$\mathcal{Q}$uantum Computing and $\mathcal{Q}$uantum Information

A Comprehensive Note

A Symphony of Theoretical Computer Science and Quantum Physics

by
Padmapriya S and Nishkal Rao

IISER Pune
BS-MS

July 2025

To our parents and teachers

Preface

Teaching is the best form of learning, and when passion meets perseverance, surprising results occur. This book is one such result of a passion project in the pursuit of knowledge and love for teaching.

There are many quantum computing and quantum information textbooks, covering the vast and deep aspects of this subject and its intricate interdisciplinary nature. It is easy for a first-time learner to get lost in this ocean of knowledge. So did we when we started learning this beautiful subject. As an attempt to navigate through this ocean, we decided to curate and bring together the vast topics in an easy-to-understand and concise form, leading to the writing of this book. Additionally, the Quantum Information and Computing course at our institute was a primary factor that intrigued and motivated us to write this text. As students and budding researchers ourselves, we understand the struggles and natural questions that arise while learning this subject. In this book, we have made our best attempts to clarify these subtle details and give a flavour of this interdisciplinary subject to anyone irrespective of their background, provided the reader is comfortable with elementary linear algebra and high school mathematics. This has been a series of motivated efforts in bringing out an extremely comprehensive review of necessary and relevant topics in the broad field of quantum computing and quantum information, in order to provide a head-start to early-career researchers, and to attempt to tackle foundational problems in the field.

This book requires dedicated reading and is not meant as a casual introduction to quantum computing and information. It is ideally suited for undergraduates, graduates, and probably motivated high schoolers who want to gain an in-depth yet first-level exposure to the broad field of quantum computing and quantum information theory. This book does not elaborate on experimental aspects and technological developments of the field. We hope that after reading this text, one will be able to delve deeper into any particular topic in quantum computing and quantum information.

Part I gives an overview of how truly interdisciplinary the field of quantum computing and quantum information is, with topics spanning from pure mathematics, theoretical physics and computer science. The first chapter is intended to be a quick recap of all the mathematical tools required to understand this textbook. The reader is expected to already know most of these, especially matrix and linear algebra. If not already familiar, there are many amazing and standard resources available to learn these topics. (Refer to Linear algebra done right by Sheldon Axler, or the lectures by Gilbert Strang)

The second chapter sets the stage and provides the necessary background for readers from mathematics, computer science, engineering or any other background who are not familiar with quantum physics. If the reader is already familiar with undergrad-level abstract algebra and quantum mechanics, then they can skip the first two chapters of Part I of this textbook. However, reading these chapters may provide a quick revision and also present these concepts through a new lens.

The third chapter provides the necessary theoretical computer science background, covering topics primarily in computational complexity theory. Even if the reader is familiar with complexity theory, we encourage them to take a look at this chapter as it touches upon quantum complexity theory alongside the relatively familiar classical complexity theory.

The final chapter of Part I introduces qubits, the fundamental unit of quantum information, serving as the quantum analogue of the classical bit. It lays out the essential concepts and overarching themes of quantum computing and information from functional and historical perspectives. With this foundation, readers are free to explore any chapter from Parts II and III in any order they prefer, as each chapter is self-contained and independent, rather than hierarchically structured. This makes the book equally suitable for readers interested in gaining a broad overview, focusing on a specific topic within quantum computing and quantum information.

For readers who wish to learn independently and work through the entire textbook, we recommend following the chapters in the order presented.

Part II of the textbook covers topics in quantum computing from basic quantum algorithms to sophisticated algorithms like Shor's prime factoring algorithm that can break the classically secure RSA cryptosystems and Grover's search algorithm that can search in an unstructured database faster than its classical counterpart.

Part III, the last part of the textbook, talks about quantum information, including topics from quantum error correction. Both Part II and Part III have a lot of visual elements, presenting each topic in an intuitive and pedagogical form. We have made an effort to naturally build the concepts from the ground up rather than directly presenting them. Throughout these two parts, wherever necessary, we have drawn detailed parallels to classical computer science to appreciate the similarities and differences in both these worlds.

Although we have not included exercises, this book offers a concise yet substantial foundation for anyone seeking to build a strong theoretical understanding in a relatively short read. Readers can use it to acquire the necessary background and then practice with problems from other well-known texts on quantum computing and information. Beyond self-study, this book also serves as an ideal companion for any quantum computing or information course. As part of a course throughout the semester, we managed to cover the various aspects that were taught to us, and presented newer insights and perspectives that we believe would help grasp some of the intricacies and inner understandings that make this field extremely interesting.

We have further provided additional references through footnotes, leading to research material, for the interested reader. We have built some examples through boxes, where we provide a natural, geometric and intuitive visualisation of some concepts. Additionally, we ensured to have all illustrations generated through LaTeX, to ensure that we can convey information through maximum flexibility, and enhance the readability of the book.

Despite having revised and refined this text multiple times, there is always room for improvement. We welcome your feedback. Please feel free to contact the authors with any suggestions or report any errors you may find. We plan to continue updating this textbook, and the latest version will always be available on the website: https://o-qcblog.github.io/QIQC/.

Happy learning!

$$\frac{1}{\sqrt{2}}\Big[\,|\mathscr{N}\text{ishkal }\mathscr{R}\text{ao}\rangle \oplus |\mathscr{P}\text{admapriya }\mathscr{S}\rangle\Big]$$

Acknowledgments

The authors express gratitude to Dr. Sreejith GJ, Department of Physics, IISER Pune, for his exceptional teaching in the Quantum Information and Computing course, which inspired us to write this book. Also, thanks to Dr. T.S. Mahesh, Chair of Physics at IISER Pune, and Dr. Anirban Hazra, Department of Chemistry, IISER Pune, whose courses helped give a holistic view of the subject and related domains, and indirectly added value to this book.

Padmapriya wishes to thank Dr. Rajat Mittal from CSE, IIT Kanpur, Dr. M.S. Santhanam, Department of Physics, IISER Pune, and Dr. Soumen Maity, Chair of Mathematics at IISER Pune, for their support in encouraging research projects on quantum computing and related areas of theoretical computer science.

Additionally, this book would not have been possible without the wonderful support of our parents, siblings, colleagues, and friends. Finally, the authors want to thank each other for the incredible fun they had during the journey of writing this book.

Navigation

Among the excellent resources for Quantum Computing and Quantum Information, here are some introductory resources that we have referred to and been inspired by.

- Quantum Computation and Quantum Information, Textbook by Isaac Chuang and Michael Nielsen
  A foundational introduction to key concepts in quantum computing, highlighting notable aspects of quantum algorithms, quantum information, and quantum error correction.

- Quantum Information and. Computation., Lecture Notes for Physics 229 by John Preskill
  A comprehensive and insightful resource for understanding quantum computing and quantum information theory.

- Principles of Quantum Computation And Information; Textbook by Giuliano Benenti, Giuliano Strini, Giulio Casati
  Comprehensive two-volume companion designed to enhance understanding of quantum computing through clear pedagogical insights and engaging problems.

- Quantum Computer Science: An Introduction; Textbook by David Mermin
  Beautiful introduction to various aspects of quantum computing. Beware of the QBits, though!

- Quantum Computing Since Democritus; Textbook by Scott Aaronson
  A philosophical understanding of quantum computing based on complexity, offering valuable insights into physics, mathematics, and theoretical computer science.

- Dancing with Qubits: How Quantum Computing Works and how it Can Change the World; Textbook by Robert S. Sutor
  Modern introduction to the concepts in quantum computing and the engineering aspects of the physical theory.

- Quantum Computing, Lecture Notes by Rajat Mittal
  A concise theoretical computer science and mathematical perspective on quantum computation targeted at an audience lacking a physics background.

Contents

Conventions

As mathematicians and physicists work with concepts, we need a concise way of conveying what they mean. Good notation can make a statement or a proof much clearer and more insightful to the reader. Over time, the symbols and expressions that prove to be most useful win out while the others fade away into the archives. In the case of Dirac's bra-ket notation, it has become ubiquitous across quantum mechanics and now quantum computing.

Dirac Notation

Vectors can come in many flavours, as $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$ as a tuple (useful for our computer scientists), equivalently, $v = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$ as a row vector, $v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$ as a column vector (daunting form in physics textbooks). While quantum mechanics is centrally captured by linear algebra, we would be needing extensive use of vectors. To add on to the flavours, let us introduce two more invented by Paul Dirac[1], a theoretical physicist, that we proceed to use extensively further.

Given a vector $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$, we denote by $\langle v|$, the *bra-v*, is defined as

$$\langle v| = \begin{bmatrix} v_1^* & v_2^* & \cdots & v_n^* \end{bmatrix}$$

where we take the complex conjugate of each entry. For a vector $\boldsymbol{w} = (w_1, w_2, \ldots, w_m)$, we have the *ket-$w$*, given by

$$|w\rangle = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}$$

as the column vector without conjugations.

[1] If you read Dirac's *Principles of Quantum Mechanics*, he says to assume the correspondence between a ket and the corresponding bra, which is actually a central area of study called the Riesz Representation Theorem, which Dirac assumed as obvious to the readers. He has no mention of it in his book, and we will respect his legacy, by doing the same. That being said, this is the same man who remained completely silent after a student said, "I don't understand the second equation," during a lecture. After being asked why Dirac didn't answer the student's question, Dirac said, "That was not a question, that was a statement." The interested reader can refer to `https://math.stackexchange.com/a/3670861` for further insight.

Inner Product

When $n = m$ for same dimensions, we can conjunct the notation for the *inner product* of the vectors $\boldsymbol{v}$ and $\boldsymbol{w}$, as

$$\langle v|w\rangle = \begin{bmatrix} v_1^* & v_2^* & \cdots & v_n^* \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} = v_1^* w_1 + v_2^* w_2 + \ldots v_n^* w_n$$

which will be very helpful in further discussions. The norm of a vector denoting its length can be seen, thereby as

$$||\boldsymbol{v}|| = \sqrt{\langle v|v\rangle} = \sqrt{|v_1|^2 + |v_2|^2 + \cdots + |v_n|^2}$$

This is why we have the complex conjugates, so that complex numbers can give the norm.

Outer Product

Further, we will be requiring the notion of the *outer product* wherein we have the operation,

$$|w\rangle\langle v| = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} \begin{bmatrix} v_1^* & v_2^* & \cdots & v_n^* \end{bmatrix} = \begin{bmatrix} w_1 v_1^* & w_1 v_2^* & \cdots & w_1 v_n^* \\ w_2 v_1^* & w_2 v_2^* & \cdots & w_2 v_n^* \\ \vdots & \vdots & \ddots & \vdots \\ w_n v_1^* & w_n v_2^* & \cdots & w_n v_n^* \end{bmatrix}$$

Asymptotic Notation

Asymptotic notation is used to compare and understand the behavior of real-valued functions having a positive integer domain as the input grows large. We will be using it extensively to compare and analyze the scale of algorithms in the classical and quantum settings to help us demarcate the separation and understand the quantum advantage.

Consider two functions $f(x)$ and $g(x)$ that map positive integers to positive real numbers. Then the asymptotic notations are defined as follows:

Big $\mathcal{O}$-notation

$f(x)$ is said to be big-$\mathcal{O}$ of g, denoted as $\mathcal{O}(g(x))$ if there exists a constant $c > 0$ and there exists a positive integer constant x_0 such that,

$$\begin{gathered} g(x) \geq f(x) + c \;\; \forall x \geq x_0 \\ \implies f(x) = \mathcal{O}(g(x) \end{gathered}$$

Big Ω-notation

$f(x)$ is said to be big-Omega of g, denoted as $\Omega(g(x))$ if there exists a constant $c > 0$ and there exists a positive integer constant x_0 such that,

$$\begin{gathered} g(x) \leq f(x) + c \;\; \forall x \geq x_0 \\ \implies f(x) = \Omega(g(x) \end{gathered}$$

Θ-notation

Θ-notation gives a tight bound. We say $f(x) = \Theta(g(x))$ when $f(x)$ is both $\Omega(g(x))$ and $O(g(x))$.

Õ-notation

$\tilde{O}$-notation is used to hide the logarithmic factors, that is, if $f(x) = \tilde{O}(g(x))$ implies $f(x) = (\log x)^c g(x)$, where c can be any real number.

Little o-notation

$f(x)$ is said to be little-o of g, denoted as $o(g(x))$ if for any constant $c > 0$, there exists a positive integer constant x_0 such that,

$$\begin{gathered} g(x) > f(x) + c \;\; \forall x \geq x_0, \forall c > 0 \\ \implies f(x) = o(g(x) \end{gathered}$$

Little-o is used to denote a looser upper bound to a function compared to big-O. Also note that the greater than or equal to is replaced with strictly greater than.

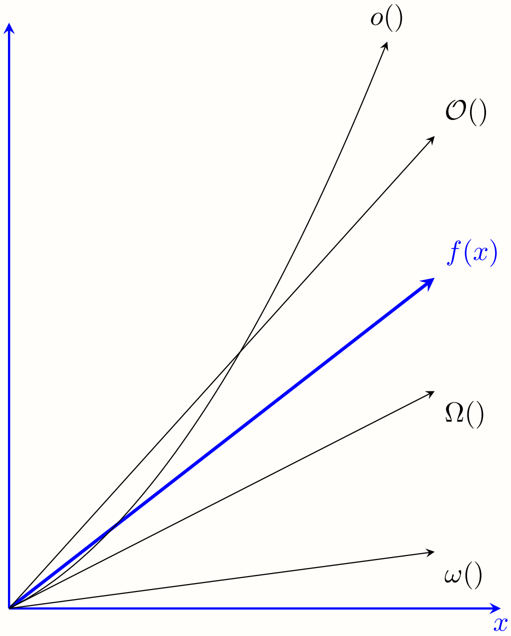


Figure 1: Asymptotic Notation

Little ω-notation

$f(x)$ is said to be little-omega of g, denoted as $\omega(g(x))$ if for any constant $c > 0$, there exists a positive integer constant x_0 such that,

$$g(x) < f(x) + c \quad \forall x \geq x_0, \forall c > 0$$
$$\implies f(x) = \omega(g(x)$$

Little-omega is used to denote a looser lower bound to a function compared to big-Omega. Also note that the less than or equal to is replaced with strictly less than.

Part I

Foundations

Chapter 1

Mathematical Background

“If all of mathematics disappeared, physics would be set back by exactly one week.”
– Richard Feynman, *Lecture at Caltech, Pasadena*

“Precisely the week in which God created the world.”
– Mark Kac, *Enigmas of Chance*

1.1 Probability Theory

At its core, quantum mechanics is a probabilistic theory[1], meaning that it predicts the likelihood of different outcomes for a given measurement. The interpretation of these probabilities has been a subject of debate among physicists and philosophers for decades. We provide with a concise recap of basic probability theory to address problems in quantum mechanics further:

1. *Conditional probability:* Let A and B be two events

$$P[A|B] := \frac{P(A \cap B)}{P(B)}$$

2. *Partition formula:* Given A and disjoint partition $B_1, B_2 \dots B_m$ of sample space,

$$P(A) = \sum_{i=1}^{m} P(B_i) P[A|B_i]$$

3. *Bayes rule:*

$$P[A|B] = \frac{P[B|A]P(A)}{P(B)}$$

[1] Refer for https://www.quantamagazine.org/where-quantum-probability-comes-from-20190909/ a very interesting insight

4. *Random variable:* Given sample space Ω of an experiment, a random variable is a function $X : \Omega \to \mathbb{R}$. In general, the range of a random variable X need not be real; it could be any other set with more structure (like real numbers are *ordered*; they can be added, multiplied, etc.)

5. *Probability mass function:* Given a probability function P on Ω, it can be naturally extended to the probability of the random variable,
$$P_X(x) := P(X = x) = \sum_{w:X(w)=x} P(w)$$
This is the *probability mass function* of a random variable. The *joint probability mass function* is defined to be $P_{X,Y}(x, y) := P(X = x \text{ and } Y = y)$

6. *Expectation* $E[X] := \sum_{x \in R} P(X(w) = x)x$ where R is the range of the random variable X. The expectation is linear, that is $E[aX + bY] = aE[X] + bE[Y]$.

7. *Variance:* $Var[X] := E[(X - E[X])^2] = E[X^2] - (E[X])^2$. *Standard deviation* is the square root of variance. If $Y = aX$, where X is a random variable, then $Var[Y] = a^2 Var[X]$.

8. Let $\{X_i\}_{i=1}^n$ be *pairwise independent family of random variables.* Then,
$$Var\left[\sum_{i=1}^n X_i\right] = \sum_{i=1}^n Var[X_i]$$

9. *Inclusion-exclusion principle:*
$$P\left(\bigcup_{i\in[n]} A_i\right) = \sum_{S\subseteq[n], S\neq\phi} (-1)^{|S|+1} P\left(\bigcap_{i\in S} A_i\right)$$

10. *Union Bound:* When we have a lot of events, it becomes hard to calculate the probability of their unions using the inclusion-exclusion principle. In these cases, a simple union bound can be used to upper bound the probability of their union.
$$P\left(\bigcup_{i\in[n]} A_i\right) \leq \sum_i P(A_i)$$

1.1.1 Law of large numbers

Let the random experiment be modelled by a random variable X. Suppose the experiment is repeated n times. Denote $X_1, X_2, \cdots, X_n$ to be n copies of X (they have the same distribution). We also assume that the family of random variables $\{X_i\}_{i=1}^n$ is pairwise independent (any two random variables are independent).

The intuition is, as n gets bigger, the average value of $X_1, X_2, \cdots, X_n$ should be close to $\mathbb{E}[X]$. So, define a new random variable,
$$\overline{X} = \frac{\sum_{i=1}^n X_i}{n}.$$

Hence, $\overline{X}$ is the average of n repetitions of X (as a random variable). By linearity of expectation $E[\overline{X}] = E[X]$.

Theorem 1.1.1. *Weak law of large numbers Define the random variable* $\overline{X} = \frac{\sum_{i=1}^{n} X_i}{n}$*, where each* X_i *has the same distribution as a random variable* X *and are pairwise independent. Then,*

$$P(|\overline{X} - E[X]| \geq a) \leq \frac{Var[X]}{na^2}$$

1.2 Linear Algebra

Linear algebra provides the language of quantum mechanics. Its concepts, from vector spaces to eigenvalue decompositions, allow us to rigorously formulate and manipulate the state spaces of quantum systems.

1.2.1 Vector Spaces and Inner Product Spaces

Definition 1.2.1 (Vector Space)**.** *A* vector space V *over a field* $\mathbb{F}$ *is a set equipped with two operations: vector addition and scalar multiplication. These operations satisfy the following axioms for all* $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$ *and all scalars* $a, b \in \mathbb{F}$*:*

1. ***Associativity of Addition:*** $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$.

2. ***Commutativity of Addition:*** $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$.

3. ***Existence of Zero:*** *There exists a unique zero vector* $\mathbf{0} \in V$ *such that* $\mathbf{u} + \mathbf{0} = \mathbf{u}$.

4. ***Existence of Additive Inverses:*** *For every* $\mathbf{u} \in V$*, there exists a vector* $-\mathbf{u}$ *such that* $\mathbf{u} + (-\mathbf{u}) = \mathbf{0}$.

5. ***Distributivity:*** $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$ *and* $(a + b)\mathbf{u} = a\mathbf{u} + b\mathbf{u}$.

6. ***Compatibility:*** $a(b\mathbf{u}) = (ab)\mathbf{u}$.

7. ***Identity:*** $1\mathbf{u} = \mathbf{u}$*, where* 1 *is the multiplicative identity in* $\mathbb{F}$.

Definition 1.2.2 (Inner Product Space)**.** *An* inner product space *is a vector space* V *endowed with an inner product* $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{F}$ *satisfying:*

1. ***Conjugate Symmetry:*** $\langle \mathbf{u}, \mathbf{v} \rangle = \overline{\langle \mathbf{v}, \mathbf{u} \rangle}$.

2. ***Linearity in the First Argument:*** $\langle a\mathbf{u} + b\mathbf{v}, \mathbf{w} \rangle = a\langle \mathbf{u}, \mathbf{w} \rangle + b\langle \mathbf{v}, \mathbf{w} \rangle$.

3. ***Positive-Definiteness:*** $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$*, with equality if and only if* $\mathbf{u} = \mathbf{0}$.

1.2.2 Linear Operators and Spectral Decomposition

The study of linear operators, particularly those that act on finite-dimensional inner product spaces (Hilbert spaces), reveals the structure behind quantum evolution. In quantum mechanics, operators such as the Hamiltonian or measurement observables are Hermitian, ensuring real eigenvalues and a well-behaved spectral decomposition.

Definition 1.2.3 (Linear Operator)**.** *A mapping* $\mathbf{A} : V \to V$ *is called a* linear operator *if for all* $\mathbf{u}, \mathbf{v} \in V$ *and scalars* $c \in \mathbb{F}$*, we have*

$$\mathbf{A}(c\mathbf{u} + \mathbf{v}) = c\,\mathbf{A}(\mathbf{u}) + \mathbf{A}(\mathbf{v}).$$

Definition 1.2.4 (Hermitian Operator)**.** *A Hermitian operator is a linear operator that is self-adjoint, that is,* $\mathbf{A}$ *is Hermitian when* $\mathbf{A} = \mathbf{A}^\dagger$*. Where* $\mathbf{A}^\dagger$ *is the conjugate transpose of* $\mathbf{A}$*.*

Theorem 1.2.1 (Spectral Theorem for Hermitian Operators)**.** *Let* $\mathbf{A}$ *be a Hermitian operator acting on a finite-dimensional inner product space. Then there exists an orthonormal basis of* V *consisting of eigenvectors of* $\mathbf{A}$*, and the operator can be expressed as*

$$\mathbf{A} = \sum_i \lambda_i \mathbb{P}_i,$$

where $\lambda_i \in \mathbb{R}$ *are the eigenvalues and* $\mathbb{P}_i$ *are the orthogonal projection operators onto the corresponding eigenspaces.*

Example 1.2.1. *Consider the operator*

$$\mathbf{A} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

acting on $\mathbb{R}^2$*. A straightforward calculation shows that its eigenvalues are* $\lambda_1 = 3$ *and* $\lambda_2 = 1$*, with corresponding normalized eigenvectors. The spectral decomposition then takes the form*

$$\mathbf{A} = 3\,\mathbb{P}_1 + 1\,\mathbb{P}_2,$$

which reveals the underlying structure of $\mathbf{A}$ *in a clear and elegant way.*

The spectral theorem tells us that every Hermitian operator can be "diagonalized" by choosing an appropriate basis. This is analogous to expressing a musical chord as a combination of pure tones, with each eigenvalue representing a "note" of the operator, and the corresponding eigenvectors provide the "directions" in the space along which these notes resonate.

1.2.3 Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD) is a powerful factorization method that generalizes the spectral decomposition to any (possibly non-square) matrix. In the context of quantum computing, SVD is instrumental in understanding state transformations and noise processes.

Theorem 1.2.2 (Singular Value Decomposition)**.** *For any $m \times n$ matrix $\mathbf{M}$, there exist unitary matrices $\mathbf{U}$ (of size $m \times m$) and $\mathbf{V}$ (of size $n \times n$) such that*

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\dagger},$$

where $\mathbf{\Sigma}$ is an $m \times n$ diagonal matrix with non-negative real numbers (the singular values*) on the diagonal.*

Example 1.2.2. *Let's look at the matrix*

$$\mathbf{M} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 0 & 0 \end{pmatrix}$$

This matrix takes a 2D vector (since it has 2 columns) and transforms it into a 3D vector (since it has 3 rows). The SVD of this matrix is

$$\mathbf{M} = \underbrace{\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{U}} \underbrace{\begin{pmatrix} 2 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}}_{\mathbf{\Sigma}} \underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^{\dagger}}_{\mathbf{V}^{\dagger}}$$

Let's understand the components: In this specific case, $\mathbf{V}$ is the matrix that swaps the standard basis vectors.

$$\mathbf{V} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

This is a simple rotation and reflection matrix. Since it's a real matrix, the dagger operation is just the transpose, so $\mathbf{V}^{\dagger} = \mathbf{V}$. When we apply $\mathbf{V}^{\dagger}$ to an input vector, it swaps its components. For example, it rotates the vector $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ to $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$.
$\mathbf{U}$ is a 3×3 unitary matrix.

$$\mathbf{U} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

This matrix also performs a rotation in 3D space. Specifically, it swaps the first two coordinates of a vector, which corresponds to a rotation in the output space.
The SVD reveals the core action of $\mathbf{M}$. The transformation first swaps the input components ($\mathbf{V}^{\dagger}$), then stretches the new second component by 2 and the new first component by 1 ($\mathbf{\Sigma}$), and finally swaps these first two components in the 3D output space ($\mathbf{U}$). The magic of SVD is that it finds the exact "input" and "output" bases ($\mathbf{V}$ and $\mathbf{U}$) where the transformation is just a simple scaling ($\mathbf{\Sigma}$).

SVD can be seen as a "best possible" diagonalization of any matrix. Imagine reshaping an arbitrary transformation into a rotation (via $\mathbf{V}^{\dagger}$), followed by a scaling (via $\mathbf{\Sigma}$), and then another rotation (via $\mathbf{U}$). This perspective is particularly useful in quantum information, where one often needs to analyze the effect of noise or perform optimal approximations of unitary evolutions.

1.2.4 Polar Decomposition

Think of any quantum process, from a perfect, idealized gate to a noisy, real-world interaction, as a linear transformation acting on a quantum state. The Polar Decomposition provides an essential and deeply intuitive way to dissect any such transformation.

It elegantly separates the process into two fundamental and distinct actions: a pure rotation (represented by the unitary operator **U**), which preserves the geometric relationships within the quantum state space, and a pure stretch or deformation (represented by the positive operator **J**). This separation is invaluable in quantum information because it allows us to isolate the ideal, coherent part of an evolution from its non-unitary components, which often correspond to noise or measurement effects. Its most vital application is in finding the closest ideal quantum gate (**U**) to an actual, imperfect experimental operation, making it an indispensable tool for analyzing gate fidelity and designing error-resilient quantum controls.

Polar decomposition says that every linear operator can be decomposed as a unitary and a unique positive operator.

Theorem 1.2.3 (Polar Decomposition)**.** *Given a linear operator A on a vector space V, there exists unitary U and unique positive matrices $J \equiv \sqrt{A^\dagger A}$ and $K \equiv \sqrt{AA^\dagger}$ such that*

$$A = UJ = KU$$

If A is invertible, then U is also unique.

Proof. J as defined in the theorem is a positive operator, so by spectral decomposition it can be written as $J = \sum_i \lambda_i \left|i\right\rangle$. Define $\left|\phi_i\right\rangle = A\left|i\right\rangle$. Notice that $\langle\phi_i|\phi_i\rangle = \lambda_i^2$, so $\left|\phi_i\right\rangle / \lambda_i^2$ (when $\lambda_i \neq 0$) is a unit vector, call it $\left|e_i\right\rangle$. By the Gram-Schmidt process, we can extend the basis as $\{e_i\}$ to form an orthonormal basis.

Define $U = \sum_i \left|e_i\right\rangle \left\langle i\right|$. Consider the action of UJ on any eigen vector $\left|i\right\rangle$ with non-zero eigenvalue λ_i,

$$UJ\left|i\right\rangle = \lambda_i U\left|i\right\rangle = \lambda_i\left|e_i\right\rangle = \left|\phi_i\right\rangle = A\left|i\right\rangle$$

For $\lambda_i = 0$,

$$UJ\left|i\right\rangle = 0 = \left|\phi_i\right\rangle$$

Thus, the action of UJ on the basis vectors is the same as that of A. So we have $A = UJ$.

$A^\dagger = JU^\dagger$, so $A^\dagger A = J^2$, giving J a unique value $\sqrt{A^\dagger A}$. When A is invertible J also is invertible, thus giving $U = AJ^{-1}$ uniquely. Similar arguments can be used to show $A = KU$ as well. ■

1.3 Group Theory

Group theory provides the language of symmetry that is pervasive in both classical and quantum systems. Its axioms encapsulate the essence of symmetry operations, which are central to understanding quantum dynamics and computational processes.

1.3.1 Fundamental Definitions

Definition 1.3.1 (Group)**.** *A* group $(G, \cdot)$ *is a set* G *together with a binary operation* $\cdot : G \times G \to G$ *satisfying:*

1. ***Closure:*** *For every* $a, b \in G$*, the product* $a \cdot b$ *is in* G*.*

2. ***Associativity:*** *For all* $a, b, c \in G$*,* $(a \cdot b) \cdot c = a \cdot (b \cdot c)$*.*

3. ***Identity:*** *There exists an element* $e \in G$ *such that for every* $a \in G$*,* $e \cdot a = a \cdot e = a$*.*

4. ***Inverses:*** *For each* $a \in G$*, there exists an element* $a^{-1} \in G$ *such that* $a \cdot a^{-1} = a^{-1} \cdot a = e$*.*

Definition 1.3.2 (Abelian Group)**.** *A group* G *is called* abelian *(or commutative) if, in addition to the group axioms, it satisfies*

$$a \cdot b = b \cdot a \quad \text{for all } a, b \in G.$$

1.3.2 Subgroups, Cosets, and Normality

Understanding substructures within a group allows us to analyze and decompose complex symmetry operations.

Definition 1.3.3 (Subgroup)**.** *A non-empty subset* $H \subseteq G$ *is a* subgroup *of* G *if* H *is itself a group under the operation inherited from* G*. We denote this by* $H \leq G$*.*

Theorem 1.3.1 (Lagrange's Theorem)**.** *If* G *is a finite group and* H *is a subgroup of* G*, then the order (number of elements) of* H *divides the order of* G*.*

Definition 1.3.4 (Normal Subgroup)**.** *A subgroup* $N \leq G$ *is* normal *(denoted* $N \triangleleft G$*) if it is invariant under conjugation; that is, for every* $n \in N$ *and every* $g \in G$*, we have*

$$gng^{-1} \in N.$$

Normal subgroups allow the construction of quotient groups, which capture the idea of symmetry modulo some invariant structure.

1.3.3 Cyclic Groups and Group Homomorphisms

Cyclic groups are the simplest examples of groups, serving as building blocks for more intricate symmetry operations.

Definition 1.3.5 (Cyclic Group)**.** *A group* G *is* cyclic *if there exists an element* $g \in G$ *such that every element in* G *can be written as a power of* g*, i.e.,*

$$G = \{g^n \mid n \in \mathbb{Z}\}.$$

Such an element g *is called a* generator *of* G*.*

Example 1.3.1. *The group* $(\mathbb{Z}/n\mathbb{Z}, +)$ *is cyclic, with* 1 *(or any element coprime to* n*) serving as a generator. This example illustrates how modular arithmetic naturally leads to cyclic group structures.*

A deeper understanding of group structure is achieved via homomorphisms.

Definition 1.3.6 (Group Homomorphism)**.** *A map $\varphi : G \to H$ between two groups $(G, \cdot)$ and $(H, *)$ is a* group homomorphism *if for all $a, b \in G$,*

$$\varphi(a \cdot b) = \varphi(a) * \varphi(b).$$

Homomorphisms preserve the algebraic structure, allowing us to relate different groups through their shared symmetries.

1.4 Fourier Transformation

Fourier transformation is a fundamental tool that decomposes functions into their constituent frequency components. In both classical and quantum contexts, it enables us to switch between time (or spatial) representations and frequency domains. This dual perspective is not only mathematically elegant but also pivotal in quantum algorithms such as Shor's algorithm.

For a function $f : \mathbb{R} \to \mathbb{C}$, the Fourier transform is defined as

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi}\, dx.$$

The inverse Fourier transform recovers the original function:

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i x \xi}\, d\xi.$$

Think of the Fourier transform as a way to "listen" to the hidden frequencies within a signal. Just as a musical chord can be decomposed into individual notes, any function can be expressed as a sum (or integral) of sinusoidal components. In quantum mechanics, this idea underpins the relationship between position and momentum representations.

1.5 Group Theoretic Perspective on Fourier Transform

The Fourier transform can be generalized to functions defined on groups, revealing deep connections between harmonic analysis and group theory. This perspective is especially fruitful in quantum computing, where symmetries play a central role in algorithm design.

1.5.1 Fourier Transform over Abelian Groups

For a finite Abelian group G, the Fourier transform decomposes a function $f : G \to \mathbb{C}$ into a sum over the group's characters. A *character* χ is a homomorphism from G to the multiplicative circle group $\mathbb{C}^\times$.

Definition 1.5.1 (Fourier Transform on Finite Abelian Groups)**.** *Let G be a finite Abelian group of order $|G|$. For a function $f : G \to \mathbb{C}$, its Fourier transform is defined as*

$$\hat{f}(\chi) = \frac{1}{\sqrt{|G|}} \sum_{g \in G} f(g) \overline{\chi(g)},$$

for every character χ in the dual group $\widehat{G}$.

The inverse Fourier transform is given by

$$f(g) = \frac{1}{\sqrt{|G|}} \sum_{\chi \in \widehat{G}} \hat{f}(\chi)\chi(g).$$

In the Abelian case, the characters serve as the "frequency modes" of the group. They allow us to express a function as a combination of these basic oscillatory components. For example, in the cyclic group $\mathbb{Z}_n$, the characters are simple exponential functions, which makes the discrete Fourier transform a natural tool for digital signal processing and quantum algorithms.

Example 1.5.1. *For the cyclic group $G = \mathbb{Z}_n$, the characters are given by*

$$\chi_k(j) = e^{2\pi ikj/n}, \quad k, j \in \{0, 1, \ldots, n-1\}.$$

Thus, the Fourier transform on $\mathbb{Z}_n$ becomes

$$\hat{f}(k) = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} f(j) e^{-2\pi ikj/n}.$$

1.5.2 Fourier Transform over Non-Abelian Groups

When the group G is non-Abelian, the Fourier transform is extended by replacing characters with the set of irreducible representations. For a finite non-Abelian group G, let $\{\rho\}$ denote the set of inequivalent irreducible representations of G, where each representation $\rho : G \to \mathrm{GL}(V_\rho)$ maps group elements to matrices acting on the vector space V_ρ.

Definition 1.5.2 (Fourier Transform on Finite Non-Abelian Groups)**.** *Let $f : G \to \mathbb{C}$ be a function. The Fourier transform of f at an irreducible representation ρ is defined by*

$$\hat{f}(\rho) = \sum_{g \in G} f(g)\rho(g)^\dagger.$$

Here, $\hat{f}(\rho)$ is a matrix of dimension $\dim(\rho) \times \dim(\rho)$.

The inversion formula is given by

$$f(g) = \frac{1}{|G|} \sum_{\rho} \dim(\rho)\, \mathrm{Tr}\Big(\rho(g)\hat{f}(\rho)\Big),$$

where the sum is taken over all inequivalent irreducible representations of G.
In non-Abelian groups, the irreducible representations generalize the notion of frequency modes. Instead of scalar oscillations, the decomposition yields matrix-valued components that capture more complex symmetries. This richer structure is central in quantum algorithms that exploit non-Abelian hidden subgroup problems or study symmetry properties of quantum systems.

Example 1.5.2. *Consider the symmetric group S_3, one of the simplest non-Abelian groups. It has three irreducible representations: two one-dimensional representations and one two-dimensional representation. When applying the Fourier transform to a function on S_3, the one-dimensional representations yield scalar components, while the two-dimensional representation provides a 2×2 matrix capturing the more intricate symmetry of the group.*

1.6 Number Theoretic Foundations

The elegant machinery and beauty of number theory, particularly the properties of modular arithmetic, provide the foundational framework for powerful algorithms, notably in cryptography and quantum computation. We will be using the following beautiful definitions and theorems in the chapters ahead, with a special emphasis on Shor's breakthrough in quantum computing.

1.6.1 Finite Groups Modulo N

Let $(\mathbb{Z}/n\mathbb{Z})^\times$ denote the multiplicative group of integers modulo N that are coprime to N. Formally $(\mathbb{Z}/n\mathbb{Z})^\times = \{a \in \mathbb{Z} \mid 1 \leq a < N \text{ and } \gcd(a, N) = 1\}$.
If n is not a prime, it has all elements of $(\mathbb{Z}/n\mathbb{Z})$ which are coprime with n. If n is prime, $(\mathbb{Z}/n\mathbb{Z})^\times$ is the same as $(\mathbb{Z}/n\mathbb{Z})$. This set forms a group under multiplication modulo N, with order $\varphi(N)$, where φ is Euler's totient function. This denotes the cardinality of numbers, which are coprime to N.

Example 1.6.1. *For $N = 15$, $(\mathbb{Z}/15\mathbb{Z})^\times = \{1, 2, 4, 7, 8, 11, 13, 14\}$ and $\varphi(15) = 8$.*

Note that the Euler totient function is a multiplicative function, that is, if two numbers m and n are relatively prime, that $\varphi(mn) = \varphi(m)\varphi(n)$.

