# COMP 458/558
# Introduction to Quantum Computing

Micah Kepe

# Contents

# Chapter 1

# Phase I: Introduction and Background

## 1.1 Lecture 1: Overview of Quantum Computing Concepts

**Definition 1.1.1: Quantum Computing**

Quantum computing is a computational paradigm leveraging quantum mechanical principles such as superposition, entanglement, and interference to perform computations that can surpass the capabilities of classical systems for specific tasks.[a]

[a] Superposition allows quantum bits (qubits) to exist in multiple states simultaneously, and entanglement enables correlations between qubits even at a distance.

### Historical Development of Quantum Computing

- **1980s-1990s:** Conception of quantum computing, with foundational ideas like the quantum Turing machine and quantum gates.
- **1990s-2000s:** Demonstration of key building blocks, such as quantum algorithms (e.g., Shor's and Grover's algorithms).
- **2016:** Emergence of quantum computing clouds, enabling access to quantum hardware via the internet.
- **2019:** First claims of **quantum advantage**, showcasing tasks where quantum computers outperform classical counterparts.
- **2024:** Increasing qubit counts and improvements in quantum error correction techniques.

### Applications of Quantum Computing

Quantum computing offers speedup in areas such as:

1. **Quantum Simulation:** Applications in chemistry, physics, and materials science, such as simulating molecular energy levels and drug discovery.
2. **Security and Encryption:** Developing quantum-safe cryptographic protocols and random number generation.
3. **Search and Optimization:** Enhancing solutions for weather forecasting, financial modeling, traffic planning, and resource allocation.

**Example 1.1.1** (Example: Quantum Speedup in Drug Discovery)
Drug discovery benefits from quantum simulation by enabling more accurate modeling of molecular interactions, which classical computers struggle to achieve efficiently.

### Classical vs. Quantum Computing Paradigms

- **Classical Computing:** Utilizes traditional processing units (CPU, GPU, FPGA) and executes deterministic computations.
- **Quantum Computing:** Employs quantum processing units (QPU) with probabilistic computation based on quantum states.

**Note:-**

Note: Classical computing paradigms still dominate in tasks that require precision and deterministic results. Quantum computing excels in probabilistic or exponentially large state-space problems.

## 1.2 Lecture 2: Review of Linear Algebra Concepts

**Definition 1.2.1: Vectors: Row and Column Vectors**

A **vector** is an ordered list of numbers, which can be represented as either a row or column vector. The components of vectors in quantum computing belong to the field of complex numbers ($\mathbb{C}$).

### Column Vectors

A column vector is a vertical arrangement of numbers:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}, \quad v_i \in \mathbb{C}.$$

### Row Vectors

A row vector is the complex conjugate transpose (adjoint) of a column vector:

$$\mathbf{v}^\dagger = \begin{bmatrix} \overline{v_1} & \overline{v_2} & \dots & \overline{v_n} \end{bmatrix}.$$

**Definition 1.2.2: Inner Product**

The **inner product** of two vectors $\mathbf{v}, \mathbf{w} \in \mathbb{C}^n$ is defined as:

$$\langle \mathbf{v}, \mathbf{w} \rangle = \mathbf{v}^\dagger \mathbf{w} = \sum_{i=1}^{n} \overline{v_i} w_i.$$

**Example 1.2.1** (Example: Inner Product)

Let $\mathbf{v} = \begin{bmatrix} 1+i \\ 2 \end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix} 3 \\ i \end{bmatrix}$. Then:

$$\langle \mathbf{v}, \mathbf{w} \rangle = (1-i)(3) + (2)(i) = 3 - 3i + 2i = 3 - i.$$

**Definition 1.2.3: Outer Product**

The **outer product** of two vectors $\mathbf{v} \in \mathbb{C}^m$ and $\mathbf{w} \in \mathbb{C}^n$ produces an $m \times n$ matrix:

$$\mathbf{v}\mathbf{w}^\dagger = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} \begin{bmatrix} \overline{w_1} & \overline{w_2} & \dots & \overline{w_n} \end{bmatrix}.$$

