Bachelor Project

Czech
Technical
University
in Prague

F4

Faculty of Nuclear Sciences and Physical Engineering
Department of Physics

Programming and simulation with quantum computers

Jiří Viták

Supervisor: Iskender Yalcinkaya, Ph.D.
Consultant: Aurél Gábor Gábris, Ph.D.
August 2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Viták** Jméno: **Jiří** Osobní číslo: **509296**

Fakulta/ústav: **Fakulta jaderná a fyzikálně inženýrská**

Zadávající katedra/ústav: **Katedra fyziky**

Studijní program: **Kvantové technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Programování a simulace na kvantových počítačích

Název bakalářské práce anglicky:

Programming and simulation with quantum computers

Pokyny pro vypracování:

1. Studium hradlového modelu kvantového počítání v rámci kvantového prostředí "Qiskit": Qubit a jeho Blochova reprezentace, jednoqubitová hradla, rotace na Blochově sféře, řízená hradla a generování provázanosti, fázová zpětná vazba, Bellova báze, univerzální kvantová hradla.
2. Studium fungování základních kvantových algoritmů: Deutch-Jozsa algoritmus, Groverův algoritmus, univerzální kvantová simulace.
3. Identifikace fyzikálního systému vhodného pro efektivní simulaci na kvantovém počítači (např. časový vývoj spinových řetízků, kvantové procházky, problém částice v krabici, nejnižší energetická hladina atomů/molekul).
4. Analýza účinnosti a stability implementované simulace.

Seznam doporučené literatury:

[1] Benenti, G., Casati, G., Rossini, D., Strini, G., “Principles of Quantum Computation and Information: A Comprehensive Textbook”, World Scientific (2019).
[2] Montanaro, A. Quantum algorithms: an overview, npj Quantum Inf 2, 15023 (2016).
[3] X. Yuan, S. Endo, Q. Zhao, and Y. Li, S. C. Benjamin, Theory of variational quantum simulation, Quantum 3, 191 (2019).
[4] Cerezo, M., Arrasmith, A., Babbush, R. et al., Variational quantum algorithms, Nat. Rev. Phys. 3, 625–644 (2021).

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Iskender Yalcinkaya, Ph.D. katedra fyziky FJFI

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Aurél Gábor Gábris, Ph.D. katedra fyziky FJFI

Datum zadání bakalářské práce: **31.10.2023** Termín odevzdání bakalářské práce: **05.08.2024**

Platnost zadání bakalářské práce: **31.10.2025**

Iskender Yalcinkaya, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

doc. Ing. Václav Čuba, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

10. 11. 2023
Datum převzetí zadání

Podpis studenta

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta jaderná a fyzikálně inženýrská

Břehová 7
115 19 Praha 1

PROHLÁŠENÍ

Já, níže podepsaný

Jméno a příjmení studenta: Jiří Viták

Osobní číslo: 509296

Studijní program: Kvantové technologie

Specializace:

prohlašuji, že jsem bakalářskou práci s názvem:

Programování a simulace na kvantových počítačích

vypracoval samostatně a uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Praze dne 5. 8. 2024 ....

podpis

Acknowledgements

First of all, I would like to thank Dr. Iskender Yalcinkaya for the professional help and time he gave me during the writing of this thesis. Furthermore, I would like to thank Dr. Aurél Gábris for consulting on topics and ideas for this thesis.

Declaration

I declare that the presented work is solely mine and that I cited all the literature used.

In Prague, 5. August 2024

Abstract

Although quantum computers that can outperform conventional computers are still in development, it is crucial to advance our understanding and development of quantum algorithms to prepare for the future of quantum computing. This bachelor project explores the growing field of quantum programming and simulation on currently available quantum computers. We study the basics of quantum computing and summarize the commonly used principles used in the field. We theoretically describe quantum walks with a special emphasis on discrete-time variants, which constitute the core part of this thesis. Furthermore, we introduce an optimized method for the efficient implementation of discrete-time quantum walks suitable for the current quantum computer architecture. We show that this method is more successful in examining quantum walk related aspects like spreading dynamics and perfect state transfer compared to conventional methods.

Keywords: quantum algorithm, quantum simulation, quantum walk

Supervisor: Iskender Yalcinkaya, Ph.D.
Katedra fyziky,
Fakulta jaderná a fyzikálně inženýrská
ČVUT v Praze

Abstrakt

Přestože kvantové počítače, které mohou překonat konvenční počítače, jsou stále ve vývoji, je pro přípravu na budoucnost kvantové výpočetní techniky zásadní, abychom pokročili v porozumění a vývoji kvantových algoritmů. Tento bakalářský projekt zkoumá rozvíjející se oblast kvantového programování a simulace na aktuálně dostupných kvantových počítačích. Studujeme základy kvantového počítání a shrnujeme běžně používané principy používané v této oblasti. Teoreticky popisujeme kvantové procházky s klíčovým důrazem na varianty s diskrétním časem, které tvoří stěžejní část této práce. Dále představujeme optimalizovanou metodu pro efektivní implementaci kvantových procházek v diskrétním čase vhodnou pro současnou architekturu kvantových počítačů. Ukazujeme, že tato metoda je ve srovnání s běžnými metodami úspěšnější při zkoumání aspektů souvisejících s kvantovými procházkami, jako je dynamika šíření a dokonalý přenos stavu.

Klíčová slova: kvantový algoritmus, kvantová simulace, kvantová procházka

Překlad názvu: Programování a simulace na kvantových počítačích

Contents

Figures

Tables

Chapter 1

Introduction

The concept of quantum computing was first proposed by Richard Feynman in 1981 when he suggested that extraordinary properties of quantum computers might be exponentially more efficient for simulating natural phenomena than classical computers. Since then, the journey from Feynman's initial vision to the development of practical quantum computers has been marked by several key milestones:

1. Deutsch's quantum Turing machine (1985): David Deutsch described the concept of a universal quantum computer that can perfectly simulate any Turing machine. Furthermore, Deutsch demonstrated that certain physical systems can be simulated with a universal quantum computer but are beyond the capabilities of a universal Turing machine [1].

2. Deutsch-Jozsa algorithm (1992): David Deutsch and Richard Jozsa introduced an algorithm that can determine if a function is constant or balanced with one query, demonstrating an exponential speed up over classical algorithms [2].

3. Shor's algorithm (1994): Peter Shor introduced a quantum algorithm for integer factorization, providing an exponential speedup over classical algorithms. This breakthrough has significant implications for cryptography, as it can potentially break widely used encryption systems like RSA[1] [3].

4. Grover's algorithm (1996): Lov Grover introduced a quantum algorithm for searching unsorted databases quadratically faster than any classical algorithm, showcasing a practical application of quantum computing [4].

The ongoing development of practical quantum algorithms continues to advance, with the aim of harnessing the unique capabilities of quantum computers. In recent years, the idea of employing quantum walks has been explored intensively in various fields. Quantum walks, analogous to classical random walks, have shown potential in developing new quantum algorithms

[1] RSA is a widely used encryption algorithm that relies on the difficulty of factoring large integers, making Shor's algorithm a potential threat to its security.

that may outperform their classical counterparts in solving problems such as database searching.

This thesis consists of both theoretical and implementation parts. It is organized as follows: Chapter 2 provides an overview of the notation and important concepts in quantum computing, such as quantum state tomography and universal quantum computing. This is followed by a discussion of the theoretical background of two historically significant quantum algorithms: the Deutsch-Jozsa algorithm and Grover's algorithm. In Chapter 3, we describe quantum walks and related topics such as spreading dynamics, state transfer, and disorder. Chapter 4 covers the simulation of these algorithms and the properties of quantum walks on IBM's quantum computers.

In conclusion, by combining a comprehensive theoretical foundation with practical implementation on contemporary quantum hardware, this project aims to demonstrate the current capabilities and future potential of quantum computing. Through the exploration of quantum walks and their applications, we hope to contribute to the ongoing development of efficient quantum algorithms and inspire further advancements in the field.

Chapter 2

Basics of quantum computing

In this chapter, we introduce fundamental concepts and important algorithms in quantum computing. Studying the necessary formalism and tools described here will provide the foundation needed to understand the quantum algorithms and more complex properties discussed in the following chapters.

2.1 Isolated quantum systems

An isolated quantum system is a quantum mechanical system that does not interact with its environment. This means that no information, energy, or particles are exchanged between the system and its surroundings. Such systems are described by the following features.

State

A quantum system is described by a *quantum state* that encapsulates all the information necessary to describe the properties and behavior of the system. In case of an isolated system, mathematically, a state (or a *pure state*) is a unit vector living in a d-dimensional Hilbert space $\mathcal{H}^d$ defined in complex numbers $\mathbb{C}$. The bra-ket notation is very useful for expressing the system's state, as it provides an easier and clearer way to write expressions in quantum mechanics. This notation was created by P. A. M. Dirac in the publication *A New Notation for Quantum Mechanics* [5], where a "ket", denoted by $|\psi\rangle$, represents the state of the quantum system with the following properties:

- For two different states $|\psi\rangle$ and $|\psi'\rangle$, the *linear combination* of these states $a\,|\psi\rangle + b\,|\psi'\rangle$ is also a state, where $a, b \in \mathbb{C}$.

- The *inner product*, $\langle\psi|\psi'\rangle \to \mathbb{C}$, is defined so that it matches every pair of quantum states with a complex number. Every ket in a given vector space coexists with a "bra", denoted by $\langle\psi|$, which is an element of the vector space $\mathcal{H}^*$ (complex conjugate of the space $\mathcal{H}$) dual to $\mathcal{H}$. If two states are *orthogonal*, then their inner product returns zero.

- The *norm* of the quantum state is defined as $|\psi| = \langle\psi|\psi\rangle^{1/2}$. Then, a normalized state has a unit norm, $|\psi| = 1$. Therefore, an overall phase (or it is called a *global phase*), $|e^{i\theta}\left|\psi\right\rangle| = |\left|\psi\right\rangle|$ has no physical significance since a quantum state is defined as a state with unit norm.

- The *dimension* d is the maximum number of linearly independent vectors that can be defined in the vector space $\mathcal{H}^d$. Therefore, each state $\left|\psi\right\rangle$ (and its dual) can be written as a linear combination of these d linearly independent vectors as $\left|\psi\right\rangle = \sum_{i=0}^{d-1} = c_i \left|i\right\rangle$ where $c_i \in \mathbb{C}$ are the components. Vectors $\left|i\right\rangle$ are called a basis for the Hilbert space $\mathcal{H}^d$. All algebraic operations, such as addition, scalar multiplication, inner product, etc., defined for states in $\mathcal{H}^d$ can be done in terms of the vector components. Physically, the dimension indicates the number of possible states that a quantum system can occupy.

- According to the Gram-Schmidt theorem, an *orthonormal basis* can be created out of a linearly independent basis. By convention, an orthonormal basis, $\langle i|j\rangle = \delta_{ij}$ is chosen to describe the vectors. We will call this basis a natural basis or *computational basis* throughout the thesis. Any state can be expressed by column and row vectors in the computational basis $\left|i\right\rangle \leftrightarrow (0, \ldots, 0, 1, 0, \ldots, 0)^T$ as [1]

$$\left|\psi\right\rangle \leftrightarrow (c_1, \ldots, c_d)^T, \qquad \left\langle\psi\right| \leftrightarrow (c_1^*, \ldots, c_d^*), \tag{2.1}$$

$$\left|\psi\right\rangle \leftrightarrow \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = c_1 \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + c_2 \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} + c_n \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}. \tag{2.2}$$

Given a basis, all vector/matrix operations are valid among row and column vectors. For example, we can express an inner product in terms of the components of the vectors as

$$\langle v|w\rangle = (\left\langle v\right|)(\left|w\right\rangle) \leftrightarrow \begin{pmatrix} v_1^* & v_2^* & \cdots & v_d* \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix} = v_1^* w_1 + \ldots + v_d^* w_d. \tag{2.3}$$

Observable

In general, an *operator* Y is a linear map from vectors to vectors, expressed as $\left|\psi\right\rangle \to Y\left|\psi\right\rangle$. From this perspective, an *observable* is a physical quantity that can be measured during an experiment and is represented by Hermitian (or self-adjoint) operators to ensure that their eigenvalues, which correspond to possible measurement outcomes, are real numbers. In a given basis in $\mathcal{H}^d$, an observable A can be represented by a $d \times d$ matrix, where the elements

[1] The transpose operation, T, flips a matrix over its diagonal. In case of vectors, it replaces a row vector with a column vector, or vice versa.

satisfy the property $\langle i|A|j\rangle = \langle j|A|i\rangle^*$ due to hermiticity. Additionally, the eigenvectors of an observable are mutually orthogonal, ensuring that any state in $\mathcal{H}^d$ can be written as a linear combination of the d eigenvectors of the observable.

Dynamics

When we are given a quantum state $|\psi(t)\rangle$ at time t, the *time evolution* of this state is described by a *unitary operator* $U(t', t)$, the final state at time t' is expressed as

$$|\psi(t')\rangle = U(t', t)\,|\psi(t)\rangle \tag{2.4}$$

where the unitary operator $U(t', t)$ satisfies the property $U^\dagger = U^{-1} = I$, which means the transpose conjugate of U equals its inverse. Here, I is the identity operator. The Schodinger equation describes how a given state $|\psi\rangle$ evolves in time as

$$i\hbar\frac{d}{dt}\,|\psi(t)\rangle = H\,|\psi(t)\rangle \tag{2.5}$$

where H is called the Hamiltonian of the system with the solution for a time-independent H

$$U(t', t) = \exp\left[-\frac{i}{\hbar}H(t' - t)\right]. \tag{2.6}$$

Measurement

When a *measurement* is performed on a quantum system, or in other words, when an observable is measured, the state of the system is transformed into one of the eigenvectors of the observable. From a mathematical point of view, before we make the measurement, we think of the state of the quantum system as being expressed in terms of the eigenvectors $|a_i\rangle$ of the observable A we want to measure, as $|\psi\rangle = \sum_i c_i\,|a_i\rangle$. Therefore, the measurement projects the quantum state onto one of the directions defined by the eigenvectors of the observable A. The result we obtain after this measurement is probabilistic due to the nature of quantum systems, such that the measurement transforms the quantum state $|\psi\rangle$ into a particular eigenvector $|a_i\rangle$ with probability $P(a_i) = |\,\langle a_i|\psi\rangle\,|^2$. Therefore, a measurement can be associated with a *projection operator* $E_i = |a_i\rangle\,\langle a_i|$, which is a matrix in a given basis. Then, the probability of obtaining the state $|a_i\rangle$ can be expressed as $P(a_i) = \langle\psi|E_i|\psi\rangle$.

Composite systems

A *composite system* refers to a system that consists of multiple subsystems, each of which can be described by its own quantum state. The general state of the composite system is represented by the *tensor product* of the states of its individual subsystems $|\psi_1\rangle$, $|\psi_2\rangle$, $\cdots$, $|\psi_n\rangle$ as

$$|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle\,. \tag{2.7}$$

This representation highlights the composite nature of the quantum register and its corresponding Hilbert space. In the following text we will use conventional notation to write tensor product of kets as follows

$$|i\rangle \otimes |j\rangle \otimes \ldots \otimes |k\rangle \equiv |ij \ldots k\rangle \equiv |i\rangle |j\rangle \ldots |k\rangle \,. \tag{2.8}$$

2.2 Density matrix

A *density matrix* generalizes the concept of a quantum state to include statistical mixtures of pure states. It is particularly useful for describing the behavior of a smaller part of a larger system (usually called an *open system*), where the smaller part exchanges information with the rest of the system, conventionally called the *environment*. Therefore, the rules we defined to describe isolated systems in Sec. 2.1, such as states being unit vectors and time evolution being unitary, are no longer valid. The density matrix $\hat{\rho}$ is defined as an operator that is Hermitian $\left(\hat{\rho} = \hat{\rho}^\dagger\right)$, positive $(\hat{\rho} \geq 0)$, and has a trace equal to one $(\mathrm{Tr}\,(\hat{\rho}) = 1)$. Every density matrix can be written in the form

$$\hat{\rho} = \sum_i p_i \, |\psi_i\rangle \langle\psi_i| \,, \tag{2.9}$$

where p_i represents the statistical probability that the quantum system is in the pure state $|\psi_i\rangle$. Therefore, p_i satisfy the normalization condition $(\sum_i p_i = 1)$. The density matrix is a generalization of pure states such that it involves the cases when we have an statistical ensemble of pure states. The mixture of pure states expressed by a density operator is called a *mixed state*. Hence, a pure state $|\psi\rangle$ can be written in density matrix form as the outer product

$$\hat{\rho} = |\psi\rangle \langle\psi| \,, \tag{2.10}$$

and its time evolution is expressed as $\hat{\rho}(t+1) = U^\dagger \hat{\rho}(t) U$. Although the density matrix formalism is essentially employed to describe the time evolution of open quantum systems, via *superoperators*, we will omit it since that is out of the scope of this thesis.