Example 1.6.2. *For $N = 3$, $(\mathbb{Z}/3\mathbb{Z})^\times = \{1, 2\}$ and for $(\mathbb{Z}/5\mathbb{Z})^\times = \{1, 2, 3, 4\}$. Note that $\varphi(3) \times \varphi(5) = \varphi(15) = 8$.*

1.6.2 Order of an Element

Definition 1.6.1 (Order)**.** *The* ***order*** *of an element $a \in (\mathbb{Z}/n\mathbb{Z})^\times$ is the smallest positive integer r such that: $a^r \equiv 1 \mod N$.*

By Lagrange's theorem, we emphasize that the order of the cyclic group generated by an element a given by $\{1, a, a^2, \ldots, a^{r-1}\}$ such that $a^r = 1 \mod N$, divides the order of the group, hence r divides $\varphi(N)$.

1.6.3 Fermat-Euler Theorem

Theorem 1.6.1. *(Fermat's Little Theorem): For any $a \in \mathbb{Z}$, p some prime,*

$$a^p \equiv a \mod p.$$

Theorem 1.6.2. *(Fermat-Euler): For any $a \in (\mathbb{Z}/n\mathbb{Z})^\times$,*

$$a^{\varphi(N)} \equiv 1 \mod N.$$

Euler's theorem is a generalization of Fermat's little theorem (For any prime $\varphi(p) = p - 1$ since there exist $p - 1$ numbers co-prime to p smaller than p, hence $a^{\varphi(p)} \mod p \equiv a^{p+1} \mod p \equiv 1 \mod p$, thereby $a^p \equiv a \mod p$), which can be understood from group theoretic principles. Since the order of any element in $(\mathbb{Z}/n\mathbb{Z})^\times$ divides the order of $(\mathbb{Z}/n\mathbb{Z})^\times$, we have $a^r \equiv 1 \mod N$ where r divides $\varphi(N)$. Thereby we have $\varphi(N) = rk$ for some k, then $a^{\varphi(N)} \mod N \equiv (a^r)^k \mod N = 1 \mod N$.

Chapter 2

Physics Formalism

> *"When searching for harmony in life, one must never forget that in the drama of existence, we are ourselves both actors and spectators."*
>
> – Niels Bohr, *Discussions with Einstein*

2.1 Postulates of Quantum Mechanics

Quantum mechanics, at its heart, is a framework for predicting the behavior of the universe at its smallest scales. While its implications can seem *bizarre*, the theory itself rests on a few fundamental postulates. These postulates provide the language and machinery for describing physical reality. Instead of merely stating them, let's explore their physical meaning.

Postulate 1. *States of a quantum system are associated with a unit vector in Hilbert space.*

Everything we can possibly know about an isolated quantum system is encoded in a single mathematical object: a state vector, denoted $|\psi\rangle$. This vector lives in a special complex vector space called a Hilbert space. Because probabilities must sum to one, this state vector is always a unit vector ($\langle\psi|\psi\rangle = 1$). The power of this postulate is the *principle of superposition*, if $|\psi_1\rangle$ and $|\psi_2\rangle$ are valid states, then so is their linear combination, $\alpha|\psi_1\rangle + \beta|\psi_2\rangle$.

Postulate 2. *Observables are associated with Hermitian operators on the system's Hilbert space.*

Every measurable property of a system, like position, momentum, or spin, is represented by a Hermitian operator acting on that system's Hilbert space. The necessity for Hermitian operators which are self adjoint, is because the outcome of any real-world measurement must be a real number, and Hermitian operators are guaranteed to have real eigenvalues.

Postulate 3. *States transform via unitary operations. The Schrodinger equation governs time evolution.*

When we aren't looking at it, a quantum system evolves in a perfectly smooth, continuous, and deterministic way. This evolution is described by a unitary transformation. A state $|\psi(t_1)\rangle$ evolves to $|\psi(t_2)\rangle = U|\psi(t_1)\rangle$. A unitary transformation is essentially a rotation in the Hilbert space that preserves the length of the state vector, ensuring that probabilities continue to make sense over time.

Postulate 4 (Quantum Measurement)**.** *When observable A is measured on state $|\psi\rangle$, the set of outcomes is the set of eigenvalues of A $\{a_i\}$*
i) The probability of obtaining outcome a_i is given by $p(a_i) = \left|\langle\psi|a_i\rangle\right|^2$
ii) The state of the system after measurement collapses to one of the eigenstates $\{|a_i\rangle\}$ of A.

The smooth deterministic evolution of the system is violently interrupted by the act of measurement. Measurement in quantum mechanics is a probabilistic event which has enormous philosophical notions and is still debated upon. When an observable A is measured, the only possible results are the eigenvalues $\{a_i\}$ of the operator A. Even if the system was in a vast superposition of states, the measurement outcome is restricted to this specific set of values. However, we cannot, in general, predict the outcome with certainty. We can only predict the probability of obtaining a specific outcome a_i. This is given by the square of the projection of the state vector $|\psi\rangle$ onto the corresponding eigenvector $|a_i\rangle$. That is, $p(a_i) = |\langle a_i|\psi\rangle|^2$. The closer the state $|\psi\rangle$ is aligned with an eigenvector $|a_i\rangle$, the more likely that outcome becomes. The measurement doesn't just report a value; it fundamentally alters the system. Immediately after obtaining the outcome a_i, the system's state is no longer $|\psi\rangle$. It abruptly *collapses* to the corresponding eigenvector $|a_i\rangle$. All information about the original superposition is lost, and the system is now defined by the result of the measurement. This is the source of the inherent randomness in the quantum world. The notion of how this happens is heavily debated upon and is termed the Measurement Problem[1].

2.2 State Vector

As stated in the first postulate, the state vector $|\psi\rangle$ is the fundamental carrier of information in quantum mechanics. It is an element of a Hilbert space $\mathcal{H}$, a complex vector space equipped with an inner product. For a two-level system, which will be of significant interest further, the Hilbert space is two-dimensional, $\mathcal{H}_2$. The standard basis for this space is given by the orthonormal vectors, represented by $|0\rangle$ and $|1\rangle$, which we shall delve into further.

A general state of a two-level system is a superposition of these basis states:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

where $\alpha, \beta \in \mathbb{C}$ are complex amplitudes. The normalization condition $\langle\psi|\psi\rangle = 1$ requires that $|\alpha|^2 + |\beta|^2 = 1$. This condition ensures that the probabilities of measuring the system to be in state $|0\rangle$ or $|1\rangle$ sum to unity.

[1] Refer https://arxiv.org/abs/2502.19278 to question the reality we live in, and review through the different notions and interpretations of one of the fundamental postulates of quantum mechanics.

2.3 Entanglement

When we consider systems composed of more than one two-level system, we encounter one of the most profound and counterintuitive features of quantum mechanics *entanglement.* A composite quantum system, say consisting of two subsystems A and B, is described by a state vector in the tensor product of their individual Hilbert spaces, $\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B$.

A state of two two-level systems $|\psi\rangle_{AB}$ is called *separable* if it can be written as a tensor product of individual states of the subsystems:

$$|\psi\rangle_{AB} = |\psi\rangle_A \otimes |\psi\rangle_B$$

A separable state implies that the properties of subsystem A are independent of subsystem B.

As we shall see further, a state is *entangled* if it is not separable. The most famous example of an entangled state is the Bell state $|\Phi^+\rangle$:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \equiv \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B)$$

This state cannot be factored into a simple product of a state for system A and a state for system B. The consequence of this is that the measurement outcomes of the two systems are perfectly correlated, no matter how far apart they are. If someone measures their system (A) and finds it in the state $|0\rangle$, they would instantly know that their friend's system (B) is also in the state $|0\rangle$. This *spooky action at a distance*, as Einstein famously called it, does not allow for faster-than-light communication, but it is a powerful resource for quantum computation and information protocols like teleportation, which we shall see further.

2.4 Measurement

Every dynamical observable q, such as position, momentum, angular momentum, etc., are associated with a Hermitian operator Q. Note that eigenvalues $\{q_i\}$ of a Hermitian operator are real and the non-degenerate eigenvectors of a Hermitian operator are orthogonal. The eigenvectors $\{|q_i\rangle\}$ of a Hermitian operator form a complete orthonormal basis with a spectral decomposition:

$$Q = \sum_{q_i} q_i \mathbb{P}_{q_i},$$

where the sum runs up to the dimension of the Hilbert space $\mathcal{H}_N$, and $\mathbb{P}_{q_i}$ are projectors. For non-degenerate eigenvalues, we have $\mathbb{P}_{q_i} = |q_i\rangle\langle q_i|$. For a degenerate subspace with $\{|q_i^{(1)}\rangle, |q_i^{(2)}\rangle, \ldots, |q_i^{(r)}\rangle\}$, with each member having the same eigenvalue q_i, then the projector is constructed using all of them as:

$$\mathbb{P}_{q_i} = \sum_j |q_i^{(j)}\rangle\langle q_i^{(j)}|.$$

Measuring an observable Q for a quantum system in state $|\Psi\rangle$ results in one of the eigenvalues q_i with probability given by the Born rule:

$$p(q_i) = \langle \mathbb{P}_{q_i} \rangle_\Psi = \langle \Psi | \mathbb{P}_{q_i} | \Psi \rangle = \left| \mathbb{P}_{q_i} | \Psi \rangle \right|^2 .$$

Interestingly, some experimentalists are still investigating if this rule is exact or a first-order approximation[2]. Note that this rule corresponds to an intuition of two different concepts. Firstly, we can regard the measurement probability as the expectation value of the projector over the state $|\Psi\rangle$ corresponding to an ensemble average. Also, this can be seen as the corresponding probability amplitude of the system to transfer from a state $|\Psi\rangle$ to a state proportional to $\mathbb{P}_{q_i}|\Psi\rangle$, with the amplitude defined from the inner product between the states as $\langle \Psi | (\mathbb{P}_{q_i} | \Psi \rangle)$. Thereby, we further claim that the post-measurement state can be thought of as evolution into the state $\mathbb{P}_{q_i}|\Psi\rangle$, given formally by the normalised form as:

$$|\Psi_{q_i}\rangle = \frac{\mathbb{P}_{q_i}|\Psi\rangle}{|\mathbb{P}_{q_i}|\Psi\rangle|} = \frac{\mathbb{P}_{q_i}|\Psi\rangle}{\sqrt{\langle \Psi | \mathbb{P}_{q_i} | \Psi \rangle}} .$$

For a non-degenerate operator, we obtain $p(q_i) = |\langle q_i | \Psi \rangle|^2$, and $|\Psi_{q_i}\rangle = |q_i\rangle$. Note that we will touch upon this in great detail, as this collapse of a state corresponds to the most intricate theories in quantum mechanics. Since we have modeled everything through unitarity so far, this non-unitary operation of measurement has very different characteristics and nuances that we will explore.

2.5 State Vector vs Density Matrix

We formulate the notions of quantum mechanics in a very different notion, which is a profound realisation of the statistical and probabilistic interpretations.

To gain a better understanding, we review the uncertainties in quantum mechanics first. There are two different kinds of uncertainty in quantum mechanics. The first is the intrinsic quantum mechanical uncertainty due to its features of superposition and probabilistic measurements. Knowing the state of the system completely still implies that we can only make probabilistic statements about the outcomes of some experiments. For example, if we have two particles in a 2-state system, then the system may be in an entangled state like

$$|\Psi\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle|1\rangle + |1\rangle|0\rangle \right) .$$

In this case, it is not possible to say with certainty if a measurement of one of the particles will yield the result 0 or 1.

The second is a classical uncertainty in the preparation of the state of the system. For example, perhaps we think there's a 50% chance that the system is in the state Ψ given above and a 50% chance that the system is in a different state $|\Phi\rangle$, given by

$$|\Psi\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle|1\rangle - |1\rangle|0\rangle \right) .$$

[2] See, for example, *Science, 329, 418-421 (2010).*

This kind of uncertainty cannot be represented nicely using only state vectors in Hilbert space. While we are already struggling with two particles, imagine the sheer amount of technicalities that need to be specified for a huge system in thermodynamic aspects.

The density matrix formulation, borrowed from Quantum Statistical Mechanics can encode both kinds of uncertainty. Because the density matrix can handle both kinds of uncertainty, the density matrix generalizes the normal quantum state vector and lets us handle a wider variety of cases.

2.6 Density Matrix Formalism

In quantum mechanics, the state of a system is conventionally described by a state vector in a Hilbert space. However, as discussed before, when we wish to account for uncertainties, whether they come from inherent quantum indeterminacy or from classical probabilistic mixtures, the state vector is no longer sufficient. In such cases, we turn to the density matrix (or density operator) formalism, which elegantly encapsulates both pure and mixed states.

Let us begin with a single qubit in a pure state, represented by the state vector $|\psi\rangle = \begin{pmatrix} a \\ b \end{pmatrix}$, where $a, b \in \mathbb{C}$ satisfy $|a|^2 + |b|^2 = 1$. For a pure state, the density matrix is defined as the projection operator onto the state:

$$\rho = |\psi\rangle\langle\psi|.$$

Writing this out in the computational basis $\{|0\rangle, |1\rangle\}$, we obtain

$$\rho = |a|^2|0\rangle\langle 0| + ab^*|0\rangle\langle 1| + a^*b|1\rangle\langle 0| + |b|^2|1\rangle\langle 1|,$$

or, equivalently, in matrix form,

$$\rho = \begin{pmatrix} |a|^2 & ab^* \\ a^*b & |b|^2 \end{pmatrix}.$$

The diagonal entries of ρ represent the probabilities of measuring the qubit in the states $|0\rangle$ and $|1\rangle$, respectively, while the off-diagonal (or coherent) terms capture the phase relationships, i.e. the quantum coherence between these basis states.

Pure states describe systems with complete knowledge of the quantum state. In practice, however, we often encounter situations where the system is in a probabilistic mixture of different states. Such a scenario is described by a *mixed state.* If the system is prepared in the state $|\varphi_i\rangle$ with probability p_i, then the density matrix is given by

$$\rho = \sum_i p_i |\varphi_i\rangle\langle\varphi_i|,$$

with the conditions,

$$p_i \geq 0, \quad \sum_i p_i = 1.$$

The states $\{|\varphi_i\rangle\}$ are (not necessarily orthogonal in a general ensemble, but can be chosen to form an orthonormal set in the spectral decomposition).

This formulation is especially useful when we have incomplete information about the system or when the system is entangled with an external environment. One of the strengths of the density matrix approach is its ability to provide a unified description of both classical uncertainty and quantum superposition. The density matrix ρ possesses several important mathematical properties:

- **Hermiticity:** $\rho^\dagger = \rho$, ensuring that its eigenvalues are real. Taking the Hermitian conjugate of ρ, we have

$$\begin{aligned}\rho^\dagger &= \left(\sum_i p_i |\varphi_i\rangle\langle\varphi_i|\right)^\dagger \\ &= \sum_i p_i \left(|\varphi_i\rangle\langle\varphi_i|\right)^\dagger \quad \text{(by linearity)} \\ &= \sum_i p_i |\varphi_i\rangle\langle\varphi_i| \quad (\text{since } (|\varphi_i\rangle\langle\varphi_i|)^\dagger = |\varphi_i\rangle\langle\varphi_i|) \\ &= \rho.\end{aligned}$$

  Thus, ρ is Hermitian. This property guarantees that all eigenvalues of ρ are real.

- **Positivity:** ρ is a positive semi-definite operator, which implies that $\langle\phi|\rho|\phi\rangle \geq 0$ for any state $|\phi\rangle$. For an arbitrary state $|\phi\rangle$, we compute

$$\begin{aligned}\langle\phi|\rho|\phi\rangle &= \left\langle \phi \left| \sum_i p_i |\varphi_i\rangle\langle\varphi_i| \right| \phi \right\rangle \\ &= \sum_i p_i \langle\phi|\varphi_i\rangle\langle\varphi_i|\phi\rangle \\ &= \sum_i p_i |\langle\varphi_i|\phi\rangle|^2.\end{aligned}$$

  Since $p_i \geq 0$ and $|\langle\varphi_i|\phi\rangle|^2 \geq 0$ for all i, it follows that

$$\langle\phi|\rho|\phi\rangle \geq 0.$$

  Thus, ρ is a positive semi-definite operator.

- **Unit Trace:** $\mathrm{Tr}(\rho) = 1$, reflecting the total probability. Using the definition of the trace in any complete orthonormal basis $\{|j\rangle\}$, we have

$$\begin{aligned}\mathrm{Tr}(\rho) &= \mathrm{Tr}\left(\sum_i p_i |\varphi_i\rangle\langle\varphi_i|\right) \\ &= \sum_i p_i \mathrm{Tr}\left(|\varphi_i\rangle\langle\varphi_i|\right).\end{aligned}$$

But for any normalised state $|\varphi_i\rangle$,

$$\mathrm{Tr}\left(|\varphi_i\rangle\langle\varphi_i|\right) = \langle\varphi_i|\varphi_i\rangle = 1.$$

Thus,

$$\mathrm{Tr}(\rho) = \sum_i p_i = 1.$$

Moreover, the purity of a state can be quantified by the trace of ρ^2. Starting from the spectral decomposition, the square of the density matrix is

$$\begin{aligned}\rho^2 &= \left(\sum_i p_i|\varphi_i\rangle\langle\varphi_i|\right)\left(\sum_j p_j|\varphi_j\rangle\langle\varphi_j|\right)\\ &= \sum_{i,j} p_i p_j |\varphi_i\rangle\langle\varphi_i|\varphi_j\rangle\langle\varphi_j|.\end{aligned}$$

If the states $\{|\varphi_i\rangle\}$ are chosen as an orthonormal eigenbasis of ρ, then

$$\langle\varphi_i|\varphi_j\rangle = \delta_{ij},$$

and we have,

$$\rho^2 = \sum_i p_i^2|\varphi_i\rangle\langle\varphi_i|.$$

Taking the trace,

$$\begin{aligned}\mathrm{Tr}(\rho^2) &= \mathrm{Tr}\left(\sum_i p_i^2|\varphi_i\rangle\langle\varphi_i|\right)\\ &= \sum_i p_i^2 \mathrm{Tr}\left(|\varphi_i\rangle\langle\varphi_i|\right)\\ &= \sum_i p_i^2.\end{aligned}$$

Since $\sum_i p_i = 1$ and $0 \leq p_i \leq 1$, by the properties of probabilities (Cauchy-Schwarz Inequality) we have

$$\sum_i p_i^2 \leq \left(\sum_i p_i\right)^2 = 1,$$

with equality if and only if one of the $p_i = 1$ (i.e., for a pure state). Hence,

$$\mathrm{Tr}(\rho^2) \leq 1.$$

For a pure state, $\mathrm{Tr}(\rho^2) = 1$ whereas for a mixed state, $\mathrm{Tr}(\rho^2) < 1$. This criterion provides a clear operational test for distinguishing between pure and mixed states.

An important aspect of the density matrix formalism is its role in computing expectation values. Let O be any observable (a Hermitian operator). The expectation value of O in the state ρ is defined as

$$\langle O \rangle = \sum_i p_i \langle \varphi_i | O | \varphi_i \rangle.$$

On the other hand, using the definition of the trace,

$$\begin{aligned}\mathrm{Tr}(\rho O) &= \mathrm{Tr}\left(\sum_i p_i |\varphi_i\rangle\langle\varphi_i| O\right) \\ &= \sum_i p_i \mathrm{Tr}\left(|\varphi_i\rangle\langle\varphi_i| O\right) \\ &= \sum_i p_i \langle\varphi_i|O|\varphi_i\rangle,\end{aligned}$$

where we used the cyclic property of the trace and the fact that

$$\mathrm{Tr}\left(|\varphi_i\rangle\langle\varphi_i| O\right) = \langle\varphi_i|O|\varphi_i\rangle.$$

Thus, we conclude that

$$\langle O \rangle = \mathrm{Tr}(\rho O).$$

2.7 Reduced Density Operator

When dealing with a composite quantum system, such as an entangled pair, we often want to describe the state of just one of its subsystems. The state vector $|\psi\rangle_{AB}$ describes the entire system, but what is the state of subsystem A alone? This question is answered by the *reduced density operator*.

Given a composite system AB described by the density operator ρ_{AB}, the reduced density operator for subsystem A is obtained by performing a partial trace over subsystem B, denoted Tr_B

$$\rho_A \equiv \mathrm{Tr}_B(\rho_{AB})$$

The partial trace is an operation that traces out the degrees of freedom of subsystem B, leaving an operator that acts only on the Hilbert space of A. If $\{|b_j\rangle\}$ is an orthonormal basis for the Hilbert space of subsystem B, the partial trace is defined as

$$\rho_A = \sum_j \langle b_j | \rho_{AB} | b_j \rangle$$

The resulting operator ρ_A completely describes all possible measurement outcomes for any observable acting solely on subsystem A.

A remarkable feature of entanglement is revealed here. If the composite system AB is in a pure entangled state, the reduced state of its subsystems will be a mixed state. Let's demonstrate this with the Bell state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. The density operator for the composite system is

$$\begin{aligned}
\rho_{AB} &= |\Phi^+\rangle\langle\Phi^+| \\
&= \frac{1}{2}\left(|00\rangle + |11\rangle\right)\left(\langle 00| + \langle 11|\right) \\
&= \frac{1}{2}\left(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|\right)
\end{aligned}$$

Now, we compute the reduced density operator for subsystem A by tracing over B,

$$\begin{aligned}
\rho_A = \mathrm{Tr}_B(\rho_{AB}) &= \langle 0|_B \rho_{AB} |0\rangle_B + \langle 1|_B \rho_{AB} |1\rangle_B \\
&= \frac{1}{2}\left(\langle 0|_B |00\rangle\langle 00|0\rangle_B + \langle 1|_B |11\rangle\langle 11|1\rangle_B\right) \\
&= \frac{1}{2}\left(|0\rangle_A\langle 0|_A + |1\rangle_A\langle 1|_A\right) \\
&= \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \frac{1}{2}I
\end{aligned}$$

The state of subsystem A is termed the *maximally mixed state*. It contains no information about the state of the system along any measurement axis, since there is an equal probability of measuring 0 or 1 for any measurement basis. The purity of this state is $\mathrm{Tr}(\rho_A^2) = \mathrm{Tr}(\frac{1}{4}I) = \frac{1}{2}$, confirming it is a mixed state. This demonstrates a fundamental principle that, for an entangled system, information is stored in the correlations between the subsystems, not in the subsystems themselves. By ignoring one part of an entangled system, we lose all the information about the whole.

Chapter 3

Theory of Computation

> *"As long as a branch of science offers an abundance of problems, so long is it alive; a lack of problems foreshadows extinction or the cessation of independent development."*
>
> – David Hilbert, *Mathematical Problems*

Suppose we are tasked with a problem to be solved on a real computer. A computer can be realised as a circuit model, with wires carrying information to be processed through a set of logical operations (gates), which allows us to implement any complex calculation. Given the limited resources of memory, time, and energy, we are tasked to find the best possible sequences of a set of rules to solve it, optimising the resources. Welcome to the field of computational complexity!

Computational complexity theory is a broad field that uses various models, *circuit model, RAM model, query model*, etc, to better understand resource optimization in every possible way. We will be looking at *circuit model* and *RAM model* in this chapter and *query model* in chapter 7.

3.1 Turing Machine

By definition, an *algorithm* is a set of instructions for solving a problem. The *Turing Machine*[1] provides a firm mathematical framework to adjoin the intuitive understanding of an algorithm. It was introduced as the fundamental 'universal computer' containing the essential elements on which any modern computer is based.

The general idea is strongly implied from what a 'human computer'[2] would do. Such a

[1] Alan Turing was a brilliant and eccentric persona, credited to have solved the Halting Problem through very intuitive arguments and formal structure of the Turing machine. Some believe the ideas of Turing inspired Kurt Gödel to formulate the *Incompleteness Theorems*. For more, refer to *Gödel, Escher, Bach: an Eternal Golden Braid.*

[2] Back in the day, a computer is one who computes (sadly, mostly women). Read `https://www.smithsonianmag.com/science-nature/history-human-computers-180972202/` for a gendered history of computing.

human computer has limited storage capacity for information but ideally has an unlimited amount of paper for reading and writing operations. Formally, a Turing machine constitutes of

- A *tape*, which can be infinite and is subdivided. Each cell division constitutes one letter $\mathscr{A}_i$ from the alphabet $\{\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_k\}$ or is blank.

- A *control unit* with states referring to the internal configuration $\{s_1, s_2, \ldots, s_l, \mathcal{H}\}$, where $\mathcal{H}$ halts and terminates the computation internal state and the symbol currently being read. Since we want this machine to be physically realizable, the number of possible internal states should be finite.

- A *read/write* head which reads and writes a new symbol in the current cell, overwriting whatever symbol is there, moving backwards or forward one cell, and switching to a new state or halt.

Turing's first result is the existence of a universal machine, whose job is to simulate any other machine described via symbols on the tape. Let us briefly emphasize why this is so groundbreaking. Imagine you were given a set of Turing machines. Through some painstaking work and tweaking, you can build a machine that can solve any problem for you, that can play games, that can watch videos, print text etc.

If this wasn't already enough, Turing's fundamental insight on the Halting problem envisioned ideas that are very simple to grasp. The halting problem questions whether a given problem halts or not. Simple, isn't it? We can't run it for ages, because we are limited on time, space and money. Although sounding very simple, this problem can give insights into profound philosophies. Think about any unsolved conjecture in math. Suppose we could test out the condition for every integer or natural number through a program sequentially, such that it halts when the conjecture fails. Then deciding whether that program ever halts is equivalent to deciding the truth of the conjecture.

But since we still have many unsolved conjectures, it gives hope to believe that there exists no program to solve the halting problem. How can we even prove such a thing? Mr. Turing to the rescue! These types of problems are frequently encountered in logic and solved by explicitly constructing a contradiction against the assumption.

Say, we have a program $\mathcal{P}$ that decides whether a given program Ω halts. We try to analyse the internal dynamics of the problem underpinning some contradiction through it. Generate another program $\mathcal{R}$ through $\mathcal{P}$, such that $\mathcal{R}$ runs forever if Ω halts given its own code as input, or $\mathcal{R}$ halts if Ω runs forever given its own code as input. In an argument inspired by Russel's paradox[3], what happens if we feed the program $\mathcal{R}$ itself. For $\Omega \equiv \mathcal{R}$, the program halts if it runs forever, and runs forever if it halts. Beautiful, isn't it? These logical arguments comprise a basis of proof called *reductio ad absurdum*.

[3] Suppose a barber who shaves all men who do not shave themselves. Who shaves the barber? For a detailed insight, refer to https://en.wikipedia.org/wiki/Russell's_paradox.

A corollary to this that easily falls out is the well-acclaimed Gödel's incompleteness theorem. This is the beauty of logic! Such a profound statement about the boundaries of mathematics and, thereby, life is contained in this beautiful and intricate yet simple argument by Turing.

3.2 Circuit Model of Computation

We proceed forward to building a real computer, through ideas from the previous sections, but by introducing the *bit*, the fundamental unit of classical information. The bit is defined as a two-valued binary variable, typically encoding 0 and 1. A circuit is made of *wires* and *gates*, with each wire carrying one bit of information, and the gates performing logical operations.

Any number $N < 2^n$ can be encoded as a binary sequence of 0's and 1's as:

$$N = \sum_{k=0}^{n-1} a_k 2^k,$$

where the value of each digit $a_k \in \{0, 1\}$. We represent $N \equiv a_{n-1}a_{n-2} \ldots a_1 a_0$. The supremacy of binary would be to enable voltage based regulations for storing information. The binary operations are embodied through logical gates, which respect the Boolean algebra.

In any model of computation, we provide a n-bit input and recover a m-bit output, represented through a logical function as:

$$f : \{0, 1\}^n \rightarrow \{0, 1\}^m.$$

The universality of some elementary logical operations is to embed any operation as a series of elementary logical operations. Any function can be constructed from the elementary gates AND, OR, NOT, and FANOUT, constituting the universal set of gates for classical computation. The number of these basic gates used in a particular algorithm determines the *circuit complexity* of the algorithm.

3.3 RAM Model of Computation

The most commonly used model of computation for the analysis of algorithms is the Random-Access Machine RAM model of computation. In this book, when we talk about asymptotic bounds, unless specified otherwise, we talk about the RAM model of computation. Informally[4], the following describes the RAM model:

- Has a finite memory that is divided into units of w bits.
- We set $w = \log n$ where n is the upper bound on the size of input received or the upper bound on the size of the computation.

[4] For a more mathematically rigorous understanding, refer to the lecture notes by Jelani Nelson or the lecture video by O'Donnell

- Basic arithmetic (+,-,×, % , /), logical (NOT, AND, OR) and relational (>,<,=) operators are considered to take one time unit.
- Function call, accessing a memory location, and bitwise operations are also one time step.
- Other complex operations, which are generally composed of the above unit time operations, have correspondingly multiple time steps.

Thus, for any algorithm, the total number of time steps, calculated as mentioned above, determines the *time complexity* of the algorithm. The size of memory used by the algorithm overall is called *space complexity*.

3.4 Bird's Eye View of Complexity Theory

The ability to compute is limited by two resources: space (memory) and time. The difficulty of computing allows problems to be categorized into different complexity classes.

Consider an algorithm that takes in an input of length n (for example, the number of digits in a number). We call this a polynomial time algorithm if it doesn't take more than Cn^k steps for some fixed C, $k \geq 0$, to compute the answer. We denote this by $\mathcal{O}(n^k)$. These are considered efficient algorithms. The class of problems solved by these algorithms is called P.

Another important complexity class is called NP. This is the class of problems whose solutions can be verified in polynomial time. For example, once we find the factorization of some number $N = PQ$, we can efficiently verify that $PQ = N$. Indeed, we have $\texttt{P} \subseteq \texttt{NP}$.

Both complexity classes presented above are bounded by time. There are also a number of complexity classes bounded by space. PSPACE is such a class that contains problems that can be solved with a polynomial number of bits in input size.

For our study, there are two more complexity classes that are important. The first is the BPP, which are problems that can be solved with a bounded probability of error in polynomial time. The second one is the BQP, which is essentially the same thing on a quantum machine. Factoring numbers using Shor's algorithm is BQP.

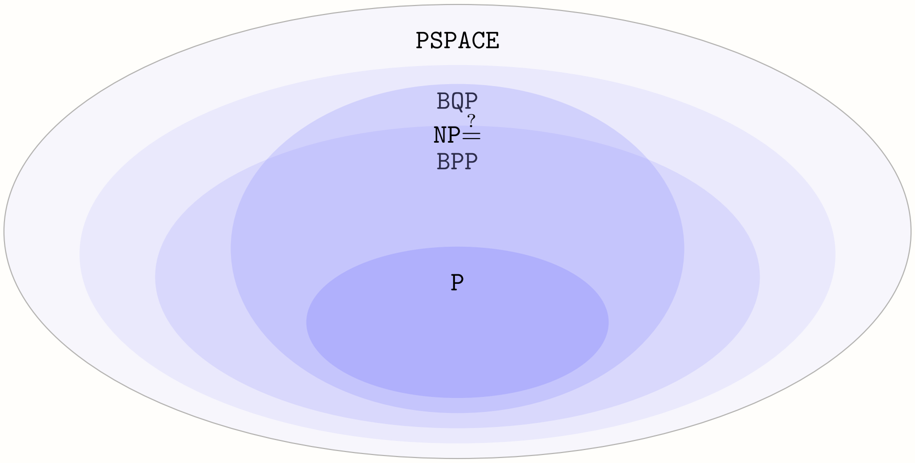
The known relationship between these complexity classes is:

$$\texttt{P} \subseteq \texttt{BPP}, \texttt{NP}, \texttt{BQP} \subseteq \texttt{PSPACE}$$

In addition, we also have $\texttt{BPP} \subseteq \texttt{BQP}$. The relationship between BPP, NP and BQP is unknown.

3.5 Church-Turing Thesis

Complexity theory classifies problems as *efficient*, which can be solved using resources that are bounded by the size of the input, and *intractable*, which are superpolynomial in the



input size. While the former is easy or feasible to solve, the latter is difficult. But how can we solve the tractable ones?

The Church-Turing thesis asserts that any model of computation can be simulated by a Turing machine, with almost a polynomial increase in the number of elementary operations involved. This profound correspondence states that if a problem cannot be solved with polynomial resources on a Turing machine, then we better lose hope!

So far, we have been seeing the story of classical computers. Can quantum computers give hope to solve even the intractable problems? The honest answer is, nobody knows. But over the past few decades, the rise of new quantum algorithms, robust quantum error correction techniques, and the advancement of qubit technology has increased the application of quantum computing to a wide range of disciplines, including machine learning, computational chemistry, biology, quantum simulation of molecules, etc, which has kept quantum computing research positive. It has also led to the introduction of brand new fields like post-quantum cryptography.

Also, it is important to remember that quantum computers won't likely replace classical computers as each of these is specialized in its own way. We will be seeing an example of this in chapter 6, how quantum computing using quantum Fourier transform can solve factoring problem which is a classical NP-hard problem, yet this algorithm can not be used to compute all Fourier coefficients of a function, a polynomial time solvable task on a classical computer. We still do not know what tasks a quantum computer is better at compared to classical computers. All we know is that for certain cherry-picked problems, we certainly have a better algorithm in the quantum world.

Thus, there are a lot of unanswered questions in the field of quantum computing and quantum information, waiting to be solved by future researchers *(intended to the readers)*!

Chapter 4

Overview of Quantum Computer and Quantum Information

> *"Computation is Physical."*
>
> – Sreejith GJ, *Lecture (Paraphrasing David Deutsch)*

4.1 Vignettes of Quantum Computing and Quantum Information

A brief history[1] of the developments leading to quantum computing and information is presented in the Table 4.1 below.

4.2 Qubit

The basic unit of information in the classical world is a bit. A bit can be 0 or 1. The quantum world analogue of a bit is a qubit. But spooky as quantum mechanics sounds, a qubit can be 0 and 1 at the same time. More precisely, it can be any linear combination of $|0\rangle$ and $|1\rangle$.

Formally, a qubit is a two-level quantum system. It resides in a 2-dimensional complex linear vector space (Hilbert space). With orthonormal basis $\{|0\rangle, |1\rangle\}$. Then the most general normalised state can be expressed as $a|0\rangle + b|1\rangle$ satisfying $|a|^2 + |b|^2 = 1$.

Consider a general qubit $|\psi\rangle$,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}\alpha\\\beta\end{pmatrix}$$

[1] Inspired from a recent talk at A Hundred Years of Quantum Mechanics, organised by ICTS, Bangalore.

1930s	Models of Computation *Classes include Recursive functions, λ-calculus, Turing Machine; early formalizations motivated by automating theorem-proving.*
1936	Church–Turing thesis *Any reasonable model of computation is equivalent in power to the Turing Machine.*
1960-70s	Quantum Communication, Cryptography, Quantum Money
1980	Computation is Physical *A paradigm shift recognizing that computation must obey the laws of physics; notions further supported by insights from information theory and thermodynamics.*
1982	Feynman's Quantum Computer *Feynman proposed using quantum systems to simulate physical processes that are infeasible for classical computers. For details, see his seminal paper Simulating Physics with Computers.*
1985	Deutsch's Physical Church–Turing thesis *Deutsch extended the Church-Turing thesis to the quantum realm, arguing that a universal quantum computer could efficiently simulate any physical process.*
1985-92	Extended Church–Turing thesis in the Classical Context *This thesis asserted that any reasonable computational model can be efficiently simulated by a probabilistic Turing machine. Early work by Deutsch and later by Josza, along with insights formalized by Bernstein and Vazirani, began to challenge this view in relativized (oracle) settings.*
1993	Bernstein-Vazirani's Quantum Turing Machine *Bernstein and Vazirani rigorously defined the Quantum Turing Machine model and demonstrated, in the oracle (or relativized) setting, a superpolynomial advantage over classical deterministic models.*
1994	Simon's Problem *Simon's algorithm provided the first exponential separation between quantum and classical query complexities, hinting at the power of quantum computation.*
1994	Shor's Algorithm *Shor introduced polynomial-time algorithms for integer factorization and discrete logarithms, exploiting periodicity via the Quantum Fourier Transform. This result, detailed in his original paper Algorithms for Quantum Computation: Discrete Logarithms and Factoring, showcased an exponential advantage over the best-known classical algorithms.*
1996	Quantum Parallelism & Formula Evaluation *Grover's algorithm showcased quantum parallelism via superposition, achieving a quadratic speedup for unstructured database search over classical randomized algorithms. This breakthrough not only demonstrated a clear quantum advantage but also inspired new modular techniques in quantum algorithm design.*

Table 4.1: Overview of the developments leading to modern-day quantum computation and information.

