quantum-Classical Algorithms C++: QPP is a C++ extension that uses Quantum Computing. It serves as a quantum computing library which is not restricted to qubit systems or specific quantum information.

2.7 Question 07

Which applications inspired you the most? Describe this application in more detail.

The application that has most inspired me is the one focused on Portfolio Optimization using Quantum Computing. Its ability to solve a problem that was previously unknown to me during the presentation was truly remarkable. Essentially, the problem pertains to the fact that when investing, the objective is to maximize rewards while minimizing losses, making perfect sense.

The approach involves utilizing D-Wave adiabatic quantum computation to select a binary portfolio of stocks in an optimal manner. This application of Quantum Computing is efficient and robust in solving the problem. The solution methodology comprises problem formulation, data collection, classical preprocessing, quantum analysis and validation of results, and finally classical postprocessing. It is important to acknowledge that this application is subject to certain constraints, which include the common limitations of hardware, hybrid algorithms, and the need for noise and error correction.

2.7.1 Topics to Research / Investigate

N. Slate, E. Matwiejew, S. Marsh, and J. B. Wang have proposed a high-quality solution for Portfolio Optimization that takes advantage of Quantum Walk algorithms. The proposal is a Quantum Walk Optimization Algorithm.

- "Quantum Walk-Based Portfolio Optimization" by N. Slate, E. Matwiejew, S. Marsh, and J. B. Wang

---

And while Portfolio Optimization research does not pique my personal interest, it has certainly motivated me to explore other unfamiliar applications that may be of interest to me.

2.8 Question 08

What have been your personal insights, aha moments, or epiphanies you experienced in the second part of your quantum learning journey?

During the second phase of my journey in quantum learning, I focused on augmenting the course material with self-learning. My research primarily involved investigating various quantum applications and exploring their implications for classical computer theory. For instance, I delved into the subject of decidable and undecidable problems concerning quantum automata, specifically whether the language recognized by a quantum finite automaton is empty or nonempty. I also began learning about Quantum Turing Machines, particularly the concept of a Universal Quantum Turing Machine.

2.8.1 Quantum Computing and Development Environments

Although the course did not largely focus on the technology surrounding Quantum Computing, I have acquired useful knowledge beyond the course's intended scope. Here are some of the topics I have learned about, in no particular order:

- BibTeX is a LaTeX tool that facilitates citation of references.
- GitHub Codespaces is a cloud-based tool that enables coding.
- Jupyter Notebook is a text file tool that enables writing and running of code.
- Pandoc is a tool that converts markdown and `.ipynb` files into PDF format.

Out of all the tools I have learned, the one that proved to be most useful is GitHub Codespaces. I encountered numerous challenges while attempting to set up Jupyter Notebook with a Q# kernel on my computer, which ultimately hindered my ability to complete some assignments satisfactorily. Consequently, I was highly motivated to establish a development environment that aligned with my workflow.

In order to setup the desired development environment, I had to learn about the following:

- `.devcontainer`
- Configuration and Customization
- Docker Images
- GitHub Codespaces
- Jupyter Notebook
- Microsoft Q# Docker Image

Although not comprehensive, the aforementioned list provides an excellent foundation for individuals seeking to establish a development environment. I opted to utilize GitHub Codespaces as my development environment due to my preference for cloud-based solutions that are speedy, free, and bypass any operating system-related complications.

GitHub Codespaces Template: github.com/FlyteWizard/verbose-octo-engine

The template is a basic development environment and lacks any supplementary VS Code plugins. However, my present GitHub Codespaces instance features additional beta VS Code plugins that are inaccessible to the general public, including Copilot Nightly, Copilot Labs, GitHub Copilot Voice.

General Setup

GitHub Codespaces leverages dev containers to host Docker images. Below is the foundational code for such an instance:

```
{
    "image": "mcr.microsoft.com/quantum/iqsharp-base:latest",
    "extensions": [ "quantum.quantum-devkit-vscode", "ms-vscode.csharp"]
}
```

Microsoft provides the Docker image that configures the remote development environment, equipped with all the essential tools required for running Jupyter Notebook with a Q# kernel.