2.3 Qubit and Bloch sphere

A *qubit*, short for quantum bit, serves as the basic unit of information in quantum computing. In contrast to classical bits, which are confined to exist in either state 0 or 1, a quantum bit can exist in a linear combination, commonly referred to as a *superposition*, of both states. The state of a qubit can, therefore, be expressed as

$$|\psi\rangle = \alpha \, |0\rangle + \beta \, |1\rangle \,, \tag{2.11}$$

where $|0\rangle$ and $|1\rangle$ form an orthonormal basis in a Hilbert space $\mathcal{H}^2$, i.e., $\langle i|j\rangle = \delta_{ij}$ with $i, j = 0, 1$. The complex numbers α and β represent *probability* *amplitudes*. As we explained in Sec. 2.1, when a measurement is made on the quantum system expressed by the qubit state, the probabilities of finding the system in state 0 or 1 are given by Born's rule as $|c|^2 = |\langle c|\psi\rangle|^2$, $c = \alpha, \beta$, respectively. This rule implies condition for the probability amplitudes according to the *normalization of probabilities*, expressed as $|\alpha|^2 + |\beta|^2 = 1$.

A collection of n qubits used in quantum computations is referred to as a *quantum register* of size n existing in a Hilbert space $\mathcal{H}$ as follows

$$\mathcal{H} = \mathcal{H}_{n-1} \otimes \mathcal{H}_{n-2} \otimes \cdots \otimes \mathcal{H}_0. \tag{2.12}$$

The state of n qubits $|\Psi\rangle$ is then described through the tensor product of their individual quantum states $|\psi_1\rangle$, $|\psi_2\rangle$, $\cdots$, $|\psi_n\rangle$ as

$$|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle\,. \tag{2.13}$$

The *Bloch sphere* (Figure 2.1) is a geometric representation used in quantum computing to describe the state of a qubit. It is depicted as a unit sphere, with all possible qubit states represented by points on its surface. Hence, we need two spherical parameters to uniquely specify a given qubit state. However, given a qubit state as described in Eq. (2.11), the complex coefficients α and β can be rewritten as:

$$\begin{aligned} \alpha &= x_1 + iy_1, \\ \beta &= x_2 + iy_2 \end{aligned} \tag{2.14}$$

where $x_1, x_2, y_1, y_2 \in \mathbb{R}$. Contrary to what we claimed earlier, there appear to be four independent parameters, but we will show in a moment that, in fact, two of them can be eliminated.

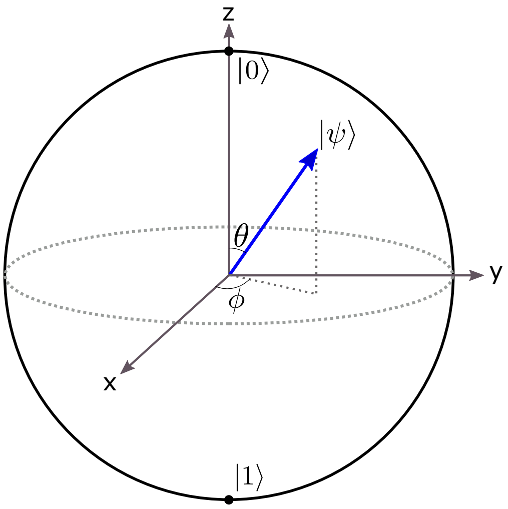


Figure 2.1: The Bloch sphere. All qubit states can be represented by the points on this unit sphere. In here, only the computational basis ($|0\rangle$, $|1\rangle$) and a random qubit state $|\psi\rangle$, which is uniquely defined by the spherical coordinates (ϕ, θ), are shown.

Let us rewrite the parameters α and β by using polar coordinates

$$\begin{aligned} \alpha &= r_1 \left(\cos\Theta_1 + i\sin\Theta_1\right) = r_1 e^{\Theta_1}, \\ \beta &= r_2 \left(\cos\Theta_2 + i\sin\Theta_2\right) = r_2 e^{\Theta_2}, \end{aligned} \tag{2.15}$$

where r_1, $r_2 \in [0, 1]$ and Θ_1, $\Theta_2 \in [0, 2\pi)$. If we rewrite Eq. (2.11) by using this transformation

$$|\psi\rangle = e^{i\Theta_1}\{r_1 |0\rangle + r_2 e^{i(\Theta_2 - \Theta_1)} |1\rangle\}, \tag{2.16}$$

the term $e^{i\Theta_1}$ can be factored out. This term is called a *global phase* in quantum mechanics and does not have any observable consequences. This is because physical measurements in quantum mechanics are based on the inner products of state vectors, and these inner products are unaffected by a global phase. Therefore, one of the four independent variables can be eliminated and the quantum state can be equivalently written as

$$|\psi\rangle = r_1 |0\rangle + r_2 e^{i\Theta} |1\rangle\,, \tag{2.17}$$

where we replaced $\Theta_2 - \Theta_1$ with Θ since both Θ_1 and Θ_2 are independent variables. Furthermore, when we switch back to Cartesian coordinates,

$$|\psi\rangle = z |0\rangle + (x + iy) |1\rangle\,, \tag{2.18}$$

and consider the normalization condition $x^2 + y^2 + z^2 = 1$, we see that these three parameters are not independent from each other and define a unit sphere. This condition reduces the dependence to two real parameters. Finally, we switch to the spherical coordinates $x = \sin\theta\cos\phi$, $y = \sin\theta\sin\phi$, $z = \cos\theta$,

$$|\psi\rangle = \cos\theta |0\rangle + e^{i\varphi}\sin\theta |1\rangle\,, \tag{2.19}$$

where $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi)$. Since the states in the lower half differ from those in the upper half only by a global phase factor of -1, we use the following transformation to neglect the states in the lower half and extend the upper half to complete the unit sphere:

$$|\psi\rangle = \cos\frac{\theta}{2} |0\rangle + e^{i\phi}\sin\frac{\theta}{2} |1\rangle\,. \tag{2.20}$$

Therefore, Eq. (2.20) is the Bloch representation of a qubit state with two independent variables, θ and ϕ.

2.4 Single-qubit gates

Quantum gates serve as fundamental building blocks of quantum circuits and facilitate the manipulation of sets of qubits. These gates are essentially unitary operators represented by $2^n \times 2^n$ unitary matrices, with n denoting the number of qubits on which the quantum gates operate.

In this section, for clarity, we will confine our focus to single-qubit gates such as the Identity gate, X, Y, Z gates, the Hadamard gate, and phase-shift gates.

2.4.1 Identity gate

The identity gate is a single qubit gate and is typically denoted by I. In the matrix form it is written as

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2.21}$$

When applied to a qubit, the identity gate has no effect on its state. However, it holds significance in quantum computing as a reference point for quantum operations.

2.4.2 X, Y, Z gates

The X, Y, and Z gates (also known as Pauli matrices, σ_x, σ_y, and σ_z) operate on single qubits and are essential in quantum computing. In the computational basis $|0\rangle$, $|1\rangle$, they are expressed as

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \; Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \; Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.22}$$

All three Pauli gates, when applied to a qubit, function as rotations around their respective axes of the Bloch sphere by π radians. They are also involutory, meaning that each Pauli gate is its own inverse:

$$I^2 = X^2 = Y^2 = Z^2. \tag{2.23}$$

The X gate is sometimes referred to as a bit-flip since it maps $|0\rangle$ to $|1\rangle$ and $|1\rangle$ to $|0\rangle$. Although it exhibits similar behaviour to the NOT gate in the classical computing, considering them identical would be incorrect, as there does not exist universal quantum NOT gate.

The Y gate changes the basis state and introduces phase to the quantum state, expressed as $Y|0\rangle = i|1\rangle$ or $Y|1\rangle = -i|0\rangle$.

Lastly, the Z gate is often called phase-flip gate. Unlike the other two gates, it does not alter the basis state but changes phase of the state as $Z|0\rangle = i|0\rangle$ and $Z|1\rangle = -i|1\rangle$.

2.4.3 Hadamard gate

The Hadamard gate could be seen as a rotation around the axis $\frac{1}{\sqrt{2}}(\hat{x} + \hat{y})$ for the angle of π. Mathematically, the Hadamard matrix is involutory matrix given by

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{2.24}$$

The Hadamard gate transforms the computational basis into a superposition of both states with equal probabilities. The transformation is represented as follows

$$
\begin{aligned}
H\left|0\right\rangle &= \frac{1}{\sqrt{2}}\left(\left|0\right\rangle + \left|1\right\rangle\right) \equiv \left|+\right\rangle, \\
H\left|1\right\rangle &= \frac{1}{\sqrt{2}}\left(\left|0\right\rangle - \left|1\right\rangle\right) \equiv \left|-\right\rangle.
\end{aligned}
\tag{2.25}
$$

2.4.4 Phase shift gates

The shift gate introduces the phase $e^{i\varphi}$ to the state $\left|1\right\rangle$, $P\left|1\right\rangle = e^{i\varphi}\left|1\right\rangle)$, and does not change the state $\left|0\right\rangle$. The matrix representing the phase shift gate is given by

$$
P(\varphi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix}.
\tag{2.26}
$$

where $\varphi \in [0, 2\pi)$. Notable examples of shift gates include the T gate for $\varphi = \frac{\pi}{8}$, the Swap gate (S) for $\varphi = \frac{\pi}{2}$ and the Z gate for $\varphi = \pi$.

2.5 Rotations on the Bloch sphere

As mentioned in Sec. 2.3, every qubit state can be represented by a point on the Bloch sphere. The time evolution of the qubit state under the influence of quantum gates or other quantum operations can be represented by rotations on the Bloch sphere so that a quantum operation transforms a point on the Bloch sphere into another point. These rotations are described by unitary operators, and the angles and axes of rotation determine the specific transformation applied to the quantum state.

2.5.1 Rotations about x, y, z axes

Firstly, we will discuss rotations on the Bloch sphere about the Cartesian axes x, y, and z. The matrix exponential can be expanded in Taylor series for a given general involutory matrix A and parameter θ representing the rotation angle as follows:

$$
\begin{aligned}
e^{i\theta A} &= I + i\theta A - \frac{\theta^2 I}{2!} - i\frac{\theta^3 A}{3!} + \frac{\theta^4 I}{4!} + i\frac{\theta^5 A}{5!} + \cdots \\
&= \left(1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} + \cdots\right) I + i\left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} + \cdots\right) A \\
&= I\cos\theta + iA\sin\theta.
\end{aligned}
\tag{2.27}
$$

Rotations around the basis axes are then derived by substituting Pauli matrices for the operator A. The result is three quantum single-qubit gates, each describing a rotation around its respective basis axis, represented by a matrix

$$R_x(\theta) = e^{-\frac{i\theta X}{2}} = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}, \tag{2.28}$$

$$R_y(\theta) = e^{-\frac{i\theta Y}{2}} = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}, \tag{2.29}$$

$$R_z(\theta) = e^{-\frac{i\theta Z}{2}} = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}, \tag{2.30}$$

where X, Y, Z are Pauli matrices.

2.5.2 Rotations about an arbitrary axis

Considering a real unit vector $\hat{n} = n_x\hat{x} + n_y\hat{y} + n_z\hat{z}$, where $\hat{x}$, $\hat{y}$ and $\hat{z}$ are unit vectors, the rotation about an arbitrary axis defined by vector $\hat{n}$ by an angle of θ is given by

$$\begin{aligned} R_{\hat{n}}(\theta) &= e^{-i\frac{\theta}{2}\hat{n}\cdot\vec{\sigma}} \\ &= I\cos\frac{\theta}{2} - i\hat{n}\cdot\vec{\sigma}\sin\frac{\theta}{2}. \end{aligned} \tag{2.31}$$

The mathematical object $\vec{\sigma} = \sigma_x\hat{x} + \sigma_y\hat{y} + \sigma_z\hat{z}$ is called the Pauli vector, where σ_x, σ_y and σ_z are Pauli matrices as defined in Sec. 2.4.2.

2.5.3 Z-Y decomposition for a single qubit

The significance of rotations around arbitrary and basis axes is evident in a phenomenon known as the Z-Y decomposition. This principle asserts that for any unitary operator U acting on a single qubit, there exist four real parameters α, β, γ and δ so the equation

$$U = e^{i\alpha}R_z(\beta)R_y(\gamma)R_z(\delta) \tag{2.32}$$

is satisfied. Therefore, any unitary operation can be decomposed into three sequential rotations about the z and y axes up to a phase constant. In matrix form, Eq. 2.32 can be represented as

$$U = \begin{pmatrix} e^{i\left(\alpha-\frac{\beta}{2}-\frac{\delta}{2}\right)}\cos\frac{\gamma}{2} & -e^{i\left(\alpha-\frac{\beta}{2}+\frac{\delta}{2}\right)}\sin\frac{\gamma}{2} \\ e^{i\left(\alpha+\frac{\beta}{2}-\frac{\delta}{2}\right)}\sin\frac{\gamma}{2} & e^{i\left(\alpha+\frac{\beta}{2}+\frac{\delta}{2}\right)}\cos\frac{\gamma}{2} \end{pmatrix}. \tag{2.33}$$

Since U is unitary, the rows and columns of U are orthonormal.

2.6 Controlled gates

Controlled gates are quantum gates that act on two or more qubits, where the action taken on a target qubit depends on the state of one or more control qubits. Let us consider a situation with two qubits; the possibilities with more qubits would be analogous. The control gate $C(U)$, when applied to a system of two qubits, acts as a unitary operator U on the second qubit only if the first qubit state is $|1\rangle$. Mathematically, it is expressed by

$$C(U) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{00} & U_{01} \\ 0 & 0 & U_{10} & U_{11} \end{pmatrix}, \tag{2.34}$$

where U_{00}, U_{01}, U_{10} and U_{11} are matrix elements of the operator U.

A typical example of a controlled gate is the controlled-NOT or CNOT gate, which has as an input of two qubits. It works as an identity gate on the target qubit, if the control qubit state is $|0\rangle$ and as an NOT gate on the target qubit, if the control qubit state is $|1\rangle$. The matrix representation of CNOT gate is

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2.35}$$

2.7 Quantum circuits

A quantum circuit is a model for quantum computation in which a computation is a sequence of quantum gates, each of which performs a specific operation on a set of qubits. Quantum circuits are analogous to classical logic circuits, but exploit the principles of quantum mechanics, such as superposition and entanglement, to process information.

The quantum computer can be thought of as a finite collection of n qubits, a *quantum register* of size n as seen in Fig. 2.2(a). Single- and multi-qubit elementary operations, such as H, and $CNOT$, can be performed on single qubits or pairs of qubits, and these operations may be combined in an ordered way to produce any given complex quantum operation. The representation of single- and multi-qubit gates is given in Figs. 2.2(b) and (c). The state of an n-qubit quantum computer then can be written as

$$|\psi\rangle = \sum_{i=0}^{2^n-1} c_i |i\rangle = \sum_{i_{n-1}}^{1} \ldots \sum_{i_0}^{1} c_{i_{n-1},\ldots,c_0} |i_{n-1}\rangle \ldots |i_0\rangle \tag{2.36}$$

where $i_0, i_1, \ldots, i_{n-1} \in \{0, 1\}$ are binary digits and $i = i_{n-1}2^{n-1} + \ldots + i_1 2 + i_0$ is the decimal representation of the binary number $i_{n-1} \ldots i_1 i_0$.

To perform a quantum computation, one should be able to [x]:

- *Preparation*: The quantum computer is initialized in a well-defined initial state; usually it is $|\psi_i\rangle = |0 \ldots 00\rangle$.
- *Manipulation*: The initial state is manipulated by a unitary operation U, which can be decomposed into a sequence of single and multi-qubit gates.

- *Measurement*: After the manipulation stage, all the qubits are measured separately on a given basis and a single probabilistic result is obtained. For example, if the quantum register consists of two qubits and we measure on the computational basis $\{|0\rangle, |1\rangle\}$ only one of the four possible results $\{00, 01, 10, 11\}$ is obtained after measurement.

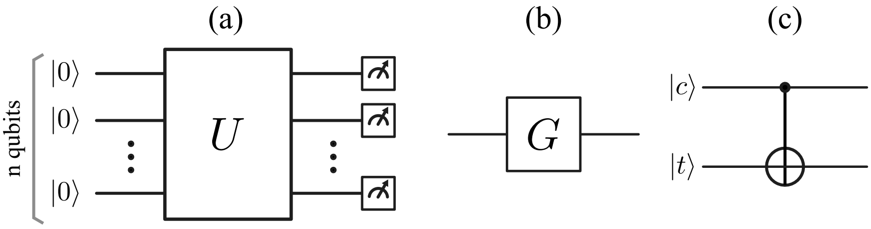


Figure 2.2: (a) General scheme of a quantum circuit where preparation, manipulation and measurement stages are represented. (b) Representation of single-qubit gates; G can be any single qubit gate such as H and I as defined in previous sections. (c) Representation of a $CNOT$ gate where $|c\rangle$ and $|t\rangle$ are the control and target qubits, respectively.

The above three steps are repeated for a predetermined number of iterations, and a measurement statistic is obtained for 2^n possible outcomes. This statistic is the result produced by the quantum circuit and is evaluated according to the desired purpose.