As $\alpha, \beta \in \mathbb{C}$ we can write them as $z_c = r_c e^{i\theta_c}$

By doing this, we get the *polar representation* of the quantum state:

$$|\psi\rangle = r_\alpha e^{i\theta_\alpha} |0\rangle + r_\beta e^{i\theta_\beta} |1\rangle$$

With some rearrangement and ignoring the overall phase (as a qubit is normalized, and during measurements and other operations, the overall phase does not matter), we get,

$$\begin{aligned}|\psi'\rangle &= e^{-i\theta_\alpha} \left(r_\alpha e^{i\theta_\alpha} |0\rangle + r_\beta e^{i\theta_\beta} |1\rangle \right) \\ &= r_\alpha |0\rangle + r_\beta e^{i(\theta_\beta - \theta_\alpha)} |1\rangle \\ &= r_\alpha |0\rangle + r_\beta e^{i\theta} |1\rangle\end{aligned}$$

$$|\alpha|^2 + |\beta|^2 = 1 \;\Rightarrow\; |r_\alpha|^2 + |x + iy|^2 = r_\alpha^2 + x^2 + y^2 = 1$$

This last equation is just a 3-dimensional sphere in real space!
Setting $z = r_\alpha$ we can write the state as,

$$\begin{aligned}|\psi'\rangle &= z|0\rangle + (x + iy)|1\rangle \\ &= \cos\theta|0\rangle + \sin\theta(\cos\phi + i\sin\phi)|1\rangle \\ &= \cos\theta|0\rangle + e^{i\phi}\sin\theta|1\rangle\end{aligned}$$

But notice that the angles are restricted,

$$0 \leq \theta \leq \pi, 0 \leq \phi \leq 2\pi$$

This gives us the general form of a qubit, which can be thought of as a point on a unit sphere specified by the coordinates (θ, ϕ)

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

This unit sphere on which the qubit lies is called the *Bloch sphere.*

Remarks. *The two orthogonal qubits* $\{|0\rangle, |1\rangle\}$ *are along the z-axis in the Bloch sphere. These two basis kets are called the* computational basis states. *Later you will learn about the* Pauli group *and realise that the computational basis is nothing but the eigen basis of the Pauli* Z *operator or the Phase gate* Z.

The state $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ *and* $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ *denoted as* $|+\rangle$ *and* $|-\rangle$ *respectively are eigen basis of Pauli* X *operator or the NOT gate* X. *And the state* $\frac{|0\rangle+i|1\rangle}{\sqrt{2}}$ *and* $\frac{|0\rangle-i|1\rangle}{\sqrt{2}}$ *denoted as* $|i\rangle$ *and* $|-i\rangle$ *respectively are eigen basis of Pauli* Y *operator or the* Y *gate which is equal to* $-iXZ$.

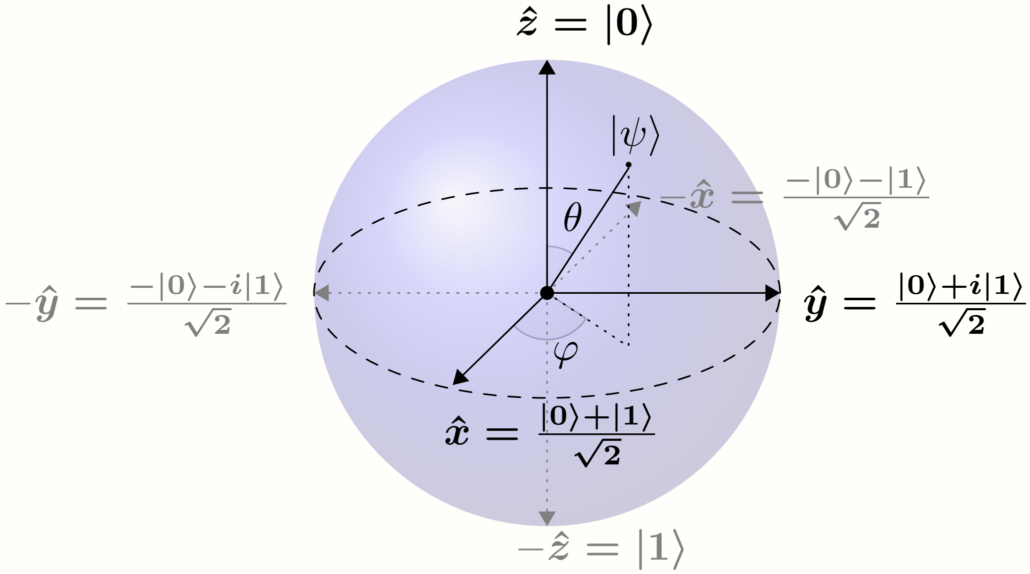

$\hat{z} = |0\rangle$
$|\psi\rangle$
$-\hat{x} = \frac{-|0\rangle - |1\rangle}{\sqrt{2}}$
θ
$-\hat{y} = \frac{-|0\rangle - i|1\rangle}{\sqrt{2}}$
$\hat{y} = \frac{|0\rangle + i|1\rangle}{\sqrt{2}}$
φ
$\hat{x} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$
$-\hat{z} = |1\rangle$

Figure 4.1: Bloch Sphere

4.3 Multi Qubits

Although a single qubit is extremely interesting, but the true power of quantum computation is unleashed when we consider multiple qubits working together. Just as classical computers use registers of many bits, quantum computers use registers of multiple qubits. The way we describe these multi-qubit systems is through a mathematical construction called the *tensor product*.

Let's consider the simplest multi-qubit system of two qubits. If the first qubit is in the state $|\psi_A\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ and the second is in the state $|\psi_B\rangle = \beta_0|0\rangle + \beta_1|1\rangle$, the state of the combined two-qubit system is given by their tensor product, denoted $|\psi_A\rangle \otimes |\psi_B\rangle$. The tensor product combines the two vector spaces into a larger one. For a single qubit, the Hilbert space is 2-dimensional ($\mathcal{H}_2$). For two qubits, the combined Hilbert space $\mathcal{H}_4 = \mathcal{H}_2 \otimes \mathcal{H}_2$ is 4-dimensional.

The tensor product is distributive, so we can expand it as follows:

$$\begin{aligned}|\psi_A\rangle \otimes |\psi_B\rangle &= (\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\beta_0|0\rangle + \beta_1|1\rangle)\\ &= \alpha_0\beta_0(|0\rangle \otimes |0\rangle) + \alpha_0\beta_1(|0\rangle \otimes |1\rangle) + \alpha_1\beta_0(|1\rangle \otimes |0\rangle) + \alpha_1\beta_1(|1\rangle \otimes |1\rangle)\end{aligned}$$

For convenience, we use a shorthand notation where $|a\rangle \otimes |b\rangle$ is written as $|ab\rangle$ or $|\psi_A\rangle|\psi_B\rangle$. Using this, the state becomes

$$\alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle$$

The four states $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ form an orthonormal basis for the two-qubit system. This generalizes powerfully: a system of n qubits is described by a state vector in a 2^n-dimensional Hilbert space. This exponential growth of the state space with the number of qubits is a key reason for the potential power of quantum computers.

It's important to remember that a general n-qubit state is a superposition of all 2^n basis states. For two qubits, an arbitrary state is:

$$|\Psi\rangle = c_{00}|00\rangle + c_{01}|01\rangle + c_{10}|10\rangle + c_{11}|11\rangle$$

where the complex coefficients must satisfy $\sum_{ij} |c_{ij}|^2 = 1$. The above 2^n basis states are also written as numbers 1 to n in base 10, corresponding to the base 2 representation. That is,

$$|\Psi\rangle = c_{00}|0\rangle + c_{01}|1\rangle + c_{10}|2\rangle + c_{11}|3\rangle$$

As we saw in the previous chapter, if a state cannot be written as a simple tensor product of its constituent parts (i.e., it is not separable), it is entangled.

In many quantum algorithms, we also make use of *ancilla qubits*. These are extra helper qubits that are used as a workspace during a computation, much like temporary variables in classical programming. They might be used to store intermediate results or to enable complex controlled operations. Typically, an ancilla qubit is initialized to a known state, like $|0\rangle$, interacts with the primary qubits of the computation through the multi-qubit operation, and is ideally returned to its initial state at the end of the algorithm so it is disentangled from the final result.

4.4 Gates and Circuits

4.4.1 Single Qubit Gates and 2-qubit Gates

As seen in the previous section, a qubit can be thought of as a unit vector in the Bloch sphere. So, what does computation mean in this setup? Given an input qubit, we can think of computation as a series of transformations done to the input qubit to get the desired output qubit state. What type of transformation? Note that the qubit is of unit norm, and norm-preserving transformations are unitary transformations. So these transformations can be captured by unitary matrices.

Some examples of single-qubit gates and two-qubit gates are given below in Fig. 4.2. Notice that all these quantum gates are unitary matrices.[2]

4.4.2 Need for n-qubit Gates?

We saw some basic single-qubit and 2-qubit gates. For a general quantum circuit with n qubits, would we need to have all possible n-qubit gates? Fortunately, we need not just keep learning multiple gates for each value of n. It turns out that arbitrary unitary

[2] For detailed explanation, refer to Quantum Computation and Quantum Information Textbook by Chuang and Nielsen.

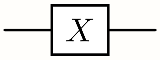
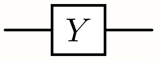
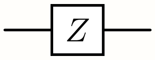
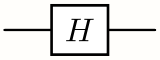
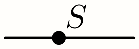
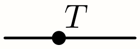
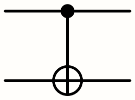
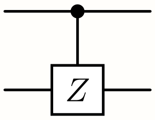
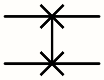
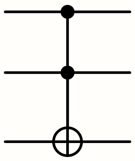
Operator	**Gate(s)**	**Matrix**
Pauli-X (X)		$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Pauli-Y (Y)		$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$
Pauli-Z (Z)		$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
Phase (S, P)		$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$
$\pi/8$ (T)		$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$
Controlled Not (CNOT, CX)		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
Controlled Z (CZ)		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$
SWAP		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$

Figure 4.2: Comprehensive list of some of the extensively used single-qubit and two-qubit gates in quantum computation.

transformations can be approximated to an arbitrary degree of precision by sufficiently many 1- and 2-qubit gates [3].

4.4.3 Constructing Arbitrary 1 and 2-qubit States

The art of quantum computation is to construct circuits out of 1 and 2-qubit gates that produce final states capable of revealing useful information when measured.
From the section 4.2, we know that for any $|\psi\rangle$ there is a 1-qubit unitary gate $\mathbf{U}$ that takes $|0\rangle$ to $|\psi\rangle$ such that $\mathbf{U}\,|0\rangle = |\psi\rangle$).
An arbitrary 2-qubit state can be written as,

$$|\Psi\rangle = \alpha_{00}|00\rangle + \alpha_{0,}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle.$$

This is of the form

$$|\Psi\rangle = |0\rangle \otimes |\psi\rangle + |1\rangle \otimes |\phi\rangle,$$

where

$$|\psi\rangle = \alpha_{00}|0\rangle + \alpha_{01}|1\rangle, \quad |\phi\rangle = \alpha_{10}|0\rangle + \alpha_{11}|1\rangle.$$

Now consider an unitary $\mathbf{U} \otimes \mathbb{I}$ acting on $|\Psi\rangle$ such that

$$\begin{aligned}(\mathbf{U} \otimes \mathbb{I})|\Psi\rangle &= \mathbf{U}|0\rangle \otimes |\psi\rangle + \mathbf{U}|1\rangle \otimes |\phi\rangle \\ &= (a|0\rangle + b|1\rangle) \otimes |\psi\rangle + (-b^*|0\rangle + a^*|1\rangle)\,|\phi\rangle \\ &= |0\rangle\,|\psi'\rangle + |1\rangle\,|\phi'\rangle\,.\end{aligned}$$

where $|\psi'\rangle = a|\psi\rangle - b^*|\phi\rangle$ and $|\phi'\rangle = b|\psi\rangle + a^*|\phi\rangle$.
Note that $\mathbf{U}$ is a unitary that we are constructing. Thus we can pick a and b such that we make $|\psi'\rangle$ and $|\phi'\rangle$ orthogonal. And choose λ and μ such that we make $|\psi''\rangle = \frac{|\psi'\rangle}{\lambda}$, $|\phi''\rangle = \frac{|\phi'|}{\mu}$ unit vectors.

As $|\psi''\rangle\,, |\phi''\rangle$ are orthonormal, they are related to $|0\rangle$ and $|0\rangle$ by unitary transformation.

$$|\psi''\rangle = V|0\rangle \quad |\phi''\rangle = V|1\rangle.$$

Now we can re-write the equations as

$$\begin{aligned}(\mathbf{U} \otimes \mathbb{I})|\Psi\rangle &= |0\rangle\,|\psi'\rangle + |1\rangle\,|\phi'\rangle \\ &= \lambda\,|0\rangle\,|\psi''\rangle + \mu\,|1\rangle\,|\phi''\rangle \\ (\mathbf{U} \otimes \mathbb{I})|\Psi\rangle &= \mathcal{U}|\Psi\rangle \text{ (say)} \\ \mathcal{U}\mathcal{U}^\dagger\,|\Psi\rangle &= |\Psi\rangle \text{ (as } \mathcal{U} \text{ is unitary)} \\ \implies |\Psi\rangle &= \mathcal{U}^\dagger(|0\rangle\,|\psi'\rangle + |1\rangle\,|\phi'\rangle) = (\mathcal{U}^\dagger \otimes V)(\lambda\,|00\rangle + \mu\,|11\rangle)\end{aligned}$$

(since $|\psi''\rangle$ and $|\phi''\rangle$ can be got from $|0\rangle$ and $|1\rangle$ by an unitary transformation)

$$\begin{aligned}&= (\mathcal{U}^\dagger \otimes V)(C_{10}(\lambda\,|0\rangle + \mu\,|1\rangle) \otimes |0\rangle) \\ &= (\mathcal{U}^\dagger \otimes V)(C_{10}(W\,|0\rangle) \otimes |0\rangle) \\ |\Psi\rangle &= \mathcal{U}^\dagger V C_{10} W\,|00\rangle\,.\end{aligned}$$

[3] The argument is given by David P. DiVincenzo in *Physical Review A 51, 1015–1022 (1995)*.

Here C_{10} is cNOT gate with 1st qubit as target and second as control and W is yet another unitary. Therefore 3-1 qubit unitary and 1 NOT gate is enough to get any general 2 -qubit state from $|00\rangle$.

4.5 Reversible Computation

As defined in chapter 3, *algorithm* is a set of instructions for solving a problem. One can think of the classical computer program as given an input x it performs the necessary instructions and outputs $f(x)$. Similarly we would expect to design a quantum computer to act on x and produce the necessary $f(x)$.

Suppose we wish to compute a function $f : \{0,1\}^n \to \{0,1\}^m$. In a quantum computer, we need at least $n+m$ qubits to compute f. Like the Turing Machine, which overwrites on the input tape, why can not we use less than $n+m$ qubits to compute f? One important reason why this can not be done is that if f assigns the same value to different values of x then this computation cannot be inverted if its only effect is to transform the contents of a single register from x to $f(x)$. Thus, the reversibility constraint forces us to have at least $n+m$ qubits to compute f. Thus, computing f is the same as applying a unitary $\mathbf{U}_f$ on the $n+m$ qubits.

$\mathbf{U}_f$ is defined by specifying its action on the basis states; by linearity, this can be extended to any arbitrary superposition of the basis vectors. The standard quantum computation protocol defines the action of $\mathbf{U}_f$ on the computational basis $|x\rangle_n |y\rangle_m$ of the $n+m$ qubits making the *input* and *output* registers as follows:

$$\mathbf{U}_f(|x\rangle_n |y\rangle_m) = |x\rangle_n |y \oplus f(x)\rangle_m$$

where $\oplus$ indicates modulo-2 bitwise addition (without carrying) or exclusive OR operation.

If the initial value represented by the output register is $y = 0$ then we have

$$\mathbf{U}_f(|x\rangle_n|0\rangle_m) = |x\rangle_n|f(x)\rangle_m$$

and we end up with $f(x)$ in the output register. Regardless of the initial value of y, the input register remains in its initial state $|x\rangle_n$.

The transformation $\mathbf{U}_f$ is clearly invertible. Also note that $\mathbf{U}_f$ is its own inverse:

$$\mathbf{U}_f\mathbf{U}_f(|x\rangle|y\rangle) = \mathbf{U}_f(|x\rangle|y \oplus f(x)\rangle)$$

$$= |x\rangle|y \oplus f(x) \oplus f(x)\rangle = |x\rangle|y\rangle,$$

since $z \oplus z = 0$ for any z.

Thus, $\mathbf{U}_f$ gives a generic way to construct any function f on a quantum computer.

4.5.1 Landauer's Principle

Though it appears as if some mental gymnastics have to be done to compute $f(x)$ by first defining $\mathbf{U}_f$ in a quantum computer, this has given an intrinsic advantage to quantum computers over classical irreversible computation.

Rolf Landauer pointed out in 1961 that erasure of information is necessarily a dissipative process. His insight is that erasure always involves the compression of phase space and so is irreversible.

For example, one can store a bit of information by placing a single molecule in a box, either on the left side or the right side of a partition that divides the box. Erasure means that we move the molecule to the left side (say) irrespective of whether it started out on the left or right. We can suddenly remove the partition and then slowly compress the one-molecule "gas" with a piston until the molecule is definitely on the left side. This procedure reduces the entropy of the gas by $\Delta S = k \ln 2$, and there is an associated flow of heat from the box to the environment. If the process is isothermal at temperature T, then work $W = kT \ln 2$ is performed on the box, work that we have to provide. So if one wants to erase information, someone will have to pay the power bill for it. In this aspect, reversible computation, which does not involve the erasure of information, is energy-effective.

The logic gates used to perform classical computation are typically irreversible, e.g., the NAND gate

$$(a, b) \rightarrow \neg(a \wedge b)$$

has two input bits and one output bit, and we can't recover a unique input from the output bit. According to Landauer's principle, since about one bit is erased by the gate (averaged over its possible inputs), at least work $W = kT \ln 2$ is needed to operate the gate. If we have a finite supply of batteries, there appears to be a theoretical limit to how long a computation we can perform.

But Charles Bennett found in 1973 that any computation can be performed using only reversible steps, and so, in principle, requires no dissipation and no power expenditure. We can actually construct a reversible version of the NAND gate that preserves all the information about the input: For example, the (Toffoli) gate

$$(a, b, c) \rightarrow (a, b, c \oplus a \wedge b)$$

is a reversible 3-bit gate that flips the third bit if the first two both take the value 1 and does nothing otherwise. The third output bit becomes the NAND of a and b if $c = 1$. We can transform an irreversible computation to a reversible one by replacing the NAND gates by Toffoli gates. This computation could in principle be done with negligible dissipation.

However, in the process we generate a lot of extra junk, and one wonders whether we have only postponed the energy cost; we'll have to pay when we need to erase all the junk. Bennett addressed this issue by pointing out that a reversible computer can run forward to the end of a computation, print out a copy of the answer (a logically reversible operation) and then reverse all of its steps to return to its initial configuration. This procedure removes the junk without any energy cost.

In principle, then, we need not pay any power bill to compute. In practice, the (irreversible) computers in use today dissipate orders of magnitude more than $kT \ln 2$ per gate, anyway, so Landauer's limit is not an important engineering consideration. But as computing hardware continues to shrink in size, it may become important to beat Landauer's limit to prevent the components from melting, and then reversible computation may be the only option.

4.5.2 Maxwell's Demon

The insights of Landauer and Bennett led Bennett in 1982 to the reconciliation of Maxwell's demon with the second law of thermodynamics. Maxwell had envisioned a gas in a box, divided by a partition into two parts A and B. The partition contains a shutter operated by the demon. The demon observes the molecules in the box as they approach the shutter, allowing fast ones to pass from A to B, and slow ones from B to A. Hence, A cools and B heats up, with a negligible expenditure of work. Heat flows from a cold place to a hot place at no cost, in apparent violation of the second law.

The resolution is that the demon must collect and store information about the molecules. If the demon has a finite memory capacity, he cannot continue to cool the gas indefinitely; eventually, information must be erased. At that point, we finally pay the power bill for the cooling we achieved. (If the demon does not erase his record, or if we want to do the thermodynamic accounting before the erasure, then we should associate some entropy with the recorded information.)

These insights were largely anticipated by Leo Szilard in 1929; he was truly a pioneer of the physics of information. Szilard, in his analysis of the Maxwell demon, invented the concept of a bit of information, (the name "bit" was introduced later, by Tukey) and associated the entropy $\Delta S = k \ln 2$ with the acquisition of one bit (though Szilard does not seem to have fully grasped Landauer's principle, that it is the erasure of the bit that carries an inevitable cost).[4]

4.5.3 Classical Reversible Computation v.s Quantum Computing

The only nontrivial reversible operation a classical computer can perform on a single bit is the NOT operation **X**. Far more operations are possible on a single qubit. The reversible operations that a quantum computer can perform upon a single qubit are represented by the action on the state of the qubit of any linear transformation that takes unit vectors into unit vectors. Such transformations **U** are called unitary and satisfy the condition

$$\mathbf{U}\mathbf{U}^\dagger = \mathbf{U}^\dagger\mathbf{U} = 1$$

Since any unitary transformation has a unitary inverse, such actions of a quantum computer on a qubit are reversible. The reason why reversibility is crucial for the effective functioning

[4] These examples illustrate that work at the interface of physics and information has generated noteworthy results of interest to both physicists and computer scientists.

of a quantum computer will emerge in Chapter 2.

The most general reversible n-bit operation in a classical computer is a permutation of the $(2^n)!$ different classical-basis states. The most general reversible operation that a quantum computer can perform upon n qubits is represented by the action on their state of any linear transformation that takes unit vectors into unit vectors - i.e. any 2^n-dimensional unitary transformation **U**, satisfying

$$\mathbf{U}\mathbf{U}^\dagger = \mathbf{U}^\dagger\mathbf{U} = 1$$

Any reversible operation on n bits - i.e. any permutation **P** of the 2^n bit states - can be associated with a unitary operation **U** on n qubits. One defines the action of **U** on the classical-basis states of the qubit to be identical to the operation of **P** on the corresponding classical states of the bit. Since the classical basis is a basis, **U** can be extended to arbitrary n-qubit states by requiring it to be linear. Since the action of **U** on the classical-basis states is to permute them, its effect on any superposition of such states $\sum \alpha_x |x\rangle_n$ is to permute the amplitudes α_x. Such a permutation preserves the value of $\sum |\alpha_x|^2$, so U takes unit vectors into unit vectors. Being norm-preserving and linear, **U** is indeed unitary.

Many important unitary operations on qubits that we shall be examining below are defined in this way, as permutations of the classical-basis states, which are implicitly understood to be extended by linearity to all qubit states. In particular, the transformations NOT, SWAP, and cNOT on bits are immediately defined in this way for qubits as well. But the available unitary transformations on qubits are, of course, much more general than straightforward extensions of classical operations. We have already encountered two such examples, the operator **Z** and the Hadamard transformation **H**. Both of these take the classical-basis states of a qubit into another orthonormal basis, so their linear extensions to all qubit states are necessarily unitary.

In designing quantum algorithms, the class of allowed unitary transformations is almost always restricted to ones that can be built entirely out of products of unitary transformations that act on only one qubit at a time, called 1-qubit gates, or that act on just a pair of qubits, called 2-qubit gates. This restriction is imposed because the technical problems of making higher-order quantum gates are even more formidable than the (already difficult) problems of constructing reliable 1-and 2-qubit gates.

4.6 Quantum Parallelism

In chapter 3, we saw the powers Turing Machines have - anything that can be computed can be captured by a Turing machine. *Turing completeness* is the ability for a computational model or a system of instructions to simulate a Turing machine. A programming language that is Turing Complete is theoretically capable of expressing all tasks accomplished by computers; nearly all classical programming languages are Turing Complete if the limitations of finite memory are ignored. So, given that everything of interest can be computed on a classical computer, why are we so keen on building a quantum computer? Even energy dissipation can be avoided by classical reversible computation.

To understand what quantum computers can offer more than a classical computer, one has to understand what are the unique features of quantum mechanics that are not described in classical mechanics. Many would answer this with two keywords, *superposition* and *entanglement.*

Consider a two-qubit state $|0\rangle|0\rangle$. If we apply to each qubit the 1-qubit Hadamard transformation $\mathbf{H}$, then we get

$$\begin{aligned}(\mathbf{H}\otimes\mathbf{H})(|0\rangle\otimes|0\rangle) &= \mathbf{H}_1\mathbf{H}_0|0\rangle|0\rangle = (\mathbf{H}|0\rangle)(\mathbf{H}|0\rangle)\\ &= \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\\ &= \frac{1}{2}(|0\rangle|0\rangle+|0\rangle|1\rangle+|1\rangle|0\rangle+|1\rangle|1\rangle)\\ &= \frac{1}{2}(|0\rangle_2+|1\rangle_2+|2\rangle_2+|3\rangle_2).\end{aligned}$$

This clearly generalizes to the n-fold tensor product of n Hadamards, applied to the n-qubit state $|0\rangle_n$:

$$\mathbf{H}^{\otimes n}|0\rangle_n = \frac{1}{2^{n/2}}\sum_{0\le x<2^n}|x\rangle_n,$$

where

$$\mathbf{H}^{\otimes n} = \mathbf{H}\otimes\mathbf{H}\otimes\cdots\otimes\mathbf{H}, \quad n \text{ times}.$$

So if the initial state of the input register is $|0\rangle_n$ and we apply an n-fold Hadamard transformation to that register, its state becomes an equally weighted superposition of all possible n-qubit inputs. If we then apply $\mathbf{U}_f$ to that superposition, with 0 initially in the output register, then by linearity, we get

$$\begin{aligned}\mathbf{U}_f(\mathbf{H}^{\otimes n}\otimes\mathbf{1}_m)(|0\rangle_n|0\rangle_m) &= \frac{1}{2^{n/2}}\sum_{0\le x<2^n}\mathbf{U}_f(|x\rangle_n|0\rangle_m)\\ &= \frac{1}{2^{n/2}}\sum_{0\le x<2^n}|x\rangle_n|f(x)\rangle_m.\end{aligned}$$

So if we have a mere hundred qubits in the input register, initially all in the state $|0\rangle_{100}$ (and m more in the output register), if a hundred Hadamard gates act on the input register before the application of $\mathbf{U}_f$, then the form of the final state contains the results of $2^{100}\approx 10^{30}$ evaluations of the function f. A billion billion trillion evaluations! This apparent miracle is called *quantum parallelism.*

So, can we now claim we have successfully computed $f(x)$ for all x in one run of our quantum circuit? The answer is no! And this clearly shows why quantum computing is not parallel computing.

Though $\frac{1}{2^{n/2}}\sum_{0\le x<2^n}|x\rangle_n|f(x)\rangle_m$ holds the superposition of all values of $f(x)$ after measurement the state of the registor reduces to $|x_0\rangle\,|f(x_0)\rangle$ and we no longer can know about

$f(x)$ for any other x other than x_0.

What if we now make a sufficiently large number of copies of the output register and measure all to get $f(x)$ at all x without running the whole computation over again? Unfortunately, this is also not possible, and we will see why in the next section.

4.7 No-Cloning Theorem

Theorem 4.7.1. *(No-Cloning Theorem) There exist no unitary* $\mathbf{U}$ *such that* $\mathbf{U}(|\psi\rangle\,|0\rangle) = |\psi\rangle\,|\psi\rangle$

Proof. If

$$\mathbf{U}(|\psi\rangle|0\rangle) = |\psi\rangle|\psi\rangle \quad \text{and} \quad \mathbf{U}(|\phi\rangle|0\rangle) = |\phi\rangle|\phi\rangle$$

then it follows from linearity that

$$\mathbf{U}(a|\psi\rangle + b|\phi\rangle)|0\rangle = a\mathbf{U}|\psi\rangle|0\rangle + b\mathbf{U}|\phi\rangle|0\rangle = a|\psi\rangle|\psi\rangle + b|\phi\rangle|\phi\rangle$$

But if $\mathbf{U}$ cloned arbitrary inputs, we would have

$$\begin{aligned}\mathbf{U}(a|\psi\rangle + b|\phi\rangle)|0\rangle &= (a|\psi\rangle + b|\phi\rangle)(a|\psi\rangle + b|\phi\rangle)\\ &= a^2|\psi\rangle|\psi\rangle + b^2|\phi\rangle|\phi\rangle + ab|\psi\rangle|\phi\rangle + ab|\phi\rangle|\psi\rangle\end{aligned}$$

Notice that these cross terms are missing in $\mathbf{U}|\phi\rangle|0\rangle = a|\psi\rangle|\psi\rangle + b|\phi\rangle|\phi\rangle$. These two equations are equal only when either a or b is zero. Thus, for an arbitrary state, we do not have any unitary that can copy that state. ■

Not only that, we can prove a stronger claim that we can not even copy an arbitrary state to a reasonable degree of approximation.

Theorem 4.7.2. *There exists no unitary* $\mathbf{U}$ *that can approximately clone an arbitrary state. That is there is no unitary* $\mathbf{U}$ *such that* $\mathbf{U}(|\psi\rangle\,|0\rangle) \approx |\psi\rangle\,|\psi\rangle$

Proof. Suppose that $\mathbf{U}$ approximately cloned both $|\phi\rangle$ and $|\psi\rangle$:

$$\mathbf{U}(|\psi\rangle|0\rangle) \approx |\psi\rangle|\psi\rangle \text{ and } \mathbf{U}(|\phi\rangle|0\rangle) \approx |\phi\rangle|\phi\rangle$$

Then, since unitary transformations preserve inner products, since the inner product of a tensor product of states is the (ordinary) product of their inner products, and since $\langle 0 \mid 0\rangle = 1$, it follows that

$$\langle \phi \mid \psi\rangle \approx \langle \phi \mid \psi\rangle^2$$

But this requires $\langle \phi \mid \psi\rangle$ to be either close to 1 or close to 0 . Hence a unitary transformation can come close to cloning both of two states $|\psi\rangle$ and $|\phi\rangle$ only if the states are very nearly the same, or very close to being orthogonal. In all other cases at least one of the two states will be badly copied. ■

If this were the whole picture, then we would not be having researchers and companies interested in quantum computing. Though we can not get all the values of $f(x)$ we can do something clever and interesting. Here, the skill and art of algorithmic thinking come into play. Alongside $\mathbf{U}_f$, one could add several other unitaries cleverly such that when the final measurement is done, one could extract useful information about *relations* between the values of f for several different x, which a classical computer could get only by making several independent evaluations. The price one inevitably pays for this relational information is the loss of the possibility of learning the actual value $f(x)$ for any individual x. This tradeoff of one kind of information for another is typical of quantum computation and typical of quantum physics in general, where it is called the uncertainty principle. The principle was first enunciated by Werner Heisenberg in the context of mechanical information - the position of a particle versus its momentum.

In the following chapters, we will have a glimpse into the art of designing algorithms in quantum computers with an advantage over their classical counterparts.

4.8 Building a Qubit

Among the diverse hardware platforms being explored for quantum computing, superconducting circuit architectures have emerged as a front-runner. These systems, formally known as *circuit quantum electrodynamics* (cQED)[5] devices, harness the quantum dynamics of microwave photons and electrical currents within superconducting circuits to create, control, and read out quantum information. Their key advantage lies in their design flexibility and potential for scaling to large numbers of qubits.

Circuit QED devices are most often formed by embedding a special kind of superconducting device, known as a Josephson junction, into complex systems of circuitry, further embedded with superconductors for minimising dissipative losses. Josephson junction can be intuitively imagined as a non-linear induction circuit, which helps in the physical realisation of quantum states. It is made by sandwiching a thin layer of a nonsuperconducting material between two layers of superconducting material.

To appreciate the role of the Josephson junction, one must first grasp the fundamentals of superconductivity. When certain metals and alloys are cooled to extremely low temperatures (typically within a few degrees of absolute zero), they undergo a phase transition. At a specific critical temperature, the material shifts from its normal, resistive state to a superconducting state, where direct electrical current flows with zero resistance. Below this temperature, the subtle interaction between electrons and the crystal lattice of the metal becomes attractive, allowing electrons to overcome their natural repulsion and bind together into what are known as Cooper pairs.

The formation of Cooper pairs opens an energy gap, creating a collective, macroscopic quantum state, a superfluid of charge that moves without resistance. It is by quantising the collective electrical degrees of freedom of this superfluid, such as the number of Cooper

[5] For an excellent in-depth review, see https://arxiv.org/pdf/2106.11352.

pairs on an isolated superconducting island or the magnetic flux threading a loop, that we can engineer a robust two-level quantum system: the physical realisation of a qubit.

Numerous types of superconducting qubits exist, including the charge qubit, flux qubit, phase qubit, and fluxonium, each differing in its design and energy scales. A particularly interesting variant is the transmon, a type of charge qubit formed by two superconducting islands connected by a Josephson junction. In its simplest form, the transmon is an LC oscillator, a parallel combination of a capacitor and an inductor.

However, a simple LC circuit is a harmonic oscillator, characterised by an infinite ladder of equally spaced energy levels. This is unsuitable for a qubit, as a control signal intended for one transition would excite all of them. This is where the Josephson junction's nonlinearity becomes critical. By replacing the standard linear inductor with a junction, the circuit's potential energy is no longer a simple quadratic function but instead follows a cosine dependence on the magnetic flux. This property, known as anharmonicity, breaks the uniform spacing of the energy levels. It ensures that the energy gap between the ground state ($|0\rangle$) and the first excited state ($|1\rangle$) is unique, allowing us to selectively address the $|0\rangle \leftrightarrow |1\rangle$ transition with precisely tuned microwave pulses, thereby realizing a high-fidelity qubit.

Part II

Quantum Computing

Chapter 5

Basic Quantum Algorithms

> *"For me, great algorithms are the poetry of computation. Just like verse, they can be terse, allusive, dense, and even mysterious. But once unlocked, they cast a brilliant new light on some aspect of computing."*
>
> – Francis Sullivan, *The Joy of Algorithms*

5.1 Some Basic Functions

As we saw in chapter 4 the way to construct a function f is by constructing a unitary $\mathbf{U}_f$ such as the one shown in figure 5.1. If $y = 0$ then the output register holds $|x\rangle \otimes |f(x)\rangle$, thus we have computed $f(x)$. In this section, we will see how to design $\mathbf{U}_f$ for some basic functions f. Also recall that the function is from $\{0,1\}^n$ to $\{0,1\}$.

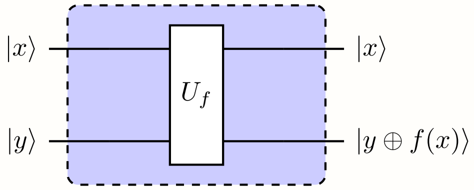


Figure 5.1: Unitary to represent a function

5.1.1 Constant Function

Let's first look at the following table:
In order to construct any function, it helps to write such a table and then think what should $\mathbf{U}_f$ be so that the output register is of the form $|x\rangle \otimes |y \oplus f(x)\rangle$.

x	$f(x)$	$0 \oplus f(x)$	$1 \oplus f(x)$
0	0	0	1
1	0	0	1

Table 5.1: $f(x) = 0$

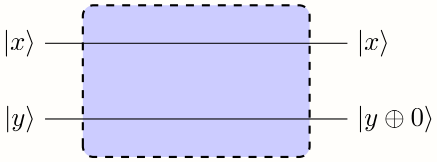


Figure 5.2: Circuit representing constant function $f(x) = 0$

Now, it is easy to see why $\mathbf{U}_f = \mathbb{I}$ works in this case.

What if f(x) is 1?

Doing the same exercise done for $f(x) = 0$ again we find:

x	$f(x)$	$0 \oplus f(x)$	$1 \oplus f(x)$
0	1	1	0
1	1	1	0

Table 5.2: $f(x) = 1$

Thus, the same circuit given for $f(x) = 0$ works for $f(x) = 1$.

5.1.2 Identity Function

Again, let's look at the table:

x	$f(x)$	$0 \oplus f(x)$	$1 \oplus f(x)$
0	0	0	1
1	1	1	0

Table 5.3: $f(x) = x$

Now, it is easy to see why $\mathbf{U}_f =$ cNOT works in this case.

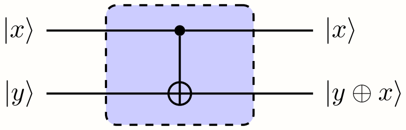


Figure 5.3: Circuit representing identity function $f(x) = x$

Is cloning achieved with the Identity function?

The answer is No! If f is the *identity function*, $f(x) = x$. Thereby, the unitary is

$$\mathbf{U}_f(|x\rangle|y\rangle) = |x\rangle|x\rangle$$

Now, let $|x\rangle = a|0\rangle + b|1\rangle$. We have the action of the unitary,

$$\begin{aligned}\mathbf{U}_f|x\rangle|0\rangle = \mathbf{U}_f((a|0\rangle + b|1\rangle)(|0\rangle)) &= \mathbf{U}_f(a|00\rangle + b|10\rangle)\\ &= a|0\rangle|0 \oplus f(0)\rangle + b|1\rangle \mid 0 \oplus f(0)\\ &= a|0\rangle|0\rangle + b|1\rangle|1\rangle = (a|00\rangle + a|11\rangle)\end{aligned}$$

However, for the product state,

$$(a|0\rangle + b|1\rangle)(a|0\rangle + b|1\rangle) = \big(a^2|00\big) + ab|01\rangle + ba|10\rangle + b^2|11\rangle$$

Thereby, we do not have the cross terms from the unitary. Therefore, we have not cloned $|x\rangle$.

5.1.3 Swap

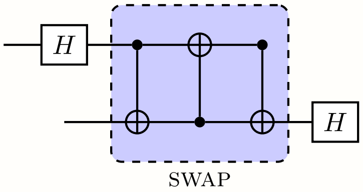


Figure 5.4: Circuit representing Swap operation

5.1.4 Is such a unitary always possible?

The computational process generally requires more than $n + m$ registers, the workspace register. A quantum unitary $\mathbf{W}_f$ applies on *all* the registers, say $n + m + r$, the input, output, and workspace. In general, the input and output registers will become entangled with the states of the additional r qubits and cannot even be assigned a state, in this case, we cannot hat a unitary $\mathbf{U}_f$ that relates only the input and output state like shown in 5.1.