As well, one could argue that Microsoft should offer their own official setup as a default GitHub Codespaces template, along with comprehensive instructions on how to configure the remote development environment in their documentation.

Resources

- https://learn.microsoft.com/en-us/azure/quantum/install-overview-qdk
- https://github.com/microsoft/iqsharp/#using-iq-as-a-container
- https://github.com/microsoft/Quantum/tree/master/.devcontainer

2.8.2 Algorithms and Proofs

Isabelle is a generic proof assistant. Isabelle/HOL, on the other hand, is a theorem proving environment that utilizes higher-order logic.

A group composed of Anthony Bordg, Hanna Lachnitt, and Yijun He created a library for quantum computing, which they called "Isabelle Marries Dirac: A Library for Quantum Computation and Quantum Information." This library formalizes the no-cloning theorem, quantum teleportation, Deutsch's Algorithm, the Deutsch-Jozsa Algorithm, and the Quantum Prisoner's Dilemma.

An example from the formalized proof…

Quantum Gate: Identity Matrix

definition $Id :: nat \Rightarrow complex\ mat$ **where**
$Id\ n \equiv 1_m(2^n)$

lemma *id-is-gate [simp]:*
gate n (Id n)
$\langle proof \rangle$

Motivation

In my journey to deepen my knowledge of formal proof systems, I came across Isabelle. Not only does it serve as a proof assistant, but it also provides access to an archive of papers and research on proof formalizations.

I am not familiar with all the technical intricacies involved in the process. However, exploring the archive has been informative and has aided in my understanding of the algorithms covered in class.

Resources

- Isabelle: https://isabelle.in.tum.de
- Isabelle Marries Dirac: https://www.isa-afp.org/entries/Isabelle_Marries_Dirac

2.9 Question 09

How would you motivate other students to join the journey into quantum computing given the motivational materials presented in class and found in the references? Your answers to this question will likely evolve during this course. Revisit regularly.

To inspire students to delve into the world of quantum computing, I would showcase quantum solutions to classical problems. Similar to the graduate students who presented on quantum computing topics during their term projects, I would encourage exploration and understanding of this exciting field to potential students.

Furthermore, I would enhance course curriculums by providing supplementary reading materials such as articles and papers that offer a quantum computing perspective on the same topics.

Moreover, I would actively encourage students to engage in research opportunities related to quantum computing, such as the CREATE Undergraduate Summer Research Top Up Scholarship. And while priority is given to students who already receive support from NSERC, USRA, or other research scholarships, the application period has been extended, and eligible students are encouraged to apply.

Resources

- https://www.nserc-crsng.gc.ca/students-etudiants/ug-pc/usra-brpc_eng.asp

Bibliography

[1] D. Charlebois, "MATH 211 - Summer 2021." May 2021. Accessed: Jan. 13, 2023. [Online]. Available: https://notes.dominiquecharlebois.com/math211

[2] 3Blue1Brown, "Essence of Linear Algebra - YouTube," *Essence of Linear Algebra*. Aug. 2016. Accessed: Jan. 13, 2023. [Online]. Available: https://www.youtube.com/playlist?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab

[3] J. Hefferon, "Free Linear Algebra text, from Jim Hefferon," *Free Math Texts*. Jun. 2021. Accessed: Jan. 13, 2023. [Online]. Available: https://hefferon.net/linearalgebra

[4] A-tA-v *et al.*, "Qiskit: An Open-source Framework for Quantum Computing." 2021. doi: 10.5281/zenodo.2573505.

[5] A. Kay, "Quantikz." Royal Holloway, University of London, 2019. doi: 10.17637/RH.7000520.