2.8 The Bell basis

A Bell basis is an orthonormal basis in the four-dimensional Hilbert space for two qubits. It consists of four specific two-qubit states, often referred to as Bell states. The Bell states

$$\begin{aligned}
|\Phi^+\rangle &= \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right), \\
|\Phi^-\rangle &= \frac{1}{\sqrt{2}}\left(|00\rangle - |11\rangle\right), \\
|\Psi^+\rangle &= \frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right), \\
|\Psi^-\rangle &= \frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right),
\end{aligned} \tag{2.37}$$

are *entangled quantum states*. If we measure either of the qubits we would obtain the state $|0\rangle$ with probability $P(0) = 1/2$ or the state $|1\rangle$ with the same probability. Although these measurement results seem independent, since the quantum state of each particle cannot be described independently of the state of the other, there is a correlation between them. Measurements to one particle instantaneously affect the other, even if separated by large distances, which is a phenomenon that A. Einstein famously called “spooky action at a distance" [6]. Consider $|\Phi^+\rangle$. If a measurement returns the result $|0\rangle$, then the second qubit can only be found in state $|1\rangle$ with certainty. However, these types of correlation still do not allow communication faster than the speed of light [x].

The Bell states can be created through quantum circuits in many different ways. The simplest may be via usage of the Hadamard, NOT, CNOT and Pauli-Z gates, as shown in Fig. 2.3.

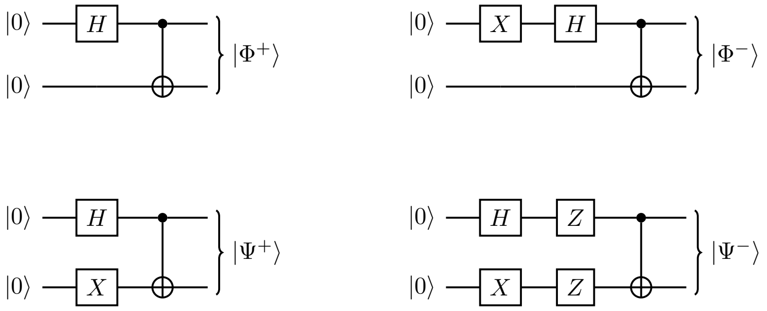


Figure 2.3: Creation of Bell states via quantum circuits

2.9 Phase kickback

In classical computing, logic gates operate independently of each other. However, in quantum computing, quantum gates, such as the CNOT gate, can simultaneously affect multiple qubits due to the principles of superposition and entanglement. This interconnected behavior is exemplified by the phenomenon of phase kickback, where the state of one qubit indirectly influences another qubit's state through controlled gates, resulting in a change in the second qubit's phase.

A simple example of this effect is application of CNOT gate on a quantum circuit where the control qubit is in the state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and the target qubit is the state $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. This means that the initial state is $|+-\rangle$. Applying the CNOT gate results in the following state transformation

$$\begin{aligned} CNOT\,(|+-\rangle) &= CNOT\left(\frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)\right) \\ &= \frac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle) = |--\rangle\,, \end{aligned} \tag{2.38}$$

where it is apparent that the controlled quantum gate, contrary to what was expected, modifies only the state of the control qubit, leaving the target qubit unchanged.

Phase kickback is a phenomenon that is highly utilized in quantum algorithms. For example, in Grover's algorithm, phase kickback is employed to amplify the amplitude of the target state while simultaneously suppressing the amplitudes of the non-target states. This selective amplification and suppression are key to the efficiency of Grover's search algorithm, as we will discuss in Sec. 2.12.2.

2.10 Universal quantum gates

In classical computation, a universal gate refers to a logic gate that can implement any Boolean function without the need for any other type of gate. Examples of universal gates include NAND[2] and NOR[3] gates, which both have two input bits and one output bit.

In quantum computing, a set of quantum gates is considered universal if any unitary operation can be approximated to arbitrary accuracy by a quantum circuit that involves only those gates [7]. It can be shown that the combination of single-qubit gates and the $CNOT$ gate are universal gates for quantum computation.

2.11 Notable 3-qubit gates

Although any quantum circuit can be constructed using a combination of single-qubit and 2-qubit gates that together form a universal set of quantum gates, it is valuable to highlight a few notable 3-qubit gates. These gates are significant because of their historic importance, their connection to classical computing, and their role in illustrating the decomposition of gates operating on a larger number of qubits into single-qubit and 2-qubit gates.

2.11.1 Toffoli gate

The Toffoli gate, also known as the CCNOT gate, was introduced in 1980 by Tommaso Toffoli. This gate operates on three qubits: two control qubits and one target qubit. The control qubits remain unchanged by the gate's action, while the state of the target qubit is flipped (i.e., a NOT operation is applied) only if both control qubits are in the state $|1\rangle$. The Toffoli gate is significant because it is a universal reversible logic gate in classical computing. Unlike many classical logic gates, such as NAND, which are irreversible, all quantum logic gates must be unitary and thus inherently reversible. Although classical circuits, which often rely on irreversible gates, cannot be directly simulated by quantum circuits, any classical computation can be performed by an equivalent circuit composed entirely of reversible gates, such as the Toffoli gate.

As mentioned earlier, any quantum circuit can be implemented using just the Hadamard, phase, CNOT, and $\frac{\pi}{8}$ gates. Consequently, a possible implementation of the Toffoli gate in practical simulations on quantum computers is depicted in Fig. 2.4.

[2] The NAND gate represents the complement of the AND operation with the algebraic function $F = (xy)'$, where x and y are input bits.

[3] The NOR gate represents the complement of the OR operation with the algebraic function $F = (x + y)'$, where x and y are input bits.

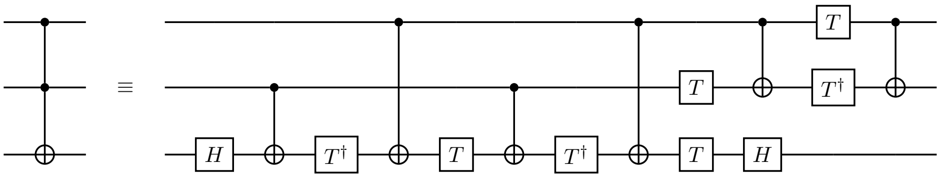


Figure 2.4: Decomposition of the Toffoli gate into one- and two-qubit gates [7]

2.11.2 Fredkin gate

The Fredkin gate, also known as the controlled SWAP gate, was conceptualized by Edward Fredkin and Tommaso Toffoli in 1982. The gate operates on three qubits, where the first qubit is the control qubit, and the other two are the target qubits. The gate swaps the states of the two target qubits only if the control qubit is in the state $|1\rangle$; otherwise, the target qubits remain unchanged. Similarly to the Toffoli gate, the Fredkin gate is reversible and can be decomposed using the same set of single-qubit and two-qubit gates, as shown in one possible decomposition in Fig. 2.5.

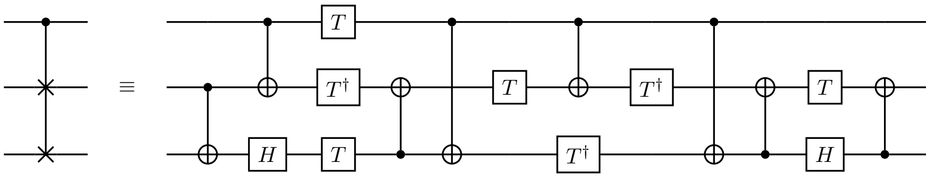


Figure 2.5: Decomposition of the Fredkin gate into one- and two-qubit gates [8]

2.12 Quantum algorithms

Quantum algorithms represent a new milestone in the quest for efficient solutions to various problems. They leverage the principles of quantum mechanics and harness the unique properties of qubits. Although many remarkable quantum algorithms have already been discovered, describing all of them would be too broad for this project. Therefore, in this chapter, we will select a few key algorithms, describe their working principles, and explore their potential future applications.

2.12.1 Deutsch-Jozsa algorithm

The Deutsch-Jozsa algorithm, developed by David Deutsch and Richard Jozsa in their 1992 paper [2], addresses the following problem: given a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ provided as an oracle[4], determine whether the function is constant or balanced. A function is constant if it produces the same output (either 0 or 1) for all possible inputs. In contrast, it is balanced if it outputs 0 for half of the possible inputs and 1 for the other half. The

[4] In quantum computation, an oracle is a subroutine that represents an unknown function $f(x)$, which is provided as an input and returns an output based on a predetermined rule.

objective is to determine which of these two cases describes the function, using as few function queries as possible.

Classically, solving the Deutsch-Jozsa problem requires the function $f(x)$ to be queried $2^{n-1}+1$ times in the worst-case scenario. However, on a quantum computer, the Deutsch-Jozsa algorithm can solve the problem with just one function evaluation.

We will focus solely on the quantum version of the algorithm. The algorithm starts with $(n+1)$ qubits initialized in the state $|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle$, where n is the number of input qubits in the state $|0\rangle$, and the remaining qubit in the state $|1\rangle$ is introduced as an ancillary qubit to facilitate computation. Next, a Hadamard gate is applied to each qubit to create a superposition of states $|0\rangle$ and $|1\rangle$, which can be expressed as

$$|\psi_1\rangle = H^{\otimes n}|0\rangle^{\otimes n} \otimes H|1\rangle = \frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle\left(|0\rangle - |1\rangle\right), \tag{2.39}$$

where x represents the n-bit strings from 0 to 2^n-1. The next step involves applying the oracle function (or black-box function) U_f to $|\psi_1\rangle$. The oracle function flips the phase of the state $|x\rangle$ if and only if $f(x)=1$[9]. Mathematically, this can be represented as

$$U_f : |x\rangle|y\rangle \rightarrow |x\rangle\left(|y \oplus f(x)\rangle\right), \tag{2.40}$$

where $\oplus$ represents addition modulo 2. After applying the oracle, the state becomes

$$\begin{aligned}|\psi_2\rangle = U_f|\psi_1\rangle &= \frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle\left(|f(x)\rangle - |1\oplus f(x)\rangle\right)\\ &= \frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|x\rangle\left(|0\rangle - |1\rangle\right),\end{aligned} \tag{2.41}$$

where the phase $(-1)^{f(x)}$ is “kicked back” from the ancillary qubit to the input qubits. To determine whether the function is constant or balanced, the ancillary qubit $\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$ can be omitted from further measurements. Then, a Hadamard gate is applied to the n input qubits, leading to the quantum state

$$|\psi_3\rangle = \sum_{y=0}^{2^n-1}\left[\frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}(-1)^{x\cdot y}\right]|y\rangle, \tag{2.42}$$

where the probability of obtaining state $|y\rangle$ from the measurement corresponds to

$$P(y) = \left|\frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}(-1)^{x\cdot y}\right|^2. \tag{2.43}$$

For illustration, considering the case where $|y\rangle = |0\rangle^{\otimes n}$, the probability of obtaining this state simplifies due to $x\cdot y = 0$. Thus, we obtain

$$P(y) = \left|\frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}\right|^2. \tag{2.44}$$

This probability equals 1 if the $f(x)$ is constant, due to constructive interference, and 0 if $f(x)$ is balanced, due to destructive interference. In other words, the final measurement result will be $|0\rangle^{\otimes n}$ with unit probability if $f(x)$ is constant and we will not obtain $|0\rangle^{\otimes n}$ at all if the $f(x)$ is balanced.

Although the Deutsch-Jozsa algorithm demonstrates a task that can be solved more efficiently with a quantum computer than with its classical counterpart, a practical application for distinguishing between constant or balanced functions is not yet known.

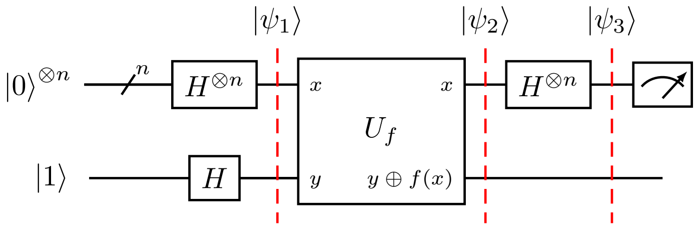


Figure 2.6: Quantum circuit diagram of Deutsch-Jozsa algorithm quantum [7]

2.12.2 Grover's algorithm

Grover's algorithm, also known as the quantum search algorithm, offers a quadratic speed-up over classical algorithms for searching an unsorted database. The problem addressed by this algorithm is as follows: Given an unsorted database containing N elements, where only one specific item (that is, the marked item) has to be retrieved/found. In classical computing, an average algorithm involves examining the items in the database one by one. Thus, the process would require the test of $\frac{N}{2}$ items on average. In contrast, Grover's algorithm can achieve the same result in only $O\left(\sqrt{N}\right)$ steps, representing a quadratic speed up.

The first step in Grover's algorithm initializes the quantum state to represent all possible solutions. This is done by using Hadamard gates, which puts the system into a uniform superposition over computational basis as

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \,. \tag{2.45}$$

For the following step, lets consider that a function $f : \{0, 1, \ldots, N\} \to \{0, 1\}$, where the domain represents indices in the database, and $f(x) = 1$ only for the index that satisfies the search criterion. This function is implemented in the algorithm through a unitary operator U_w known as Grover's oracle, which acts as follows

$$U_w |x\rangle = (-1)^{f(x)} |x\rangle \,. \tag{2.46}$$

This operation introduces the phase π to the basis state which is being searched for, leaving all other bases states unchanged. Essentially, this operator "marks" the target state in the search space. Following this, the diffusion operator is applied, which can be represented as

$$U_G = 2 |\psi\rangle \langle\psi| - I \,. \tag{2.47}$$

The diffusion operator amplifies the amplitude of the desired state while simultaneously reducing the amplitudes of all other states. This operation increases the likelihood of finding the target state upon measurement. Both operators U_w and U_G are applied multiple times before the desired state is found. The number of Grover iterations k depends on the size of the searched database, specifically $k = \frac{\pi}{4}\sqrt{N}$ for the case of the database with only one target item. This ensures that we can find the marked item with very high probability in $O\left(\sqrt{N}\right)$ steps. Grover's algorithm can be modified for the situations where there is more than one target item. The number of necessary iterations is then reduced to $k = \frac{\pi}{4}\sqrt{\frac{N}{M}}$, where M stands for the number of targeted items.

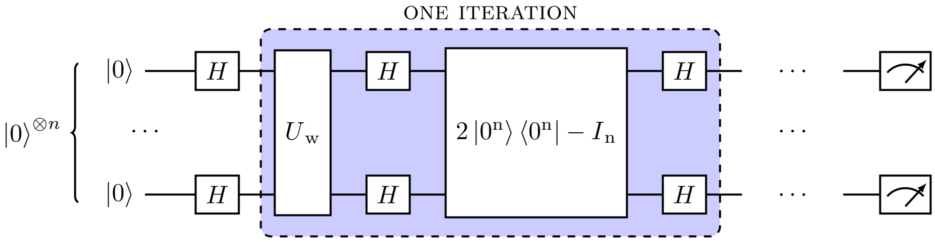


Figure 2.7: Scheme of the Grover's algorithm quantum circuit

Geometrical interpretation

To have more intuition on how Grover's algorithm works, we can examine it from a geometrical point of view. Considering a two-dimensional space spanned by the initial vector $|\psi\rangle$ and the state consisting of a uniform superposition of solutions to the search problem and the general initial state

$$|\psi\rangle = \sqrt{\frac{N-M}{N}}\,|\alpha\rangle + \sqrt{\frac{N}{M}}\,|\beta\rangle\,, \tag{2.48}$$

where N is the number of all indexes, M represents the number of solutions and $|\alpha\rangle$, $|\beta\rangle$ are balanced superposition of all states p and all states representing solution s defined as

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}}\sum_p |p\rangle \tag{2.49}$$

$$|\beta\rangle = \frac{1}{\sqrt{M}}\sum_s |s\rangle\,, \tag{2.50}$$

the effect of applying Grover iteration $G = U_G U_w$ can be understood as that the oracle operation U_w performs a reflection about the vector $|\alpha\rangle$ and $|\beta\rangle$ in the plane defined by $|\alpha\rangle$ and $|\beta\rangle$ as follows

$$U_w\left(a\,|\alpha\rangle + b\,|\beta\rangle\right) = a\,|\alpha\rangle - b\,|\beta\rangle\,. \tag{2.51}$$

Reflection in the plane defined by $|\alpha\rangle$ and $|\beta\rangle$ about vector $|\psi\rangle$ is also performed by the Grover diffusion operator $U_G = 2\,|\psi\rangle\,\langle\psi| - I$ as shown on a general initial state:

$$U_G \sum_i \alpha_i \left|i\right\rangle = \sum_i \left(2\langle\alpha\rangle - \alpha_i\right)\left|i\right\rangle . \tag{2.52}$$

The Grover iteration is then given as a product of two reflections which is a rotation, therefore the state $G^k \left|\psi\right\rangle$ after $k \in \mathbf{N}$ iterations remains in the space spanned by the vectors $\left|\alpha\right\rangle$ and $\left|\beta\right\rangle$. Consider now $\cos\frac{\theta}{2} = \sqrt{\frac{N-M}{N}}$ and $\sin\frac{\theta}{2} = \sqrt{\frac{M}{N}}$, the initial state can be rewritten as

$$\left|\psi\right\rangle = \cos\frac{\theta}{2}\left|\alpha\right\rangle + \sin\frac{\theta}{2}\left|\beta\right\rangle , \tag{2.53}$$

where $\theta/2$ is the angle between vectors $\left|\psi\right\rangle$ and $\left|\alpha\right\rangle$. The application of Grover iteration on a state $\left|\psi\right\rangle$ leads to state

$$G\left|\psi\right\rangle = \cos\frac{3\theta}{2}\left|\alpha\right\rangle + \sin\frac{3\theta}{2}\left|\beta\right\rangle , \tag{2.54}$$

and the k times repeated application results in a state

$$G^k\left|\psi\right\rangle = \cos\frac{(2k+1)\,\theta}{2}\left|\alpha\right\rangle + \sin\frac{(2k+1)\,\theta}{2}\left|\beta\right\rangle , \tag{2.55}$$

which implies that the angle of the rotation equals to a real number $\theta \in \langle 0, 2\pi\rangle$.