If the action of the computer on all $n+m+r$ qubits has a special form such that at the end of the computation, the workspace registers are not entangled with the input-output qubits and have a state that is independent of the initial state of the input and output qubit, the having such unitary $\mathbf{U}_f$ is possible. One can achieve this by simply taking advantage of the fact that unitary transformational are reversible.

- Apply an unitary $\mathbf{V}_f$ only on $n+r$ qubits and computer $f(x)$ in m-qubits of the $n+r$ qubits. As the m output qubits are untouched, they are not entangled with the input and workspace qubits.
- Change output register y to $y \oplus f(x)$ by applying m cNOT gates,$\mathbf{C}_m$.
- Since the state of the $n+r$ Qbits is not altered by the application of $\mathbf{C}_m$, we can inverse the transformation $\mathbf{V}^\dagger$ to restore them to their original state.

Note that in the above process (also shown in figure 5.5), as the workspace registers are restored back, they are neither entangled nor dependent on the input and output states. Thus, we can safely use the above trick and talk about $\mathbf{U}_f$ (as in figure 5.1) every time.

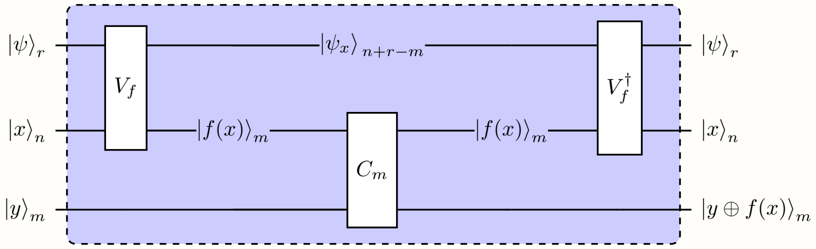


Figure 5.5: Circuit to unentangle workspace registers

5.2 Deutsch's Algorithm

Problem Statement: Given a function $f : \{0,1\} \to \{0,1\}$ find if f is a constant function or not.

Note that there are only 4 possible functions $f : \{0,1\} \to \{0,1\}$ represented by the 4 circuits given below 5.7. Suppose we are given this $\mathbf{U}_f$ as a black box; that is, we do not know which one of these four is our function; how will we find f? We can let the black box act twice, once on the state $|0\rangle\,|0\rangle$ and once on $|1\rangle\,|0\rangle$ and find f. Similarly, when we have a classical black box that computes the value of f, we need to query it twice to find $f(0)$ and $f(1)$ in order to say if the function is constant or not. One can not do any better on a classical computer. Can we do better, that is, find if $f(0)$ and $f(1)$ are different just with one query to $\mathbf{U}_f$?

By creating an equal superposition of $|0\rangle$ and $|1\rangle$, we can create an equal superposition of $f(0)$ and $f(1)$ with one call to the oracle. But how do we capture whether the functional values are the same or different? Note that if you measure the state now, it collapses to either $f(0)$ or $f(1)$, and from the 4 circuits in figure 5.7, we can only narrow down to 2, but still, we have equal probability of the function being constant or not.

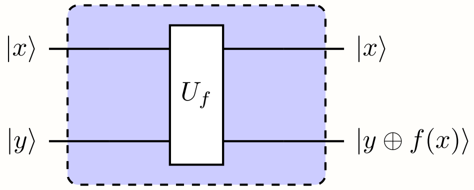


Figure 5.6: Deutsch's Algorithm

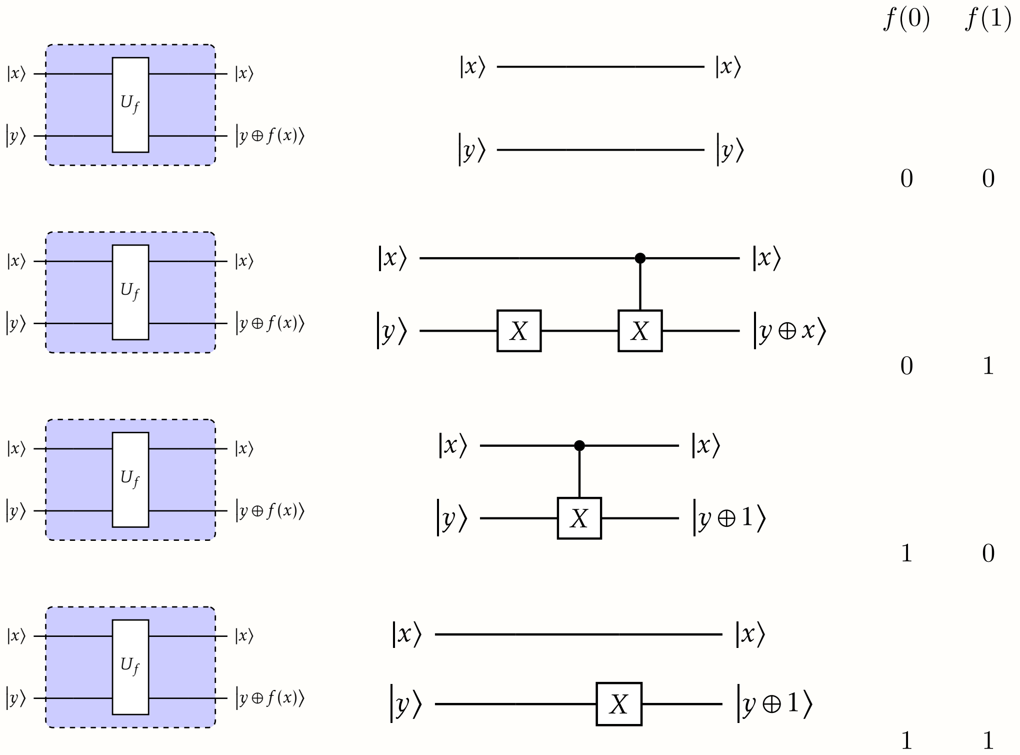


Figure 5.7: Circuits showing all possible functions $f : \{0,1\} \to \{0,1\}$.

Instead, if we do,

$$(\mathbf{H}\otimes\mathbf{H})(\mathbf{X}\otimes\mathbf{X})(|0\rangle|0\rangle) = (\mathbf{H}\otimes\mathbf{H})(|1\rangle|1\rangle) = \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)$$
$$= \frac{1}{2}(|0\rangle|0\rangle - |1\rangle|0\rangle - |0\rangle|1\rangle + |1\rangle|1\rangle).$$

And now apply $\mathbf{U}_f$

$$\frac{1}{2}\Big(\mathbf{U}_f(|0\rangle|0\rangle) - \mathbf{U}_f(|1\rangle|0\rangle) - \mathbf{U}_f(|0\rangle|1\rangle) + \mathbf{U}_f(|1\rangle|1\rangle)\Big).$$

$$\frac{1}{2}\Big(|0\rangle|f(0)\rangle - |1\rangle|f(1)\rangle - |0\rangle|\tilde{f}(0)\rangle + |1\rangle|\tilde{f}(1)\rangle\Big),$$

where, $\tilde{x} = 1\oplus x$ so that $\tilde{1} = 0$ and $\tilde{0} = 1$, and $\tilde{f}(x) = 1\oplus f(x)$. So if $f(0) = f(1)$ the output state is

$$\frac{1}{2}(|0\rangle - |1\rangle)(|f(0)\rangle - |\tilde{f}(0)\rangle), \quad f(0) = f(1),$$

but if $f(0) \neq f(1)$ then $f(1) = \tilde{f}(0)$, $\tilde{f}(1) = f(0)$, and the output state becomes

$$\frac{1}{2}(|0\rangle + |1\rangle)(|f(0)\rangle - |\tilde{f}(0)\rangle), \quad f(0) \neq f(1). \tag{2.21}$$

If, finally, we apply a Hadamard transformation to the input register, these become

$$|1\rangle\frac{1}{\sqrt{2}}(|f(0)\rangle - |\tilde{f}(0)\rangle), \quad f(0) = f(1), \tag{2.22}$$

$$|0\rangle\frac{1}{\sqrt{2}}(|f(0)\rangle - |\tilde{f}(0)\rangle), \quad f(0) \neq f(1). \tag{2.23}$$

Therefore, the overall action of these gates is,

$$(\mathbf{H}\otimes\mathbf{1})\mathbf{U}_f(\mathbf{H}\otimes\mathbf{H})(\mathbf{X}\otimes\mathbf{X})(|0\rangle|0\rangle)$$
$$= \begin{cases} |1\rangle\frac{1}{\sqrt{2}}(|f(0)\rangle - |\tilde{f}(0)\rangle), & f(0) = f(1), \\ |0\rangle\frac{1}{\sqrt{2}}(|f(0)\rangle - |\tilde{f}(0)\rangle), & f(0) \neq f(1). \end{cases}$$

Thus, the state of the input register determines whether the function is constant or not.

Remarkably, with a quantum computer, we did not have to run $\mathbf{U}_f$ twice to determine whether or not f is constant. We could do this in a single run. Interestingly, when we did this, we learned nothing about the individual values of $f(0)$ and $f(1)$, but we were nevertheless able to answer the question about their relative values: whether or not they are the same. Thus, we get less information than we get in answering the question with a classical computer, but by renouncing the possibility of acquiring that part of the information that is irrelevant to the question we wish to answer, we can get the answer with only a single application of the black box.

5.2.1 Circuit Theory Approach

In figure 5.7, we saw the equivalent circuit representation of the 4 possible functions $f : \{0,1\} \rightarrow \{0,1\}$. Applying Hadamard gates to each qubit, both before and after the application of $\mathbf{U}_f$, must produce exactly the same result as it would if the Hadamards were applied to the equivalent circuits. After applying Hadamad, the resulting circuit will look like the circuit shown in figure 5.8

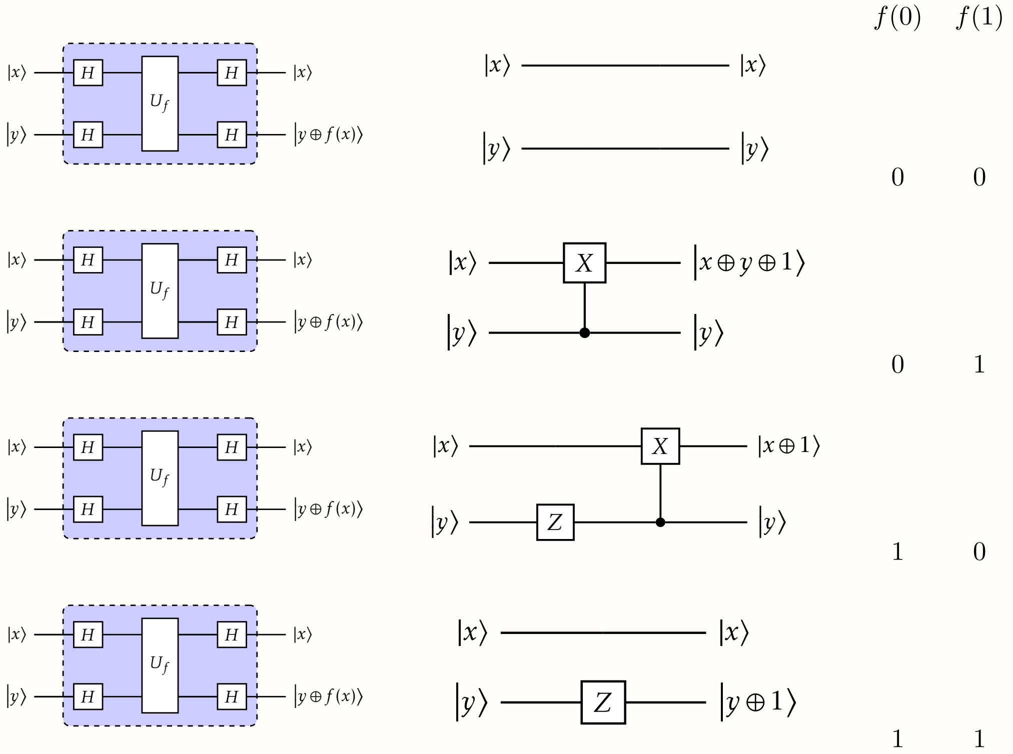


Figure 5.8: Equivalent circuit for Deutsch algorithm

The figure 5.8 shows explicitly that when $\mathbf{U}_f$ is sandwiched between Hadamards, the input register ends up in the state $|0\rangle$ if $f(0) = f(1)$ and in state $|1\rangle$ if $f(0) \neq f(1)$

Hadamard swaps the control and target

Notice that in the circuit 5.8 after application of Hadamard the control and target qubits of the cNOT gate has swapped.

5.3 Deutsch–Jozsa algorithm

Hadamard on n-qubits

The action of $\mathbf{H}$ on a single qubit can be compactly summarized as

$$\mathbf{H}|x\rangle_1 = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^x|1\rangle\right) = \frac{1}{\sqrt{2}}\sum_{y=0}^{1}(-1)^{xy}|y\rangle$$

If we apply $\mathbf{H}^{\otimes n}$ to an n-qubit computational-basis state $|x\rangle_n$ we can therefore express the result as

$$\begin{aligned}\mathbf{H}^{\otimes n}|x\rangle_n &= \frac{1}{2^{n/2}}\sum_{y_{n-1}=0}^{1}\cdots\sum_{y_0=0}^{1}(-1)^{\sum_{j=0}^{n-1}x_jy_j}|y_{n-1}\rangle\cdots|y_0\rangle \\ &= \frac{1}{2^{n/2}}\sum_{y=0}^{2^n-1}(-1)^{x\cdot y}|y\rangle_n\end{aligned}$$

Deutsch–Jozsa algorithm is just the application of Deutsch algorithm but on n input qubits as opposed to 1.

5.4 Bernstein Vazirani Problem

Problem Statement: $f : \{0,1\}^n \to \{0,1\}^n$ such that $f(x) = x.a$ ($x.a$ is bitwise modulo-2 inner product). The goal is to find a.

Suppose we have a black box that evaluates $f(x) = x.a$ how many times do we have to call it to determine a?

The m th bit of a is $a \cdot 2^m$, since the binary expansion of 2^m has 1 in position m and 0 in all the other positions. So with a classical computer, we can learn the n bits of a by applying f to the n values $x = 2^m, 0 \leq m < n$. But with a quantum computer, a single invocation is enough to determine a completely, regardless of how big n is!

This time let us take a circuit theory approach to design the necessary quantum circuit. Suppose $a = 10011$ here our function f is assumed to be $n = 4$ bit function. When $f(x) = a \cdot x$, the action of $\mathbf{U}_f$ on the computational basis is to flip the 1-qubit output register once, whenever a bit of x and the corresponding bit of a are both 1. When the state of the input register is $|x\rangle_n$ this action can be performed by a collection of cNOT gates all targeted on the output register. There is one cNOT for each nonzero bit of a, controlled by the qubit representing the corresponding bit of x. The combined effect of these cNOT gates on every computational basis state is precisely that of $\mathbf{U}_f$. Therefore, the effect of any other transformations preceding and/or following $\mathbf{U}_f$ can be understood by examining their effect on this equivalent collection of cNOT gates, even though $\mathbf{U}_f$ may actually be implemented in a completely different way.

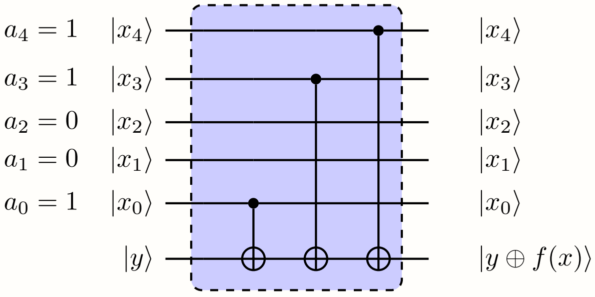


Figure 5.9: Circuit of $f(x) = x \cdot a$

Since we have no control over what the value of a can be, we wish to take away the control from a (pun intended!). We have seen that we can flip the control and target qubits by application of Hadamard gates.

After this reversal of target and control qubits, the output register controls every one of the cNOT gates, and since the state of the output register is $|1\rangle$, every one of the NOT operators acts. That action flips just those qubits of the input register for which the corresponding bit of a is 1 . Since the input register starts in the state $|0\rangle_n$, this changes the state of each qubit of the input register to $|1\rangle$, if and only if it corresponds to a nonzero bit of a. As a result, the state of the input register changes from $|0\rangle_n$ to $|a\rangle_n$.

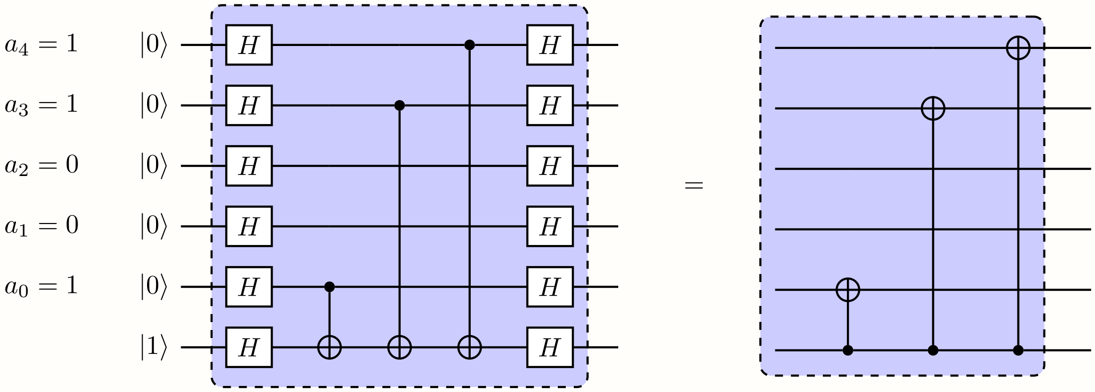


Figure 5.10: Sandwitching cNOT between Hadamard

Algebraically, as always, we start with an equal superposition of the input registers and apply $\mathbf{U}_f$. Now we apply Hadamard again as this will give the form $x \cdot (a + y)$ as an exponent to -1, helping to capture only those ys that are equal to a.

$$
\begin{aligned}
&\left(\mathbf{H}^{\otimes n}\otimes \mathbf{1}\right)\mathbf{U}_f\left(\mathbf{H}^{\otimes n}\otimes \mathbf{H}\right)|0\rangle_n|1\rangle_1\\
&\quad=\left(\mathbf{H}^{\otimes n}\otimes \mathbf{1}\right)\mathbf{U}_f\left(\frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x\rangle\right)\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)\\
&\quad=\frac{1}{2^{n/2}}\left(\mathbf{H}^{\otimes n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|x\rangle\right)\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)\\
&\quad=\frac{1}{2^n}\sum_{x=0}^{2^n-1}\sum_{y=0}^{2^n-1}(-1)^{f(x)+x\cdot y}|y\rangle\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle).
\end{aligned}
$$

We do the sum over x first. If the function $f(x)$ is $a\cdot x$ then this sum produces the factor

$$\sum_{x=0}^{2^n-1}(-1)^{(a-x)}(-1)^{(y-x)}=\prod_{j=1}^{n}\sum_{x_j=0}^{1}(-1)^{(a_j+y_j)x_j}$$

At least one term in the product vanishes unless every bit y_j of y is equal to the corresponding bit a_j of $a-$, i.e. unless $y=a$. Therefore, the entire computational process reduces to

$$\mathbf{H}^{\otimes(n+1)}\mathbf{U}_f\mathbf{H}^{\otimes(n+1)}|0\rangle_n|1\rangle_1=|a\rangle_n|1\rangle_1,$$

Final $\mathbf{H}$ to the 1-qubit output register to make the final expression look neater and more symmetric.

Thus, all n bits of the number a can now be determined by measuring the input register, even though we have called the subroutine only once!

Why can not we do the same swapping cNOT technique classically?

Interestingly, quantum computers can do this only because it allows the reversal of the control and target qubits of a cNOT operation solely by means of 1-qubit (Hadamard) gates. One can also reverse control and target bits of a cNOT classically, but this requires the use of 2-qubit SWAP gates, rather than 1-qubit Hadamards. You can confirm for yourself that this circuit-theoretic solution to the Bernstein-Vazirani problem no longer works if one tries to replace all the Hadamard gates by any arrangement of SWAP gates.

5.5 Simon's Problem

Problem Statement: $f:\{0,1\}^n\rightarrow\{0,1\}^{n-1}$, a two to one function, such that $f(x\oplus a)=f(x)$. Find the period a.

As always, we have the black box that computes f. With a classical computer, we can keep computing f until we by chance encounter x_i and x_j such that both give the same $f(x)$.

Then we know that $a = x_i \oplus x_j$. At any stage of this process, if we picked m different x_ks such that none have the same functional value, the all we can say is, for any pair $x_i \oplus x_j \neq a$. In this way we can reject at most $\binom{m}{2} = \frac{m(m+1)}{2}$ values of a. There are 2^{n-1} possible values for a, so m should be as big as 2^{n-1} to narrow down to one value of a. So, it need exponentially many calls to the black box to compute a. Where as we can compute a with just linear calls using a quantum computer.

Let's start with equal superposition of all input states. On applying $\mathbf{U}_f$ to this, we get each of the terms in the equal superposition to be of the form $|x\rangle\,|x \cdot a\rangle$. Now if we measure it collapses to a particular $f(x_0)$ and two different x, that is x_0 and $x_0 \cdot a$ gives the same $f(x_0)$. So, on measurement of the output qubit the input qubit now collapses to an equal super position of these two values of x.

$$\frac{1}{\sqrt{2}}\,|x_0\rangle\,|x_0 \cdot a\rangle$$

With the input register in the above state, we apply the n-fold Hadamard transformation $\mathbf{H}^{\otimes n}$.

$$\mathbf{H}^{\otimes n}\frac{1}{\sqrt{2}}\left(|x_0\rangle + |x_0 \oplus a\rangle\right) = \frac{1}{2^{(n+1)/2}}\sum_{y=0}^{2^n-1}\left((-1)^{x_0\cdot y} + (-1)^{(x_0\oplus a)\cdot y}\right)|y\rangle.$$

Since $(-1)^{(x_0\oplus a)\cdot y} = (-1)^{x_0\cdot y}(-1)^{a\cdot y}$, the coefficient of $|y\rangle$ is 0 if $a \cdot y = 1$ and $2(-1)^{x_0\cdot y}$ if $a \cdot y = 0$. Therefore

$$\frac{1}{2^{(n-1)/2}}\sum_{a\cdot y=0}(-1)^{x_0\cdot y}|y\rangle$$

where the sum is now restricted to those y for which the modulo- 2 bitwise inner product $a \cdot y$ is 0 rather than 1 . So, if we now measure the input register, we learn (with equal probability) any of the values of y for which $a \cdot y = 0-$ i.e. for which

$$\sum_{i=0}^{n-1} y_i a_i = 0(\mathrm{mod} 2)$$

where a_i and y_i are corresponding bits in the binary expansions of a and y.

With $n+x$ invocations of $\mathbf{U}_f$ the probability q of acquiring enough information to determine a is

$$q = \left(1 - \frac{1}{2^{n+x}}\right)\left(1 - \frac{1}{2^{n+x-1}}\right)\cdots\left(1 - \frac{1}{2^{x+2}}\right) > 1 - \frac{1}{2^{x+1}}$$

Thus the odds are more than a million to one that with $n + 20$ invocations of $\mathbf{U}_f$ we will learn a, no matter how large n may be.

5.6 Approximate Quantum Algorithms

Suppose a quantum system starts in the state $|\psi\rangle$, and we perform either the unitary operation U, or the unitary operation V. Following this, we perform a measurement. Let

M be a projective element associated with the measurement, and let P_U (or P_V) be the probability of obtaining the corresponding measurement outcome if the operation U (or V) was performed. Then

$$|P_U - P_V| = \left|\langle\psi|U^\dagger MU\right|\psi\rangle - \langle\psi|V^\dagger MV|\psi\rangle\ |$$

Let $|\Delta\rangle \equiv (U - V)|\psi\rangle$. Simple algebra and the Cauchy-Schwarz inequality show that

$$\begin{aligned}|P_U - P_V| &= \left|\langle\psi|U^\dagger M\right|\Delta\rangle + \langle\Delta|MV|\psi\rangle\ | \\ &\leq \left|\langle\psi|U^\dagger M\right|\Delta\rangle\ | + |\langle\Delta|MV|\psi\rangle\ | \\ &\leq \||\Delta\rangle\| + \||\Delta\rangle\| \\ &\leq 2E(U, V)\end{aligned}$$

The inequality $|P_U - P_V| \leq 2E(U, V)$ gives quantitative expression to the idea that when the error $E(U, V)$ is small, the difference in probabilities between measurement outcomes is also small. Suppose we perform a sequence $V_1, V_2, \ldots, V_m$ of gates intended to approximate some other sequence of gates, $U_1, U_2, \ldots, U_m$. Then it turns out that the error caused by the entire sequence of imperfect gates is at most the sum of the errors in the individual gates,

$$E\left(U_m U_{m-1} \ldots U_1, V_m V_{m-1} \ldots V_1\right) \leq \sum_{j=1}^{m} E\left(U_j, V_j\right)$$

To prove this we start with the case $m = 2$. Note that for some state $|\psi\rangle$ we have

$$\begin{aligned}E\left(U_2U_1, V_2V_1\right) &= \|\left(U_2U_1 - V_2V_1\right)|\psi\rangle\| \\ &= \|\left(U_2U_1 - V_2U_1\right)|\psi\rangle + \left(V_2U_1 - V_2V_1\right)|\psi\rangle\|\end{aligned}$$

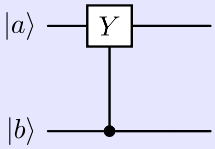
Using the triangle inequality $\||a\rangle + |b\rangle\| \leq \||a\rangle\| + \||b\rangle\|$, we obtain

$$\begin{aligned}E\left(U_2U_1, V_2V_1\right) &\leq \|\left(U_2 - V_2\right)U_1|\psi\rangle\| + \|V_2\left(U_1 - V_1\right)|\psi\rangle\| \\ &\leq E\left(U_2, V_2\right) + E\left(U_1, V_1\right)\end{aligned}$$

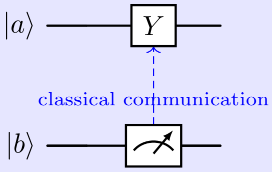
which was the desired result. The result for general m follows by induction.

To avoid non-local application of unitary

Most often, it is hard to have control gates between qubits that are spatially separated. In this case, we can measure the control qubit and classically communicate its value to the target qubit. This can be done because figures (i) and (ii) are essentially equivalent.



(i) non-local controlled unitary



(ii) classical communication

To see their equivalence, consider a general 2-qubit state:

$$\begin{aligned}
&a_{00}|00\rangle + a_{10}|10\rangle + a_{01}|01\rangle + a_{11}|11\rangle \\
&= \left(a_{00}|0\rangle + a_{10}|1\rangle\right)|0\rangle + \left(a_{01}|0\rangle + a_{11}|1\rangle\right)|1\rangle \\
&\equiv A_0 \frac{(\,a_{00}|0\rangle + a_{10}|1\rangle)}{A_0}|0\rangle + A_1\left(\frac{a_{01}|0\rangle + a_{11}|1\rangle}{A_1}\right)|1\rangle \\
&|\psi\rangle \equiv A_0\,|\phi_0\rangle\,|0\rangle + A_1\,|\phi_1\rangle\,|1\rangle
\end{aligned}$$

Where A_0 and A_1 are appropriate normalization constants.

Non-local application of controlled unitary

$$U|\psi\rangle = A_0\,|\phi_0\rangle\,|0\rangle + A_1V_1\,|\phi_1\rangle\,|1\rangle$$

where $U = \mathbb{I} \otimes V_1$. If $b = 1$, then the 1^{st} qubit collapses to $V_1\,|\phi_1\rangle$ and this happens with probability $|A_1|^2$. If $b = 0$, then the 1^{st} qubit collapses to $|\phi_0\rangle$ with probability $|A_0|^2$.

Measurement and classical communication

If we measure $|b\rangle$ then we will get 1 with probability $|A_1|^2$ and 0 with probability $|A_)|^2$. When we communicate this information classically to Y. This also gives the same probabilities for $V_1\,|\phi_1\rangle$ and $|\phi_0\rangle$, showing the equivalence.

Chapter 6

Quantum Fourier Transform and Shor's Algorithm

> *"There cannot be a language more universal and more simple, more free from errors and obscurities, more worthy to express the invariable relations of all natural things[than mathematics]. [It interprets] all phenomena by the same language, as if to attest the unity and simplicity of the plan of the universe, and to make still more evident that unchangeable order which presides over all natural causes."*
>
> – Joseph Fourier, *The Analytical Theory of Heat*

The problem of how to factor a large integer efficiently has been studied extensively in number theory. It is generally believed that factorization of a number n is hard to do in an efficient way. That is, it cannot be done in a number of steps which is polynomial in the length of the integer we're trying to factor. The RSA cryptosystem, among others, relies on the presumed difficulty of this task. Classically, the fastest known algorithm is the Number Field Sieve algorithm, which works in super-polynomial but sub-exponential time, $\mathcal{O}(\exp\bigl(n^{1/3}(\log n)^{2/3}\bigr))$.

In 1994, Peter Shor discovered an algorithm that can factor numbers in polynomial time, $\mathcal{O}(n^2 \log n \log\log n)$, using a quantum computer, a drastic improvement over the existing classical algorithms. That is, a quantum computer can factor a number exponentially faster than the best-known classical algorithms.

6.1 RSA Cryptography

RSA falls under public key cryptography. Suppose Alice and Bob want to communicate. Both Alice and Bob have two keys each, one a public key, that is publicly available to everyone and a private key that no one other than the owner knows. One possible scheme to communicate securely is as follows:
Bob encrypts his message through the function f, invoking the public key E, and sends the encrypted message C to Alice. Alice uses the function g to decrypt the message with the help of her private key D to recover the message. The functions f and g are released publicly

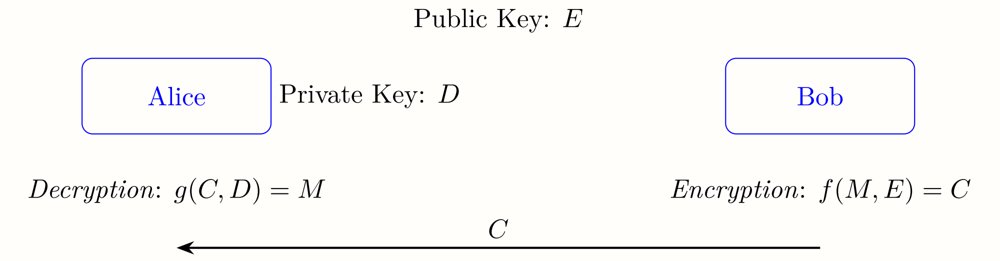


Figure 6.1: Public Key Cryptography

as a part of the protocol. Public key cryptography works on the basis that the function f is extremely difficult to invert; that is, getting the message M from the chipper text C is extremely hard. But this becomes easy with D, the private key. Thus, such protocols heavily rely on the computational hardness of a problem.

The keys for the RSA algorithm are generated in the following way:

- Choose P and Q very large primes. Compute $N = PQ$.

- N is released as a part of the public key. N will be used as the modulo arithmetic for both public and private keys. Its length, usually expressed in bits, is the *key length*.

- Let $R = (P-1)(Q-1)$ the *totient function*. Note that as $\varphi(N) = \varphi(P) \times \varphi(Q) = (P-1)(Q-1)$ since $\varphi(P)$ and $\varphi(Q)$ are $P-1$ and $Q-1$ respectively. (Refer to chapter 1 section 1.6 for the definition of *totient function* and other basics of number theory to better understand this section.)

- Choose integer E such that $1 < E < R$ and E is coprime with R. Note that this means $E \in \varphi(R)$. E is released as a part of the public key.

- Determine D such that $ED \mod R = 1$, that is $D = E^{-1} \mod R$, the modular multiplicative inverse of $E \mod R$. D is the private key.

The RSA scheme is as follows:

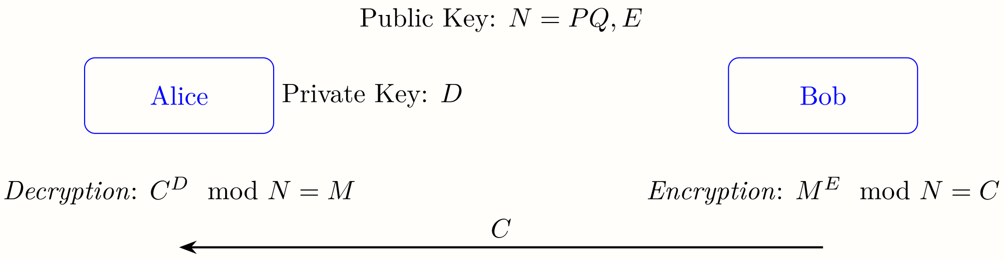

Public Key: $N = PQ, E$
Alice
Private Key: D
Bob
Decryption: $C^D \mod N = M$
Encryption: $M^E \mod N = C$
C

Figure 6.2: RSA

The decryption works as D is chosen such that $ED \mod R = 1 \implies ED = 1 + xR$ where $x \in \mathbb{Z}$. Hence, we have,

$$\begin{aligned} C^D \mod N &= (M^E)^D \mod N \\ &= M^{ED} \mod N = (M \mod N)(M^{xR} \mod N) \\ &= M \mod N \end{aligned}$$

The last line follows from the fact that $M^R \mod N = 1$ as R is totient of N.

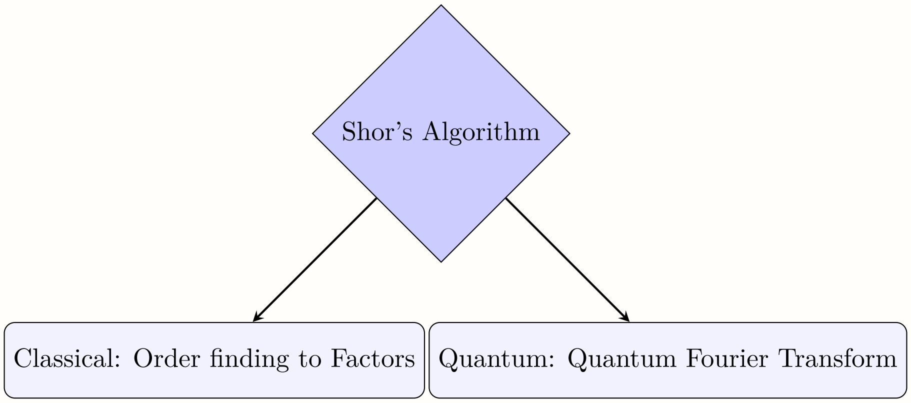
A malicious Eve can eavesdrop on Alice and Bob's conversation and get C. But what guarantees that she can not get M from C given the protocol N and E?

Classical computers can efficiently compute D such that $ED \mod R = 1$, provided R is known. So, the real difficulty lies in computing R from E, N, and C, that is, finding the prime factors of N. So, the security of RSA lies in the fact that factoring is a computationally very hard problem. This is no longer true in the case of a quantum computer.

6.2 Overview of Shor's Algorithm

6.2.1 Idea behind Shor's Algorithm

Shor's algorithm consists of a classical and a quantum part.


Shor's Algorithm
Classical: Order finding to Factors
Quantum: Quantum Fourier Transform

Shor's algorithm does not allow us to factor a number directly. Instead, it allows us to find the order of an element a modulo n in polynomial time. This, in quantum computers, is done using inverse Quantum Fourier Transform as one of the subroutines.

We will see that finding a factor of n, given the order of some element in $\mathbb{Z}/n\mathbb{Z}$ can be done efficiently even on a classical computer, but no efficient algorithm is known for finding the order of the element.

6.3 Shor's Algorithm

6.3.1 Shor's Algorithm Pseudo-code

Input: $N = PQ$ where P and Q are primes
Output: P, Q

1. Pick a number a that is coprime with N i.e. their gcd is 1.
2. Find the order R of the function $a^R \mod N$.
3. If R is even:
   - Define $x \equiv a^{R/2} \mod N$
   - If $x + 1 \not\equiv 0 \mod N$:
     Then the factors P and Q which we are looking for, at least one of them is contained in $\{\gcd(x+1, N),\ \gcd(x-1, N)\}$
4. If either of the above two conditions fails, then pick another a and repeat this all over again.

Remarks. *Note that given $a^R = 1 \mod N$ and r is even we can factor $a^R - 1$ as $(a^{\frac{R}{2}} - 1)(a^{\frac{R}{2}} + 1) = 0 \mod N$. If $x \equiv a^{R/2} \mod N$, then possibly either $x - 1$ or $x + 1$ divides N. But note that the former is not possible as we started with the assumption that the orbit of a is of size R, so it can not be $R/2$. If $x + 1$ divides N we just repeat the process again (as said in point 4.)*

If both the above fails, then either $x - 1$ or $x + 1$ is a multiple of Q and P, where $N = QP$. Thus finding $\{\gcd(x+1, N),\ \gcd(x-1, N)\}$ gives P and Q.