[6] I. Quantum, "Grover's algorithm," *IBM Quantum*. Accessed: Feb. 19, 2023. [Online]. Available: https://quantum-computing.ibm.com/lab/docs/iqx/guide/grovers-algorithm

[7] D. S. Abrams and S. Lloyd, "A Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors," *Physical Review Letters*, vol. 83, no. 24, pp. 5162–5165, Dec. 1999, doi: 10.1103/PhysRevLett.83.5162.

[8] A. Ajagekar and F. You, "Quantum Computing for Energy Systems Optimization: Challenges and Opportunities," *Energy*, vol. 179, pp. 76–89, Jul. 2019, doi: 10.1016/j.energy.2019.04.186.

[9] V. D. Blondel, E. Jeandel, P. Koiran, and N. Portier, "Decidable and Undecidable Problems about Quantum Automata," *SIAM Journal on Computing*, vol. 34, no. 6, pp. 1464–1473, Jan. 2005, doi: 10.1137/S0097539703425861.

[10] A. Bordg, H. Lachnitt, and Y. He, "Isabelle Marries Dirac: A Library for Quantum Computation and Quantum Information," *Archive of Formal Proofs*, Nov. 2020, Available: https://isa-afp.org/entries/Isabelle_Marries_Dirac.html

[11] D. Charlebois, "FlyteWizard/verbose-octo-engine: v0.1.0-alpha." Zenodo, Apr. 2023. doi: 10.5281/zenodo.7792051.

[12] M.-C. Chen *et al.*, "Quantum-Teleportation-Inspired Algorithm for Sampling Large Random Quantum Circuits," *Physical Review Letters*, vol. 124, no. 8, Feb. 2020, doi: 10.1103/physrevlett.124.080502.

[13] A. M. Childs and W. van Dam, "Quantum Algorithm for a Generalized Hidden Shift Problem." Jul. 2005. doi: 10.1145/1283383.1283515.

[14] C. University, "arXiv.org e-Print archive." Accessed: Feb. 18, 2023. [Online]. Available: https://arxiv.org/

[15] A. Cross *et al.*, "OpenQASM 3: A Broader and Deeper Quantum Assembly Language," *ACM Transactions on Quantum Computing*, vol. 3, no. 3, pp. 1–50, Sep. 2022, doi: 10.1145/3505636.

[16] A. Cross *et al.*, "OpenQASM 3: A Broader and Deeper Quantum Assembly Language," *ACM Transactions on Quantum Computing*, vol. 3, no. 3, pp. 1–50, Sep. 2022, doi: 10.1145/3505636.

[17] C. Figgatt, D. Maslov, K. A. Landsman, N. M. Linke, S. Debnath, and C. Monroe, "Complete 3-Qubit Grover Search on a Programmable Quantum Computer," *Nature Communications*, vol. 8, no. 1, Dec. 2017, doi: 10.1038/s41467-017-01904-7.

[18] Gergely Csáji, "A New Quantum Algorithm for the Hidden Shift Problem in $\mathbb{Z}_{2^t}^n$." arXiv, Feb. 2021. doi: 10.48550/arXiv.2102.04171.

[19] L. K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search." arXiv, 1996. doi: 10.48550/ARXIV.QUANT-PH/9605043.

[20] Guen Prawiroatmodjo, "Hybrid Quantum-Classical Computing Models," *Q# Blog*. Dec. 2021. Accessed: Apr. 02, 2023. [Online]. Available: https://devblogs.microsoft.com/qsharp/hybrid-quantum-classical-models/

[21] H. Gupta, "Quantum Fourier Transform and Basic QPE — QPE Algorithms," *Quantum Untangled*. Jun. 2021. Accessed: Apr. 01, 2023. [Online]. Available: https://medium.com/quantum-untangled/quantum-fourier-transform-and-basic-qpe-qpe-algorithms-1af4d08b661c