2.13 Quantum state tomography

At the end of each quantum circuit or algorithm, information from the simulated system is obtained by measurement, which in quantum mechanics produces probabilistic outcomes. For example, measuring a qubit in the computational basis ($\left|0\right\rangle$ and $\left|1\right\rangle$) provides the probability of finding the qubit in one of these states, and gives very little information about the actual quantum state, which could be a superposition or an entangled state. To obtain more information about the state, we need to implement quantum state tomography. This technique aims to reconstruct the density matrix of a quantum state by performing a series of measurements on multiple copies of the system.

Quantum state tomography typically involves three key steps: preparation, measurement, and reconstruction. First, the quantum state is prepared in the state designated for measurement. Since quantum states cannot be directly observed without altering them, multiple identical copies of the system are prepared to ensure accurate measurements. In practice, it means running the exact same experiment several times. Next, a set of measurements is performed on these copies in chosen bases, often the Pauli measurement bases. For example, for a single qubit, measurements might be made in the σ_x, σ_y and σ_z, requiring at least 3 copies of the measured system. This generalizes to an n-qubit system, which would require 3^n copies for measurement in the Pauli measurement bases. After collecting the measurement data, the final step is the reconstruction of the density matrix. Various algorithms have already been studied for this purpose, including linear inversion, maximum likelihood estimation, and Bayesian interference.

In practice, the density matrix obtained from quantum state tomography will be influenced by errors that inevitably occur during the simulation on a quantum device. The quality of the obtained results can be quantified by fidelity, which is a measure of state overlap. Considering two states described by density matrices $\hat{\rho_1}$ and $\hat{\rho_2}$, the fidelity of these states is given by

$$F\left(\rho_1,\rho_2\right)=\left[\mathrm{Tr}\left(\sqrt{\sqrt{\rho_1}\rho_2\sqrt{\rho_1}}\right)\right]^2. \tag{2.56}$$

If $\hat{\rho_1}$ or $\hat{\rho_2}$ is a pure state, the equation simplifies to $F\left(\rho_1,\rho_2\right)=\mathrm{Tr}\left(\rho_1\rho_2\right)$ [10].

2.14 Universal quantum simulation

A universal quantum simulator is a theoretical device capable of simulating any physical quantum system with high efficiency and accuracy. The need for such a device was first proposed by the physicist Richard Feynman in 1982 [11]. Feynman suggested that classical computers might not be able to simulate quantum phenomena due to the exponential scaling of time and memory resources required to accurately describe the dynamics of physical variables as the system size increases. He also established a rule for simulation: the number of computer elements required to simulate a large physical system should only be proportional to the space-time volume of the physical system. Thus, any simulation with exponential complexity violates this rule. Since nature behaves quantum mechanically, only a computing machine (quantum simulator) that obeys the same quantum mechanical laws would be able to simulate physical systems efficiently.

Two different types of quantum simulators have been proposed: analog and digital. Analog quantum simulators often use continuous variables such as the position and momentum of quantum particles. A key characteristic of analog quantum simulators is their utilization of continuous evolution to reproduce the simulated quantum system in the simulating system as closely as possible. As a result, the Hamiltonian of the simulated systems is made as similar as possible to the Hamiltonian of the analog quantum simulator. In contrast, digital quantum simulators decompose the evolution associated with a complex Hamiltonian of the simulated system in terms of efficiently implementable single-qubit and two-qubit gates. To illustrate why this decomposition is possible, consider a quantum system governed by Schrödinger's equation:

$$i\hbar\frac{d}{dt}\left|\psi\right\rangle=H\left|\psi\right\rangle \tag{2.57}$$

with a solution for a time-independent Hamiltonian H

$$\left|\psi\left(t\right)\right\rangle=e^{-iHt}\left|\psi\left(0\right)\right\rangle, \tag{2.58}$$

where we have absorbed $\hbar$ into H. For most of the physical system of n particles, the Hamiltonian can be rewritten as a sum over local interactions as follows

$$H = \sum_{k=1}^{L} H_k, \tag{2.59}$$

where each Hamiltonian H_k acts on at most a constant c number of systems, and L is polynomial in n [7]. Finally, by utilizing the Trotter-Suzuki expansion

$$e^{-iHt} \simeq \left(e^{\frac{-iH_1 t}{l}} \dots e^{\frac{-iH_M t}{l}} \right) + \sum_{i<j} \frac{[H_i, H_j]\, t^2}{2l}, \tag{2.60}$$

where l represents the total number of Trotter steps and the increase in the value of l increases the accuracy of digitized dynamics, we obtain the described decomposition. This approximation implies that to achieve acceptable fidelities, the number of digital steps must increase with the simulation time due to the error in the approximation that scales with $\frac{t^2}{l}$. The main benefit of digital quantum simulators is their ability to incorporate quantum error correction techniques to improve the accuracy of simulations over long timescales. However, achieving practical scalability for digital quantum simulators remains rather difficult.

Chapter 3
Quantum walks

Quantum walks are quantum analogues of classical random walks, and have become an important research topic in recent years [12]. In classical random walks, a particle moves step by step in a certain direction with certain probabilities, and after a given number of steps it follows a certain trajectory. In contrast, quantum walks take advantage of the principles of quantum mechanics, particularly superposition and interference, which lead to different and often more powerful behaviors in fulfilling certain tasks. Unlike in classical random walks where the walker would move randomly, in a quantum walk, the walker can exist in a superposition in position space, allowing the walker to explore multiple paths simultaneously. Two main models of quantum walks have been formulated based on the approach to dealing with time: discrete-time quantum walks and continuous-time quantum walks.

In this chapter, we examine quantum walks that can be effectively realized using available quantum computers. Our exploration is framed by current technological constraints, particularly the complexity of quantum circuits that contemporary quantum computers can simulate with acceptable fidelity. Despite these challenges, significant progress has been made in identifying and simulating physical systems within the reach of existing quantum computers. We will first define classical random walks for completeness and for comparison with quantum walks in the following sections. Then, we will briefly describe the continuous variant of quantum walks. Finally, we will define discrete-time quantum walks that we will be simulating on a quantum computer.

3.1 Random walks

Random walks model the motion of a particle consisting of a succession of random steps in a discrete space, such as points in a regular lattice or integers on a line. The simplest random walk is a one-dimensional walk on an integer line $\mathbb{Z}$. Suppose that the particle starts at position 0 and with each step it moves one position left or right based on the result of a coin toss. After the N number of steps, the probability P of finding the particle at position

$m \in \mathbb{Z}$ is given by the following expression.

$$P(m,N) = \frac{N!}{\left(\frac{N+m}{2}\right)!\left(\frac{N-m}{2}\right)!} p^{\frac{1}{2}(N+m)} q^{\frac{1}{2}(N-m)}, \tag{3.1}$$

where p be the probability of right step and $q = 1 - p$ is the probability of a left step. This probability can be expressed by a binomial distribution of a form

$$b(n,p,k) = \binom{n}{k} p^k (1-p)^{n-k} \tag{3.2}$$

that calculates number of successes in n fixed trial. The p then stands for probability of success and k is the number of successes. Considering the fact that $n = \frac{1}{2}(N+m)$, the probability $P(m,N)$ can be then rewritten as

$$P(m,N) = \binom{N}{n} p^n q^{N-n}. \tag{3.3}$$

Based on the probabilities p and q the random walk can be either balanced when $p = q$, thus the probabilities of walker going right or left are equal, or unbalanced random walk when those probabilities differ. A balanced random walk embodies symmetry and often manifests in scenarios such as unbiased coin flipping or the diffusion of particles in a homogeneous medium. In contrast, unbalanced random walk entails the disruption of symmetry and introduction of bias.

The final image of the probability distribution of the random walk also depends on the number of steps taken, because of the fact that the distribution spreads over time, which can be quantified by the standard deviation in a form

$$\sigma(N) = \sqrt{\langle n^2 \rangle - \langle n \rangle^2}. \tag{3.4}$$

Since the binomial distribution (3.3) can be approximated with normal distribution

$$w(n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp - \frac{(n - \overline{n})^2}{2\sigma^2} \tag{3.5}$$

with the same mean $\overline{n}$ and variance σ^2 as n gets large. The variance increases linearly as shown on Fig. 3.1 and 3.4.

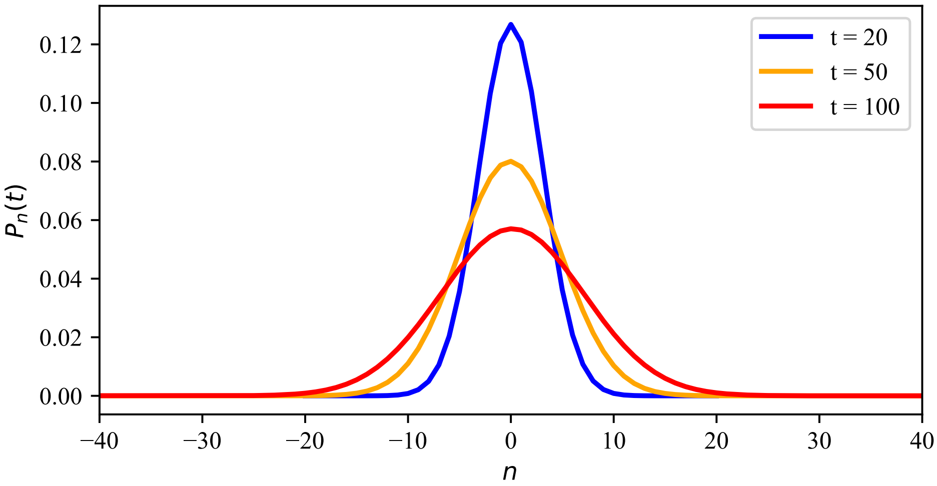


Figure 3.1: Probability distribution of a classical random walk on a line with the initial position $n = 0$ after t number of steps.

3.2 Continuous-time quantum walks

In this model of quantum walks, the entire evolution of the state of the walker takes place in the position space $\mathcal{H}_p$. The model of a continuous-time quantum walk takes inspiration from the continuous-time classical random walk on a graph $G = (V, E)$ where V is the vertex set in which some pairs are connected by links E. In order to define continuous evolution of a state in time, we need to introduce the *adjacency matrix* A of the graph G, which is a square $n \times n$ matrix containing information about links between vertices with elements given as

$$A_{jk} = \begin{cases} 1 & \text{if } (v_j, v_k) \in E \\ 0 & \text{if } (v_j, v_k) \notin E \end{cases} . \tag{3.6}$$

and the *Laplacian* of the graph, which is a diagonal matrix of the degrees of each vertex defined as

$$L = D - A, \tag{3.7}$$

where $D_{jk} = deg(v_j)\delta_{jk}$.

Suppose now that the particle has a probability rate γ of jumping to the neighboring vertices and that the transition can occur at all times. The infinitesimal generator matrix H describing the transition of a particle between vertices is then given by $H = -\gamma L$. If we now utilize the idea of Fahri and Gutmann [13], then the generator matrix can be considered as the *Hamiltonian* of the process which generates an evolution $U(t)$ as

$$U(t) = \exp(-iHt). \tag{3.8}$$

Thus, considering an initial state $|\psi_0\rangle$, the time evolution of this state is then given by

$$|\psi_t\rangle = e^{-iHt} |\psi_0\rangle \tag{3.9}$$

and the probability of measuring the particle at a position m at time $t > 0$ is given by

$$P_{n,m}(t) = |\langle m| e^{-iHt} |n\rangle|^2. \tag{3.10}$$

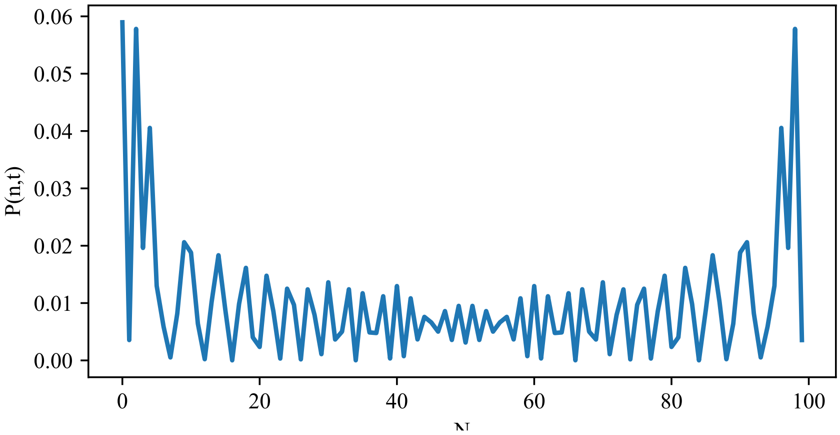


Figure 3.2: Probability distribution of continuous-time quantum walk with time step $t = 0.1$ till $t = 100$ and hoping rate $\gamma = 0.25$.

3.3 Discrete-time quantum walks

Discrete-time quantum walks (or simply quantum walks), akin to their classical random-walk counterparts, involve discrete steps between neighboring positions on a lattice or graph. Unlike continuous-time quantum walks, the total Hilbert space in discrete-time quantum walks consists of a coin space $\mathcal{H}_c$ as well as a position space $\mathcal{H}_p$. Also, the time evolution of discrete-time quantum walks takes place in discrete-time steps, contrary to continuous-time quantum walks. First, a unitary operator is applied solely to the coin space (analogous to a coin toss), and then a step is taken according to the coin state. From this perspective, there is a direct analogy between discrete-time quantum walks and classical random walks.

For simplicity, we will define the model only in one dimension, corresponding to a walk on a line or a circle. The Hilbert space $\mathcal{H}_p$ is spanned by the natural basis states $\{|n\rangle \mid n \in \mathbb{Z}\}$ representing the possible positions in which the walker can be found. Since there are two possible directions to move in one dimension, the coin space $\mathcal{H}_c$ is chosen to be two-dimensional and is spanned by $\{|0\rangle, |1\rangle\}$ representing an internal degree of freedom corresponding to the directions in which the walker can move. Thus, the total state $|\psi\rangle$ lies in the Hilbert space $\mathcal{H} = \mathcal{H}_\text{C} \otimes \mathcal{H}_\text{P}$. *A step* of a quantum walk $U_s = S(C \otimes I)$ consists of a consecutive application of a *coin operator* C in the coin space and a *shift operator* S in the total space. The most general coin operator can be written as

$$C(\theta, \phi_1, \phi_2) = \begin{pmatrix} \cos\theta & \sin\theta e^{i\phi_1} \\ \sin\theta e^{i\phi_1} & -\cos\theta e^{i(\phi_1+\phi_2)} \end{pmatrix} \tag{3.11}$$

There are many options in which this operator can be defined, but the popular choice is $H = C(\frac{\pi}{4}, 0, 0)$ which represents the Hadamard coin operator

$$H = \frac{|0\rangle \langle 0| + |0\rangle \langle 1| + |1\rangle \langle 0| - |1\rangle \langle 1|}{\sqrt{2}}. \tag{3.12}$$

We will use Hadamard coin operator throughout the thesis. The conditional shift operator is given by

$$S = \sum_{n=-\infty}^{+\infty} (|0\rangle\langle 0| \otimes |n+1\rangle\langle n| + |1\rangle\langle 1| \otimes |n-1\rangle\langle n|), \tag{3.13}$$

which acts on both the coin and position space simultaneously moving the walker one position right when the coin component is in the state $|0\rangle$ and one position left when the coin state is $|1\rangle$. The unitary evolution of a quantum walk in discrete time is specified by the product of the coin flip operator C and the conditional shift operator S, which are applied repeatedly as follows:

$$|\psi(t)\rangle = [S(C \otimes I)]^t \, |\psi(0)\rangle = U_s^t \, |\psi(0)\rangle \,, \tag{3.14}$$

where $|\psi(0)\rangle$ is the initial state and $|\psi(t)\rangle$ is the evolved state after t steps. The coin operator only acts on the coin state. Although there is no restriction on the choice of the initial position state, we usually start with a quantum walker localized in a specified position with an arbitrary initial coin state as

$$|\psi(0)\rangle = |\psi_c\rangle \otimes |0\rangle \,, \tag{3.15}$$

where the origin is chosen to be the initial position of the quantum walker. Although the time evolution of a discrete-time quantum walk is deterministic, the random nature of the discrete quantum walk comes from measurement. If we consider a state after t steps

$$|\psi(t)\rangle = \sum_{c,n} a_{c,n}(t) \, |c, n\rangle, \tag{3.16}$$

where $a_{c,n}(t)$ stands for the site amplitude and $c \in \{0, 1\}$. Then the probability of finding walker on any position at time t is given by summation over the probabilities in the coin space as follows

$$P_n(t) = \sum_c |a_{c,n}(t)|^2. \tag{3.17}$$

The most commonly used measure for determining how fast quantum walks propagate in position space is the variance of the position distribution, defined as

$$\sigma^2(t) = \sum_n (x_n - \bar{x})^2 P_n(t), \tag{3.18}$$

where x_n represents the position, $\bar{x}$ is the mean position, and $P_n(t)$ is the probability of being at position x_n at time t.