Example 6.3.1. *Consider factoring 15:*

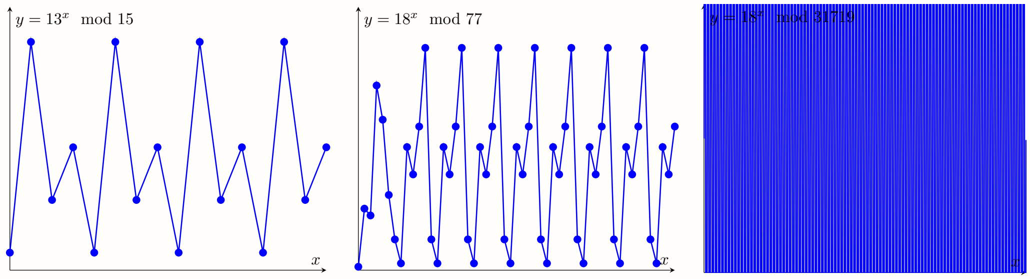
1. *Let us pick $a = 13$, as 13 is coprime with 15.*
2. *We need to find the order of $13^x \mod 15$. Since R is the smallest number such that*

x	0	1	2	3	4	5	6	...
$13^x \mod 15$	①	13	4	7	①	13	4	...

   $a^r \equiv 1 \mod N$, here $r = 4$ since the values are periodic about $x = 0, 4, 8, \ldots$.
3. *$R = 4$ is even,*
   Define $x = a^{R/2} \mod N = 13^{4/2} \mod 15 = 13^2 \mod 15 = 4 \mod 15$.
   Therefore, $x \equiv 4 \mod 15$, hence $x + 1 \equiv 4 + 1 \mod 15 \equiv 5 \mod 15 \not\equiv 0 \mod 15$

   This implies P or Q is in $\{\gcd(x+1,\ N),\ \gcd(x-1,\ N)\}$
   Here $\gcd(4+1,\ 15),\ \gcd(4-1,\ 15) = 5,\ 3$. So, $P = 5$ and $Q = 3$.

Why can not we implement the above algorithm completely classically?
The reason is that it becomes progressively harder to find the order. We can see this by looking at the plot between $a^z \mod N$ and z. As the number N grows, the period grows very quickly, and this function appears more and more aperiodic. For $N = 314191$, classical computer runs for about 2 hours in real-time computing. This order-finding part is expedited by using quantum computers.



$N = 15 = 3 \times 5$, $r = 4$ $N = 77 = 7 \times 11$, $r = 30$ $N = 314191$, $r = 17388$

6.3.2 Classical Part of Shor's Algorithm

In the below section, through Lemma (6.3.1) and Theorem (6.3.3), we will see that, given a composite number n and the order r of some $x \in \mathbb{Z}/n\mathbb{Z}$, we can compute $\gcd(x^{r/2} \pm 1, n)$ efficiently using Euclid's algorithm. This gives a non-trivial factor of n unless r is odd or $x^{r/2} \equiv -1 \mod n$. In particular, if n is a semi-prime, i.e., it is a product of two primes p and q, then Theorem (6.3.3) implies that n will be factored with probability $\frac{1}{2}$.

6.3.2.1 Factoring as Order finding

We will show that the problem of finding a non-trivial factor to n can be reduced (efficiently) to finding the order of a non-trivial element in $\mathbb{Z}/n\mathbb{Z}$.

Lemma 6.3.1. *Given a composite number n, and x non-trivial square root of 1 modulo n, i.e. $x^2 \equiv 1 \mod N$ but x is neither 1 nor $-1 \mod n$, then either $\gcd(x-1,\ n)$ or $\gcd(x+1,\ n)$ is a non-trivial factor of n.*

Proof. Since $x^2 \equiv 1 \mod n$, we have $x^2 - 1 \equiv 0 \mod n$. Factoring, we get $(x-1)(x+1) \equiv 0 \mod n$. This implies that n is a factor of $(x+1)(x-1)$. Since $(x \pm 1) \not\equiv 0 \mod n$, n has a non-trivial factor with $x+1$ or $x-1$. To find this common factor efficiently, we apply Euclid's algorithm to get $\gcd(x-1,\ n)$ or $\gcd(x+1,\ n)$. ■

Example 6.3.2. *Let $n = 55 = 5 \times 11$. We find that 34 is a square root of $1 \mod n$ since $342 = 1156 = 1 + 21 \times 55$. Computing, we get $\gcd(33,\ 55) = 11$ and $\gcd(35, 55) = 5$.*

Lemma 6.3.2. *Let n be odd, then at least half the elements in $(\mathbb{Z}/n\mathbb{Z})^\times$ have even order.*

Proof. Suppose $\operatorname{ord}(x) = r$ is odd. Then $(-x)^r = (-1)^r x^r = (-1)^r = -1 \mod n$. Hence, $-x$ must have order $2r$, which is even. Therefore, at least half the elements in $(\mathbb{Z}/n\mathbb{Z})^\times$ have even order. ■

Equipped with these tools, we will proceed to prove the main result that allows us to reduce the factorisation of n to find the order of an element in $\mathbb{Z}/n\mathbb{Z}$.

Theorem 6.3.3. *Let n be an odd integer and let $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ be the prime factorization of n. Then the probability that a uniformly randomly chosen $x \in \mathbb{Z}/n\mathbb{Z}$ has even order r and $x^{r/2} \not\equiv -1 \mod n$ is at least $1 - \frac{1}{2^{k-1}}$.*

Proof. By the Chinese Remainder Theorem, choosing $x \in (\mathbb{Z}/n\mathbb{Z})^\times$ (uniform) randomly is equivalent to choosing $x_i \in (\mathbb{Z}/p_i^{e_i}\mathbb{Z})^\times$ for each p_i randomly. Let r be the order of x and let r_i be the order of x_i. In particular, $x^{r/2}$ is never $1 \mod n$. We want to show that the probability of either r being odd or $x^{r/2} \equiv -1 \mod n$ is at most $\frac{1}{2^{k-1}}$.
Note that $r = \operatorname{lcm}(r_1, r_2, \ldots, r_k)$ (where lcm denotes the least common multiple). To see this, $x^r \equiv 1 \mod n$, $x^r \equiv 1 \mod p_i^{e_i}$, hence r is a multiple of each r_i. It is the least such number and hence the least common multiple of the r_i's.
Suppose that r is odd. This happens only if all of the r_i's are odd. r_i is odd with probability at most one-half by Lemma (6.3.2). Hence, r is odd with probability at most $\frac{1}{2^k}$.
Now, suppose that r is even. We still have to worry about the possibility that $x^{r/2} \equiv \pm 1 \mod n$. By the Chinese Remainder Theorem, this happens only if $x^{r/2} \equiv \pm 1 \mod p_i^{e_i}$ for every p_i. We need to avoid these cases since $\equiv +1$ means r wasn't the order, and $\equiv -1$ doesn't yield a useful factorisation. The probability of choosing an x such that one of these two cases happens is $2 \cdot 2^{-k} = 2^{-k+1}$.
Combining the probabilities, we get a success probability of at least $(1 - 2^{-k})(1 - 2^{-k+1}) \geq 1 - 3 \cdot 2^{-k}$.

■

By Lemma (6.3.1) and Theorem (6.3.3) , given a composite number n and the order r of some $x \in \mathbb{Z}/n\mathbb{Z}$, we can compute $\gcd(x^{r/2} \pm 1, n)$ efficiently using Euclid's algorithm. This gives a non-trivial factor of n unless r is odd or $x^{r/2} \equiv -1 \mod n$. In particular, if n is a semi-prime, i.e., it is a product of two primes p and q, then Theorem (6.3.3) implies that n will be factored with probability $\frac{1}{2}$.

6.3.3 Quantum part of Shor's Algorithm

6.3.3.1 Discrete Fourier Transform (DFT)

Let's start with a familiar idea. Imagine you're listening to a piece of music. The music is made up of different notes (frequencies) that together create a melody. Now, if you wanted to analyze which notes are present, you'd try to pick apart the sound into its individual frequencies. This is essentially what the Fourier transform does by breaking down a complex signal into a sum of simple sinusoidal waves, each with its own frequency, amplitude, and phase.

In the classical setting, when we have a periodic function, say, one that repeats every T units, the Fourier transform will show us spikes at specific frequencies. The most prominent spike is at the fundamental frequency, which is $f_0 = \frac{1}{T}$. This is the basic beat of the function, the frequency at which the pattern repeats. But a typical periodic function isn't just a simple sine wave, and might be a more complex shape. This complexity is reflected in the presence of harmonics, which are spikes at frequencies that are integer multiples of the fundamental frequency (i.e., $2f_0, 3f_0, \dots$).

In practice, especially when working with digital data, we use the Discrete Fourier Transform (DFT). The DFT algorithm takes a sequence of data points (samples of our function) and computes how much of each frequency is present in the signal. When you run a DFT on a periodic function, you see peaks in the output at the frequencies where the function has a strong periodic component.

Here is an illustration for $y = \sin(2\pi\nu x)$, whose Fourier transform DFT $\tilde{y}$ has the peak at ν. Note that the broadening of the unique peak occurs due to the finiteness of the data size. There is a single peak since $\sin(2\pi\nu x)$ has the fundamental mode and no additional harmonics.

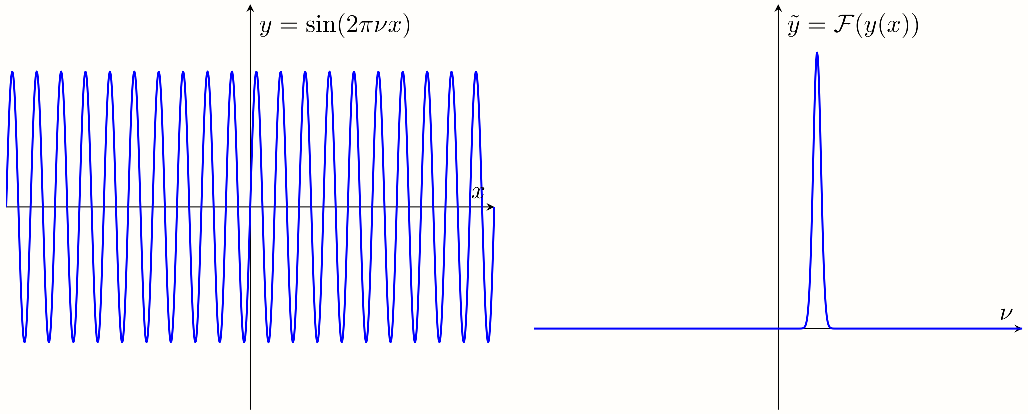


Figure 6.3: Discrete Fourier Transform of sin function

Now consider the function $f(x) = a^x \mod N$ where $a \in \mathbb{Z}$ and $N \in \mathbb{N}$, which is periodic over the scales of N. Decomposing as a DFT, we have the following interpretation.

We note that the peaks of the DFT correspond to the Fourier fundamental frequencies, which are integral multiples of the period.

We generalize this idea of hunting for fundamental frequencies to a general vector by describing the DFT as the tool that decomposes a vector of complex numbers into its intrinsic frequency components. The formal definition of the algorithm is as follows:

Input: A vector of complex numbers $x_0, x_1, \dots, x_{N-1}$, where N is a fixed parameter (assuming $N = 2^n$).
Output: A vector of complex numbers $y_0, y_1, \dots, y_{N-1}$, such that

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi ijk/N} x_j.$$

Let's build up the picture step by step. The DFT decomposes the input vector into a linear

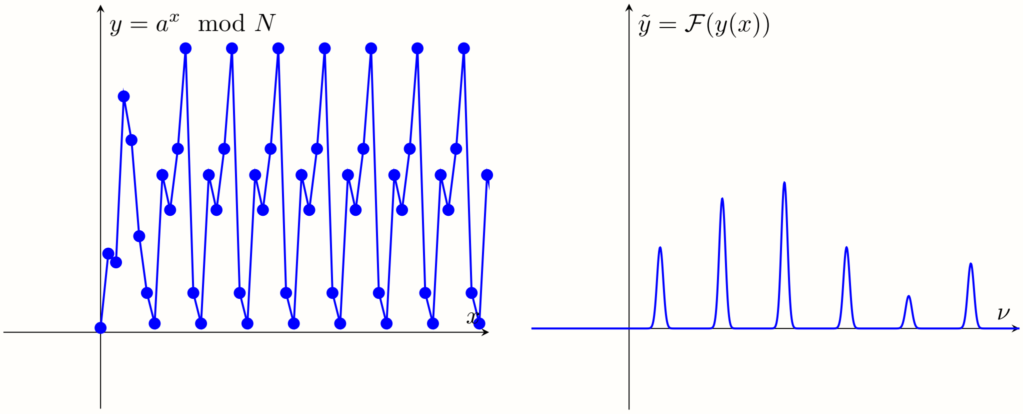


Figure 6.4: Discrete Fourier Transform of $f(x) = a^x \mod N$

combination of complex exponentials. These exponentials, given by the factors $e^{2\pi ijk/N}$, serve as basis functions that oscillate at specific frequencies. For each k, we can think of $e^{2\pi ik}/N$ as a complex vector with N entries, individually by

$$v_k = \left(\frac{1}{N}, \frac{1}{N}e^{2\pi ik/N}, \frac{1}{N}e^{4\pi ik/N}, \ldots, \frac{1}{N}e^{2\pi ik(N-1)/N}\right).$$

These N vectors form an orthonormal basis of $\mathbb{C}^N$, and can be used to decompose any vector into components along these vectors. We can directly compute the dot product of our vector to note the component along the suitable basis vector. When the input signal has a periodic structure, these basis vectors align with the natural periodicities of the signal, producing prominent peaks in the output. The term $e^{2\pi ijk/N}$ can be viewed as a rotating phase factor. For a fixed k, as j runs over the values 0 to $N-1$, these exponentials trace out a complete cycle. When the input signal x_j resonates with this cycle (that is, when the signal contains a frequency component matching k/N), the sum in Equation (6.3.3.1) reinforces this frequency component, leading to a peak in the output y_k.

Remarks. *Using the Fast Fourier Transform algorithm (FFT)*[1]*, we can do DFT faster. We will see that during the exercise of making a Quantum Fourier Transform algorithm, FFT will appear as a byproduct.*

6.3.4 Quantum Fourier Transform (QFT)

Similar to DFT definition, QFT on an orthonormal basis $|0\rangle \ldots |N-1\rangle$ is defined to be a linear operator with the following action on the basis states,

$$QFT|j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi ijk/2^n}|k\rangle.$$

[1] Refer `https://youtu.be/nmgFG7PUHfo` for an intuitive discussion of the algorithm.

The action on an arbitrary state may be written as

$$\sum_{j=0}^{N-1} x_j|j\rangle = \sum_{k=0}^{N-1} y_k|k\rangle,$$

where y_k are DFT of the amplitudes x_j. This expression strengthens our understanding of the basis transformation nature of the algorithm.

Remarks. *It is not obvious from the definition of QFT, but this transformation is a unitary transformation and thus can be implemented as the dynamics for a quantum computer. The theorem below will show this fact and also give an expression that can be easily interpreted when designing a quantum circuit for the QFT algorithm.*

Can you prove QFT as defined above is unitary?

$U|j\rangle = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{2\pi i j\cdot k/N}|k\rangle$ QFT *Claim:* U is Unitary

$$\begin{aligned} U^{+}U &= \frac{1}{N}\left(\sum_{k=0}^{N-1} e^{-2\pi i j\cdot k/N}\langle k|\right)\left(\sum_{k'=0}^{N-1} e^{+2\pi i j\cdot k'/N}|k'\rangle\right) \\ &= \frac{1}{N}\sum_{k=0}^{N-1}\sum_{k'=0}^{N-1} e^{\frac{2\pi i j}{N}(k'-k)j}\,\langle k \mid k'\rangle \\ &= \frac{1}{N}\sum_{k=0}^{N-1} \mathbb{I}\cdot 1 = \frac{N}{N}\mathbb{I} = \mathbb{I} \end{aligned}$$

We consider $N = 2^n, n \in \mathbb{Z}$. As earlier, the basis $|0\rangle \ldots |N-1\rangle$ is $|0\rangle \ldots |2^n-1\rangle$ thus can be represented using n bit-string. Thus, $|j\rangle = |j_1 \ldots j_n\rangle$ where $j = j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_n 2^0$. Also $0.j_l j_{l+1} \ldots j_m$ is binary fraction $\frac{j_l}{2} + \frac{j_l}{2^2} + \cdots + \frac{j_m}{2^{m-l+1}}$.

Theorem 6.3.4. *(QFT Representation)*

$$|j_1, \ldots, j_n\rangle \xrightarrow{QFT} \frac{1}{2^{n/2}}\left[\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.j_1 j_2\cdots j_n}|1\rangle\right)\right].$$

Proof.

$$
\begin{aligned}
QFT|j\rangle &= \frac{1}{2^{n/2}}\sum_{k=0}^{2^n-1} e^{\frac{2\pi ijk}{2^n}}|k\rangle = \frac{1}{2^{n/2}}\sum_{k_1=0}^{1}\cdots\sum_{k_n=0}^{1} e^{\frac{2\pi ij(k_1 2^{n-1}+k_2 2^{n-2}+\cdots+k_n 2^0)}{2^n}}|k\rangle \\
&= \frac{1}{2^{n/2}}\sum_{k_1=0}^{1}\cdots\sum_{k_n=0}^{1} e^{2\pi ij(k_1 2^{-1}+k_2 2^{-2}+\cdots+k_n 2^{-n})}|k\rangle \\
&= \frac{1}{2^{n/2}}\sum_{k_1=0}^{1}\cdots\sum_{k_n=0}^{1} e^{2\pi ij\sum_{l=1}^{n} k_l 2^{-l}}|k_1\cdots k_n\rangle \\
&= \frac{1}{2^{n/2}}\sum_{k_1=0}^{1}\cdots\sum_{k_n=0}^{1} \mathcal{O}times_{l=1}^{n} e^{2\pi ijk_l 2^{-l}}|k_l\rangle \\
&= \frac{1}{2^{n/2}}\mathcal{O}times_{l=1}^{n}\left[\sum_{k_l=0}^{1} e^{2\pi ijk_l 2^{-l}}|k_l\rangle\right] \\
&= \frac{1}{2^{n/2}}\mathcal{O}times_{l=1}^{n}\left(|0\rangle + e^{2\pi ij2^{-l}}|1\rangle\right) \\
&= \frac{1}{2^{n/2}}\left[|0\rangle + e^{2\pi i0.j_n}|1\rangle)(|0\rangle + e^{2\pi i0.j_{n-1}j_n}|1\rangle)\cdots(|0\rangle + e^{2\pi i0.j_1j_2\cdots j_n}|1\rangle\right]
\end{aligned}
$$

■

Is there a relation between Hadamard operator and QFT?

Consider $U|00\ldots0\rangle$, where $U \to$ QFT

$$= \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{2\pi ijk/N}|k\rangle$$

$$= \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} 1\cdot|k\rangle$$

The coefficients have become 1 as $j\cdot k = j_1\cdot k_1 + j_2\cdot k_2\cdots j_N\cdot k_N = 0$
So, $U|00\ldots0\rangle = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}|k\rangle$ which is the equal superposition of all basis, which is nothing but Hadamard on $|00\ldots0\rangle$.

6.3.4.1 Quantum circuit for implementing QFT

Let the gate R_k denote the unitary transformation, $R_k \equiv \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix}$.

To see that the pictured circuit Figure 6.5, computes the quantum Fourier transform, consider what happens when the state $|j_1\cdots j_n\rangle$ is input. Applying the Hadamard gate to the first bit produces the state

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i0.j_1}|1\rangle\right)|j_2\cdots j_n\rangle,$$

since $e^{2\pi i 0.j_1} = -1$ when $j_1 = 1$, and is $+1$ otherwise. Applying the controlled R_2 gate produces the state,

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2}|1\rangle\right)|j_2 \cdots j_n\rangle.$$

We continue applying the controlled R_3, R_4 through R_n gates, each of which adds an extra bit to the phase of the coefficient of the first $|1\rangle$. At the end of this procedure, we have the state,

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\right)|j_2 \cdots j_n\rangle.$$

Next, we perform a similar procedure on the second qubit. The Hadamard gate puts us in the state,

$$\frac{1}{\sqrt{2^2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_2}|1\rangle\right)|j_3 \cdots j_n\rangle,$$

and the controlled-R_2 through R_{n-1} gates yield the state,

$$\frac{1}{\sqrt{2^2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_2 \cdots j_n}|1\rangle\right)|j_3 \cdots j_n\rangle.$$

We continue in this fashion for each qubit, giving a final state,

$$\frac{1}{\sqrt{2^n}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_2 \cdots j_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right).$$

Swap operations (which are not shown in the figure) are then used to reverse the order of the qubits, which are simple transmutations of the elements leading to a reverse permutation. After the swap operations, the state of the qubits is,

$$\frac{1}{\sqrt{2^n}}\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_{n-1} j_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\right).$$

Comparing with Equation (6.3.4), we see that this is the desired output from the quantum Fourier transform. This construction also proves that the quantum Fourier transform is unitary since each gate in the circuit is unitary.

How many gates does this circuit use? We start by doing a Hadamard gate and $n-1$ conditional rotations on the first qubit which is a total of n gates. This is followed by a Hadamard gate and $n-2$ conditional rotations on the second qubit, for a total of $n+(n-1)$ gates. Continuing in this way, we see that $n + (n-1) + \cdots + 1 = \frac{n(n+1)}{2}$ gates are required, plus the gates involved in the swaps. At most $\frac{n}{2}$ swaps are required, and each swap can

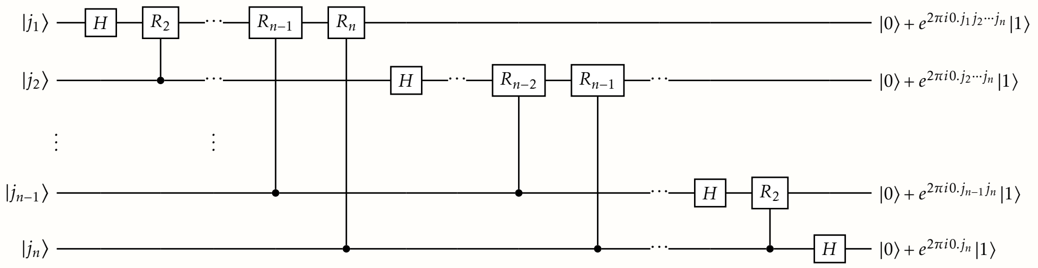

$|j_1\rangle$
$|j_2\rangle$
$|j_{n-1}\rangle$
$|j_n\rangle$
H
R_2
R_{n-1}
R_n
H
R_{n-2}
R_{n-1}
H
R_2
H
$|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle$
$|0\rangle + e^{2\pi i 0.j_2 \cdots j_n}|1\rangle$
$|0\rangle + e^{2\pi i 0.j_{n-1} j_n}|1\rangle$
$|0\rangle + e^{2\pi i 0.j_n}|1\rangle$

Figure 6.5: Quantum Circuit for QFT

be accomplished using three controlled-X gates. Therefore, this circuit provides a $\mathcal{O}(n^2)$ algorithm for performing the quantum Fourier transform.

In contrast, the best classical algorithms for computing the discrete Fourier transform on 2^n elements are algorithms such as the Fast Fourier Transform (FFT), which compute the discrete Fourier transform using $\mathcal{O}(n2^n)$ gates. That is, it requires exponentially more operations to compute the Fourier transform on a classical computer than it does to implement the quantum Fourier transform on a quantum computer.

Approximate Quantum Fourier Transform

The quantum circuit construction of the quantum Fourier transform apparently requires gates of exponential precision in the number of qubits used. However, such precision is never required in any quantum circuit of polynomial size. For example, let U be the ideal quantum Fourier transform on n qubits, and V be the transform that results if the controlled $-R_k$ gates are performed to a precision $\Delta = 1/p(n)$ for some polynomial $p(n)$. Then the error $E(U,V) \equiv \max_{|\psi\rangle} \|(U-V)|\psi\rangle\|$ scales as $\Theta\left(n^2/p(n)\right)$, and thus polynomial precision in each gate is sufficient to guarantee polynomial accuracy in the output state.

How can we prove that the error scales as $n^2 \ /p(n)$?

In the circuit we have $m = \frac{n(n+1)}{2} = \Theta\left(n^2\right) R_k$ gates. Using the result stated in section 5.6, $E(U,V) \leq m\frac{1}{p(n)} = \Theta\left(\frac{n^2}{p(n)}\right)$.

6.3.4.2 FFT from QFT

DFT takes $\Theta(2^{2n})$ operations on an input with $2n$ components. This is quite easy to see if we look at the $2^n \times 2^n = 2^{2n}$ matrix of DFT:

$$W = \frac{1}{\sqrt{2n}} \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{2n-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(2n-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(2n-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{2n-1} & \omega^{2(2n-1)} & \omega^{3(2n-1)} & \cdots & \omega^{(2n-1)(2n-1)} \end{pmatrix}.$$

If we multiply W with a vector and count the operations, we get the result.

Equation (6.3.4) allows you to take advantage of the fact that the Fourier transformed $|j_1, j_2, \ldots, j_n\rangle$ is made out of n tensored 2×1 vectors. So, we process each 2×1 vector independently by performing the following n mappings:

$$\frac{1}{\sqrt{2}} \begin{pmatrix} |0\rangle + |1\rangle \\ \vdots \\ |0\rangle + |1\rangle \end{pmatrix} \rightarrow \frac{1}{\sqrt{2}} \begin{pmatrix} |0\rangle + e^{2\pi i 0.j_n}|1\rangle \\ \vdots \\ |0\rangle + e^{2\pi i 0.j_1 \ldots j_n}|1\rangle \end{pmatrix}.$$

Each mapping takes a constant number of operations in n as it is simply multiplying a 2×1 vector by a 2×2 phase matrix.

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix}.$$

Hence, we perform n matrix-vector multiplication to process a single $|j_1 \ldots j_n\rangle$.

We know that an arbitrary vector $|\psi\rangle$ on n qubits can be written as a linear combination of 2^n binary kets $|j_1, j_2, \ldots, j_n\rangle$. For example, for $n = 2$, an arbitrary state can be written as a linear combination of 2^2 binary kets as follows:

$$|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle.$$

Therefore, to transform $|\psi\rangle$ on n qubits, we need to process 2^n binary vectors $|j_1, \ldots, j_n\rangle$ by performing n mappings described above. Since each such binary vector requires n matrix-vector multiplications, and there are 2^n of them, it takes $\Theta(n2^n)$ operations.

6.3.5 QFT in Shor's algorithm

For the following section, we will assume that N' is a composite odd integer which is not a power of prime (the algorithm fails otherwise). If N' is even, we can just factor out all the powers of 2 until we get an odd integer, then run the algorithm on the resulting integer. We can test whether N' is a prime efficiently using classical primality tests such as the AKS test and the Miller-Rabin test [2]. We can also test if N' is a power of prime efficiently by

[2] Refer to the phenomenal papers by our fellow Indians Agrawal Manindra, Kayal Neeraj, and Saxena Nitin. Primes is in P.

taking the k^{th} root of N' until $\sqrt[k]{N'} < 2$.

Given N', we choose $N = 2^n$ such that $N' < N < 2N'$ (i.e., choose the unique power of 2 in that range). We will be working with two registers (two arrays of qubits), such that each of them holds n qubits. At first, the registers are $|0\rangle \otimes |0\rangle$.

We put the first register in the uniform superposition of numbers $x \mod N$ by using the QFT (This is equivalent to applying Hadamard gate to all qubits in the first register),

$$|0\rangle \xrightarrow{QFT} \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \otimes |0\rangle.$$

Now suppose $f(x) = a^x \mod N$. Note that the period of f is the same as the order of a, given by r. Given some base a, Can we compute $f(x)$ efficiently? The answer is yes; we can just exponentiate by repeated squaring!

We need to apply f to the contents of the first register and store the result of $f(x)$ in the second register. To do so, we can construct f as a quantum function. It turns out that this is the bottleneck of the algorithm since implementing f on a quantum computer requires a lot of quantum gates[3]. Still, Shor's algorithm is much faster than factoring on a classical computer.

We have the state $\frac{1}{N} \sum_{x=0}^{N-1} |x\rangle \otimes |f(x)\rangle$. Apply the inverse QFT to the first register, and we get

$$QFT^{-1} \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \otimes |f(x)\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} (QFT^{-1}|x\rangle) \otimes |f(x)\rangle = \frac{1}{N} \sum_{x,y=0}^{N-1} e^{-\frac{2\pi i x y}{N}} |y\rangle \otimes |f(x)\rangle$$

Remarks. *Note that inverse QFT is equivalent to $QFT^\dagger$, which is the case for every quantum gate.*

Measure the second register, then after applying inverse QFT, measure the first register. Depending on the value do classical processing, as mentioned in section 6.3.1.

Example 6.3.3. *Again consider the number 15 ($|1111\rangle$ in 4 qubits representation). This time we will use the circuit to factor the number.*

1. *Start with set of 2 registers at the state $|0\rangle^{\otimes 4} |0\rangle^{\otimes 4}$.*

2. *Now apply Hadamard on the first set of register,*

$$\left[H^{\otimes 4} |0\rangle^{\otimes 4}\right] |0\rangle^{\otimes 4} = \frac{1}{4} \left[|0\rangle + |1\rangle + \cdots + |15\rangle\right] |0\rangle^{\otimes 4}.$$

   Here the numbers inside ket are in base 10 representation. In base 2, they are all possible 4 bitstrings.

[3] Refer to Shor's orginal paper, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer.

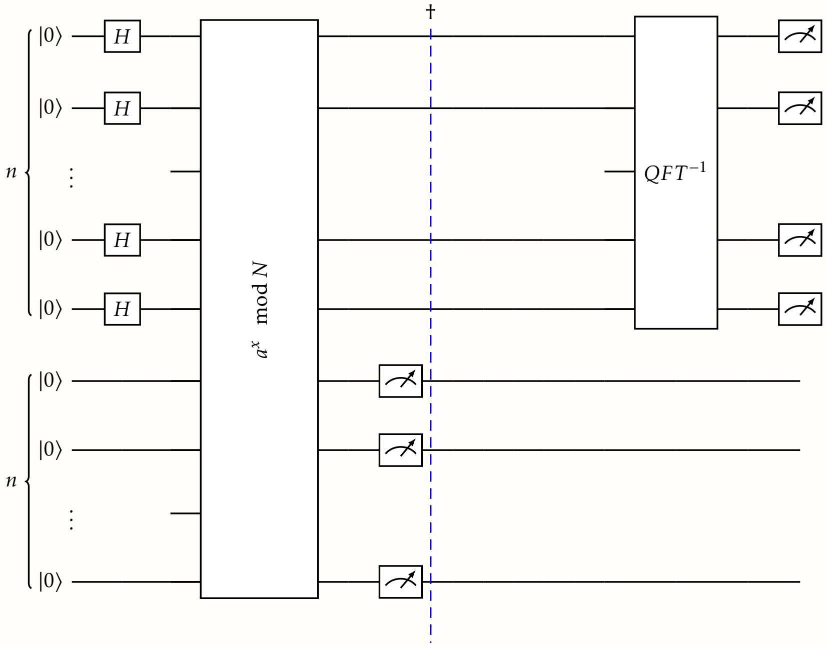


Figure 6.6: Quantum Circuit for Shor's Algorithm

3. *Applying $f(x)$ on the second register*

$$= \frac{1}{4}\left[|0\rangle\,|0 \oplus 13^0 \quad \mod 15\rangle + |0\rangle\,|0 \oplus 13^1 \quad \mod 15\rangle + \cdots\right].$$

Note that 0 $\oplus$ (i.e. XOR) something is the number itself

$$= \frac{1}{4}\Big[\,|0\rangle\,|1\rangle + |1\rangle\,|13\rangle + |2\rangle\,|4\rangle + |3\rangle\,|7\rangle + \tag{6.1}$$

$$|4\rangle\,|1\rangle + |5\rangle\,|13\rangle + |6\rangle\,|4\rangle + |7\rangle\,|7\rangle + \tag{6.2}$$

$$|8\rangle\,|1\rangle + |9\rangle\,|13\rangle + |10\rangle\,|4\rangle + |11\rangle\,|7\rangle + \tag{6.3}$$

$$|12\rangle\,|1\rangle + |13\rangle\,|13\rangle + |14\rangle\,|4\rangle + |15\rangle\,|7\rangle\,\Big]. \tag{6.4}$$

$$\tag{6.5}$$

4. *We now measure the second register (This measurement happens before applying inverse QFT)*
Suppose after measuring second register, we get $|7\rangle$. Implies, we have the superposition $\frac{1}{2}\left[|3\rangle + |7\rangle + |11\rangle + |15\rangle\right] \otimes |7\rangle$. Note the normalisation, $\frac{1}{2}$, i.e, probabilities have changed.

5. *Now apply inverse QFT (Equation (6.3.5)) to the first register.*
If we apply and compute, we will find that phases will interfere and cancel out. The

only terms which will remain are

$$= \frac{1}{8}\left[4\left|0\right\rangle + 4i\left|4\right\rangle + 4\left|8\right\rangle + 4i\left|12\right\rangle\right].$$

6. *The final step is to measure the first register.*
   We will get $\left|0\right\rangle, \left|4\right\rangle, \left|8\right\rangle$ *or* $\left|12\right\rangle$ *with equal probability of* $\frac{1}{4}$.

We have completed the quantum part of Shor's algorithm. After this, all that is left is doing the classical post-processing. The measurement results peak near $j \times \frac{N}{R}$ *for some integer* $j \in \mathbb{Z}$.
Analysing the measurement results:

- $\left|0\right\rangle$ *is trivial. If we measure* $\left|0\right\rangle$*, restart.*
- $\left|4\right\rangle$ $j^{16/R} = 4$ *One possiblity (the lowest one) is* $j = 1$
  Implies $R = 4$ *even, which is good.*
  $x = a^{R/2} \mod N = 13^{4/2} \mod 15 = 13^2 \mod 15 = 4 \mod 15.$
  Therefore, $x \equiv 4 \mod 15$ *and* $x + 1 \equiv 4 + 1 \mod 15 \equiv 5 \mod 15 \not\equiv 0 \mod 15$
  Thereby, P *or* Q *is in* $\{\gcd(x+1,\ N),\ \gcd(x-1,\ N)\}$
  Here $\gcd(4+1,\ 15),\ \gcd(4-1,\ 15) = 5,\ 3$. *So,* $P = 5$ *and* $Q = 3$.
- *For* $\left|8\right\rangle$ *and* $\left|12\right\rangle$*, we get one of the factors, and the algebra works just like above.*

Remarks. *Note that the above phase cancellations were possible because of interference which is a quantum phenomenon. This enables a drastic reduction of terms, thus giving an exponential speed-up compared to classical computers.*

It is known that if we repeat the above algorithm $\mathcal{O}(\log\log(n))$ times and almost guarantee that we find R[4].

6.4 How complex is Shor's Algorithm?

The bottleneck in the quantum factoring algorithm, i.e., the piece of the factoring algorithm that consumes the most time and space, is computing the function $f(x) = a^r \mod N$ modular exponentiation. The modular exponentiation problem is, given N, x, and r, find x^r mod N. The best classical method for doing this is to repeatedly square of $x \mod N$ to get $x^{2^m} \mod N$ for $m \leq \log_2 r$, and then multiply a subset of these powers (mod N) to get $x^r \mod N$. If we are working with $n-$bit numbers, this requires $\mathcal{O}(n)$ squaring and multiplications of $n-$bit numbers (mod N). Asymptotically, the best classical result for gate arrays for multiplication is the Schönhage–Strassen[5] algorithm. This gives a gate array for integer multiplication that uses $\mathcal{O}(n \log n \log\log n)$ gates to multiply two $n-$bit numbers. Thus, asymptotically, modular exponentiation requires $\mathcal{O}(n^2 \log n \log\log n)$ time. Making this reversible would naively cost the same amount in space[6]. However, one can reuse the

[4] This non-trivial calculation can be found out in great detail in Nielsen and Chuang's Quantum computation and quantum information textbook.

[5] Refer to Schönhage and V. Strassen. Concentration inequalities. Schnelle Multiplikation grosser Zahlen, Computing, 7(281–292), 1971.

[6] Refer to Shor's prime factorization and discrete logarithms paper for more details

space used in the repeated squaring part of the algorithm and thus reduce the amount of space needed to essentially require for multiplying two n-bit numbers. Thus, modular exponentiation can be done in $\mathcal{O}(n^2 \log n \log\log n)$ time and $\mathcal{O}(n \log n \log\log n)$ space.

As seen earlier, QFT is $\mathcal{O}(n^2)$ and repeating the above algorithm 6.3.1 $\mathcal{O}(\log\log(n))$ times can almost guarantee that we find r. Overall the time complexity of Shor's algorithm is $\mathcal{O}(n^2 \log n \log\log n)$, which is exponential speed up compared to all classically known algorithms!

6.5 Quantum Phase Estimation

Problem Statemenrt: Given a unitary operator $\mathbf{U}_f$ has an eigenvector $|u\rangle$ with eigenvalue $e^{2\pi i \phi}$, where the value of ϕ is unknown. The goal of the phase estimation algorithm is to estimate ϕ.