[22] A. Hagen, "Full Stack Ahead: Pioneering Quantum Hardware Allows for Controlling up to thousands of Qubits at Cryogenic Temperatures," *Microsoft Research*. Jan. 2021. Accessed: Apr. 02, 2023. [Online]. Available: https://www.microsoft.com/en-us/research/blog/full-stack-ahead-pioneering-quantum-hardware-allows-for-controlling-up-to-thousands-of-qubits-at-cryogenic-temperatures/

[23] M. Heinkenschloss, "LATEX Mathematical Symbols." Available: https://www.cmor-faculty.rice.edu/~heinken/latex/symbols.pdf

[24] F. S. Khan and N. Bao, "Quantum Prisoner's Dilemma and High Frequency Trading on the Quantum Cloud," *Frontiers in Artificial Intelligence*, vol. 4, Nov. 2021, doi: 10.3389/frai.2021.769392.

[25] F. X. Lin, "Shor's Algorithm and the Quantum Fourier Transform."

[26] J. Liu *et al.*, "Quantum Hoare Logic," *Archive of Formal Proofs*, Mar. 2019, Available: https://isa-afp.org/entries/QHLProver.html

[27] X. Lu and H. Lin, "Unbiased Quantum Phase Estimation." arXiv, Oct. 2022. Accessed: Apr. 01, 2023. [Online]. Available: http://arxiv.org/abs/2210.00231

[28] Marco Lanzagorta and Jeffrey K. Uhlmann, "Hybrid Quantum-Classical Computing with Applications to Computer Graphics," in *ACM SIGGRAPH 2005 Courses*, Jul. 2005, pp. 2–es. doi: 10.1145/1198555.1198723.

[29] N. Meyer, C. Ufrecht, M. Periyasamy, D. D. Scherer, A. Plinge, and C. Mutschler, "A Survey on Quantum Reinforcement Learning." arXiv, Nov. 2022. Accessed: Apr. 02, 2023. [Online]. Available: http://arxiv.org/abs/2211.03464

[30] M. Mottonen and J. J. Vartiainen, "Decompositions of General Quantum Gates."

[31] M. Muller, "Strongly Universal Quantum Turing Machines and Invariance of Kolmogorov Complexity," *IEEE Transactions on Information Theory*, vol. 54, no. 2, pp. 763–780, Feb. 2008, doi: 10.1109/TIT.2007.913263.

[32] "Hidden Subgroup Problem," *Wikipedia*. Mar. 2023. Accessed: Apr. 01, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Hidden_subgroup_problem&oldid=1143426160

[33] "IBM Quantum Composer," *IBM Quantum*. Accessed: Apr. 01, 2023. [Online]. Available: https://quantum-computing.ibm.com/composer/docs/iqx

[34] "Quantum Phase Estimation in Qiskit," *Quantum Computing UK*. Oct. 2020. Accessed: Apr. 01, 2023. [Online]. Available: https://quantumcomputinguk.org/tutorials/quantum-phase-estimation-with-code

[35] F. Phillipson, N. Neumann, and R. Wezeman, "Classification of Hybrid Quantum-Classical Computing."

[36] C. Shao, "Computing Eigenvalues of Diagonalizable Matrices in a Quantum Computer." arXiv, Sep. 2020. Accessed: Apr. 01, 2023. [Online]. Available: http://arxiv.org/abs/1912.08015

[37] N. Slate, E. Matwiejew, S. Marsh, and J. B. Wang, "Quantum Walk-Based Portfolio Optimisation," *Quantum*, vol. 5, p. 513, Jul. 2021, doi: 10.22331/q-2021-07-28-513.

[38] Weiyuan Gong, Si Jiang, and Dong-Ling Deng, "No-Go Theorem and a Universal Decomposition Strategy for Quantum Channel Compilation," *Physical Review Research*, vol. 5, no. 1, p. 013060, Jan. 2023, doi: 10.1103/PhysRevResearch.5.013060.

[39] XanaduAI, "Xanadu Quantum Codebook." Xanadu, Mar. 2023. Accessed: Apr. 01, 2023. [Online]. Available: https://github.com/XanaduAI/Xanadu-Quantum-Codebook