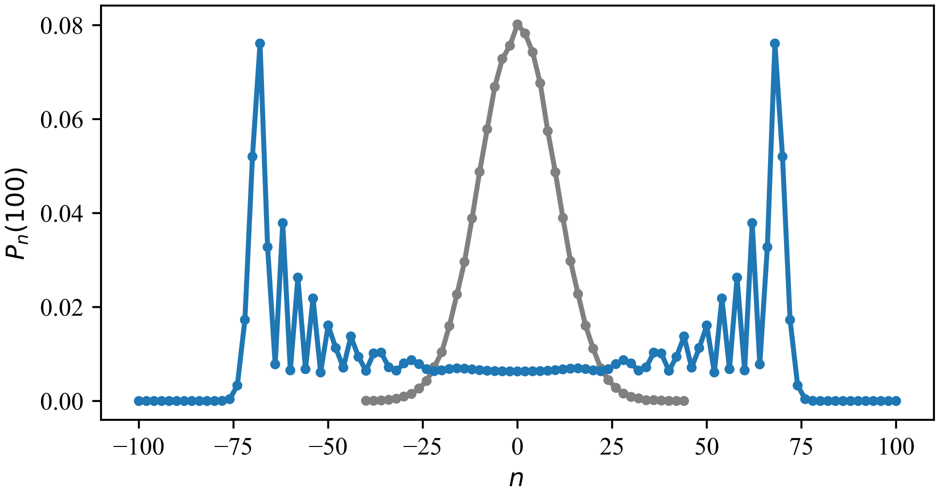


Figure 3.3: Probability distribution of a discrete-time quantum walk (blue) and random walk (grey) on a line for a walker which is localized at the position $n = 0$ at time $t = 0$ after 100 steps. The graph shows only nonzero values of $P_n(100)$.

An important feature of a discrete-time quantum walk is that the variance σ^2has a quadratic dependence on the number of steps, which is significantly faster than the linear dependence on time observed in classical random walks. This speedup is one of the main reasons why quantum walks have been studied so extensively in recent years. Their applications include the development of quantum search algorithms based on quantum walks [x].

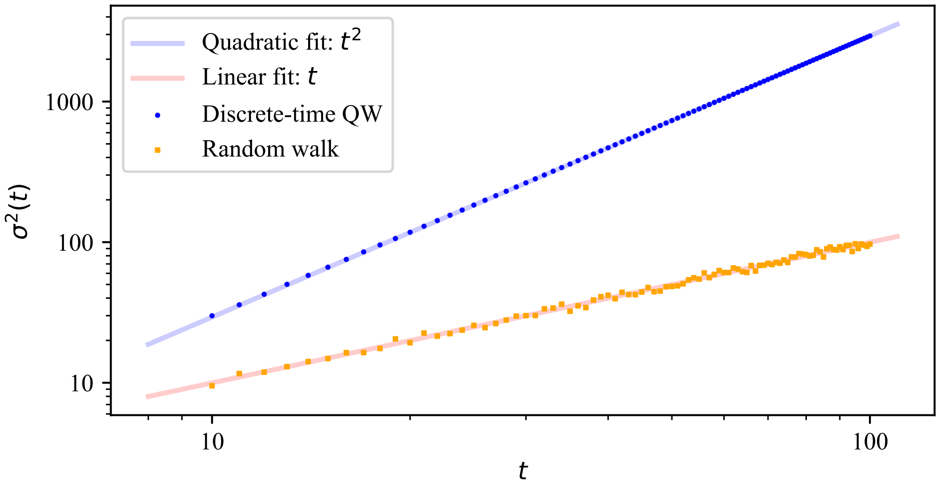


Figure 3.4: Comparison of a variance for a discrete-time quantum walk and a random walk for first $t = 100$ steps. Plot shown on a graph with logarithmic scales.

3.4 Circuit diagrams for discrete-time quantum walks

To realize a quantum walk on a quantum computer, we first need to determine how to define the position space. As explained in Sec. 2.7, for a quantum register of size n, the total Hilbert space is 2^n dimensional and is spanned by the computational basis $|n\rangle$ where $n \in \{0, \ldots, 2^n - 1\}$. In this way, the shift operator will perform binary addition and subtraction over this space, corresponding to the shift-right and shift-left operations, respectively. Since the shift operator depends on the coin space, as discussed in Sec.3.3, we also require an additional quantum register for the coin state as seen in Fig. 3.5(a). Therefore, the total dimension of Hilbert space is 2^{n+1}. Given a quantum register of size n for the position space, we will restrict our attention to N-cycles, where $N = 2^n$. This is a natural choice when N is the total number of discrete positions that a quantum walker can occupy on a cycle. Here, a cycle is a one-dimensional discrete space with periodic boundary conditions.

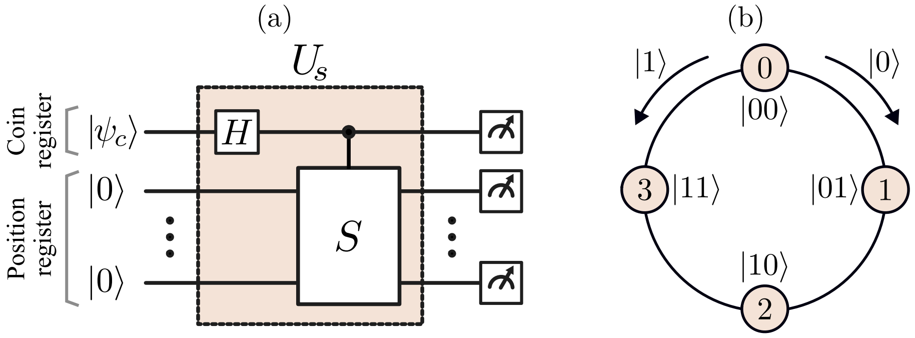


Figure 3.5: The circuit diagram for a discrete-time quantum walk in one dimension with a Hadamard coin operator and initial coin state $|\psi_c\rangle$. The step operator U_s consists of a coin operator H acting only on the coin register and a shift operator S acting on the position register, controlled by the coin register. Once the quantum circuit is run and the individual qubits are measured, the classical information obtained provides information about the position and coin state of the quantum walk. For example, for $n = 3$, a possible outcome of the quantum circuit could be 0101. If the coin register corresponds to the most significant bit in the measurement result, then this result indicates that the quantum walker is found at position 6 with coin state $|0\rangle$. After running the quantum circuit many times, a statistical distribution is obtained for $2^4 = 16$ possible outcomes. (b) A schematic description of a 4-cycle and the association of positions with quantum states. The $|0\rangle$ ($|1\rangle$) coin state represents clockwise (counterclockwise) direction.

For simplicity, let us consider 4-cycles as shown in Fig. 3.5(b). However, the discussions here can be generalized to any N-cycles with $N = 2^n$. In general, as long as we keep track of which position corresponds to which position state, the positions in a 4-cycle can be associated with position quantum states randomly. However, in Fig. 3.5(b), we have enumerated the positions

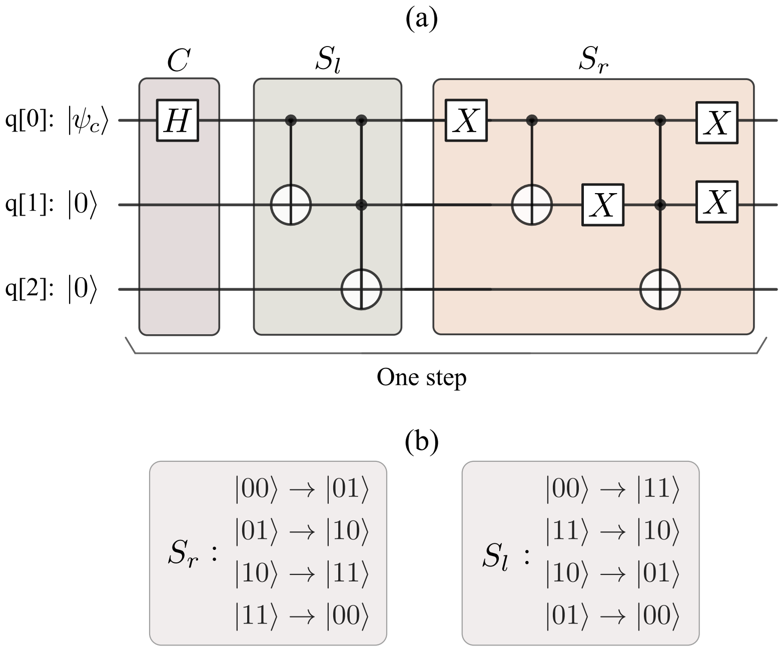

(a)
C
S_l
S_r
q[0]: $|\psi_c\rangle$
q[1]: $|0\rangle$
q[2]: $|0\rangle$
H
X
X
X
X
One step
(b)
S_r :
$|00\rangle \rightarrow |01\rangle$
$|01\rangle \rightarrow |10\rangle$
$|10\rangle \rightarrow |11\rangle$
$|11\rangle \rightarrow |00\rangle$
S_l :
$|00\rangle \rightarrow |11\rangle$
$|11\rangle \rightarrow |10\rangle$
$|10\rangle \rightarrow |01\rangle$
$|01\rangle \rightarrow |00\rangle$

Figure 3.6: (a) Circuit diagram of the single step of quantum walk on 4-cycle with initial coin state $|\psi_c\rangle$ and initial position $|00\rangle$. (b) Basis state transformations with periodic boundary conditions for shift-right S_r and shift-left S_l operators.

with increasing position states in a clockwise direction for simplicity. The shift operator as given in Eq. 3.12, consists of two parts: the shift-right (shift-clockwise) operator $S_r = \sum_n |n+1\rangle \langle n|$, and shift-left (shift-counterclockwise) operator $S_l = \sum_n |n-1\rangle \langle n|$. In Fig. 3.6(b), the proper basis transformations for S_r and S_l operators are given. For S_l, we see that if the coin state is $|1\rangle$, q[1] is always flipped, and q[2] is flipped only if the new state of q[1] is $|1\rangle$. For S_r, if the coin state is $|0\rangle$, q[1] is always flipped, and q[2] is flipped only if the new state of q[1] is $|0\rangle$. These transformations are binary subtraction and addition operations, respectively, are given in Fig. 3.6(a). Note that in the figure, q[1] and q[2] are the least significant and the most significant qubits, respectively.

In Sec. 2.10 we discussed that single-qubit gates and two-qubit $CNOT$ gate are enough to realize any unitary operation up to an arbitrary approximation. We also stated in Sec. 2.11 that we need 6 $CNOT$ gates and 9 single-qubit gates to implement a Toffoli gate. Therefore, a single step of quantum walk can be realized with 14 $CNOT$ gates and 23 single-qubit operations. This is very costly when implementing a quantum walk on a noisy quantum computer with non-negligible error rates in the realization of quantum gates and readouts. Therefore, to perform quantum walks on a quantum computer, we need to find optimized methods that depend on certain initial conditions.

3.5 Optimized formulation of discrete-time quantum walks on cycles

The shift operator, as defined in Eq. 3.12 splits the position space into two distinct subspaces: one containing only odd-numbered positions and the other containing only even-numbered positions. If the quantum walk is initially localized in either of these subspaces, the system's total quantum state will switch between the two subspaces at each step. For simplicity, consider a quantum walk on a cycle with four vertices as shown in Fig. 3.5(b), where the walker begins at position 0 with the initial state $|00\rangle$. To realize a discrete-time quantum walk on this cycle, we need an 8-dimensional Hilbert space because the coin and position spaces are two- and four-dimensional, respectively. Thus, the realization of this quantum walk on a quantum computer requires $\log_2 8 = 3$ qubits. However, the quantum walker can occupy only half of the sites in the positional space at any given step, reducing the positional space to two dimensions and the necessary Hilbert space to four dimensions. By closely analyzing how the probability amplitudes are transformed between these subspaces during the walk, we can effectively describe the quantum walk as occurring within a space half the original size. Therefore, we can optimally simulate a discrete-time quantum walk on a 4-cycle with $\log_2(4) = 2$ qubits.

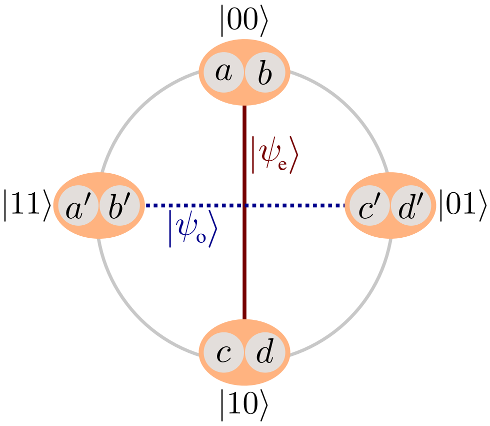


Figure 3.7: Schematic description of the optimized discrete-time quantum walk

A simple demonstration of how we keep track of the probability amplitudes in each step is given in Fig. 3.7. Given that the walker starts its motion from $|00\rangle$, it can be found either in 0 and 2, or 1 and 3. The probability amplitudes (a, b), (c, d), (a', b'), and (c', d') are associated with positions 0, 2, 3 and 1, respectively. These amplitudes are actually nothing but the coin states associated with the corresponding position $a_{c,n}(t)$ as given in Eq. 3.16. Therefore, we divide the total wave function into two parts living in complementary subspaces and define the states $|\psi_e\rangle$ and $|\psi_o\rangle$, where the subindices indicate the even and odd vertices. These states are described by the following column vectors:

$$|\psi_e\rangle \equiv \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}, \qquad |\psi_o\rangle \equiv \begin{pmatrix} a' \\ b' \\ c' \\ d' \end{pmatrix}. \tag{3.19}$$

Therefore, a step in the quantum walk transforms the state $|\psi_e\rangle$ to $|\psi_o\rangle$, or the state $|\psi_o\rangle$ to $|\psi_e\rangle$ depending on how the walker was initially localized,

$$|\psi_e\rangle \xrightarrow{U_{eo}} |\psi_o\rangle, \qquad |\psi_o\rangle \xrightarrow{U_{oe}} |\psi_e\rangle \tag{3.20}$$

where U_{eo} and U_{oe} are transformation operators between the subspaces. After applying the coin operator as given in Eq. 3.11, the correct transformation between these states are given as

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \xrightarrow{U_{eo}} \begin{pmatrix} c\cos\theta + d\sin\theta e^{i\phi_1} \\ a\sin\theta e^{i\phi_2} - b\cos\theta e^{i(\phi_1+\phi_2)} \\ a\cos\theta + b\sin\theta e^{i\phi_1} \\ c\sin\theta e^{i\phi_2} - b\cos\theta e^{i(\phi_1+\phi_2)} \end{pmatrix} = \begin{pmatrix} a' \\ b' \\ c' \\ d' \end{pmatrix}, \tag{3.21}$$

and

$$\begin{pmatrix} a' \\ b' \\ c' \\ d' \end{pmatrix} \xrightarrow{U_{oe}} \begin{pmatrix} a'\cos\theta + b'\sin\theta e^{i\phi_1} \\ c'\sin\theta e^{i\phi_2} - d'\cos\theta e^{i(\phi_1+\phi_2)} \\ c'\cos\theta + d'\sin\theta e^{i\phi_1} \\ a'\sin\theta e^{i\phi_2} - b'\cos\theta e^{i(\phi_1+\phi_2)} \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}. \tag{3.22}$$

Therefore, the transformation matrices are given as

$$U_{eo} = \begin{pmatrix} 0 & 0 & \cos\theta & \sin\theta e^{i\phi_1} \\ \sin\theta e^{i\phi_2} & -\cos\theta e^{i(\phi_1+\phi_2)} & 0 & 0 \\ \cos\theta & \sin\theta e^{i\phi_1} & 0 & 0 \\ 0 & 0 & \sin\theta e^{i\phi_2} & -\cos\theta e^{i(\phi_1+\phi_2)} \end{pmatrix} \tag{3.23}$$

and

$$U_{oe} = \begin{pmatrix} \cos\theta & \sin\theta e^{i\phi_1} & 0 & 0 \\ 0 & 0 & \sin\theta e^{i\phi_2} & -\cos\theta e^{i(\phi_1+\phi_2)} \\ 0 & 0 & \cos\theta & \sin\theta e^{i\phi_1} \\ \sin\theta e^{i\phi_2} & -\cos\theta e^{i(\phi_1+\phi_2)} & 0 & 0 \end{pmatrix}. \tag{3.24}$$

Assuming that we start with the initial state $|\psi_e(0)\rangle$ at time $t = 0$, the time evolution of the optimized quantum walk is given as

$$\begin{aligned} |\psi_o(1)\rangle &= U_{eo}\,|\psi_e(0)\rangle \\ |\psi_e(2)\rangle &= U_{oe}\,|\psi_e(1)\rangle \\ |\psi_o(3)\rangle &= U_{eo}\,|\psi_e(2)\rangle \\ &\vdots \end{aligned} \tag{3.25}$$

The step operators in Eqs. 3.23 and 3.24 can be realized by one CNOT gate and four single-qubit operations. Compared to the straightforward implementation of the shift operator as given in Sec. 3.4 where we needed 14 $CNOT$ gates and 23 single-qubit operations, optimized method significantly simplifies the quantum circuits for discrete-time quantum walks. Furthermore, the optimization method we introduced in this section for 4-cycles can be generalized to N-cycles with $N = 2^n$ where $n = \log_2 N$. The reduction in the number of qubits is important in quantum programming because it significantly reduces the number of quantum gates needed to complete each step of a quantum walk. The practical implementation of each quantum logic gate in the quantum computer introduces unwanted noise that devalues the measured results. The current architecture of quantum computers permits only single- and 2-qubit gates, so all operators operating on higher dimensions need to be decomposed into gates operating with only 1 or 2 qubits. This decomposition requires more quantum gates as the dimension of the necessary Hilbert space increases.