Note that given a unitary U ($UU^* = I$); we know that all its eigenvalues have norm 1. Since any complex number can be written as $re^{2\pi i\theta}$, all eigenvalues of U should be of the form $e^{2\pi i\theta}$ for some θ. To determine the eigenvalue, it is enough to find this θ. called the phase of the eigenvalue or eigenbasis.

Quantum phase estimation is a useful subroutine in quantum computing that uses quantum Fourier transform. Suppose we have black boxes capable of preparing the state $|u\rangle$ and performing the controlled-$\mathbf{U}^{2j}$ operation for suitable non-negative integers j.

The phase estimation subroutine, given a unitary U and its eigenvector $|u\rangle$, finds the phase of the eigenvalue corresponding to the eigenvector $|u\rangle$. To be precise, the algorithm will take the eigenvector $|u\rangle$ as input, and it needs the ability to perform controlled $U^{2^i} (i \leq k)$ operations; using those, it determines the corresponding eigenphase.

To start with, we will also assume that we have the ability to perform U^l for all $l \leq 2^k = n$ $\left(\text{instead of just controlled } U^{2^k}\right)$. Later we will show that controlled $U^{2^i} (i \leq k)$ operators can be used to perform U^l for all $l \leq 2^k$.

We will start with the state $|0, u\rangle$, where the first part of the register holds k qubits and second register holds the eigenvector $|u\rangle$. Then we will apply Hadamard on the first part and obtain,

$$\frac{1}{2^{k/2}} \sum_{l=1}^{2^k} |l, u\rangle$$

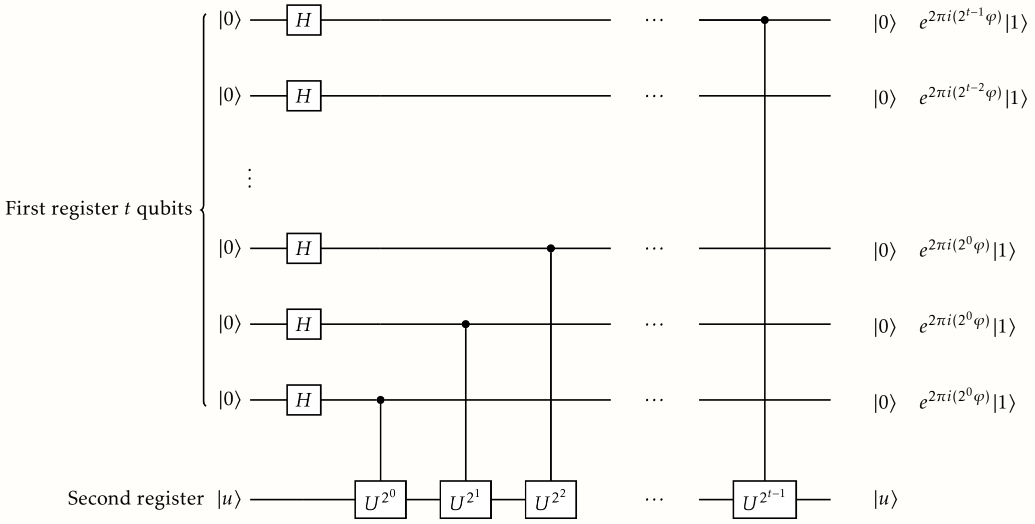
Now we can perform the operation $|l, u\rangle \rightarrow |l\rangle U^l |u\rangle$. Notice that this can be done classically on the basis states and hence can be done quantumly.

This gives us the state,

$$\frac{1}{2^{k/2}} \sum_{l=1}^{2^k} |l\rangle U^l |u\rangle = \frac{1}{2^{k/2}} \sum_{l=1}^{2^k} e^{2\pi i \theta l} |l, u\rangle.$$

Some thought shows that the first part of the register is the Fourier transform of $2^k\theta$. Hence applying inverse Fourier transform, we get the state $\left|2^k\theta\right\rangle$.

$$\frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} e^{2\pi i \varphi j} |j\rangle|u\rangle \rightarrow |\tilde{\varphi}\rangle|u\rangle$$



First register t qubits
$|0\rangle$
H
Second register $|u\rangle$
U^{2^0}
U^{2^1}
U^{2^2}
$U^{2^{t-1}}$
$\cdots$
$|0\rangle$ $e^{2\pi i(2^{t-1}\varphi)}|1\rangle$
$|0\rangle$ $e^{2\pi i(2^{t-2}\varphi)}|1\rangle$
$|0\rangle$ $e^{2\pi i(2^{0}\varphi)}|1\rangle$
$|0\rangle$ $e^{2\pi i(2^{0}\varphi)}|1\rangle$
$|0\rangle$ $e^{2\pi i(2^{0}\varphi)}|1\rangle$
$|u\rangle$

Figure 6.7: Application of controlled-U^k

If we are only given the controlled versions of U^{2^l} where $l \leq k$, then how can we achieve the same phase estimation? Notice that l now varies only up to k. Essentially, we are given the power to apply $U, U^2, U^4 \ldots, U^{2^k}$.

The simple idea is to break any integer $0 \leq h \leq 2^k$ as powers of 2 . Then using the controlled version, we can apply U^h.
Let us see how to take care of the assumptions we made, there are only k bits in the expansion of θ and we have the eigenvector as a quantum state $|u\rangle$.

Most of the time, it is not possible to know the number of digits in the binary expansion of θ beforehand. What can be done in this case? If we want to approximate θ up to k bits of accuracy, using the same circuit with $k + f(\epsilon)$ qubits instead of k qubits will give us the answer with probability $1 - \epsilon$. Here, ϵ should be treated as a parameter, and $f(\epsilon)$ is some

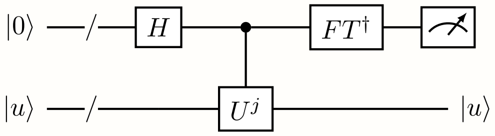


Figure 6.8: Quantum Phase Estimation

function of ϵ.

Suppose we don't have the eigenvector $|u\rangle$. If the same procedure is done over $|\psi\rangle = \sum_i \alpha_i |u_i\rangle$, we will get the phase corresponding to $|u_i\rangle$ with probability $|\alpha_i|^2$.

Chapter 7

Grover's Search Algorithm

> *"There are only two tragedies in life: one is not getting what one wants, and the other is getting it."*
>
> – Oscar Wilde, *Lady Windermere's Fan*

7.1 Introduction

The problem of searching through an unstructured database is ubiquitous, and any improvement will help a lot of applications. If it were an ordered list, we could exploit the order and search faster (like using binary search). But when there is no order in the list, we have no other option, in a classical computer, other than going through all elements one by one, making $\Theta(n)$ queries to see each element. In quantum computing, we can do it in $O(\sqrt{n})$ queries[1]. When the data set is huge, this quadratic 'speed-up' over its classical analog can save a lot of time. The genius behind the quantum search algorithm is Lov Kumar Grover, an Indian-American computer scientist.

In order to better understand the quantum search problem, let us first classically define the search problem and then look at its quantum counterpart, comparing both using the *query complexity model*.

7.2 Query Model of Computation

Imagine your friend Alice has a number in her mind. All you know is that the number is between 1 and 100. Your task is to guess that number by asking Alice only yes or no questions. For example, you can ask her, "Is the number 5?" or "Is the number between 30 and 40?" etc. What is the least number of questions you should ask to find the number?

[1] Even with the advantage of randomness in classical computing, we can show that we need at least $\Omega(n)$ queries by using Yao's minmax theorem.

In the above scenario, you are trying to *search* in a *structured database*. It is structured, as you can imagine, an ordered list of numbers 1 to 100 in your head while asking Alice the questions. Also, Alice represents what is called a classical *oracle*[2]. We know that for such ordered data we can search classically in $O(\log n)$ time (here you need to ask at most 7 questions to find the number as $\log 100 \approx 6.64$), but this is not true for unstructured data. In the upcoming section, we will formalize the notion of an oracle and look into unstructured search.

7.2.1 Classical Oracle

Consider a database with N elements; for convenience, let $N = 2^n$. One can imagine the data stored as a list with consecutive elements in a contiguous memory location. So, associated with each element, there is an ID. One of the elements in this database is marked, and we are interested in finding that element. (For now, let us focus on a single element. There are variants of Grover's algorithm where multiple elements can be marked as well. But the essential idea remains the same in both cases.)

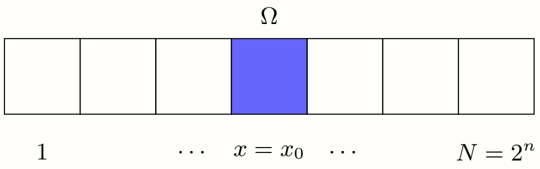


Figure 7.1: Unordered database with N elements

To find the marked element, we will have to query the database by requesting the ID number x of an element in the database and checking if it is equal to the target ID number x_0. Note that as $N = 2^n$, the IDs can be n-bit long.

A classical algorithm to find x_0 is to simply go over the database and check whether each element is the one we are looking for. In terms of the above example, where you were questioning Alice, this time she has already arranged the numbers in boxes randomly and randomly marked one of those boxes. Now there is no order. Every time you can just ask her, "Is the number in the first box?" or "twelfth box," etc. Only she has access to those boxes, and she can open and say yes or no.

Formally, this can be captured by the following function:

$$f(x) = \begin{cases} 1, & \text{if } x = x_0 \ . \\ 0, & \text{otherwise.} \end{cases}$$

Here $f : \{0,1\}^n \to \{0,1\}$, a function that maps the n-bit strings IDs to 1 or 0, corresponding to the yes or no answer given by Alice. Such a function is known as *black box function.*

[2] The word has origins from ancient Greece, where it means advice or information from the gods and often had a hidden meaning.

Classically, we have to make, on average, $O(N)$ queries to f in order to find x_0.

Note that when checking the ID number of an element we do not know what the target ID x_0 is. We just have some method of determining if the queried ID number x is equal to x_0. Thus, 'finding the marked element' is also equivalent to determining what x_0 actually is.

Remarks. *Note that here we are considering the query model of computation. In this model, only the number of queries matters and not any other computational cost. Note that query complexity and time complexity are not equivalent, but query complexity gives a lower bound on the time complexity.*

Also, one must note that separations in the query complexity model do not directly imply separations in the time complexity model.

7.2.2 Quantum Oracle

How to start thinking about the search problems in a quantum computing setup? How to store or represent the elements and what does searching mean here?

In quantum mechanics, everything happens on a Hilbert space. A natural setup is to encode the elements as an orthonormal basis of the Hilbert space. So, given $N = 2^n$ elements, let us consider a Hilbert space of 2^n dimensions with each element encoded as one of the orthonormal bases. Without loss of generality, these elements can be encoded as $\{|0\rangle, |1\rangle \ldots |N\rangle\}$ (tensor product of n-qubits written in shorthand notation). One of the basis vectors is marked, and our task is to find this. Note the similarity of this setup with the concept of searching for ID numbers mentioned in the previous section.

Initially, we have no clue about this marked state. So, a natural starting point is to start with the state $|00\ldots0\rangle$ or an equal superposition of all the basis vectors, say $|\phi\rangle$.

$$|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_i |i\rangle$$

The task now is to devise unitary transformations that will transform the state $|\phi\rangle$ and take it *sufficiently* close to the marked state so that when measured, we get the marked state with a *high* probability.

In the classical case, each time we made a guess, we had the classical oracle to say if it was correct or not. To have a quantum analog of the classical oracle function f, we need to have a unitary that indicates the state we are looking for. Given $f(x)$, we can construct a unitary $\mathbf{O}_f$ such that,

$$\mathbf{O}_f |x, b\rangle \rightarrow |x, b \oplus f(x)\rangle$$

Note that to query the *ith* position, we can set the input qubits to $|i, 0\rangle$. If we set b as the $|+\rangle$ or $|-\rangle$ state, then the action of $\mathbf{O}_f$ is equivalent to putting a phase to the input state depending on the $f(x)$ value.

$$\mathbf{O}_f|x\rangle = (-1)^{f(x)}|x,-\rangle$$
$$\mathbf{O}_f|x\rangle = |x,+\rangle$$

Thus by applying Hadamard gate to the target qubit b we can change $\mathbf{O}_f$ into an oracle that does the following:

$$\mathbf{O'}_f\,|x,b\rangle = (-1)^{f(x).b}\,|x,b\rangle$$

$\mathbf{O'}_f$ is known as the *phase oracle.*

Thus, when $b=1$, the quantum oracle will act on an $n-$qubit quantum state $|x\rangle$ and add a negative phase to the state if it is equal to the target state $|x_0\rangle$ and leaves it unchanged otherwise. Like how the classical oracle returned 1 when the state is $|x_0\rangle$, the action of adding negative phase can be thought of as an indication given by the quantum oracle. We will be using the oracle $\mathbf{O'}_f$ in Grover's algorithm.

7.3 Grover's Search Algorithm

With the base set-up, our task is to find a unitary transformation that takes $|\phi\rangle$ to the marked state, $|x_0\rangle$, given the quantum oracle $\mathbf{O'}_f$.

Let the marked state be,

$$|M\rangle \equiv |x_0\rangle$$

As $|M\rangle$ is also one of the basis state, if we measure $|\phi\rangle$ without doing anything, the probability that we get $|M\rangle$ is $\frac{1}{N}$, where $N=2^n$. This probability is very small, and to increase this, we must increase the coefficient of $|M\rangle$ and thus decrease all other basis elements' coefficients.

In other words, if $|U\rangle$ is an equal superposition of all the unmarked elements,

$$|U\rangle = \frac{1}{\sqrt{N-1}}\sum_{i:x_i\neq x_0}|i\rangle$$

and our goal it to increase c_1 while decreasing c_2.

Note that $|U\rangle$ is orthogonal to $|M\rangle$ and we can write $|\phi\rangle$ as,

$$|\phi\rangle = c_1\,|M\rangle + c_2\,|U\rangle$$

So increasing c_1 corresponds to moving $|\phi\rangle$ closer to $|M\rangle$ on the plane spanned by $|M\rangle$ and $|U\rangle$, as shown in figure 7.2.

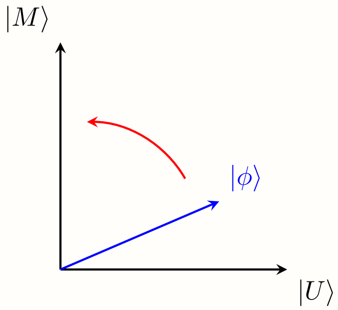


Figure 7.2: Goal is to move $|\phi\rangle$ closer to $|M\rangle$

Reflections and Rotations

- $2\,|a\rangle\,\langle a| - \mathbb{I}$ is rotation about $|a\rangle$:
  One can write out the matrix corresponding to this expression and prove it is a reflection. Otherwise, intuitively, think about what it means to reflect a vector about the y-axis in an $x - y$ plane. It can be seen as adding a negative sign to the orthogonal x-component of that vector. Similarly, in any dimension, reflection about any state $|a\rangle$ is the same as adding a negative sign to all its orthogonal states. That is precisely what $2\,|a\rangle\,\langle a| - \mathbb{I}$ does.

- 2 reflections gives a rotation:
  Imagine two lines, L_1 and L_2 in a 2D plane that intersect at an angle θ between them. For simplicity, let's align L_1 with the x-axis. This means the angle between L_1 and L_2 is simply the angle of L_2, which we call θ. Now, take a vector $\vec{v}$. Let's say it makes an angle α with our first reflection line, L_1. When we reflect $\vec{v}$ across L_1 (the x-axis), its angle flips from α to $-\alpha$. Let's call this new vector $\vec{v'}$. Now, we reflect $\vec{v'}$ (at angle $-\alpha$) across the second line, L_2 (at angle θ). A reflection flips a vector's angle relative to the reflection axis. The angle of $\vec{v'}$ relative to L_2 is $(\theta - (-\alpha)) = \theta + \alpha$. To reflect it, we swing it to the other side of L_2 by that same amount. So, the final vector $\vec{v''}$ will be at an angle of $\theta + (\theta + \alpha) = 2\theta + \alpha$. Our original vector $\vec{v}$ started at an angle α. The final vector $\vec{v''}$ is at an angle $2\theta + \alpha$. The total change is $(2\theta + \alpha) - \alpha = 2\theta$.

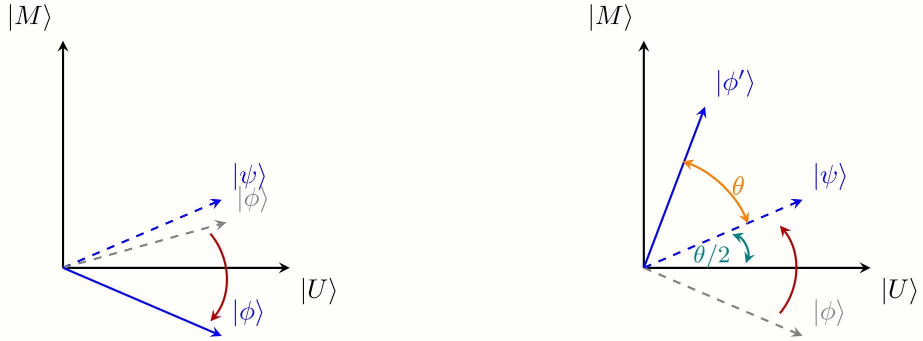
  The state vector lies in the 2D plane spanned by the marked state $|M\rangle$ and the unmarked superposition $|U\rangle$. The two reflections are about the axis $|U\rangle$ (performed by the oracle) and the axis $|\psi\rangle$ (the initial state). If the angle between these two reflection axes is α, then one Grover iteration rotates the state vector by an angle of 2α, moving it closer to the target state $|M\rangle$.

As the phase oracle adds a negative phase only to $|x_0\rangle$ component, its action is precisely $\mathbb{I} - 2\,|x_0\rangle\,\langle x_0|$, equivalently $2\,|U\rangle\,\langle U| - \mathbb{I}$. This is nothing but a reflection about the state $|U\rangle$.

Our goal is to rotate $|\phi\rangle$, and we have one reflection at hand, so all we need is another reflection. As two reflections result in a rotation, as explained in the above box. As we do not have any other information, one natural choice for another axis is to reflect on the equal superposition state $2|\psi\rangle\langle\psi| - \mathbb{I} = H^{\otimes n}(2|0\rangle\langle 0| - \mathbb{I})H^{\otimes n}$, where $|\psi\rangle = \frac{1}{\sqrt{2^n}}\sum_i |i\rangle$.

This rotation that we get by combining both the reflections is called a *Grover's iteration*,

$$G = H^{\otimes n}(2|0\rangle\langle 0| - \mathbb{I})H^{\otimes n}\mathbf{O'}_f$$



(a) Reflection of $|\phi\rangle$ about $|U\rangle$.

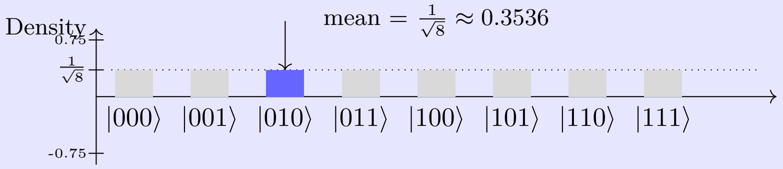
(b) Reflection of $|\phi\rangle$ about $|\psi\rangle$

Figure 7.3: Overall rotation after one iteration of Grover's algorithm.[3]

Now we just need to keep rotating the state a "sufficient number" of times and then measure it to get $|M\rangle$, the state we are searching for. Note that for each application of the Grover iteration G we invoke the oracle $\mathbf{O}_x$ once. Thus, the number of times we rotate determines the query complexity of Grover's algorithm.

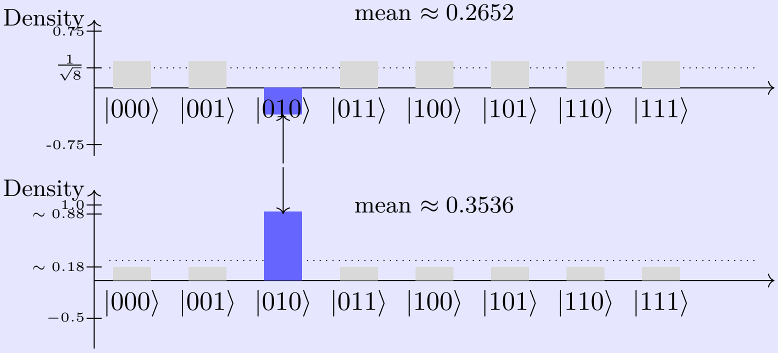
A simple application of Grover's Search Algorithm

For the sake of understanding, let us assume the coefficients of the basis states are real (in general, these are complex valued). If we assume we have 8 elements and one is marked, then our initial $|\phi\rangle$ starts off with equal amplitude and has coefficients like those shown in figure below.



[3] In Figure 7.3(a), $|\phi\rangle$ and $|\psi\rangle$ are exactly the same initially. For clarity, they are drawn slightly apart. Also, the angle θ is exaggerated for illustration. In reality, $|\phi\rangle$ starts very close to $|U\rangle$, so the rotation towards $|M\rangle$ at each iteration is generally small.

The two operations in Grover's iteration are phase flip and reflection about the mean (i.e, $|\psi\rangle$). This corresponds to adding a negative sign to the marked amplitude and flipping about the dotted line shown in figures below.



Notice how, after inverting the phase, flipping about the mean increases the amplitude of the marked state while decreasing the amplitude of all other states. This procedure is applied over and over again, increasing the amplitude of the marked state till it becomes more than $1/2$. As shown earlier, one can prove that this method takes $\sqrt{N}$ steps to do this[a].

[a] A more detailed explanation of this way of visualizing Grover's algorithm can be found in the textbook Dancing with Qubits by Robert S. Sutor.

7.4 Query complexity of Grover's Search Algorithm

Suppose after k Grover iterations we want the state to be very close to $|M\rangle$, this implies the angle between the rotated state and $|U\rangle$ is very close to $\pi/2$. After one Grover iteration, from the figure 7.3, we see that the rotated state is at an angle $\theta + \theta/2$ from $|U\rangle$. Thus, after k iterations we have,

$$\begin{aligned} k\theta + \frac{\theta}{2} &\approx \frac{\pi}{2} \\ \implies k &\approx \frac{\pi - \theta}{2\theta} \end{aligned}$$

As all the state vectors are unit vector $|\psi\rangle \cdot |U\rangle = \cos\theta/2$, this gives $\theta/2 = \cos^{-1}\sqrt{\frac{N-1}{N}}$. We are safe to assume the angle θ is small, therefore

$$\frac{1}{\sqrt{N}} = \sin\theta/2 \approx \theta/2$$

Substituting this we get $k = O(\sqrt{n})$. Thus, in just $O(\sqrt{n})$ queries, we can search the element, giving a quadratic speed-up compared to the classical computer.

Chapter 8

Variational Quantum Algorithms

> *"If one proves the equality of two numbers a and b by showing first that a is less than or equal to b; and then a is greater than or equal to b, it is unfair, one should instead show that they are really equal by disclosing the inner ground for their equality."*
>
> – Emmy Noether, *Biography*

A common goal of variational algorithms is to find the quantum state with the lowest or highest eigenvalue of a certain observable. A key insight is taken from the variational theorem of quantum mechanics.

8.1 Variational Theorem

By the spectral theorem, Hamiltonian being Hermitian can be written as,

$$\hat{\mathcal{H}} = \sum_{k=0}^{N-1} \lambda_k \left|\phi_k\right\rangle \left\langle\phi_k\right|$$

where N is the dimensionality of the space of states, λ_k is the k-th eigenvalue or, physically, the k-th energy level, and $\left|\phi_k\right\rangle$ is the corresponding eigenstate: $\hat{\mathcal{H}} \left|\phi_k\right\rangle = \lambda_k \left|\phi_k\right\rangle$, the expected energy of a system in the (normalized) state $\left|\psi\right\rangle$ will be:

$$\begin{aligned}
\langle\psi|\hat{\mathcal{H}}|\psi\rangle &= \langle\psi| \left(\sum_{k=0}^{N-1} \lambda_k \left|\phi_k\right\rangle \left\langle\phi_k\right| \right) \left|\psi\right\rangle \\
&= \sum_{k=0}^{N-1} \lambda_k \left\langle\psi \mid \phi_k\right\rangle \left\langle\phi_k \mid \psi\right\rangle \\
&= \sum_{k=0}^{N-1} \lambda_k \left|\left\langle\psi \mid \phi_k\right\rangle\right|^2
\end{aligned}$$

If we take into account that $\lambda_0 \leq \lambda_k, \forall k$, we have:

$$
\begin{aligned}
\langle\psi|\hat{\mathcal{H}}|\psi\rangle &= \sum_{k=0}^{N-1} \lambda_k \left|\langle\psi \mid \phi_k\rangle\right|^2 \\
&\geq \sum_{k=0}^{N-1} \lambda_0 \left|\langle\psi \mid \phi_k\rangle\right|^2 \\
&= \lambda_0 \sum_{k=0}^{N-1} \left|\langle\psi \mid \phi_k\rangle\right|^2 \\
&= \lambda_0
\end{aligned}
$$

Since $\{|\phi_k\rangle\}_{k=0}^{N-1}$ is an orthonormal basis, the probability of measuring $|\phi_k\rangle$ is $p_k = \left|\langle\psi \mid \phi_k\rangle\right|^2$, and the sum of all probabilities is such that $\sum_{k=0}^{N-1} \left|\langle\psi \mid \phi_k\rangle\right|^2 = \sum_{k=0}^{N-1} p_k = 1$. In short, the expected energy of any system is higher than the lowest energy or ground state energy:

$$
\langle\psi|\hat{\mathcal{H}}|\psi\rangle \geq \lambda_0
$$

The above argument applies to any valid (normalized) quantum state $|\psi\rangle$, so it is perfectly possible to consider parametrized states $|\psi(\vec{\theta})\rangle$ that depend on a parameter vector $\vec{\theta}$. This is where the "variational" part comes into play. If we consider a cost function given by $C(\vec{\theta}) := \langle\psi(\vec{\theta})|\hat{\mathcal{H}}|\psi(\vec{\theta})\rangle$ and want to minimize it, the minimum will always satisfy:

$$
\min_{\vec{\theta}} C(\vec{\theta}) = \min_{\vec{\theta}} \langle\psi(\vec{\theta})|\hat{\mathcal{H}}|\psi(\vec{\theta})\rangle \geq \lambda_0
$$

The minimum value of $C(\vec{\theta})$ will be the closest that one can get to λ_0 using the parametrized states $|\psi(\vec{\theta})\rangle$, and equality will only be reached if there exists a parameter vector $\vec{\theta}^*$ such that $\left|\psi\left(\vec{\theta}^*\right)\right\rangle = |\phi_0\rangle$.

If the (normalized) state $|\psi\rangle$ of a quantum system depends on a parameter vector $\vec{\theta}$, then the optimal approximation of the ground state (i.e. the eigenstate $|\phi_0\rangle$ with the minimum eigenvalue λ_0) is the one that minimizes the expectation value of the Hamiltonian $\hat{\mathcal{H}}$:

$$
\langle\hat{\mathcal{H}}\rangle(\vec{\theta}) := \langle\psi(\vec{\theta})|\hat{\mathcal{H}}|\psi(\vec{\theta})\rangle \geq \lambda_0
$$

The reason why the variational theorem is stated in terms of energy minima is that it includes a number of mathematical assumptions: - For physical reasons, a finite lower bound to the energy $E \geq \lambda_0 > -\infty$ needs to exist, even for $N \rightarrow \infty$. - Upper bounds do not generally exist.

However, mathematically speaking, there is nothing special about the Hamiltonian $\hat{\mathcal{H}}$ beyond these assumptions, so the theorem can be generalized to other quantum observables and their eigenstates provided they follow the same constraints. Also, note that if finite upper bounds exist, the same mathematical arguments could be made for maximizing eigenvalues by swapping lower bounds for upper bounds.If the (normalized) state $|\psi\rangle$ of a quantum system depends on a parameter vector $\vec{\theta}$, then the optimal approximation of the ground state (i.e. the eigenstate $|\phi_0\rangle$ with the minimum eigenvalue λ_0) is the one that minimizes the expectation value of the Hamiltonian $\hat{\mathcal{H}}$:

$$\langle\hat{\mathcal{H}}\rangle(\vec{\theta}) := \langle\psi(\vec{\theta})|\hat{\mathcal{H}}|\psi(\vec{\theta})\rangle \geq \lambda_0$$

The reason why the variational theorem is stated in terms of energy minima is that it includes a number of mathematical assumptions: - For physical reasons, a finite lower bound to the energy $E \geq \lambda_0 > -\infty$ needs to exist, even for $N \to \infty$. - Upper bounds do not generally exist.
However, mathematically speaking, there is nothing special about the Hamiltonian $\hat{\mathcal{H}}$ beyond these assumptions, so the theorem can be generalized to other quantum observables and their eigenstates, provided they follow the same constraints. Also, note that if finite upper bounds exist, the same mathematical arguments could be made for maximizing eigenvalues by swapping lower bounds for upper bounds.

8.2 Quantum Approximation Optimisation Algorithm

Quantum Approximate Optimisation Algorithm (QAOA) is a variational quantum algorithm designed for combinatorial optimisation problems.[1]

General working :

- The optimisation problem is encoded in the Hamiltonian of a quantum system, using some objective function C_z
- A quantum circuit is constructed that encodes the potential solution
- The parameters in the circuit are optimised classically to extremise the expectation value of the Hamiltonian
- Final measured state provides solutions to the original optimisation problem
- This is iterated to improve the quality of the solution
- Final solution is represented in the computational basis, with the combinatorial solutions having the highest probabilities
- The goal is to find a solution such that $\frac{C(z)}{C} \geq r$, where r is some approximation ratio

8.2.1 Max Cut

Problem Statement: Given a graph, label the nodes as A and B such that the number of edges connecting A to B is maximized.

Decision Version: Given a graph G and an integer k, determine whether there is a cut of size at least k in G.

[1] It was introduced first by Farhi and Goldstone in 2014 https://arxiv.org/pdf/1411.4028

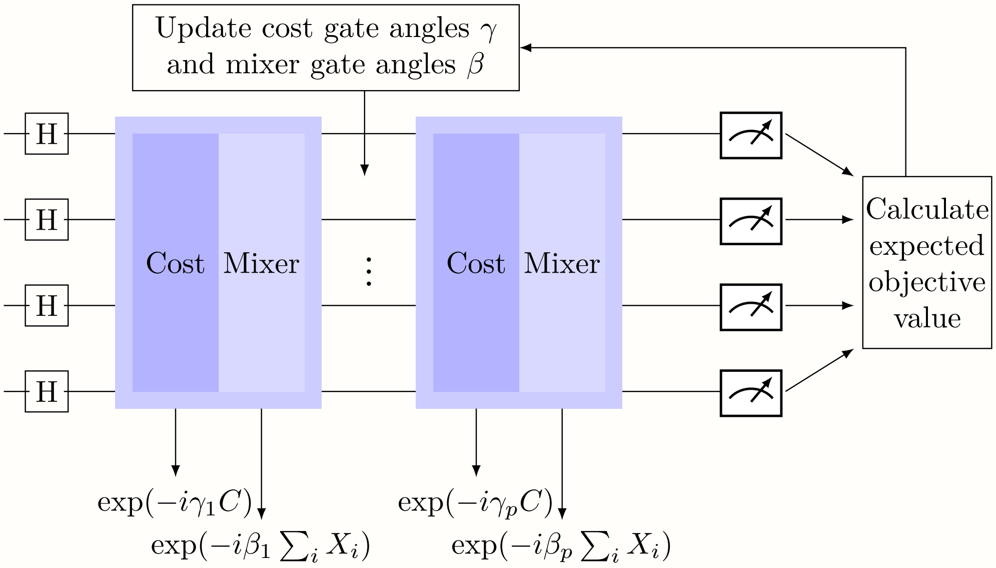

Update cost gate angles γ and mixer gate angles β
H
H
H
H
Cost
Mixer
Cost
Mixer
Calculate expected objective value
$\exp(-i\gamma_1 C)$
$\exp(-i\beta_1 \sum_i X_i)$
$\exp(-i\gamma_p C)$
$\exp(-i\beta_p \sum_i X_i)$

Figure 8.1: General working of QAOA

If we define *cut size* as the number of edges connecting A to B, then the goal here is to find the maximum value of cut across all partitioning into A and B. The decision version of the max cut problem is an NP-Hard problem.

The task is to model this problem as a suitable Hamiltonian. Let $x_i \in \{-1, 1\}$ be the label on the i^{th} node. The quantity $1 - x_i x_j$ is 0 if i, j have the same label and 2 if they have opposite.

$$H = -\frac{1}{2} \sum_{(i,j) \in E(G)} 1 - x_i x_j$$

Note that the negative sign in the Hamiltonian converts the maximization problem to a minimization problem.

8.2.2 Max Independent Set

Problem Statement: Given a graph, a subset A of nodes is an *independent set* if there are no edges connecting elements of A to elements of A. The largest among all independent sets is the *maximum independent set.* The goal is to find the size of the maximum independent set.

Decision Version: Does a given graph have an independent set of size k?

The decision version of the maximum independent set problem is NP-Hard.

Define a variable n_i for each node. It is 1 if the node is in the set A and) otherwise.

$$H = \sum_{(i,j)\in E(G)} n_i n_j$$

It is 0 if A is an independent set. There are multiple possible independent sets and all get value 0 here.

Now, we add a term to split this degeneracy and separate out the maximum independent set.

$$H = \sum_{(i,j)\in E(G)} n_i n_j - \Delta \sum_{k\in V(G)} n_k$$

Here again, minimizing H corresponds to the maximum independent set.

Part III

Quantum Information

Chapter 9

Generalising Operations

> *"If someone gave me a practical quantum computer tomorrow, then I confess that I can't think of anything that I, personally, would want to use it for: only things that other people could use it for!"*
>
> – Scott Aaronson, *Quantum Computing since Democritus*

9.1 Schmidt Decomposition and Purification

In quantum mechanics, the Schmidt decomposition is a wonderfully insightful tool. It's essentially the Singular Value Decomposition (SVD) as mentioned earlier, applied to a quantum state that is shared between two separate systems, often called a bipartite state. Its real power lies in how it elegantly reveals the fundamental connections the entanglement between the two parts of the system.

Theorem 9.1.1 (Schmidt Decomposition)**.** *Take any quantum state, $|\Psi\rangle$, that is shared between two systems, A and B (living in Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$ respectively). The theorem guarantees that we can always find a special pair of orthonormal bases, $\{|u_i\rangle\}$ for system A and $\{|v_i\rangle\}$ for system B, which align perfectly with each other. Along with a set of positive real numbers $\{\lambda_i\}$, called the Schmidt coefficients, we can write the shared state in a remarkably simple form:*

$$|\Psi\rangle = \sum_i \lambda_i |u_i\rangle \otimes |v_i\rangle,$$

where the coefficients are normalized such that $\sum_i \lambda_i^2 = 1$.

The Schmidt decomposition theorem for a pure state $|\psi\rangle \in \mathcal{H} = \mathcal{H}_A \oplus \mathcal{H}_B$ of a bipartite quantum system, then there would exist orthonormal states $\{|l_A\rangle\}$ for $\mathcal{H}_A$, and $\{|l_B\rangle\}$ for $\mathcal{H}_B$, such that

$$|\psi\rangle = \sum_{l=1}^{\mathscr{R}} \sqrt{\lambda_l} |l_A\rangle |l_B\rangle$$

with λ_i positive real numbers satisfying $\sum_{i=1}^{\mathscr{R}} \lambda_i = 1$, where $\mathscr{R}$ denotes the Schmidt rank of the state $|\psi\rangle$ given by the number of non-zero eigenvalues of the reduced density matrices

$\rho_A = \mathrm{Tr}_B \rho$ and $\rho_B = \mathrm{Tr}_A \rho$. Naively, the Schmidt number can be a criterion for entanglement, but not a measure of entanglement, since a bipartite pure state is entangled if and only if its Schmidt number is greater than one.