**Definition 1.2.4: Orthogonality**

Two vectors $\mathbf{v}, \mathbf{w} \in \mathbb{C}^n$ are **orthogonal** if their inner product is zero:

$$\langle \mathbf{v}, \mathbf{w} \rangle = 0.$$

**Example 1.2.2** (Example: Orthogonality)

Let $\mathbf{v} = \begin{bmatrix} 1 \\ i \end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix} i \\ 1 \end{bmatrix}$. Then:

$$\langle \mathbf{v}, \mathbf{w} \rangle = (1)(i) + (i)(1) = i - i = 0.$$

**Definition 1.2.5: Eigenvalues and Eigenvectors**

For a square matrix $A \in \mathbb{C}^{n \times n}$, a vector $\mathbf{v} \neq \mathbf{0}$ is an **eigenvector** if:

$$A\mathbf{v} = \lambda \mathbf{v},$$

where $\lambda \in \mathbb{C}$ is the **eigenvalue**.

## 1.3 Lecture 3: Quantum Bits and States

**Definition 1.3.1: Qubit**

A **qubit** is the fundamental unit of quantum information. Unlike a classical bit, which is either 0 or 1, a qubit can exist in a superposition of states:

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

**Definition 1.3.2: Dirac Notation**

Quantum states are commonly represented in **Dirac notation** (bra-ket notation). For example:

- $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ form the computational basis.
- Any qubit state can be expressed as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

**Definition 1.3.3: Bloch Sphere Representation**

The **Bloch sphere** visualizes a qubit state $|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\phi}\sin(\theta/2)|1\rangle$ as a point on a unit sphere, where:

- $\theta$: Polar angle, $[0, \pi]$.
- $\phi$: Azimuthal angle, $[0, 2\pi)$.

**Note:-**

Note: Measurement collapses a qubit into one of the basis states with probabilities proportional to the square of the amplitudes.

**Example 1.3.1** (Example: Measurement Probabilities)

For $|\psi\rangle = \frac{1}{\sqrt{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle$:

- Probability of $|0\rangle$: $P(0) = \left|\frac{1}{\sqrt{3}}\right|^2 = \frac{1}{3}$.
- Probability of $|1\rangle$: $P(1) = \left|\sqrt{\frac{2}{3}}\right|^2 = \frac{2}{3}$.

# Chapter 2

# Phase II: Fundamentals of Quantum Algorithms

## 2.1 Lecture 4: Quantum State Representation and Qubits

**Definition 2.1.1: Quantum States**

Define quantum states.

**Definition 2.1.2: Qubits**

Define qubits and their representation.

## 2.2 Lecture 5: Quantum Gates and Transformations

**Definition 2.2.1: Unitary Matrices**

Discuss unitary matrices.[a]

[a] A **unitary matrix** $U$ satisfies $U^\dagger U = I$, where $U^\dagger$ is the conjugate transpose of $U$.

**Example 2.2.1** (Hadamard Gate)
Provide an example and matrix representation of the Hadamard gate.

## 2.3 Lecture 6: Quantum Circuits and Entanglement

**Definition 2.3.1: Entanglement**

Define entanglement.

**Example 2.3.1** (Bell State)
Example of a Bell state and its significance.

# Chapter 3

# Phase III: Advanced Quantum Algorithms

## 3.1 Lecture 7: Grover's Search Algorithm

**Definition 3.1.1: Grover's Algorithm**

Overview of Grover's search algorithm.

**Question 1**

Provide a pseudocode or example implementation in Python.

## 3.2 Lecture 8: Variational Quantum Algorithms

**Definition 3.2.1: VQE**

Introduce the Variational Quantum Eigensolver (VQE).

# Chapter 4

# Phase IV: Special Topics in Quantum Computing

## 4.1 Lecture 9: Quantum Annealing

**Definition 4.1.1: Quantum Annealing**

Discuss the basics of quantum annealing.

## 4.2 Lecture 10: Quantum Machine Learning

**Definition 4.2.1: QML**

Define quantum machine learning and its applications.

# Chapter 5

# Phase V: Concluding Lectures

## 5.1 Lecture 11: Project Presentations and Feedback

**Note:-**

Placeholder for project presentation notes.