IBM's quantum processing units are composed of transmon qubits which are a type of superconducting charge qubit. The main benefit of the transmon is its reduced sensitivity to charge noise, achieved by using a Josephson junction [1] in combination with the shunting capacitor. This setup significantly increases the qubit coherence time[2] by reducing the sensitivity to charge noise. Despite this, transmon qubits still behave like open quantum systems, interacting with their environment to some extent. These interactions lead to decoherence, the process by which quantum information is lost over time due to the entanglement of the qubit states of the environment.

3.6 Perfect state transfer

The transfer of a quantum state from one location to another without interrupting the encoded information is an important challenge for future quantum technologies. Since its first proposal by Bose [14], perfect state transfer has found applications in communication between quantum processors and in other areas such as the implementation of universal quantum gate sets using both continuous-time and discrete-time quantum walks [15, 16].

One of the most common scenarios for perfect state transfer involves quantum systems organized in a network, such as a line, a n-cycle, or a more complex graph structure. Given such a graph $G = (V, E)$ with vertices $a, b \in V(G)$ and a quantum walker that is initially localized at a vertex a. Than graph G has perfect state transfer from vertex a to vertex b at time t for a continuous-time quantum walk if

$$\left| \langle b| e^{-iHt} |a\rangle \right| = 1, \tag{3.26}$$

[1] A Josephson junction is a device constructed by a thin layer of a non-superconducting material between two layers of a superconducting material. According to the Josephson effect, current flows continuously across the device without any voltage applied.

[2] Time over which the quantum information is preserved.

where H is the Hamiltonian described in Sec. 3.2, and for a discrete-time quantum walk if

$$\sum_{c,c_0} |\langle b,c| U^k |a,c_0\rangle| = 1, \tag{3.27}$$

where c are coin states at vertex b, c_0 are the chosen initial coin states on vertex a and U is the evolution operator from section 3.3. In other words, the conditions (3.26) and (3.27) mean that fidelity, defined in (2.56), between the initial state at vertex a and the final state at vertex b is equal to one. In practical implementations, the perfect state transfer condition may be unreachable due to errors occurring with implementation of quantum gates. Thus, for the purpose of our simulations, we will consider a *high amplitude transfer*, where we will consider a lower bound $\lambda = 0.9$ for the transmitted amplitude as a sufficient level of a transfer [17]. The the conditions for continuous-time and discrete-time quantum walks can be rewritten as follows

$$|\langle b| e^{-iHt} |a\rangle| \geq \lambda \qquad \sum_{c,c_0} |\langle b,c| U^k |a,c_0\rangle| \geq \lambda. \tag{3.28}$$

Additionally, a graph G is periodic in case of a continuous quantum walk if for any state $|\psi\rangle$ exists time $t > 0$ so that $|\langle\psi| e^{-iHt} |\psi\rangle| = 1$ [18]. Same definition applies for discrete-time quantum walk if exists k steps after which condition $\sum_{c,c_0} |\langle\psi,c| U^k |\psi,c_0\rangle| = 1$ is fulfilled.

In recent years, there have been studied many types of graphs that would enable perfect state transfer, but simulating all of them would be above the capacity of this project, therefore we are going to restrict ourselves on the study of a perfect state transfer in n-cycle graphs. An n-cycle graph, denoted C_n, is a graph with n vertices where each vertex i is connected to vertices $(i-1)$ mod n and $(i+1)$ mod n. The evolution of a initial quantum state of the walker is given by a Hamiltonian $H = A$, where A is the adjacency matrix A. For an n-cycle graph A is an symmetric $n \times n$ matrix with entries defined as follows

$$A_{ij} = \begin{cases} 1 & \text{if vertices } i \text{ and } j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{3.29}$$

The eigenvalues λ_k of this matrix are

$$\lambda_k = 2\cos\left(\frac{2\pi k}{n}\right),\ k = 0, 1, \dots, (n-1) \tag{3.30}$$

with the corresponding eigenvectors

$$v_k = \left(1, \omega^k, \omega^{2k}, \dots, \omega^{(n-1)k}\right)^T, \tag{3.31}$$

where $\omega = e^{\frac{2\pi i}{n}}$ is the nth root of unity [19]. The time evolution operator U can then be expressed in terms of the eigenvalues and eigenvectors as

$$U = e^{-iHt} = \sum_{k=0}^{n-1} e^{-i\lambda_k t} |v_k\rangle \langle v_k| \tag{3.32}$$

and the condition (3.26) for perfect state transfer from vertex a to vertex b can be rewritten as follows

$$|\langle b|\, U\, |a\rangle\,| = \frac{1}{n}\sum_{k=0}^{n-1} e^{-i\lambda_k t}\omega^{k(b-a)}. \tag{3.33}$$

In the case of n-cycles the condition (3.33) is not satisfied by every arbitrary n, few examples where the perfect state transfer is enabled are $n = 4$, 8. Additionally, the number of steps that is required for a perfect state transfer depends not only on the size of the graph but also often on the chosen coin operator. Example given in [20] include Hadamard coin operator $\mathcal{H} = \mathcal{C}\left(\frac{\pi}{4}, 0, 0\right)$ when the perfect state transfer occurs after 4 steps.

3.7 Disorder

In the context of the quantum walks on a lattice or graph structure, disorder, manifesting as random potential fluctuations or irregularities in the lattice spacing, can lead to phenomena like particle localization at certain vertices of the graph. The disorder can be introduced through the evolution operator U of the discrete-time quantum walk, either through a position-dependent coin operator or a disorder operator.

The position-dependent coin operator C is a method in which a different coin operator is applied at each vertex. The variability in the coin operator introduces disorder into the quantum walk by altering the transition probabilities at each step depending on the particle's location. For example, consider a quantum walk on a 1-dimensional lattice with vertices labeled $n \in \mathbb{N}$. An illustrative position-dependent coin operator at vertex n could be represented by the matrix

$$C_n = \begin{pmatrix} \cos\theta_n & \sin\theta_n \\ \sin\theta_n & -\cos\theta_n \end{pmatrix}, \tag{3.34}$$

where $\theta_n \in [-\pi, \pi]$ is a random angle specific to each vertex n.

The second method involves expanding the evolution operator U with the disorder operator D as $U = D\,(S \cdot C)$, where S is the shift operator and C represents coin operator described in Sec. 3.3. At each step of the discrete-time quantum walk, the disorder operator modifies the evolution by adding random phases to the probability amplitudes of different states, altering the particle's propagation behavior. For a quantum walk on a graph with vertices labeled as $n \in \mathbb{N}$, an illustrative disorder operator D could be represented by a diagonal matrix

$$D = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} e^{i\phi_1} & 0 & 0 & \cdots & 0 \\ 0 & e^{i\phi_2} & 0 & \cdots & 0 \\ 0 & 0 & e^{i\phi_3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & e^{i\phi_n} \end{pmatrix}, \tag{3.35}$$

where each diagonal element is a phase factor $e^{i\phi_n}$ with $\phi_n \in [-\pi, \pi]$ being a random phase associated with vertex n.

The study of disorder in quantum walks can be categorized into two distinct scenarios: static disorder and dynamic disorder. In the static disorder scenario, the phase shifts introduced by the disorder operator remain constant throughout the evolution of the quantum walk. As discovered by Anderson in 1958 [21] and later experimentally verified by [22], static disorder leads to an absence of diffusion and localization of the particle's wave function as shown on Fig. 3.8, where the quantum walker on the line keeps being localized at the initial position. From a physical perspective, localization in condensed matter physics refers to the confinement of waves, such as those of electrons or photons, within a medium exhibiting potential fluctuations. In a perfectly ordered medium, waves propagate without significant scattering. However, introducing disorder, such as impurities or defects, disrupts the propagation, causing waves to become trapped in localized regions.

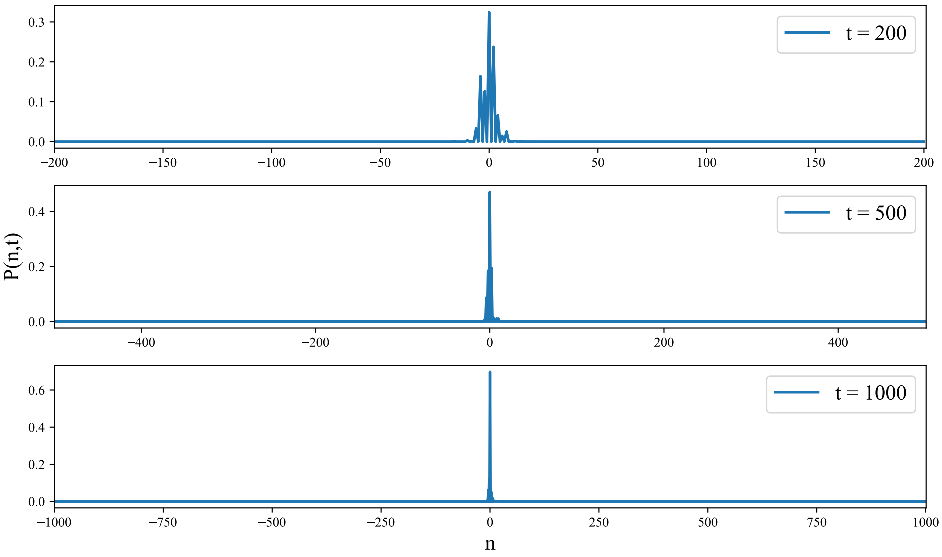


Figure 3.8: Probability distribution $P(n, t)$ for a quantum walk on a line initially localized in the origin with static disorder after $t = 200$, $t = 500$ and $t = 1000$ steps on a line with positions $n \in \mathbb{Z}$.

In the case of temporal disorder, the disorder operator is step-dependent, as the phase shifts change over time. Temporal disorder disrupts the coherence of the quantum walk by introducing randomness into the system, thus suppressing interference effects. As a result, with increasing steps, the quantum walk begins to behave more like a classical random walk. As shown in Fig. 3.9, the probability distribution exhibits a loss of ballistic peaks within a small number of steps and transitions to diffusive behavior over a larger number of steps. For a large number of steps, the shape of the probability distribution approaches a Gaussian distribution as in the classical random walk.

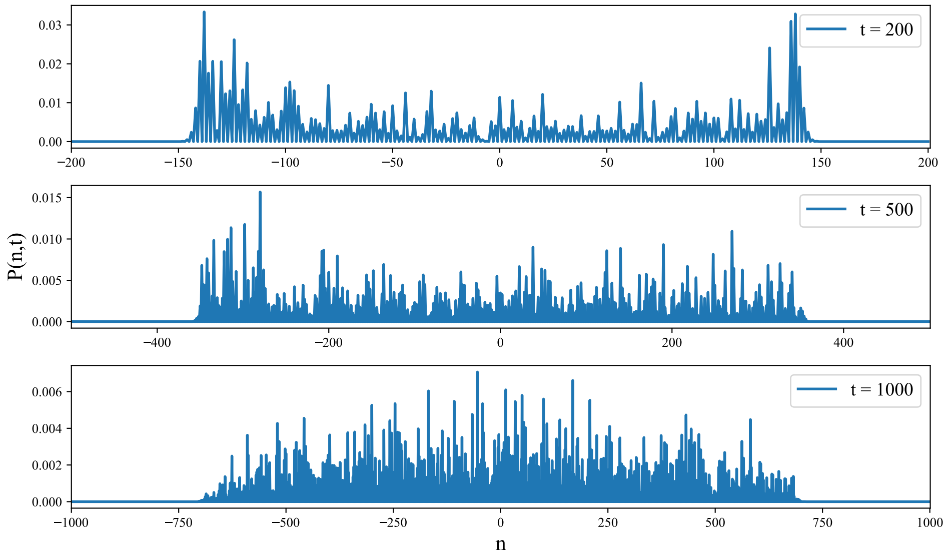


Figure 3.9: Probability distribution $P(n, t)$ for a quantum walk on a line with temporal disorder after $t = 100$, $t = 1000$ and $t = 2000$ steps on a line with positions $n \in \mathbb{Z}$. The disorder was multiplied by a factor of 0.05 to make the transition to diffusive behavior more noticeable.

Chapter 4

Simulations with a quantum computer

In this chapter, we present the results obtained from our realizations of the theoretical topics we described in previous chapters. These experiments were conducted on currently available IBM quantum computers, utilizing their advanced quantum hardware and software platforms. We also provide analysis of the data, including error rates and algorithmic efficiency.

4.1 Deutsch-Jozsa algorithm

As mentioned in Sec. 2.12.1, the Deutsch-Jozsa algorithm determines whether a given function is balanced or constant using an oracle. We realized this algorithm with a 4-qubit register for both a balanced and a constant function. The results, displayed in Fig. 4.2, convincingly demonstrate the ability to identify the type of function in both scenarios. We repeated each experiment ten times for each type of function and calculated the average and the standard deviation. The consistent detection of the function type across these simulations highlights the robustness and accuracy of the Deutsch-Jozsa algorithm when applied to a 4-qubit register.

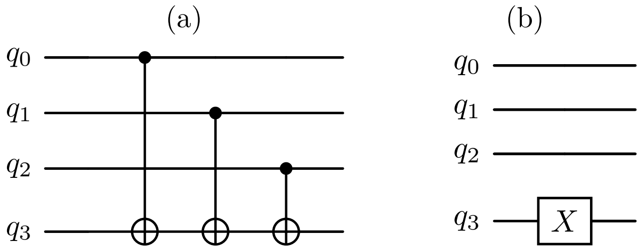


Figure 4.1: The quantum circuits for (a) balanced and (b) constant oracles.

4.2 Grover's algorithm

To verify theoretical assumptions, we realized Grover's algorithm, described in Sec. 2.12.2, on IBM Fez quantum computer. The experiment utilized an 8-qubit register and a targeted state $|101\rangle$. Based on the initial size of the

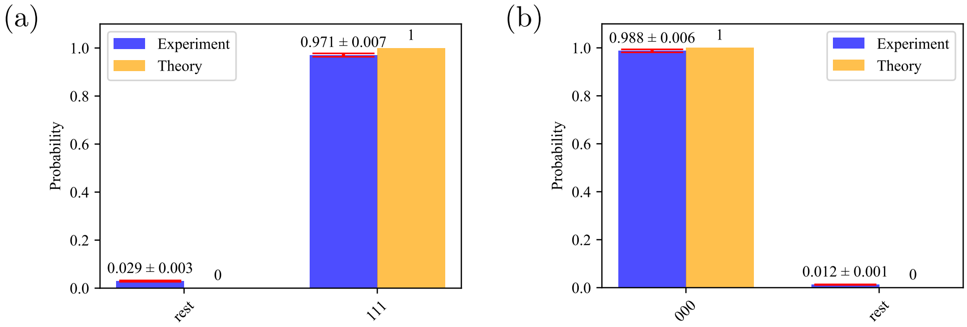


Figure 4.2: Simulation of the Deutsch-Jozsa algorithm on a 4-qubit register. (a) For a balanced function, where only the state $|111\rangle$ can be measured. (b) For a constant function, where only the state $|000\rangle$ can be measured. The simulations were performed on an IBM Fez quantum computer. The "rest" is calculated by the summing over the probabilities for all the other results.

register and the number of target states, the optimal number of iterations was calculated as $k = \lfloor \frac{\pi}{4}\sqrt{\frac{8}{1}} \rfloor = 2$. The experiment was repeated ten times, after which we calculated the average probability and standard deviation for each state.

The results shown in Fig. 4.3 confirm the theoretical prediction. After two iterations of Grover's algorithm, the probability of measuring the target state $|101\rangle$ was significantly higher than that of any other state, demonstrating the algorithm's efficiency in locating the target state within the search space.