On a related note, given a quantum system described by a density matrix ρ_A, we can introduce another system ρ_B such that the state of the composite system is a pure state and $\rho_A = \mathrm{Tr}_B \rho = \mathrm{Tr}_B\{|\psi\rangle\langle\psi|\}$. To see this, consider a generic pure state for the global system given by the expression

$$|\psi\rangle = \sum_{lk} c_{lk} |l_A\rangle\langle k_B|$$

with $\{|l_A\rangle\}$ and $\{|k_B\rangle\}$ as basis sets for the subsystems. The corresponding density matrix is thereby,

$$\rho = \sum_{lk} \sum_{l'k'} c_{lk} c^*_{l'k'} |l_A\rangle|k_B\rangle\langle l'_A|\langle k'_B|$$

whose trace can be evaluated as

$$\begin{aligned}
\rho_A = \mathrm{Tr}_B \rho &= \sum_{k''} \langle k''_B| \left(\sum_{lk} \sum_{l'k'} c_{lk} c^*_{l'k'} |l_A\rangle|k_B\rangle\langle l'_A|\langle k'_B| \right) |k''_B\rangle \\
&= \sum_{k''} \sum_{lk} \sum_{l'k'} c_{lk} c^*_{l'k'} |l_A\rangle\langle k''_B|k_B\rangle\langle l'_A|\langle k'_B|k''_B\rangle \\
&= \sum_{lk} \sum_{l'k'} c_{lk} c^*_{l'k'} |l_A\rangle\langle l'_A| \underbrace{\langle k'_B| \left(\sum_{k''} |k''_B\rangle\langle k''_B| \right) |k_B\rangle}_{=\langle k'_B|k_B\rangle = \delta_{kk'}} \\
&= \sum_{k} \sum_{ll'} c_{lk} c^*_{l'k} |l_A\rangle\langle l'_A|
\end{aligned}$$

Thus, the coefficients of the density matrix in the expansion in its subsystem must obey the relation

$$(\rho_A)_{ll'} = \sum_k c_{lk} c^*_{l'k}$$

which attributes correctly and has solutions, provided the Hilbert space of the adjoint system is large enough, with at least the same dimension.

Purifying a qubit

Consider a qubit with density matrix ρ_A, and we adjoin another ancillary qubit for its purification. From the above condition, we have the following set of equations:

$$\begin{aligned}
(\rho_A)_{00} &= c_{00} c^*_{00} + c_{01} c^*_{01}, \\
(\rho_A)_{01} &= c_{00} c^*_{10} + c_{01} c^*_{11} = (\rho_A)^*_{10} \\
(\rho_A)_{11} &= c_{10} c^*_{10} + c_{11} c^*_{11}
\end{aligned}$$

which can be solved to give,

$$c_{00} = \sqrt{(\rho_A)_{00}}, \quad c_{01} = 0, \quad c_{10} = \frac{(\rho_A)_{01}^*}{\sqrt{(\rho_A)_{00}}}, \quad c_{11} = \sqrt{\frac{(\rho_A)_{10}(\rho_A)_{11} - |(\rho_A)_{01}|^2}{(\rho_A)_{00}}}$$

leading us to the purification. For a two-qubit system, it is thus possible to generate any density matrix ρ_A for one of the two qubits through unitary operations on that system.

9.2 Kraus Representation

To motivate the idea of a superoperator transforming density operators to density operators, we first look into the evolution of the density matrix under unitary evolution. To simplify things, we work with simple mixed states that evolve under unitary transforms as $|\varphi\rangle \longmapsto |\varphi'\rangle = \mathbf{U}|\varphi\rangle$. Equivalently, we have the density matrix of the new state as

$$\begin{aligned}\rho' &= \sum_i p_i |\varphi_i'\rangle\langle\varphi_i'| \\ &= \sum_i p_i \mathbf{U}|\varphi_i\rangle\langle\varphi_i|\mathbf{U}^\dagger \\ &= \mathbf{U}\left(\sum_i p_i|\varphi_i\rangle\langle\varphi_i|\right)\mathbf{U}^\dagger \quad \text{(by linearity)} \\ &= \mathbf{U}\rho\mathbf{U}^\dagger\end{aligned}$$

As we see above, density matrices transform critically through

$$\rho \longmapsto \rho' = \xi(\rho)$$

The quantum operation generalises the dynamic change to a state that occurs as the result of some physical process, with ρ being the initial state before the process, and $\xi(\rho)$ the final state after the process occurs, possibly up to some normalisation factor.

A natural way to describe the dynamics of an open quantum system is to regard it as arising from an interaction between the system of interest, which we shall call the principal system, and an environment, which together form a closed quantum system.

We can generalise this notion of density operator mapping through the *Kraus operator-sum representation* as the map defined by a set of $\{E_k\}$ operators, with

$$\xi : \rho \to \rho' = \sum_k E_k \rho E_k^\dagger$$

such that the completeness relation is satisfied

$$\sum E_k^\dagger E_k = \mathbb{I}$$

which maps density operators to density operators obeying

- Hermiticity preserving:

$$\rho'^{\dagger} = \left(\sum_k E_k \rho E_k^{\dagger}\right)^{\dagger} = \sum_k (E_k^{\dagger})^{\dagger} \rho^{\dagger} E_k^{\dagger} = \sum_k E_k \rho E_k^{\dagger} = \rho'$$

- Unit Trace preserving:

$$\text{Tr}(\rho') = \text{Tr}\left\{\sum_k E_k \rho E_k^{\dagger}\right\} = \sum_k \text{Tr}\{E_k \rho E_k^{\dagger}\} = \text{Tr}\left\{\rho \sum_k E_k E_k^{\dagger}\right\} = \text{Tr}\rho = 1$$

- Positive semi-definiteness preserving:

$$\langle\phi|\rho'|\phi\rangle = \sum_k \langle\phi|E_k \rho E_k^{\dagger}|\phi\rangle \equiv \sum_k \underbrace{\langle\varphi_k|}_{\langle\varphi_k|=E_k\langle\phi|} \quad \rho \quad \underbrace{|\varphi_k\rangle}_{|\varphi_k\rangle=E_k^{\dagger}|\phi\rangle} \quad \geq 0$$

Analogous to the natural notion of a Kraus operator arising for a unitary evolution of the system, we can solve the converse problem of given a Kraus representation for a specific system, it is possible to introduce an auxiliary system for a global unitary evolution. Given a state $|\psi_A\rangle$, we can construct an inner product preserving unitary operator U acting on a bigger system, through the superoperator as

$$U|\psi_A\rangle|0_B\rangle = \sum_k E_k|\psi_A\rangle|k_B\rangle$$

where $\{k_B\}$ represents an orthonormal basis for the extended system, whose unitarity arises from the completeness relation of the Kraus operators.

We can further provide a physical understanding of the process of quantum operation through the equivalence of unitary operations on the global system. This structure gives rise to a probabilistic notion of a noisy channel, which we shall explore further later. Naively, we can define the probability of the k^{th} operator through $p(k) = \text{Tr}(E_k \rho E_k^{\dagger})$, and the k^{th} density matrix can be normalized to give

$$\rho_k = \frac{E_k \rho E_k^{\dagger}}{\text{Tr}(E_k \rho E_k^{\dagger})}$$

such that the quantum operation beautifully maps to a noisy communication channel where a state ρ is probabilistically replaced by the state ρ_k, with

$$\xi(\rho) = \sum_k E_k \rho E_k^{\dagger} = \sum_k \text{Tr}(E_k \rho E_k^{\dagger}) \frac{E_k \rho E_k^{\dagger}}{\text{Tr}(E_k \rho E_k^{\dagger})} = \sum_k p(k)\rho_k$$

There exists an intricate structure to the notion of superoperators, where we can compose two Kraus operators ξ_A and ξ_B to give rise to

$$\xi = \xi_B \xi_A, \quad \xi(\rho) = \xi_B(\xi_A(\rho))$$

which gives rise to a *semigroup* structure due to the non existence of an invertible structure unless unitarity is maintained. The non-existence of invertibility gives rise to a physical notion of describing an evolution from t_0 to t_1, but not the reverse. This can be seen as a loss of information from the system to the auxiliary adjoint system: environment, and we can't run the evolution backwards. This phenomenon gives rise to *decoherence* and will be delved in detail further.

We also note that the different representations can give rise to the same superoperator. If two superoperators coincide $\xi(\rho) = \sum_k E_k \rho E_k^\dagger$ and $\xi'(\rho) = \sum_k F_k \rho F_k^\dagger$, if and only if there exists an unitary matrix $\mathscr{U}$ such that

$$F_i = \sum_j \mathscr{U}_{ij} E_j$$

This can be shown by noting that two states produce the same density operator if there exists an unitary matrix transforming one state to the other, understood as an effective change of basis, but representing the same state, as

$$|\psi_i\rangle = \sum_j \mathscr{U}_{ij} |\varphi_j\rangle$$

This results in an untiary freedom in the operator sum representaion.

Now, we are equipped to tackle the fundamental representation theorem:

Theorem 9.2.1. *A map $\xi : \rho \to \rho'$ satisfying the following requirements:*

- *linearity: $\xi(p_A \rho_A + p_B \rho_B) = p_A \xi(\rho_A) + p_B \xi(\rho_B)$,*
- *preserves hermiticity,*
- *preserves trace,*
- *is completely positive,*

  has an operator-sum representation given by

$$\rho' = \sum_k E_k \rho E_k^\dagger$$

where E_k satisfy $\sum_k E_k^\dagger E_k = \mathbb{I}$,

For clarification, the *completely positive* is a stronger property than positive, which primarily constrains the positive nature of the density matrices. Complete positivity implies, for any extension of the Hilbert space $\mathcal{H}_{\mathcal{A}}$, to $\mathcal{H}_A \otimes \mathcal{H}_B$, the superoperator $\xi \otimes \mathbb{I}$ must be positive.

Proof. Given a system $\mathcal{H}_A$ satisfying the above axioms, we shall consider an auxiliary system $\mathcal{H}_B$ of the same dimension. Let $|l_A\rangle$ and $|l_B\rangle$ be the orthonormal basis to define the maximally entangled state of $\mathcal{H}_A \otimes \mathcal{H}_B$ as

$$|\ell\rangle := \sum_l |l_A\rangle |l_B\rangle$$

Given a quantum operation ξ, we further define the operator on the maximally entangled state given by

$$\tilde{\xi} := (\mathbb{I}_A \otimes \xi)(|\ell\rangle\langle\ell|) = \sum_{ll'} (|l_A\rangle|\langle l'_A|)\xi(|l_B\rangle\langle l'_B|)$$

The beauty of the argument relies on the notion that the operator $\tilde{\xi}$ provides a complete description of the quantum operation ξ. To understand the effect of ξ on an arbitrary state of $\mathcal{H}_B$, it is sufficient to know the action on the single maximally entangled state with the auxiliary system. To recover ξ from $\tilde{\xi}$, we note, for a state $|\psi_B\rangle = \sum_k c_k|k_B\rangle$ in $\mathcal{H}_B$, we define a corresponding state $|\psi_A\rangle$ in $\mathcal{H}_A$ through

$$|\psi_A\rangle = \sum_k c_k^*|k_A\rangle$$

such that the effect of ξ can be recovered from $\tilde{\xi}$ through the partial trace as

$$\begin{aligned}
\langle\psi_A|\tilde{\xi}|\psi_A\rangle = \langle\psi_A|\left((\mathbb{I}_A \otimes \xi)(|\ell\rangle\langle\ell|)\right)|\psi_A\rangle &= \langle\psi_A|\left(\sum_{ll'}(|l_A\rangle|\langle l'_A|)\xi(|l_B\rangle\langle l'_B|)\right)|\psi_A\rangle \\
&= \sum_k c_k\langle k_A|\left(\sum_{ll'}(|l_A\rangle|\langle l'_A|)\xi(|l_B\rangle\langle l'_B|)\right)\sum_{k'} c_{k'}^*|k'_A\rangle \\
&= \sum_{ll'}\sum_{kk'} c_k c_{k'}^* \underbrace{\langle k_A|l_A\rangle}_{\delta_{lk}} \xi(|l_B\rangle\langle l'_B|) \underbrace{\langle l'_A|k'_A\rangle}_{\delta_{l'k'}} \\
&= \sum_{ll'} c_l c_{l'}^* \xi(|l_B\rangle\langle l'_B|) = \xi\left(\sum_l c_l|l_B\rangle \sum_{l'} c_{l'}^*\rangle l'_B|\right) = \xi(|\psi_B\rangle\langle\psi_B)
\end{aligned}$$

Suppose now there exists some decomposition of $\tilde{\xi}$ as $\tilde{\xi} = \sum_i p_i|j_i\rangle\langle j_i|$ and we consequently define the map

$$E_i|\psi_B\rangle := \sqrt{p_i}\langle\psi_A|j_i\rangle$$

This linear map can be decomposed as

$$\begin{aligned}
\sum_i E_i|\psi_B\rangle\langle\psi_B|E_i^\dagger &= \sum_i p_i\langle\psi_A|j_i\rangle\langle j_i|\psi_A\rangle \\
&= \langle\psi_A|\left(\sum_i p_i|j_i\rangle\langle j_i|\right)|\psi_A\rangle \\
&= \langle\psi_A|\tilde{\xi}|\psi_A\rangle = \xi(|\psi_B\rangle\langle\psi_B|)
\end{aligned}$$

Thereby, we have

$$\xi(|\psi_B\rangle\langle\psi_B|) = \sum_i E_i|\psi_B\rangle\langle\psi_B|E_i^\dagger$$

for all pure states $|\psi_B\rangle$ of $\mathcal{H}_B$. By convex linearity, it follows that

$$\xi(\rho) = \sum_i E_i\rho E_i^\dagger$$

$\blacksquare$

In other words, the Kraus representation theorem infers that, if the evolution of a density matrix $\rho_B \to \rho'_B = \xi(\rho_B)$ preserves hermiticity and trace, is linear and completely positive, then the evolution can be realised by the unitary transformation, acting on a larger Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$.

The limitation of the formalism of a quantum operation is that systems that interact with degrees of freedom are used to prepare the system, even after the preparation is complete.

9.3 Generalised Measurements

A generalised measurement is described by a set $\{M_i\}$ of measurement operators, not necessarily self-adjoint, that satisfy the completeness relation

$$\sum_i M_i^\dagger M_i = \mathbb{I}$$

with the post-measurement state with outcome i is the

$$|\psi'_i\rangle = \frac{M_i|\psi\rangle}{\sqrt{\langle\psi|M_i^\dagger M_i|\psi\rangle}}$$

with probability of measurement

$$p_i = \langle\psi|M_i^\dagger M_i|\psi\rangle$$

We realise that the completeness relation results in unit probabilities. Further, the projective measurements described earlier are a special case of generalised measurements in which the operators M_i are orthogonal projectors with $M_i^\dagger = M_i$ and $M_i m_j = \delta_{ij} M_i$ with completeness $\sum_i M_i = \mathbb{I}$. Projective measurements together with unitary operations are equivalent to generalised measurements, in a bigger Hilbert space.

Positive Operator-Valued Measure (POVM)

Alice and Bob decide to play a quantum game. Alice has multiple copies of two qubits, each one either in the $|0\rangle$ state or the $|+\rangle$ state. She sends Bob one of these qubits and challenges him to find out which qubit was sent. What strategy do you think Bob can use to find out the qubit was sent correctly every single time? In the first place, is it possible for Bob to win this game every single time?

Suppose Bob decides to measure the qubit in the computational basis. Note that if the outcome is $|1\rangle$, then he can certainly say that $|0\rangle$ was not sent. But what if the outcome was $|0\rangle$? In this case, Bob will not be able to say for certain whether the qubit was $|0\rangle$ or $|+\rangle$. The same holds if he chooses to do the measurement in Pauli X eigenbasis (i.e. $\{|+\rangle, |-\rangle\}$ basis). Regardless of what set of orthogonal projective measurements Bob chooses, he will not be able to distinguish $|0\rangle$ and

$|+\rangle$. Thus, non-orthogonal states can not be perfectly distinguished. So what can he best do? Is there a way to go beyond these orthonormal projective measurements? This brings us to the concept to POVM a broader class of measurements.

As mentioned in the last section of this chapter, generalized measurements need not require the condition $\Pi^2 = \Pi$ which is satisfied by the projective measurements. All we need is that the set of measurement operators give well-defined probabilities.

Consider a set of positive operators $\mathcal{M} = \{E_1, E_2 \ldots E_d\}$ such that $0 \leq E_i \leq \mathbb{I}$ for all $i \in \{1, 2 \ldots, d\}$ and $\sum_i E_i = \mathbb{I}$. (Notice that the inequality is between matrices. $E_i \leq \mathbb{I} \implies \mathbb{I} - E_i \leq 0$, that is we require $\mathbb{I} - E_i$ to be a positive operator. The summation condition ensures that $\mathcal{M}$ gives a valid probability distribution. Such a set of measurements is called *Positive Operator-Valued Measure (POVM).*

Now with the power of POVM, if not distinguish every time, Bob can at least come up with a *zero error discrimination strategy.* Consider the following set of measurements:

$$\mathcal{M} = E_1, E_2, E_3$$

where $E_1 = \frac{\sqrt{2}}{1+\sqrt{2}}|1\rangle\langle 1|$, $E_2 = \frac{\sqrt{2}}{1+\sqrt{2}}|-\rangle\langle -|$ and $E_3 = \mathbb{I} - (E_1 + E_2)$, coefficients chosen such that the condition $0 \leq E_i \leq \mathbb{I}$ for all $i \in \{1, 2 \ldots, d\}$.

Using this when Bob does the measurement, if the outcome is that of E_1, he can certainly say Alice sent $|+\rangle$ state, and if it was E_2 again with certainty, he can say she sent $|0\rangle$ state. But if the outcome is that of E_3, then he will not be able to say which state it was. Thus, whenever he is able to find out the state, he can do it with complete certainty. Such strategies are called *zero error discrimination strategy*, which is one of many advantages opened up by POVM measurements.

Chapter 10

Quantum Entropy

"The theory 'All crows are black' is refuted by the single observation of a white crow, while the theory 'Some crows are black' is not refuted by the observation of a thousand white crows."

– F. Bavaud, *Information theory (paraphrasing Popper)*

10.1 Shannon Entropy

Suppose a random variable X can take values $x_1, x_2 \dots x_n$ with some probabilities $p_1, p_2 \dots p_n$ respectively. How can one quantify how much information is gained by knowing the value of X?

Intuitively, we would want the information function, say $I(X)$, to depend on p_i's and not the labels x_i's, as the event of 50% head and 50% tail occurrence should contain the same information as the event of 50% one and 50% zero. Also, the information content should not have drastic jumps or falls with slight tweaks in the probabilities. And it is also reasonable to expect that the information gained when two independent events occur with individual probabilities p and q is the sum of the information gained from each event alone.

One can show that the function $I(p) = k \log p$ for some constant k satisfies all the above-stated intuitive conditions. More formally $I(p)$ follows:

- I is a function of $p_1, p_2 \dots p_n$ and not $x_1, x_2 \dots x_n$,
- $I(p)$ is a smooth function,
- $I(pq) = I(p) + I(q)$.

Given the above intuition, let us see the definition of *Shannon entropy.* Also note that when we say $\log N$ we always mean logarithm to the base 2. This is adopted as in the most basic form, information in the current day digital computers is represented as 1 or 0, in other words as *bits.*

Definition 10.1.1. *(Shannon entropy) Given a probability distribution $p_1, p_2 \dots p_n$, the Shannon entropy associated with this probability distribution is*

$$H(X) \equiv H(p_1, p_2, \dots, p_n) \equiv -\sum_i p_i \log p_i$$

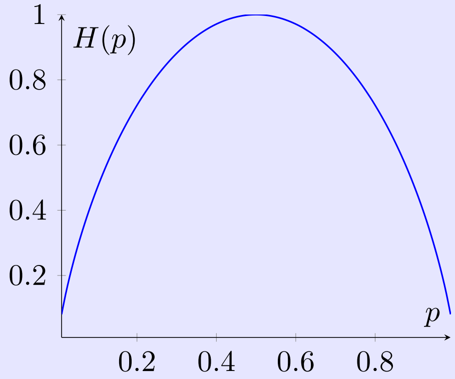
Events that never occur, $p_i = 0$, are not considered in the calculation of entropy as intuitively they do not add to the information content of E (More rigorously, one can also argue that $\lim\limits_{p_i \to 0} p_i \log p_i = 0$). On average, Shannon entropy quantifies the information gain when we learn the value of a contextual bit in a message.

Shanon Binary entropy

Consider a two-state system for $n = 2$ and define $p_1 = p$ where $0 \leq p \leq 1$, hence $p_2 = 1 - p$, thereby, the Shanon binary entropy is a function of p alone as

$$H(p_1 = p, p_2 = 1 - p) = -p \log p - (1 - p) \log(1 - p)$$

which can be visualized graphically, in a simple plot



Note that the entropy equals zero at $p = 0$ or $p = 1$, and attains the maximal value when $p = \frac{1}{2}$. This is consistently well defined as a notion of entropy since it processes the average information content of each letter in a message. Information is a measure of *a priori ignorance*. If we already know that we shall receive a with certainty ($p = 1$), then no information is gained from its reception, and similarly for receiving b when $p = 0$. For the equiprobable case, ignorance is maximum; hence, the maximum possible information is available.

The above example can be elaborated to a general case to understand that $H(p_1, p_2, \dots, p_n)$ is maximum when $p_1 = p_2 = \dots = p_n = \frac{1}{n}$.

Defining a quantity called *relative entropy* is equally useful, which can measure how close two probability distributions are.

Definition 10.1.2. *(Relative entropy) Given two probability distributions p and q, the relative intensity is defined as*

$$H(p||q) \equiv \sum_i p_i \log \frac{p_i}{q_i} \equiv -H(X) - \sum_i p_i \log q_i$$

Conventionally, $\lim_{q_i \to 0} -p_i \log q_i \to \infty$. The motivation for defining relative entropy as above comes from the following theorem.

Theorem 10.1.1. *The relative entropy is non-negative and is zero if and only if the two probability distributions are equal.*

Proof. Using the identity that $\log x = \frac{\ln x}{\ln 2} \leq x - 1$ we can write

$$\begin{aligned} H(p||q) &= -\sum_i p_i \log \frac{q_i}{p_i} \geq \frac{1}{\ln 2} \sum_i p_i \left(1 - \frac{q_i}{p_i}\right) \\ &= \frac{1}{\ln 2} \sum_i (p_i - q_i) = \frac{1}{\ln 2}(1-1) = 0 \end{aligned}$$

We can see that equality is held when $p_i = q_i$ for all x_i's. ■

Corollary 10.1.1.1. *If X has d outcomes, then $H(X) \leq \log d$ with equality if and only if X is a uniform distribution.*

10.2 Classical Data Compression

Can information be efficiently stored by compressing the bit string when given in a series of bit strings? If possible, to what fraction can we compress? As a first insight into classical data compression, we employ a lossless data compression technique that assigns variable-length codes to characters based on their frequency of occurrence in the data.

Huffman Data Encoding

Huffman coding uses a greedy algorithm to build a prefix tree that optimises the encoding scheme so that the most frequently used symbols have the shortest encoding. Consider a message written in the alphabet such that the frequency of occurrence of different letters is different, due to a probability distribution. To send a code word, we need $\sum_i p_i l_i$ bits where l_i is the length, in bits, of the coded letter. Note that the good strategy, here as in any other useful compression code, is to encode the most probable strings in the shortest sequences and the less probable strings in the longest sequences.

To address these questions mathematically rigorously, let us consider an information source that produces independent and identically distributed bits $X_1, X_2 \dots$ each of which is zero with probability p and one otherwise. Though sources often do not behave in the real world in this fashion, this is a good approximation and works well in most cases.

For a more concrete understanding, consider X_i as the i^{th} coin toss with a head occurring with $p = 0.4$. We know that in the large n limit, we will likely find 0.4 fraction of the tosses to be heads and the remaining tails. We call such sequences *typical sequences.* Formally defined as follows.

Definition 10.2.1. *(Typical sequence) Given $X_1 \dots X_n$ with each X_i equal to 0 with probability p and 1 with probability $1 - p$. In the large n limit, we expect with high probability a fraction p of the X_i's to be zero and the remaining ones. A sequence $x_1 \dots x_n$ for which this is true is called a typical sequence. Those that do not follow this are called atypical sequences.*

Using the fact that the information source produces independent X_i that will highly likely be typical sequences with large n, we get

$$p(x_1, \dots, x_n) = p(x_1)p(x_2) \dots p(x_n) \approx p^{np}(1-p)^{(1-p)n}$$

How many bits do we need to represent this sequence? Taking logarithms on both sides, we find that

$$-\log p(x_1, \dots x_n) \approx -np\log p - n(1-p)\log(1-p) \equiv nH(X)$$

Thus, $p(x_1, \dots, x_n) \approx 2^{-nH(X)}$ from which we can say that there are at most $2^{nH(X)}$ typical sequences (as total probability of all typical sequences ≤ 1). Therefore, we can use only $nH(X) \leq n$ bits to identify these typical sequences uniquely. In this sense, we can say that the information content is not n bits but $nH(X)$, and per bit it is $H(X)$.

Definition 10.2.2. *(Entropy rate) Given a random variable X distributed according to the source distribution, $H(X) = -p\log p - (1-p)\log(1-p)$ is called the entropy of the source distribution or the entropy rate of the source.*

In other words, for such an independent and identically distributed information source, in the large n limit, the data from n bits can be compressed to $nH(X)$ bits. One could make this idea more general by defining ϵ-typical strings.

Definition 10.2.3. *Given a $\epsilon > 0$ we say a string is ϵ-typical if*

$$2^{-n(H(X)+\epsilon)} \leq p(x_1, \dots x_n) \leq 2^{-n(H(X)-\epsilon)}$$

$|T(n, \epsilon)|$ denotes the set of all ϵ-typical sequences.

Definition 10.2.4. *(Compression and decompression scheme) A compression scheme, $C^n(x)$, of rate R maps the possible sequences of $x = x_1, \dots, x_n$ to $\lfloor nR \rfloor$ length bit strings. The corresponding decompression scheme, $D^n(x)$, takes the nR length string to n length string. A compression-decompression scheme is called reliable if the probability of $D^n(C^n(x)) = x$ approaches one as n tends to ∞.*

The following theorem shows that $H(X)$ is necessary and sufficient to store the output from the source reliably. Before that, we will see a useful lemma whose proof relies on the law of large numbers.

Lemma 10.2.1. *For a fixed $\epsilon > 0$ and any $\delta > 0$ with sufficiently large n, the probability that the sequence is ϵ-typical is at least $1 - \delta$. When we fix both $\epsilon > 0$ and $\delta > 0$ then*

$$(1-\delta)2^{n(H(X)-\epsilon)} \leq |T(n,\epsilon)| \leq 2^{n(H(X)+\epsilon)}$$

If $S(n)$ is a collection at most 2^{nR} strings of length n, where $R < H(X)$ and n large, then for any $\delta > 0$

$$\sum_{x \in S(n)} p(x) \leq \delta$$

The probability of finding a string from this set goes to zero with large n.

Theorem 10.2.2. *(Shannon's noiseless channel coding theorem) Suppose $X = x_1, \ldots, x_n$ is an independent and identically distributed information source, and $H(X)$ is the entropy rate, then a reliable compression-decompression scheme exists if and only if $R > H(X)$.*

Proof. As noted, a typical sequence is an n-letter message, $X = x_1, \ldots, x_n$, where $x_i \in \mathcal{A}$, and we have an independent distribution of letters with specific probabilities, such that we have np_i times the i^{th} letter on average. The number of such strings can be enumerated as

$$\frac{n!}{\prod_{-=1}^{k}(np_i)!}$$

, which represents the number of distinct strings, having the requisite number of parameters. We can show that this number must approximate

$$\frac{n!}{\prod_{-=1}^{k}(np_i)!} \approx 2^{nH(p_1,\ldots,p_k)}$$

explicitly shown using the Stirling's formula. Thus, the probability of obtaining such a typical sequence is the inverse.

Thereby, we obtain,

$$-\frac{1}{n}\log p(x_1,\ldots,x_k) = -\frac{1}{n}\sum_{i=1}^{n}\log(p(x_i)) \approx H(p_1,\ldots,p_k)$$

where the last (approximate) equality is guaranteed by the law of large numbers. The frequency $\frac{n_j}{n}$ of the letter j in the message is substituted by the a priori probability p_j, such that we obtain the number of times j appears in the message.

The law of large numbers also leads us to, for $\epsilon > 0$, we say a sequence is ϵ-typical, when

$$\left| -\frac{1}{n}\log p(x_1,\ldots,x_n) - H(p_1,\ldots,p_k) \right| < \epsilon$$

as defined earlier. Then, for any $\delta > 0$, the probability that a given sequence is ϵ-typical is larger than $1 - \delta$, for sufficiently large n. Therefore, most of the sequences are ϵ-typical in the limit of large n.

Since there are $2^{nH(X)}$ typical sequences, asymptotically in n, each occurring with a probability $2^{-nH(X)}$, we can identify which one of these sequences actually occurred using $nH(X)$ bits. Thus, asymptotic compression to $H(X)$ bits per letter is optimal. ■

10.3 Von Neumann Entropy

Like how Shannon's entropy measures the information content of classical probability distributions, *Von Neumann entropy* is defined for quantum states.

Definition 10.3.1. *(Von Neumann entropy) Given a quantum state's density matrix ρ, its Von Neumann entropy is defined as*

$$\mathcal{S}(\rho) := -\mathrm{Tr}\{\rho \log \rho\}$$

The above definition is motivated by the fact that it resembles the classical Shannon's entropy when expressed in terms of the eigenvalues of ρ, say λ_i's, as

$$\mathcal{S}(\rho) = -\mathrm{Tr}\{\rho \log \rho\} = -\sum_i \lambda_i \log \lambda_i$$

To compare the entropy of two density matrices, similar to the notion of classical relative intensity, *quantum relative entropy* is defined.

Definition 10.3.2. *(Quantum relative density) Given two density matrices ρ and σ the quantum relative density is defined as*

$$\mathcal{S}(\rho||\sigma) := \mathrm{Tr}\{\rho \log \rho\} - \mathrm{Tr}\{\rho \log \sigma\}$$

Asymmetry in the relative entropy measure

The above defined relative entropy measures, both classical $H(p||q)$ and quantum $\mathcal{S}(\rho||\sigma)$, is asymmetric in p and q (ρ and σ). In some cases, the logarithm diverges. Thus, given two probability distributions (or density matrices), we choose one to be p (ρ) and the other to be q (σ) in such a way that the logarithm term makes sense. The asymmetry in relative entropy arises because it measures the difference in information between two probability distributions, not just the distance between them. In essence, this asymmetry is not a deficiency but a feature, arising from the inherent asymmetry in the mathematical models from which both concepts emerge.

Definition 10.3.3. *(Kernel and Support of density matrix) The vector space spanned by the eigenvectors of the density matrix ρ with eigenvalue zero is called the kernel, and that spanned by the non-zero eigenvectors is called the support.*

If the kernel of σ intersects the support of ρ non-trivially, then relative entropy is $+\infty$.

Theorem 10.3.1. *The quantum relative entropy is non-negative and is zero if and only if the two density matrices are equal.*

Proof. Let the spectral decomposition of the density matrix ρ be $\rho = \sum_i \lambda_i |u_i\rangle\langle u_i|$, where $|u_i\rangle$ are the orthonormal eigenvectors and λ_i are the corresponding non-negative eigenvalues summing to one ($\sum_i \lambda_i = 1$).

We can write the relative entropy as:

$$\mathcal{S}(\rho||\sigma) = \mathrm{Tr}\{\rho(\log\rho - \log\sigma)\}$$

Let's evaluate the two terms separately in the eigenbasis of ρ:

$$\mathrm{Tr}\{\rho\log\rho\} = \mathrm{Tr}\left\{\left(\sum_i \lambda_i|u_i\rangle\langle u_i|\right)\left(\sum_j (\log\lambda_j)|u_j\rangle\langle u_j|\right)\right\} = \sum_i \lambda_i \log\lambda_i$$

$$\mathrm{Tr}\{\rho\log\sigma\} = \mathrm{Tr}\left\{\left(\sum_i \lambda_i|u_i\rangle\langle u_i|\right)\log\sigma\right\} = \sum_i \lambda_i\langle u_i|(\log\sigma)|u_i\rangle$$

Now, let the spectral decomposition of σ be $\sigma = \sum_j \mu_j|v_j\rangle\langle v_j|$. Then $\log\sigma = \sum_j(\log\mu_j)|v_j\rangle\langle v_j|$. Substituting this into the expression for $\mathrm{Tr}\{\rho\log\sigma\}$:

$$\langle u_i|(\log\sigma)|u_i\rangle = \sum_j\langle u_i|(\log\mu_j)|v_j\rangle\langle v_j||u_i\rangle = \sum_j(\log\mu_j)|\langle u_i|v_j\rangle|^2$$

Let's define $\mathcal{P}_{ij} = |\langle u_i|v_j\rangle|^2$. Note that for any fixed i, $\sum_j \mathcal{P}_{ij} = \sum_j\langle u_i|v_j\rangle\langle v_j|u_i\rangle = \langle u_i|(\sum_j |v_j\rangle\langle v_j|)|u_i\rangle = \langle u_i|I|u_i\rangle = 1$.

The relative entropy is then:

$$\mathcal{S}(\rho||\sigma) = \sum_i \lambda_i\log\lambda_i - \sum_i\lambda_i\left(\sum_j \mathcal{P}_{ij}\log\mu_j\right)$$

The function $\log(x)$ is strictly concave. By Jensen's inequality, for each i:

$$\sum_j \mathcal{P}_{ij}\log\mu_j \leq \log\left(\sum_j \mathcal{P}_{ij}\mu_j\right)$$

Let's define a probability distribution $q_i = \sum_j \mathcal{P}_{ij}\mu_j = \langle u_i|\sigma|u_i\rangle$. The set $\{q_i\}$ forms a probability distribution since $\sum_i q_i = \sum_i\langle u_i|\sigma|u_i\rangle = \mathrm{Tr}(\sigma) = 1$.

Substituting this back into the expression for relative entropy, we get a lower bound:

$$\mathcal{S}(\rho||\sigma) \geq \sum_i \lambda_i\log\lambda_i - \sum_i\lambda_i\log q_i = \sum_i\lambda_i\log\left(\frac{\lambda_i}{q_i}\right)$$

This final expression is exactly the classical relative entropy (or Kullback-Leibler divergence) $H(\lambda||q)$ between the probability distribution of eigenvalues of ρ, $\{\lambda_i\}$, and the distribution of the diagonal elements of σ in the eigenbasis of ρ, $\{q_i\}$. As established in the chapter, the classical relative entropy is non-negative, $H(\lambda||q) \geq 0$.

For the equality $\mathcal{S}(\rho||\sigma) = 0$ to hold, two conditions must be met

1. The inequality $H(\lambda||q) \geq 0$ must be an equality. This happens if and only if $\lambda_i = q_i$ for all i. So, $\lambda_i = \langle u_i|\sigma|u_i\rangle$.
2. The Jensen's inequality for the concave log function must be an equality for every i. This occurs if and only if for each i, all the values of μ_j for which $\mathcal{P}_{ij} = |\langle u_i|v_j\rangle|^2 > 0$ are identical.

The second condition implies that for any given eigenvector $|u_i\rangle$ of ρ, all eigenvectors $|v_j\rangle$ of σ that it has a non-zero projection on must share the same eigenvalue. This is only possible if each $|u_i\rangle$ is also an eigenvector of σ. Since $\{|u_i\rangle\}$ forms a basis, this means that ρ and σ must commute and are thus simultaneously diagonalizable.

If they share the same set of eigenvectors, let this basis be $\{|k\rangle\}$. Then $\rho = \sum_k \lambda_k |k\rangle\langle k|$ and $\sigma = \sum_k \mu_k |k\rangle\langle k|$. In this case, the quantum relative entropy simplifies to the classical relative entropy of their eigenvalues

$$\mathcal{S}(\rho||\sigma) = \sum_k \lambda_k \log\left(\frac{\lambda_k}{\mu_k}\right)$$

This is zero if and only if $\lambda_k = \mu_k$ for all k. Since they have the same eigenvalues and the same corresponding eigenvectors, the density matrices must be identical, $\rho = \sigma$. ■

Theorem 10.3.2. *The following are some properties of $\mathcal{S}(\rho)$:*

1. *$\mathcal{S}(\rho)$ is non-negative. It is zero if and only if ρ is pure state.*
2. *In a d-dimentional Hilbert space the entropy is at most $\log d$. It is equal to $\log d$ if and only if the state is a completely mixed state.*
3. *Suppose a composite system $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$ is in pure state then $\mathcal{S}(A) = \mathcal{S}(B)$.*
4. *Suppose p_i are probabilities of the state being in ρ_i then*

$$\mathcal{S}\left(\sum_i p_i\rho_i\right) = H(p_i) + \sum_i p_i \mathcal{S}(\rho_i)$$

Proof. 1. The von Neumann entropy is defined in terms of the eigenvalues $\{\lambda_i\}$ of the density matrix ρ as $\mathcal{S}(\rho) = -\sum_i \lambda_i \log \lambda_i$. For a density matrix, the eigenvalues satisfy $0 \leq \lambda_i \leq 1$. For any λ_i in this range, $\log \lambda_i \leq 0$. Thus, each term $-\lambda_i \log \lambda_i$ is non-negative. The sum of non-negative terms is also non-negative, so $\mathcal{S}(\rho) \geq 0$.

The equality $\mathcal{S}(\rho) = 0$ holds if and only if every term in the sum is zero. A term $-\lambda_i \log \lambda_i$ is zero if $\lambda_i = 0$ or $\lambda_i = 1$. Since the eigenvalues must sum to one ($\sum_i \lambda_i = 1$), it must be that exactly one eigenvalue is 1 and all others are 0. A density matrix with this eigenvalue distribution describes a pure state, $\rho = |\psi\rangle\langle\psi|$. Conversely, if ρ is a pure state, its eigenvalues are $\{1, 0, \ldots, 0\}$, and its entropy is $\mathcal{S}(\rho) = -1 \log 1 - \sum 0 \log 0 = 0$.