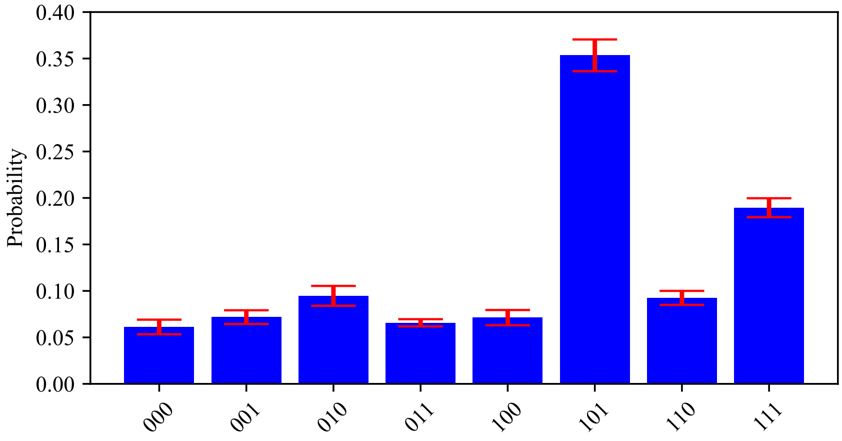


Figure 4.3: Realization of the Grover's algorithm with one target state $|101\rangle$ and size of the database $N = 8$. Simulated on the IBM Fez quantum computer.

4.3 Discrete-time quantum walks

Building on the introduction of discrete-time quantum walks presented in Chapter 3, we now present our results from the implementation of discrete-time quantum walks on IBM quantum computers. Our experiments on quantum walks will cover the time dynamics of the walk, probability distribution, perfect state transfer, and the effects of static and temporal disorder on quantum walk dynamics.

4.3.1 Probability distribution and the comparison of conventional and optimized methods

We realized discrete-time quantum walks on quantum computers using the methods described in Sec. 3.4 and Sec. 3.5. We will refer to these as the conventional and optimized methods, respectively. For both methods, we performed 10 experiments for each step, and the probability averages and their standard deviations at each position were calculated and displayed in the Fig. 4.4. We see that the probability distributions obtained by the optimized method closely match the expected theoretical probability distributions on a 4-cycle, unlike those of the conventional method. These results meet our expectations, as the optimized method significantly reduces the number of quantum gates and the depth of the circuit, as detailed in Table 4.1. We note here that the number of gates given in this table for conventional and optimized methods is much higher than what we have mentioned in Sec. 3.4 and 3.5. For example, to realize one step of the discrete-time quantum walk with optimized method, we need 30 gates, which looks contrary to what we claimed in Sec. 3.5 that we only need one CNOT gate and four single-qubit operations. The reason for this is that the four single-qubit unitary operations are decomposed into the single-qubit gates that are available in the architecture of the quantum processor. This is automatically done in an optimized way by the IBM API (Qiskit).

-	Conventional		Optimized	
Step	Gates	Depth of circuit	Gates	Depth of circuit
1	260	155	30	18
2	517	324	57	35
3	669	412	84	52
4	968	583	111	69

Table 4.1: Number of quantum gates and depth of quantum circuits for simulation of first four steps of the discrete-time quantum walk on a 4-cycle with conventional and optimized methods.

To see the success of the optimized method compared to the conventional method, we use the following measure:

$$\Sigma(t) = \sqrt{\frac{1}{N}\sum_{n}\left[P_n^{\mathrm{ex}}(t) - P_n^{\mathrm{th}}(t)\right]^2} \tag{4.1}$$

This measure averages the difference between the experimental probability values and the theoretical probability values at each position over the position. Therefore, the more the experimentally obtained probability distribution is

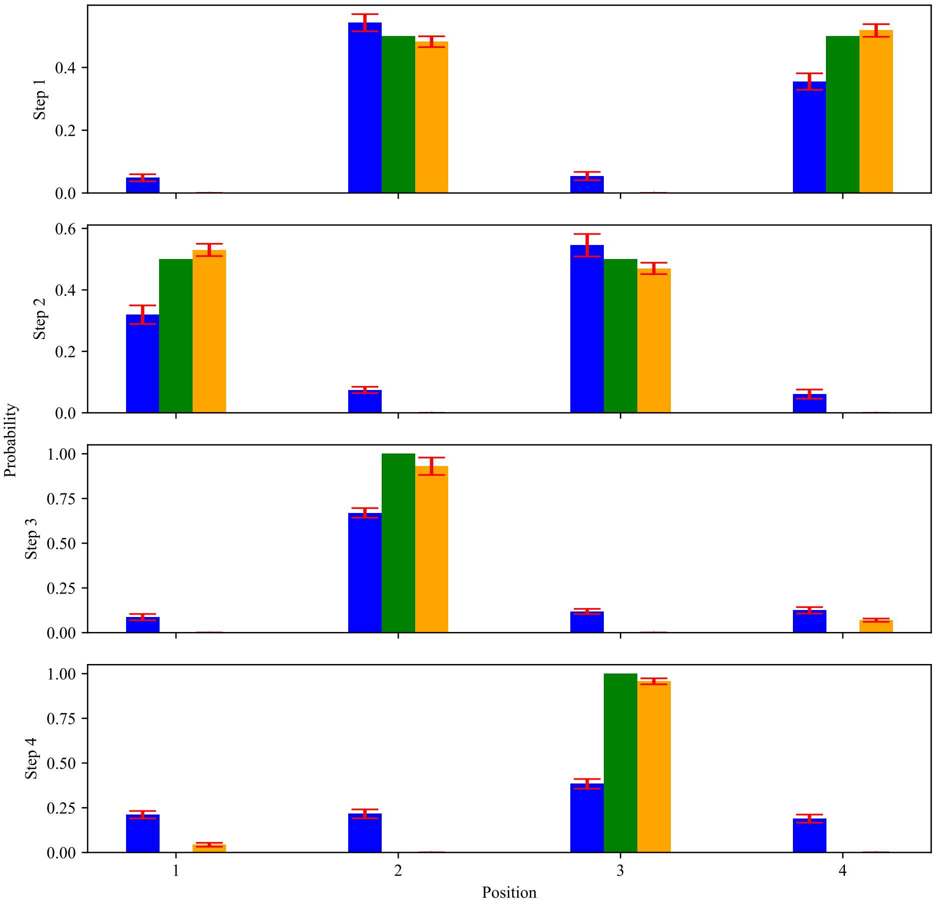


Figure 4.4: Probability distributions for the first four steps of a quantum walk on the 4-cycle with a quantum walker initially localized at position 1 with initial coin state $|0\rangle$. Blue: simulation on an IBM Torino quantum computer with conventional method, green: theoretical results, orange: simulation on IBM Torino quantum computer using the optimized formulation.

incompatible with the theoretical values, the higher the value of this measure will be. As shown in Fig. 4.5, the $\Sigma(t)$ calculated for the conventional method is generally higher and it fluctuates more compared to the optimized formulation method. This discrepancy is due to the reduced number of quantum gates required in the optimized approach, which in turn reduces the introduction of decoherence and errors. We simulated only up to 20 steps due to the limitations imposed by the number of quantum gates needed; however, additional steps would likely show an increasing trend in standard deviation due to growing decoherence, eventually stabilizing.

We clearly demonstrate that the optimized method for the n-cycles improves the accuracy of the results compared to a straightforward implementation of the quantum walk. The improvement results from a reduced number of quantum gates in the circuit, minimizing noise and errors. Consequently, this

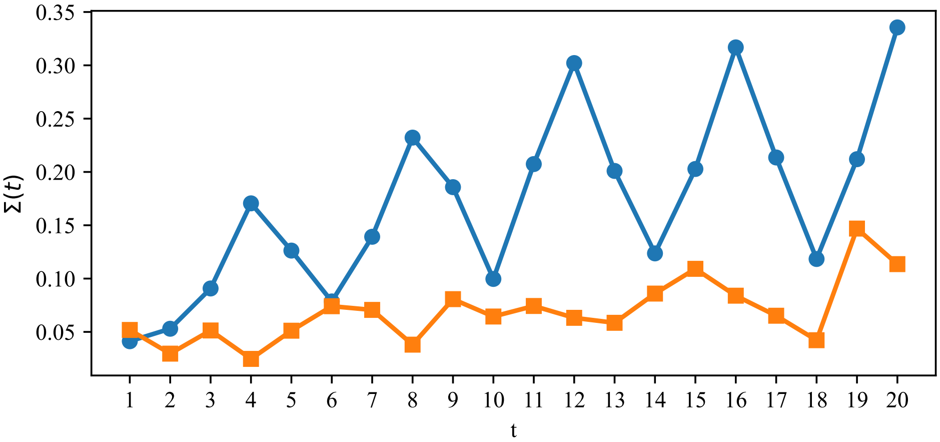


Figure 4.5: The average of the differences between experimental and theoretical results of quantum walk with a conventional (orange) and optimized method (blue) for the first 20 steps t on a 4-cycle. Simulated on IBM Torino quantum computer.

leads to more accurate measurements.

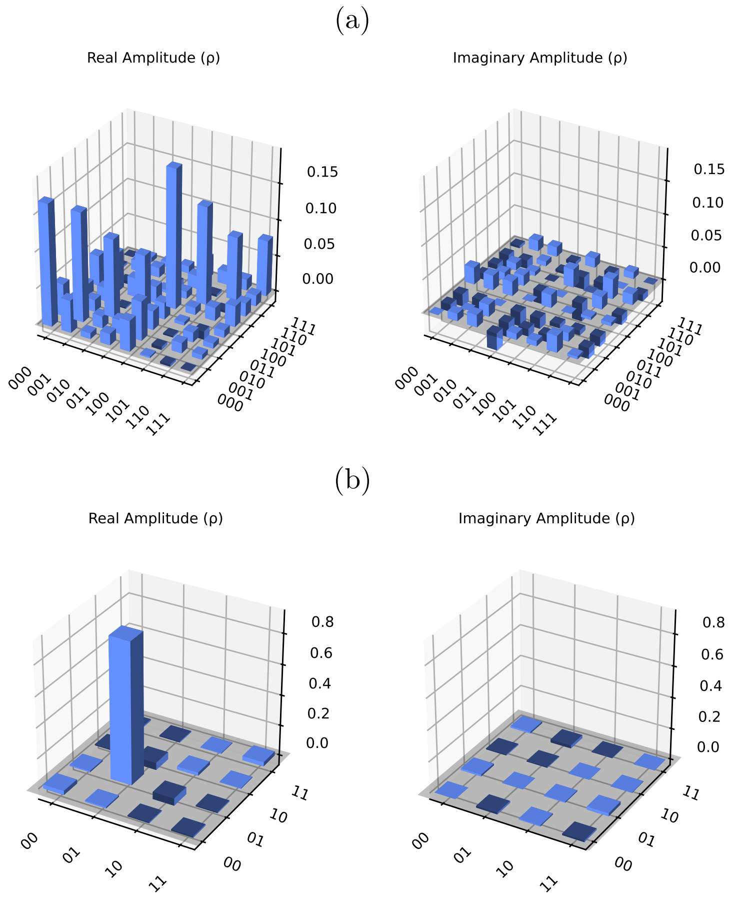
The reduction in the number of gates is also beneficial for quantum state tomography, which is crucial for reconstructing quantum states. As detailed in Sec. 2.13, the number of circuits required for tomography scales as 3^n with the number of qubits n. For a 4-cycle, this reduction means that only 9 2-qubit circuits are needed instead of 27 3-qubit circuits, enhancing the accuracy of the density matrix reconstruction and increasing the state fidelity between the measured and expected states, as illustrated in Fig. 4.6 and Tab. 4.2.

Method	Expected final state	State fidelity
Conventional	$\vert 001\rangle$	0.150
Optimized	$\vert 01\rangle$	0.925

Table 4.2: State fidelity between measured state and expected state after four steps of a discrete-time quantum walk on a 4-cycle with initial state $|000\rangle$, respectively $|00\rangle$. Simulated on IBM Sherbrooke quantum computer.

4.3.2 Perfect state transfer

To realize the perfect state transfer on a quantum computer for n-cycle, we considered an example given in [20]. This example involves a 4-step quantum walk on a 4-cycle with a Hadamard coin operator. To minimize potential noise from quantum gates implemented on a real device, we utilized an optimized method of discrete-time quantum walks on cycles. The quantum walk was repeated nine times to prepare sufficient copies of the quantum circuit for a quantum state tomography with three qubits, from which we obtained density matrices for each initial coin state. The results in Table 4.3 show


(a)

(b)

Figure 4.6: Real and imaginary amplitudes of density matrices of a quantum walk on a 4-cycle after four steps with expected final position: (a) $|001\rangle$ for a conventional approach, (b) $|01\rangle$ for a reduced Hilbert space approach.

that state fidelity between the obtained density matrices shown in Fig. 4.7 and theoretical density matrices satisfy the high amplitude transfer condition (3.28) for any initial coin state. This proves perfect state transfer on a 4-cycle after 4 steps.

Initial coin state	Expected final state	State fidelity
$\|0\rangle$	$\|01\rangle$	0.925
$\|1\rangle$	$\|11\rangle$	0.952
$\frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$	$\frac{1}{\sqrt{2}}(\|01\rangle + \|11\rangle)$	0.961
$\frac{1}{\sqrt{2}}(\|0\rangle + i\|1\rangle)$	$\frac{1}{\sqrt{2}}(\|01\rangle + i\|11\rangle)$	0.956

Table 4.3: Results of the quantum state tomography for the perfect state transfer on the 4-cycle after four steps with initial positional state $|0\rangle$ and Hadamard coin operator. State fidelity, defined in (2.56) shows overlap between expected final state and state measured on a IBM Sherbrooke quantum computer.

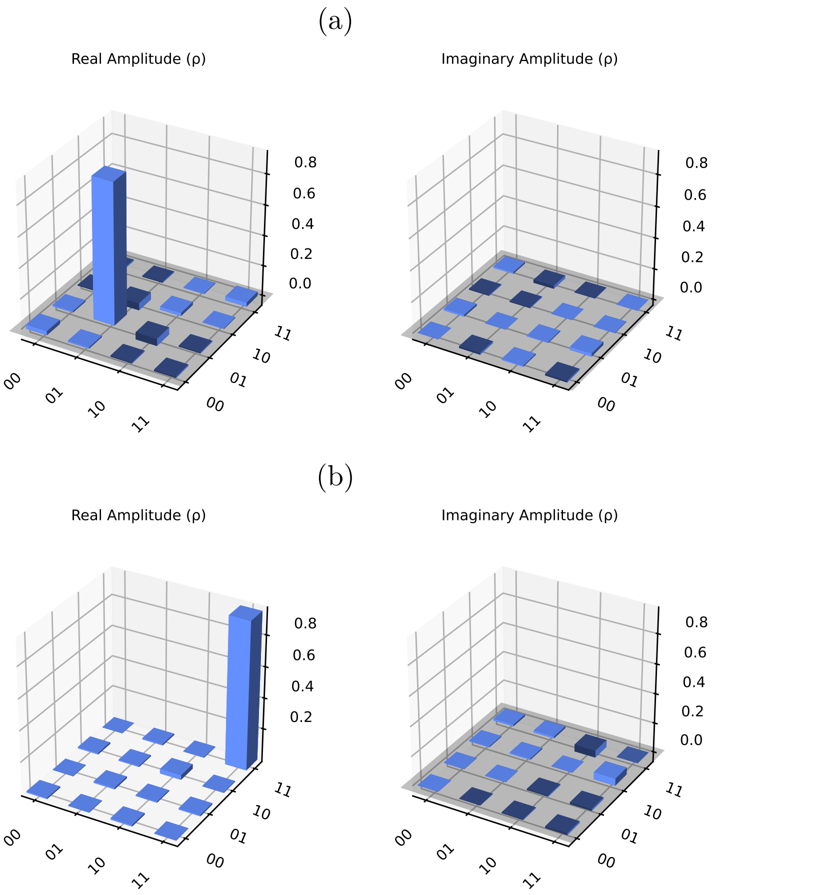
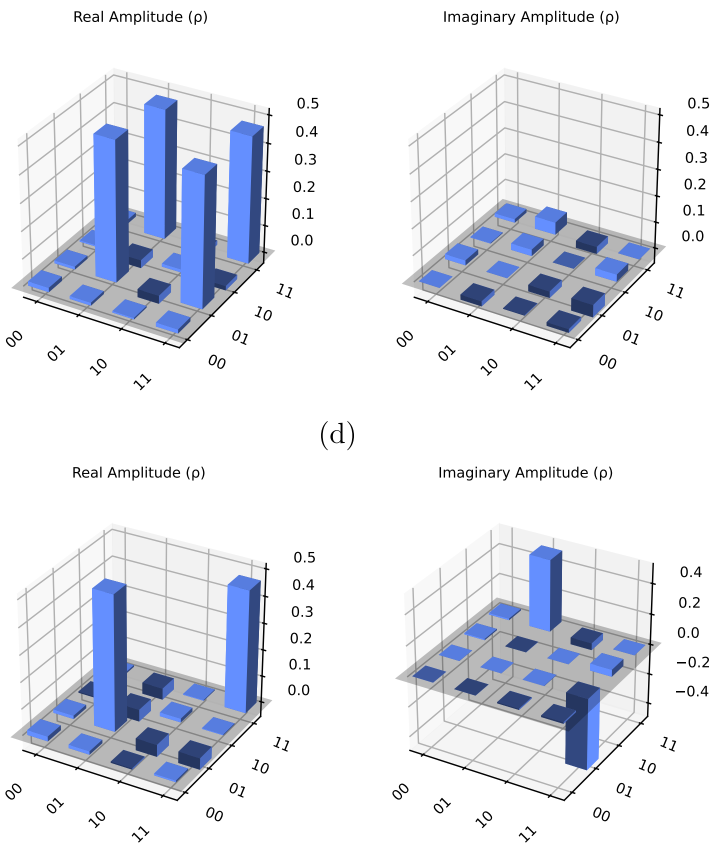


Figure 4.7: Real and imaginary amplitudes of density matrices for a discrete-time quantum walk on a 4-cycle with initial positional state $|0\rangle$. The initial coin state was chosen as: (a) $|0\rangle$, (b) $|1\rangle$, (c) $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and (d) $\frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$. Simulations were conducted on IBM Sherbrooke quantum computer.



(c)

(d)

Figure 4.7: (continued) Real and imaginary amplitudes of density matrices for a discrete-time quantum walk on a 4-cycle with initial positional state $|0\rangle$. The initial coin state was chosen as: a) $|0\rangle$, b) $|1\rangle$, c) $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and d) $\frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle)$. Simulations were conducted on IBM Sherbrooke quantum computer.

4.3.3 Disorder

In Sec. 3.7, we described two possible types of disorder in quantum walks: static and temporal. To effectively simulate both scenarios on 16-cycles, we utilized optimized formulation of discrete-time quantum walks on cycles, introducing disorder via a disorder operator at each step of the quantum walk. In both scenarios, the experiment was repeated forty times, with new random disorder values in range $[-\pi, \pi]$ generated each times. From the experimental results averages and standard deviations at each position were calculated and later presented in graphs.

In the case of static disorder, the same disorder values used for the first step of the quantum walk were utilized for all following steps. The results were measured after six steps of the discrete-time quantum walk on a 16-cycle with initial position at position 9. As shown in Fig. 4.8, the probability distribution from the simulation shows an expected peak at the initial position

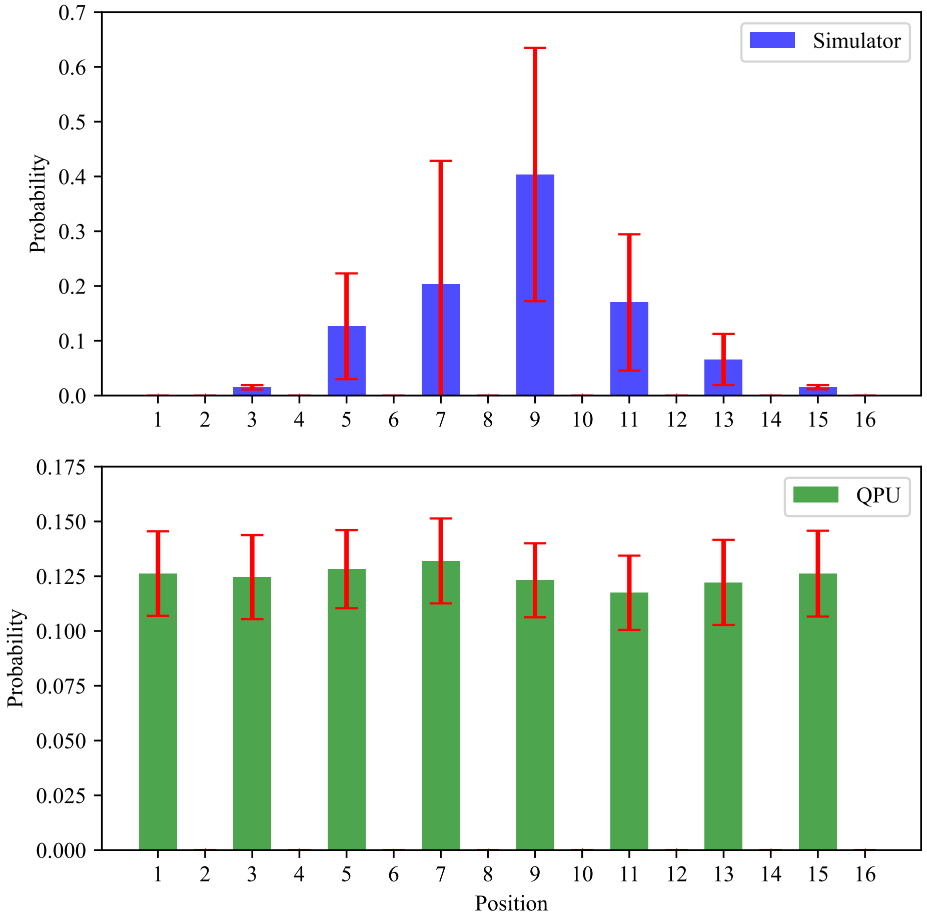


Figure 4.8: Probability distribution of a discrete-time quantum walk on a 16-cycle after 6 steps with static disorder introduced at each position via disorder operator. The initial position was at 9. Simulation was conducted on a simulator of a quantum computer and an IBM Fez quantum computer.

followed by exponential decline on both sides, proving the localization of the quantum walker. However, the results from the IBM Fez quantum computer show no localization due to strong noise effects, that lead to measurements with no outlier value.

In the case of temporal disorder, random disorder values were generated for each step. After measuring the discrete-time of the quantum walk on 16-cycle with initial position at 9, we expected a Gaussian probability distribution centered at position 9, as shown by the simulator results in Fig. 4.9. However, for the same reasons as in the static disorder scenario, the simulation on the quantum computer yielded measurements heavily affected by noise.

To improve the quality of simulations with disorder, we need to implement a solution with fewer quantum gates and lower circuit depth[1] to reduce the

[1] Circuit depth is the maximum number of sequential operations on a qubit within a quantum circuit, indicating its complexity and execution time.

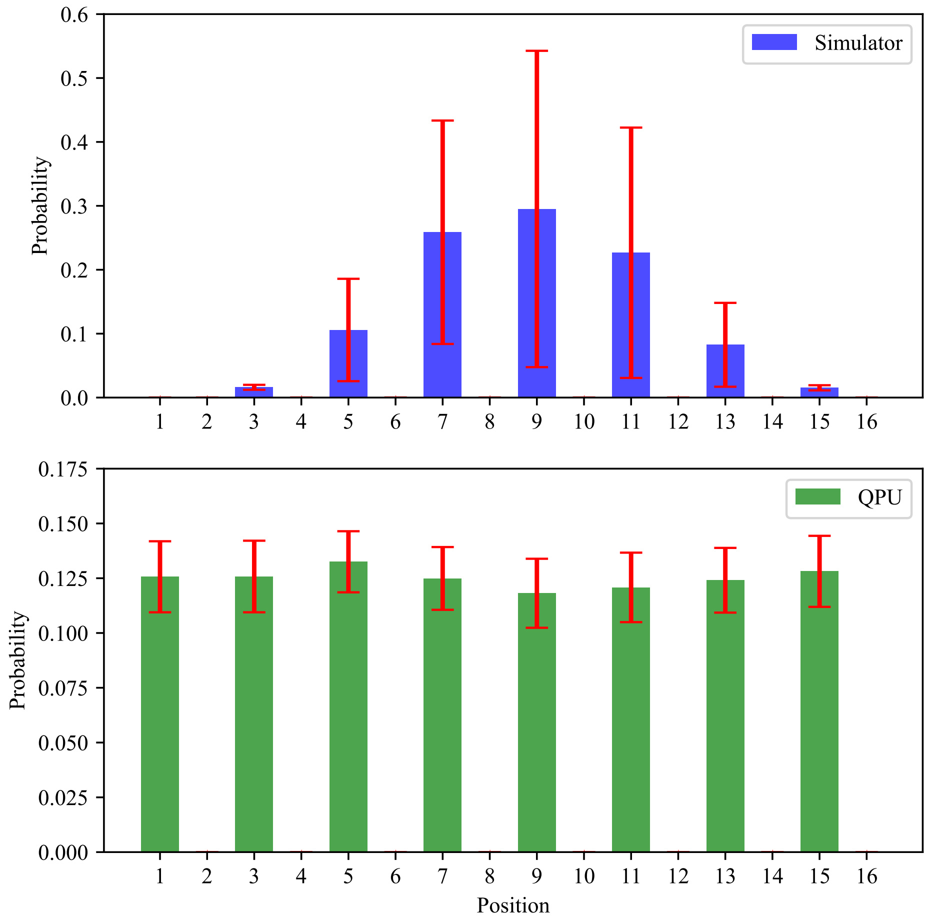


Figure 4.9: Probability distribution of a discrete-time quantum walk on a 16-cycle after 6 steps with temporal disorder introduced at each position via disorder operator. The initial position was at 9. Simulation was conducted on a simulator of a quantum computer and an IBM Fez quantum computer.

potential noise sources. We considered two possible solutions for this issue: reducing the number of positions in the n-cycle (making the cycle smaller) or introducing disorder at only one position.

Using the optimized formulation method, the first solution resulted in insufficient data to prove localization. For an 8-cycle, we obtained only 4 nonzero values, and for a 4-cycle, only 2 nonzero values. The second solution, introducing disorder at only one position, did not reduce the number of required quantum gates from the implementation of the disorder operator. Although a less complex disorder operator could be used, the disorder operator itself represents only a small portion of the implemented code. Most of the quantum gates come from the implementation of the shift operator matrices, so this solution did not yield any improvements.

Overall, we did not develop sufficient improvements in the implementation of discrete-time quantum walks with disorder on quantum computers that would be error-tolerant enough to show significant results on current IBM quantum computers.

Chapter 5

Conclusion

In this project, we explore both the theoretical foundations and main principles of quantum computing. Through simulations on currently available quantum computers, we investigated the current possibilities for realizing some quantum algorithms and quantum walks. This work focuses particularly on the simulation of quantum walks and related phenomena, such as perfect state transfer and the effects of disorder in quantum walks.

Chapter 2 provides a broad introduction to quantum computing, starting with the notation, the basics of quantum mechanics, and a description of the commonly used quantum gates in quantum circuits. The chapter then describes historically significant algorithms, namely the Deutsch-Jozsa algorithm and Grover's algorithm. It also covers essential concepts for a basic theoretical understanding of quantum computing, such as quantum state tomography and universal quantum simulations.

Chapter 3 is dedicated to the description and simulation of quantum walks. It starts by discussing random walks as a classical predecessor to quantum walks and proceeds to describe both discrete-time and continuous-time quantum walks. The chapter also covers the phenomenon of perfect state transfer and the impact of disorder in quantum walks.

In chapter 4 we simulated on real quantum computers all the concepts described in the previous chapters. We compared our results for discrete-time quantum walks on n-cycles realized with conventional and optimized methods by examining the probability distributions, number of gates, depth of circuits, deviations from expected theoretical values ($\Sigma(t)$) and state fidelities via quantum state tomography. Based on our results, we suggest that the optimized method may be beneficial in noise reduction in future applications of quantum walks. Although the optimized method is more efficient in the early steps of the quantum walk, requiring fewer resources to build quantum circuits, current hardware capabilities are still insufficient for implementing longer quantum walks. However, future improvements in reducing decoherence in quantum computers may still take advantage of the optimized method to realize quantum walks more effective on a larger scale.

Building on these findings, we used the optimized method to simulate perfect state transfer and the effects of disorder in quantum walks. For perfect state transfer, we verified the transfer of a quantum walker on a

4-cycle after four steps with a Hadamard coin, as demonstrated by [20]. We achieved high state fidelity (over 90%) between the expected final state and the state measured with quantum state tomography for four different initial coin states: $\left(\left|0\right\rangle, \left|1\right\rangle, \frac{1}{\sqrt{2}}\left(\left|0\right\rangle + \left|1\right\rangle\right), \frac{1}{\sqrt{2}}\left(\left|0\right\rangle + i\left|1\right\rangle\right)\right)$.

Finally, we simulated a discrete-time quantum walk with disorder on a quantum computer in the 16-cycle. However, due to the noise introduced by the quantum computer after implementing several quantum gates, we were unable to demonstrate the expected localization due to static disorder, as observed in the experimental results by [22], or the expected Gaussian probability distribution due to temporal disorder.

For future work, it may be beneficial to explore quantum algorithms based on quantum walks that optimize the number of required quantum gates and better control the occurrence of errors. This approach could further enhance the simulation of complex physical systems.

Bibliography

[1] David Deutsch. "Quantum theory, the Church–Turing principle and the universal quantum computer". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400 (1985), pp. 117–97. URL: https://api.semanticscholar.org/CorpusID:1438116.

[2] David Deutsch and Richard Jozsa. "Rapid solution of problems by quantum computation". In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439.1907 (1992), pp. 553–558. DOI: 10.1098/rspa.1992.0167. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.1992.0167. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1992.0167.

[3] Peter W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science* (1994), pp. 124–134. URL: https://api.semanticscholar.org/CorpusID:15291489.

[4] Lov K. Grover. *A fast quantum mechanical algorithm for database search*. 1996. arXiv: quant-ph/9605043 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/9605043.

[5] P. A. M. Dirac. "A new notation for quantum mechanics". In: *Proceedings of the Cambridge Philosophical Society* 35.3 (Jan. 1939), p. 416. DOI: 10.1017/S0305004100021162.

[6] A. Einstein, B. Podolsky, and N. Rosen. "Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?" In: *Phys. Rev.* 47 (10 May 1935), pp. 777–780. DOI: 10.1103/PhysRev.47.777. URL: https://link.aps.org/doi/10.1103/PhysRev.47.777.

[7] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[8] Michele Mosca and Priyanka Mukhopadhyay. "A polynomial time and space heuristic algorithm for T-count". In: *Quantum Science and Technology* 7.1 (Oct. 2021), p. 015003. ISSN: 2058-9565. DOI: 10.1088/

2058-9565/ac2d3a. URL: http://dx.doi.org/10.1088/2058-9565/ac2d3a.

[9] Niklas Johansson and Jan-Åke Larsson. “Efficient classical simulation of the Deutsch–Jozsa and Simon’s algorithms”. In: *Quantum Information Processing* 16.9 (Aug. 2017). ISSN: 1573-1332. DOI: 10.1007/s11128-017-1679-7. URL: http://dx.doi.org/10.1007/s11128-017-1679-7.

[10] Matteo G. A. Paris and Jaroslav Rehacek. “Qubit Quantum State Tomography”. In: *Quantum State Estimation*. Ed. by Matteo G. A. Paris and Jaroslav Rehacek. Springer, 2004, pp. 235–272. DOI: 10.1007/978-3-540-44481-7_12.

[11] Richard P. Feynman. “Simulating physics with computers”. In: *International Journal of Theoretical Physics* 21.6 (1982), pp. 467–488. ISSN: 1572-9575. DOI: 10.1007/BF02650179. URL: https://doi.org/10.1007/BF02650179.

[12] Salvador Elías Venegas-Andraca. “Quantum walks: a comprehensive review”. In: *Quantum Information Processing* 11.5 (2012), pp. 1015–1106.

[13] Andrew M. Childs, Edward Farhi, and Sam Gutmann. “An Example of the Difference Between Quantum and Classical Random Walks”. In: *Quantum Information Processing* 1/2 (2002), pp. 35–43. ISSN: 1570-0755. DOI: 10.1023/a:1019609420309. URL: http://dx.doi.org/10.1023/A:1019609420309.

[14] Sougato Bose. “Quantum Communication through an Unmodulated Spin Chain”. In: *Phys. Rev. Lett.* 91 (20 Nov. 2003), p. 207901. DOI: 10.1103/PhysRevLett.91.207901. URL: https://link.aps.org/doi/10.1103/PhysRevLett.91.207901.

[15] Andrew M. Childs. “Universal Computation by Quantum Walk”. In: *Physical Review Letters* 102.18 (May 2009). ISSN: 1079-7114. DOI: 10.1103/physrevlett.102.180501. URL: http://dx.doi.org/10.1103/PhysRevLett.102.180501.

[16] Neil B. Lovett et al. “Universal quantum computation using the discrete-time quantum walk”. In: *Physical Review A* 81.4 (Apr. 2010). ISSN: 1094-1622. DOI: 10.1103/physreva.81.042330. URL: http://dx.doi.org/10.1103/PhysRevA.81.042330.

[17] K. Barr et al. *Periodicity and perfect state transfer in quantum walks on variants of cycles*. 2013. arXiv: 1204.5937.

[18] Vivien Kendon and Christino Tamon. “Perfect State Transfer in Quantum Walks on Graphs”. In: *Journal of Computational and Theoretical Nanoscience* 8 (Apr. 2010). DOI: 10.1166/jctn.2011.1706.

[19] Steve Butler. “Eigenvalues and structures of graphs”. In: 2008. URL: https://api.semanticscholar.org/CorpusID:118295461.

[20] İskender Yalçınkaya. “Spreading and transport properties of quantum walks”. PhD thesis. Sabanci University, Istanbul, 2016.

[21] P. W. Anderson. “Absence of Diffusion in Certain Random Lattices”. In: *Phys. Rev.* 109 (5 Mar. 1958), pp. 1492–1505. DOI: 10.1103/PhysRev.109.1492. URL: https://link.aps.org/doi/10.1103/PhysRev.109.1492.

[22] A. Schreiber et al. “Decoherence and Disorder in Quantum Walks: From Ballistic Spread to Localization”. In: *Physical Review Letters* 106.18 (May 2011). ISSN: 1079-7114. DOI: 10.1103/physrevlett.106.180403. URL: http://dx.doi.org/10.1103/PhysRevLett.106.180403.