2. We want to maximize $\mathcal{S}(\rho) = -\sum_{i=1}^{d} \lambda_i \log \lambda_i$ in a d-dimensional space. We can use the non-negativity of the relative entropy. Let ρ be any state and let $\sigma = \frac{1}{d}I$ be the completely mixed state. From Klein's inequality, $\mathcal{S}(\rho||\sigma) \geq 0$.

$$\mathrm{Tr}\{\rho \log \rho\} - \mathrm{Tr}\{\rho \log \sigma\} \geq 0$$

$$-\mathcal{S}(\rho) - \mathrm{Tr}\left\{\rho \log\left(\frac{1}{d}I\right)\right\} \geq 0$$

$$-\mathcal{S}(\rho) - \mathrm{Tr}\{\rho(\log(1/d))I\} \geq 0$$

$$-\mathcal{S}(\rho) - (\log(1/d))\mathrm{Tr}\{\rho\} \geq 0$$

Since $\mathrm{Tr}\{\rho\} = 1$ and $\log(1/d) = -\log d$:

$$-\mathcal{S}(\rho) + \log d \geq 0 \implies \mathcal{S}(\rho) \leq \log d$$

The equality holds if and only if $\mathcal{S}(\rho||\sigma) = 0$, which implies $\rho = \sigma$. Therefore, the entropy is maximal and equal to $\log d$ if and only if the state is the completely mixed state, $\rho = \frac{1}{d}I$.

3. This property is a direct consequence of the Schmidt decomposition. Any pure state $|\Psi\rangle$ of a composite system $\mathcal{H}_A \otimes \mathcal{H}_B$ can be written as:

$$|\Psi\rangle = \sum_i \sqrt{\lambda_i} |u_i\rangle_A \otimes |v_i\rangle_B$$

where $\{|u_i\rangle_A\}$ and $\{|v_i\rangle_B\}$ are orthonormal sets in $\mathcal{H}_A$ and $\mathcal{H}_B$ respectively, and $\lambda_i > 0$ with $\sum_i \lambda_i = 1$.

The reduced density matrix for subsystem A is $\rho_A = \mathrm{Tr}_B(|\Psi\rangle\langle\Psi|)$.

$$\rho_A = \mathrm{Tr}_B\left(\sum_{i,j} \sqrt{\lambda_i \lambda_j} |u_i\rangle_A \langle u_j|_A \otimes |v_i\rangle_B \langle v_j|_B\right)$$

$$\rho_A = \sum_{i,j} \sqrt{\lambda_i \lambda_j} |u_i\rangle_A \langle u_j|_A \mathrm{Tr}(|v_i\rangle_B \langle v_j|_B)$$

Since $\mathrm{Tr}(|v_i\rangle\langle v_j|) = \langle v_j|v_i\rangle = \delta_{ij}$, we get:

$$\rho_A = \sum_i \lambda_i |u_i\rangle_A \langle u_i|_A$$

The non-zero eigenvalues of ρ_A are precisely the coefficients $\{\lambda_i\}$. The entropy is $\mathcal{S}(A) = -\sum_i \lambda_i \log \lambda_i$.

Similarly, the reduced density matrix for subsystem B is $\rho_B = \mathrm{Tr}_A(|\Psi\rangle\langle\Psi|)$.

$$\rho_B = \sum_i \lambda_i |v_i\rangle_B \langle v_i|_B$$

The non-zero eigenvalues of ρ_B are also $\{\lambda_i\}$. The entropy is $\mathcal{S}(B) = -\sum_i \lambda_i \log \lambda_i$. Thus, $\mathcal{S}(A) = \mathcal{S}(B)$.

4. The equality $\mathcal{S}(\sum_i p_i\rho_i) = H(p_i) + \sum_i p_i \mathcal{S}(\rho_i)$ holds under the specific condition that the density matrices ρ_i have orthogonal support. This means that the vector spaces on which each ρ_i acts non-trivially are mutually orthogonal.

   Let this condition hold. We can choose a basis for the total Hilbert space that respects this block structure. In this basis, the total density matrix $\rho = \sum_i p_i\rho_i$ is block-diagonal:

$$\rho = \begin{pmatrix} p_1\rho_1 & 0 & \cdots \\ 0 & p_2\rho_2 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

   The set of eigenvalues of ρ is the union of the sets of eigenvalues of each block $p_i\rho_i$. If $\{\lambda_{ij}\}_j$ are the eigenvalues of ρ_i, then $\{p_i\lambda_{ij}\}_j$ are the eigenvalues of the i-th block.

   The von Neumann entropy of ρ is the sum over all its eigenvalues:

$$\mathcal{S}(\rho) = -\sum_{i,j} (p_i\lambda_{ij}) \log(p_i\lambda_{ij})$$

   Using the property of logarithms, $\log(ab) = \log a + \log b$:

$$\mathcal{S}(\rho) = -\sum_{i,j} p_i\lambda_{ij} (\log p_i + \log \lambda_{ij})$$

$$\mathcal{S}(\rho) = -\sum_{i,j} p_i\lambda_{ij} \log p_i - \sum_{i,j} p_i\lambda_{ij} \log \lambda_{ij}$$

   We can split this into two parts. For the first part:

$$-\sum_{i,j} p_i\lambda_{ij} \log p_i = -\sum_i (p_i \log p_i) \left(\sum_j \lambda_{ij}\right)$$

   Since $\sum_j \lambda_{ij} = \mathrm{Tr}(\rho_i) = 1$, this simplifies to:

$$-\sum_i p_i \log p_i = H(p_i)$$

For the second part:

$$-\sum_{i,j} p_i \lambda_{ij} \log \lambda_{ij} = \sum_i p_i \left(-\sum_j \lambda_{ij} \log \lambda_{ij} \right)$$

The term in the parenthesis is the entropy of ρ_i, $\mathcal{S}(\rho_i)$. So this part becomes:

$$\sum_i p_i \mathcal{S}(\rho_i)$$

Combining the two parts gives the desired result:

$$\mathcal{S}\left(\sum_i p_i \rho_i \right) = H(p_i) + \sum_i p_i \mathcal{S}(\rho_i)$$

■

10.4 Quantum Data Compression

As a natural extension to Shnanon's noiseless coding theorem, we have the quantum analogue presented below. For a message transmission of n letters, each letter being chosen at random from the alphabet $\mathcal{A}$, which here is an ensemble of pure states, defined by

$$\mathcal{A} = \{|\psi_1\rangle, |\psi_2\rangle, \ldots |\psi_k\rangle\}$$

The state $|\psi_i\rangle$ is extracted *a priori* with probability p_i, such that $\sum\limits_{i=1}^{k} p_i = 1$. Thereby, for each letter in the message, we have the density matrix

$$\rho = \sum_{i=1}^{k} p_i |\psi_i\rangle\langle\psi_i|$$

thereby, for the entire message, we have the tensor product,

$$\rho_{(n)} = \rho^{\otimes n}$$

Here we have assumed that all the letters in the message are statistically independent and described by the same density matrix ρ.

Schumacher's theorem, similar to Shannon's coding theorem, entails us the machinery to encode the message, in the sense that we can compress the data, with the optimal compression rate directed by the von Neumann entropy.

Theorem 10.4.1. *(Schumacher's's quantum noiseless coding theorem) Suppose we have a message whose letters are drawn independently from the ensemble $\mathcal{A} = \{|\psi_1\rangle, \ldots, |\psi_k\rangle\}$ with prior probabilities $\{p_1, \ldots . p_k\}$, there exists an optimal and reliable code compressing the message to $\mathcal{S}(\rho)$ qubits per letter where $\rho = \sum_{i=1}^{k} p_i |\psi_i\rangle\langle\psi_i|$, asymptotically in the length of the message.*

Proof. The proof of this theorem closely resembles the techniques used in the proof of Shannon's noiseless coding theorem described earlier[1]. We illustrate the idea here, by spectrally decomposing the density operator ρ as

$$\rho = \sum_{i=1}^{k} \lambda_i |a_i\rangle\langle a_i|,$$

Further, we have the von Neuman entorpy relating the classical optimal compression rate, as

$$H(\lambda_1, \dots, \lambda_k) = -\sum_i \lambda_i \log \lambda_i = -\mathrm{Tr}\rho \log \rho = \mathcal{S}(\rho)$$

The ensemble $\tilde{\mathcal{A}}$ defined from the spectral decomposition states $\{|a_1\rangle, \dots, |a_k\rangle\}$ constitutes an alphabet of orthogonal pure quantum states.

We rework the definition of a ϵ-typical sequence, for a state $|x_1\rangle \otimes \cdots \otimes |x_n\rangle$, with $|x_i\rangle \in \tilde{\mathcal{A}}$ is ϵ-typical, when

$$\left| -\frac{1}{n} \log[\lambda(x_1) \cdots \lambda(x_n)] - \mathcal{S}(\rho) \right| < \epsilon$$

where $\lambda(x_i) = \lambda_j$ if $|x_i\rangle$ is in the letter $|a_j\rangle$. We define the ϵ-typical subspace as the subspace spanned by the ϵ-typical states.

As before, the dimension of the subspace can be shown to be of the order of $2^{n\mathcal{S}(\rho)}$. For any projector Π_{typical} on this typical subspace, we have

$$\mathrm{Tr}\{\Pi_{\text{typical}} \rho^n\} > 1 - \delta$$

provided asymptotically large n, as we proved for the compression scheme in the classical case. Therefore, as $n \to \infty$, the density matrix $\rho_{(n)}$ has its support on a typical subspace of dimension $2^{n\mathcal{S}(\rho)}$. A typical n state message can then be encoded using $n\mathcal{S}(\rho)$ qubits, thus constraining the optimal rate by the von Neumann entropy. ■

Compression of an n qubit message

Consider the binary alphabet $\mathcal{A} = \{|\psi_0\rangle, |\psi_1\rangle\}$, where $|\psi_0\rangle = \cos\theta|0\rangle + \sin\theta|1\rangle$, and $|\psi_1\rangle = \sin\theta|0\rangle + \cos\theta|1\rangle$, which are not necessarily orthogonal.

Say we want to transfer the n qubit message,

$$|\Psi_K\rangle = |\psi_{k_1}\rangle \otimes |\psi_{k_2}\rangle \otimes \cdots \otimes |\psi_{k_n}\rangle$$

where $K = \{k_1, \dots, k_n\}$ singles out the message, which each k_i being either 0 or 1.

The states $|\psi_0\rangle$ and $|\psi_1\rangle$ are drawn from the alphabet $\mathcal{A}$ with probabilities p and $1-p$ respectively. Any n letter message $|\Psi_K\rangle$ is in the combined Hilbert space $\mathcal{H}^{\otimes n}$, for the Hilbert space $\mathcal{H}$ of a single qubit.

[1] Interested readers can refer to Quantum Coding, Schumacher.

We decompose the message into the typical subspace through a projector as used in the proof, such that we can express

$$|\Psi_K\rangle = \alpha_K|\tau_K\rangle + \beta_K|\tau_K^{\perp}\rangle$$

where we say $|\tau_K\rangle$ to belong to the typical subspace $\mathcal{H}_{\text{typical}}$, and $|\tau_K^{\perp}\rangle$ belongs to the orthogonal complement space.

For a measurement to determine whether $|\Psi_K\rangle$ belongs to the typical subspace, such that the message is encoded, we realise that we need only $n\mathcal{S}(\rho)$ wubits for encoding, since the typical subspace has dimension $\approx 2^n\mathcal{S}(\rho)$. If instead $|\Psi_K\rangle$ belongs to the atypical subspace (given by the orthogonal complement), we substitute it with some reference state $|\mathcal{R}\rangle$ residing in the typical subspace.

On decoding the $n\mathcal{S}(\rho)$ qubits, we have the effective density matrix, given by

$$\tilde{\rho}_K = |\alpha_K|^2|\tau_K\rangle\langle\tau_K| + |\beta_K|^2|\mathcal{R}\rangle\langle\mathcal{R}|$$

As evidently seen, there is some notion of loss of information through the reference state. We can compute the effective reliability of compression through a physical quantity, termed the *fidelity* $\mathcal{F}$, given by

$$\mathcal{F} = \langle\Psi_K|\tilde{\rho}_K|\Psi_K\rangle$$

where we can clearly see that, if optimal compression, $\tilde{\rho} = |\Psi_K\rangle\langle\Psi_K|$, and we have $\mathcal{F} = 1$. If we have orthogonal initial and final states, then the fidelity vanishes, $\mathcal{F} = 0$.

We obtain the average fidelity $\bar{\mathcal{F}}$ by weighting over the probability of occurrence of the possible messages, such that, we have

$$\begin{aligned}
\bar{\mathcal{F}} &= \sum_K p_K\langle\Psi_K|\tilde{\rho}_K|\Psi_K\rangle \\
&= \sum_K p_K\langle\Psi_K|\left(\alpha_K|^2|\tau_K\rangle\langle\tau_K| + |\beta_K|^2|\mathcal{R}\rangle\langle\mathcal{R}|\right)|\Psi_K\rangle \\
&= \sum_K p_K|\alpha_K|^4 + \sum_K |\beta_K|^2\left(|\langle\Psi_K|\mathcal{R}\rangle|^2\right)
\end{aligned}$$

Thereby, we have average fidelity tending close to 1 as $n \to \infty$, such that messages overlap with the typical subspace. Hence, we can code only the typical subspace and still achieve good fidelity.

Chapter 11

Exploiting Quantum Entanglement

> *"Feeling insignificant because the universe is large has exactly the same logic as feeling inadequate for not being a cow.Or a herd of cows. The universe is not there to overwhelm us; it is our home, and our resource. The bigger the better."*
>
> – David Deutsch, *The Beginning of Infinity*

11.1 Introduction

The superposition principle illustrates the existence of entangled states in two or more quantum systems. These entangled states are characterised by cross-correlations between the systems, which any classical theory cannot satisfactorily explain. Such phenomena have played a central role in the development of quantum theory, beginning with the famous paradox posed by Einstein, Podolsky, and Rosen (EPR) and followed by the fascinating work of John Stewart Bell. This paradox exemplifies the seemingly absurd implications of entanglement when applied to the macroscopic world. The EPR dilemma challenges classical reasoning by presenting a conflict between the reality of physical properties and the locality implied by the finite speed of light. This challenge, along with subsequent developments, has refined our understanding of entanglement. In the field of quantum information, entanglement is considered a valuable resource to be utilised.

11.2 Local Operations Classical Communication

Say we play a game of quantum state exchange, starting with an entangled pure state $|\psi\rangle$ between us. Suppose we perform arbitrary operations on our local systems and can only communicate using classical communication channels. This exploration closely links with ideas of entanglement and a measure to quantify it through the different possible entanglement states $|\varphi\rangle$ it can transform into. These types of operations with intrinsic richness in the class of transformations correspond to the class of *local operations and classical communication* (LOCC), which help us disentangle the ideas of bipartite quantum entanglement.

Quantum Teleportation

Quantum teleportation[a] is an important task that can be completed by LOCC. Following the convention, this process requires 2 communication nodes or parties, namely A (Alice) and B (Bob). For simplicity, we only consider transferring a single-qubit quantum state $|\psi\rangle_C$ and this requires 3 qubits in total including the pre-shared maximally entangled state $|\Phi^+\rangle_{AB} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Alice holds systems A and C, Bob holds system B. Note that only quantum information is transferred, not the physical qubits. The workflow proceeds in the following steps:

1. At the very beginning, the system state can be described as $|\varphi_0\rangle = |\psi\rangle_C \otimes |\Phi^+\rangle_{AB} = \frac{1}{\sqrt{2}}\big[\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|00\rangle + |11\rangle)\big]$ where the quantum state Alice want to transmit is $|\psi\rangle_C = \alpha|0\rangle_C + \beta|1\rangle_C$ and the coefficients $\alpha, \beta \in \mathbb{C}$.

2. Alice applies a CNOT gate, and the resulting state $|\varphi_1\rangle = \frac{1}{\sqrt{2}}\big[\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|10\rangle + |01\rangle)\big]$

3. Alice applies a Hadamard gate, and the system state becomes $|\varphi_2\rangle = \frac{1}{2}\big[\alpha(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + \beta(|0\rangle - |1\rangle)(|10\rangle + |01\rangle)\big]$. The above state can be rearranged to $|\varphi_2\rangle = \frac{1}{2}\big[|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)\big]$.

4. Alice measures both of her qubits in the computational basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ and send the results m_1m_2 to Bob with a classical channel. There are 4 distinct possibilities: $m_1m_2 \in \{00, 01, 10, 11\}$. Then, Bob implements certain operations correspondingly on his qubit based on the received messages.

   - If the measurement result is $m_1m_2 = 00$, Bob's state will be $\alpha|0\rangle + \beta|1\rangle$, which is the state Alice want to transmit $|\psi\rangle_C$. No operations are needed and the teleportation is finished.
   - If the measurement result is $m_1m_2 = 01$, Bob's state will be $\alpha|1\rangle + \beta|0\rangle$. Bob needs to act the X gate on his qubit.
   - If the measurement result is $m_1m_2 = 10$, Bob's state will be $\alpha|0\rangle - \beta|1\rangle$. Bob needs to act the Z gate on his qubit.
   - If the measurement result is $m_1m_2 = 11$, Bob's state will be $\alpha|1\rangle - \beta|0\rangle$. Bob needs to act the X gate followed by the Z gate on his qubit.

[a] Refer to the original paper by Bennett, Charles H., et al. on "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels." Physical Review Letters 70.13 (1993): 1895.

In short, LOCC transfers quantum information between two spatially separated communication nodes (only a classical communication channel is allowed) with the help of entanglement. At the heart of entanglement theory is the notion of LOCC, since global quantum operations are unfeasible in regions separated physically.

We formalise the notion of LOCCs through the following theorem.

Theorem 11.2.1. *Let the state $|\varphi\rangle$ be transformed to $|\psi\rangle$ through the virtue of local operations and classical communication. This transformation expects a series of generalised measurement operators $\{M_i^A\}$ in virtue of A, transferring the measurement to B, who can transform the state by a pre-assigned unitary U_i to respect the change.*

Proof. Say that B performs a measurement with generalised measurement operators M_j^B on a pure state $|\varphi\rangle$. Let this state be Schmidt-decomposed as

$$|\varphi\rangle = \sum_{l=1}^{\mathcal{R}} \sqrt{\lambda_l}|l_A\rangle|l_B\rangle$$

with the Schmidt decompsition with $\mathrm{rank}(\rho) = \mathcal{R}$. In this basis, we can define

$$M_j^B = \sum_{l}^{\mathcal{R}}\sum_{k}^{\mathcal{R}} M_{j,kl}|k_B\rangle\langle l_B|$$

and we denote the specialized operator for A which is the same as the matrix representation with respect to A's Schmidt basis as

$$M_j^A \equiv \sum_{l}^{\mathcal{R}}\sum_{k}^{\mathcal{R}} M_{j,kl}|k_A\rangle\langle l_A|.$$

Now let B perform the measurement defined by these operators M_j^B, with post measurement state defined by

$$\begin{aligned}
|\psi_j^B\rangle \propto M_j^B|\varphi\rangle &= \sum_{l}^{\mathcal{R}}\sum_{k}^{\mathcal{R}} M_{j,kl}|k_B\rangle\langle l_B|\varphi\rangle \\
&= \sum_{l}^{\mathcal{R}}\sum_{k}^{\mathcal{R}} M_{j,kl}|k_B\rangle\langle l_B| \left(\sum_{l'=1}^{\mathcal{R}} \sqrt{\lambda_{l'_A}}|l'\rangle|l'_B\rangle\right) \\
&= \sum_{l}^{\mathcal{R}}\sum_{k}^{\mathcal{R}} M_{j,kl}\sqrt{\lambda_l}|l_A\rangle|k_B\rangle.
\end{aligned}$$

The probability of measurement is given by the norm as

$$\begin{aligned}
p^B(j) = \left|\left|M_j^B|\varphi\rangle\right|\right|^2 &= \left(\sum_{l'}^{\mathcal{R}}\sum_{k'}^{\mathcal{R}} M^*_{j,k'l'}\sqrt{\lambda_{l'}}\langle l'_A|\langle k'_B|\right)\left(\sum_{l}^{\mathcal{R}}\sum_{k}^{\mathcal{R}} M_{j,kl}\sqrt{\lambda_l}|l_A\rangle|k_B\rangle\right) \\
&= \sum_{l}^{\mathcal{R}}\sum_{k}^{\mathcal{R}} |M_{j,kl}|^2\lambda_l,
\end{aligned}$$

since we have the states $|l_A\rangle|k_B\rangle$ orthonormal in the Schmidt basis. Similarly, for A, we have the post-measurement state given by

$$\begin{aligned}
|\psi_j^A\rangle \propto M_j^A|\varphi\rangle &= \sum_{l}^{\mathcal{R}}\sum_{k}^{\mathcal{R}} M_{j,kl}|k_A\rangle\langle l_A|\varphi\rangle \\
&= \sum_{l}^{\mathcal{R}}\sum_{k}^{\mathcal{R}} M_{j,kl}|k_A\rangle\langle l_A| \left(\sum_{l'=1}^{\mathcal{R}} \sqrt{\lambda_{l'_A}}|l'\rangle|l'_B\rangle\right) \\
&= \sum_{l}^{\mathcal{R}}\sum_{k}^{\mathcal{R}} M_{j,kl}\sqrt{\lambda_l}|k_A\rangle|l_B\rangle.
\end{aligned}$$

The probability of measurement is given by the norm as

$$\begin{aligned}
p^A(j) = \left|\left|M_j^A|\varphi\rangle\right|\right|^2 &= \left(\sum_{l'}^{\mathcal{R}}\sum_{k'}^{\mathcal{R}} M^*_{j,k'l'}\sqrt{\lambda_{l'}}\langle k'_A|\langle l'_B|\right)\left(\sum_{l}^{\mathcal{R}}\sum_{k}^{\mathcal{R}} M_{j,kl}\sqrt{\lambda_l}|k_A\rangle|l_B\rangle\right) \\
&= \sum_{l}^{\mathcal{R}}\sum_{k}^{\mathcal{R}} |M_{j,kl}|^2\lambda_l,
\end{aligned}$$

since we have the states $|k_A\rangle|l_B\rangle$ orthonormal in the Schmidt basis.

Thereby, the probabilities are inherently equivalent $p^A(j) = p^B(j)$ and we have the states $|\psi_j^A\rangle$ and $|\psi_j^B\rangle$ are related by an unitary transformation admitting the change of basis from $|l_A\rangle|k_B\rangle$ to $|k_A\rangle|l_B\rangle$ as

$$\begin{aligned}
\psi_j^B &= (U_j^A \otimes V_j^B)\psi_j^A \\
&= (U_j^A \otimes V_j^B)\sum_{l}^{\mathcal{R}}\sum_{k}^{\mathcal{R}} M_{j,kl}\sqrt{\lambda_l}|k_A\rangle|l_B\rangle \\
&= \sum_{l}^{\mathcal{R}}\sum_{k}^{\mathcal{R}} M_{j,kl}\sqrt{\lambda_l}(U_j^A|k_A\rangle)(V_j^B|l_B\rangle),
\end{aligned}$$

such that $U_j^A|k_A\rangle = |l_A\rangle$ and $V_j^B|k_B\rangle = |k_B\rangle$.
Therefore, B performing a measurement described by measurement operators M_j is equivalent to A performing the measurement described by measurement operators $U_j^A M_j^A$ followed by B performing the unitary transformation V_j^B. In summarising, a measurement by B on a known pure state can be simulated by a measurement by A, up to a unitary transformation by B. ■

Further, we note the resulting post-measurement density matrix due to B's measurement given by

$$
\begin{aligned}
\rho' = \frac{M_j^B \rho M_j^{B\dagger}}{\mathrm{Tr}(\rho M_j^{B\dagger} M_j^B)} &\propto M_j^B |\varphi\rangle\langle\varphi| M_j^{B\dagger} \\
&= \left(\sum_{l}^{\mathcal{R}} \sum_{k}^{\mathcal{R}} M_{j,kl} \sqrt{\lambda_l} |l_A\rangle |k_B\rangle \right) \left(\sum_{l'}^{\mathcal{R}} \sum_{k'}^{\mathcal{R}} M^*_{j,k'l'} \sqrt{\lambda_{l'}} \langle l'_A| \langle k'_B| \right) \\
&= \sum_{l}^{\mathcal{R}} \sum_{k}^{\mathcal{R}} \sum_{l'}^{\mathcal{R}} \sum_{k'}^{\mathcal{R}} M_{j,kl} M^*_{j,k'l'} \sqrt{\lambda_l \lambda_{l'}} \; |l_A\rangle\langle l'_A| \; |k_B\rangle\langle k'_B|,
\end{aligned}
$$

which implies, the reduced density matrices are

$$
\begin{aligned}
\rho_j'^A &= \mathrm{Tr}_B(\rho'_j) = \sum_{l}^{\mathcal{R}} \sum_{l'}^{\mathcal{R}} \sum_{k}^{\mathcal{R}} M_{j,kl} M^*_{j,kl'} \sqrt{\lambda_l \lambda_{l'}} |l_A\rangle\langle l'_A|, \\
\rho_j'^B &= \mathrm{Tr}_A(\rho'_j) = \sum_{l}^{\mathcal{R}} \sum_{k}^{\mathcal{R}} \sum_{k'}^{\mathcal{R}} M_{j,kl} M^*_{j,k'l} \lambda_l |k_B\rangle\langle k'_B|.
\end{aligned}
$$

Further, due to A's measurements, we have

$$
\begin{aligned}
\rho''_j = \frac{M_j^A \rho M_j^{A\dagger}}{\mathrm{Tr}(\rho M_j^{A\dagger} M_j^A)} &\propto M_j^A |\varphi\rangle\langle\varphi| M_j^{A\dagger} \\
&= \sum_{l}^{\mathcal{R}} \sum_{k}^{\mathcal{R}} \sum_{l'}^{\mathcal{R}} \sum_{k'}^{\mathcal{R}} M_{j,kl} M^*_{j,k'l'} \sqrt{\lambda_l \lambda_{l'}} \; |k_A\rangle\langle k'_A| \; |l_B\rangle\langle l'_B|,
\end{aligned}
$$

which implies, the reduced density matrices are

$$
\begin{aligned}
\rho_j''^A &= \mathrm{Tr}_B(\rho''_j) = \sum_{l}^{\mathcal{R}} \sum_{k}^{\mathcal{R}} \sum_{k'}^{\mathcal{R}} M_{j,kl} M^*_{j,k'l} \lambda_l |k_A\rangle\langle k'_A|, \\
\rho_j''^B &= \mathrm{Tr}_A(\rho''_j) = \sum_{l}^{\mathcal{R}} \sum_{l'}^{\mathcal{R}} \sum_{k}^{\mathcal{R}} M_{j,kl} M^*_{j,kl'} \sqrt{\lambda_l \lambda_{l'}} |l_B\rangle\langle l'_B|.
\end{aligned}
$$

We note that ρ''_j and ρ''_j can be related by the change of basis matrices as before by the transformation

$$
\begin{aligned}
\rho'_j &= (U_j^A \otimes V_j^B) \rho''_j (U_j^{A\dagger} \otimes V_j^{B\dagger}) \\
&= (U_j^A \otimes V_j^B) \sum_{l}^{\mathcal{R}} \sum_{k}^{\mathcal{R}} \sum_{l'}^{\mathcal{R}} \sum_{k'}^{\mathcal{R}} M_{j,kl} M^*_{j,k'l'} \sqrt{\lambda_l \lambda_{l'}} \; |k_A\rangle\langle k'_A| \; |l_B\rangle\langle l'_B| (U_j^{A\dagger} \otimes V_j^{B\dagger}) \\
&= \sum_{l}^{\mathcal{R}} \sum_{k}^{\mathcal{R}} \sum_{l'}^{\mathcal{R}} \sum_{k'}^{\mathcal{R}} M_{j,kl} M^*_{j,k'l'} \sqrt{\lambda_l \lambda_{l'}} \; U_j^A |k_A\rangle\langle k'_A| U_j^{A\dagger} \; V_j^B |l_B\rangle\langle l'_B| V_j^{B\dagger}.
\end{aligned}
$$

such that $U_j^A|k_A\rangle = |l_A\rangle$ and $V_j^B|k_B\rangle = |k_B\rangle$. Note that the same does not hold for $\rho_j'^A$ and $\rho_j'^B$ defined by the partial trace. In fact, we have $\rho_j'^A = \rho_j''^B$ and $\rho_j'^B = \rho_j''^A$. Further, we note

$$U_j^A \rho_j''^A U_j^{A\dagger} = \sum_l^{\mathcal{R}} \sum_k^{\mathcal{R}} \sum_{k'}^{\mathcal{R}} M_{j,kl} M^*_{j,k'l} \lambda_l U_j^A |k_A\rangle\langle k'_A| U_j^{A\dagger} \neq \rho_j'^A$$

$$V_j^B \rho_j''^B V_j^{B\dagger} = \sum_l^{\mathcal{R}} \sum_{l'}^{\mathcal{R}} \sum_k^{\mathcal{R}} M_{j,kl} M^*_{j,kl'} \sqrt{\lambda_l \lambda_{l'}} V_j^B |l_B\rangle\langle l'_B| V_j^{B\dagger} \neq \rho_j'^B.$$

11.3 Majorization

Majorization is a purely mathematical concept with surprisingly far-reaching applications. Consider two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, where we define a sorted (in non-ascending manner) version of a vector $\boldsymbol{a}$ as $\boldsymbol{a}^{\downarrow}$, such that

$$a_1^{\downarrow} \geq a_2^{\downarrow} \geq \cdots \geq a_n^{\downarrow}$$

Note that, the sorted vector is a permutation of the elements of $\boldsymbol{v}$, hence we can relate the entries of the descending vector through a permutation matrix $\mathcal{P} \in S_n$, such that

$$\boldsymbol{v}^{\downarrow} = \mathcal{P}\boldsymbol{v}, \quad \mathcal{P} \in S_n$$

We define $\boldsymbol{x}$ *majorizes* $\boldsymbol{y}$, written as $\boldsymbol{x} \succ \boldsymbol{y}$, if

$$\boldsymbol{x} \succ \boldsymbol{y} \implies \sum_{j=1}^{k} x_j^{\downarrow} = \sum_{j=1}^{k} y_j^{\downarrow} \quad \forall\ 1 \leq k \leq n$$

The central insight into majorization theory relies on the idea that

$$\boldsymbol{x} \succ \boldsymbol{y} \quad \iff \quad \boldsymbol{y} = \sum_j p_j \mathcal{P}_j \boldsymbol{x}$$

for a probability distribution p_j over the permutation matrices $\mathcal{P}_j$. This can be understood through the inductive reasoning that, for $\boldsymbol{x} \succ \boldsymbol{y}$, the biggest element of $\boldsymbol{x}^{\downarrow}$ must exceed the last element of $\boldsymbol{y}^{\downarrow}$ and the difference of their sums, such that a convex combination of $\boldsymbol{x}$'s are obtained for $\boldsymbol{y}$. Thus, $\boldsymbol{x} \succ \boldsymbol{y}$ if and only if $\boldsymbol{y}$ can be written as a convex combination of permutations of $\boldsymbol{x}$, resulting in a more disordered sense and intermixing the elements of the vectors.

These matrices, written as a convex combination of permutation matrices, give rise to rich physical insight. The entries of these matrices are non-negative, and the sums of columns and rows are identity. Through the implications of Birkhoff's phenomenal theorem[1], we can rewrite

$$\boldsymbol{y} = \mathcal{D}\boldsymbol{x}$$

[1] The proof of this theorem involves beautiful implications of graph theory and mapping doubly stochastic matrices to an associated graph, further using Hall's marriage theorem. Refer `https://webpages.charlotte.edu/~ghetyei/courses/old/F07.3116/birkhofft.pdf`

where $\mathcal{D}$ is doubly-stochastic, which has all columns and rows as simultaneously probability distributions, that is

$$\mathcal{D}_{ij} \geq 0, \quad \sum_{i=1}^{n} \mathcal{D}_{ij} = \sum_{j=1}^{n} \mathcal{D}_{ij} = 1$$

11.4 Entanglement Transformations

Through the ideas of majorization, we can uncover the intricate aspects of quantum entanglement by understanding when we can transform a given copy of a pure bipartite quantum state $|\psi\rangle$ to another quantum state $|\varphi\rangle$ using LOCC[2]. Symbolically, we shall investigate

$$|\psi\rangle \xrightarrow{\text{LOCC}} |\varphi\rangle$$

As a first ingredient, we shall extend the definition of majorization to general density matrices that are Hermitian. We define majorization of $\rho_\psi = \text{Tr}_B\{|\psi\rangle\langle\psi|\}$ and $\rho_\varphi = \text{Tr}_B\{|\varphi\rangle\langle\varphi|\}$ such that

$$\rho_\psi \succ \rho_\varphi \qquad \Longleftrightarrow \qquad \boldsymbol{\lambda}_\psi \succ \boldsymbol{\lambda}_\varphi$$

where $\boldsymbol{\lambda}$ is a vector containing the eigenvalues.

From the above analogy of doubly stochastic matrices, we proceed to prove that $\rho_\psi \succ \rho_\varphi$ if and only if we have a stochastic unitary transformation of ρ_ψ to ρ_φ.

Theorem 11.4.1. *For Hermitian operators ρ_ψ, ρ_φ, we have $\rho_\psi \succ \rho_\varphi$ if and only if there exists a probability distribution p_j and unitary matrices U_j such that*

$$\rho_\varphi = \sum_j p_j U_j \rho_\psi U_j^\dagger$$

Proof. ($\implies$): Let ρ_ψ, ρ_φ. By definition, $\rho_\psi \succ \rho_\varphi$ implies $\boldsymbol{\lambda}_\psi \succ \boldsymbol{\lambda}_\varphi$, hence there exists a convex combination transformation through permutation matrix $\mathcal{P}_j \in S_n$ from the above proposition such that

$$\boldsymbol{\lambda}_\varphi = \sum_j p_j \mathcal{P}_j \boldsymbol{\lambda}_\psi$$

To transform from the eigenvalues to the density matrix, consider the diagonalisations through unitary transformations

$$\rho_\psi = \mathcal{S}_\psi^\dagger \Lambda_\psi \mathcal{S}_\psi, \quad \rho_\varphi = \mathcal{S}_\varphi^\dagger \Lambda_\varphi \mathcal{S}_\varphi$$

Now, note that the vectorial equation $\boldsymbol{\lambda}_\varphi = \sum_j p_j \mathcal{P}_j \boldsymbol{\lambda}_\psi$ can be expressed as

$$\Lambda_\varphi = \sum_j p_j \mathcal{P}_j \Lambda_\psi \mathcal{P}_j^\dagger$$

[2] The original idea explaining what tasks may be accomplished using a given physical resource and the ideas for entanglement transformations was first presented by Nielsen, Michael A. in "Conditions for a class of entanglement transformations." Physical Review Letters 83.2 (1999): 436.

Through the inverse transformation, we recover the density matrices

$$\begin{aligned}\rho_\varphi = \mathcal{S}_\varphi \Lambda_\varphi \mathcal{S}_\varphi^\dagger &= \mathcal{S}_\varphi \left(\sum_j p_j \mathcal{P}_j \Lambda_\psi \mathcal{P}_j^\dagger \right) \mathcal{S}_\varphi^\dagger \\ &= \sum_j p_j \mathcal{S}_\varphi \mathcal{P}_j \Lambda_\psi \mathcal{P}_j^\dagger \mathcal{S}_\varphi^\dagger = \sum_j p_j \mathcal{S}_\varphi \mathcal{P}_j (\mathcal{S}_\psi^\dagger \rho_\psi \mathcal{S}_\psi) \mathcal{P}_j^\dagger \mathcal{S}_\varphi^\dagger \\ &= \sum_j p_j (\mathcal{S}_\varphi \mathcal{P}_j \mathcal{S}_\psi^\dagger) \rho_\psi (\mathcal{S}_\psi \mathcal{P}_j^\dagger \mathcal{S}_\varphi^\dagger) \equiv \sum_j p_j \tilde{\mathcal{P}}_j \rho_\psi \tilde{\mathcal{P}}_j^\dagger\end{aligned}$$

where we define $\tilde{\mathcal{P}}_j := \mathcal{S}_\varphi \mathcal{P}_j \mathcal{S}_\psi^\dagger$, such that $\tilde{\mathcal{P}}_j^\dagger = \mathcal{S}_\psi \mathcal{P}_j^\dagger \mathcal{S}_\varphi^\dagger$. Note that the composition of unitary matrices with a permutation matrix, still results in another permutation matrix. We have completed the proof in the forward direction.
($\Longleftarrow$): Let $\rho_\varphi = \sum_j p_j U_j \rho_\psi U_j^\dagger$. On diagonalising, we have

$$\begin{aligned}\Lambda_\varphi = \mathcal{S}_\varphi^\dagger \rho_\varphi \mathcal{S}_\varphi &= \mathcal{S}_\varphi^\dagger \left(\sum_j p_j U_j \rho_\psi U_j^\dagger \right) \mathcal{S}_\varphi \\ &= \sum_j p_j \mathcal{S}_\varphi^\dagger U_j (\mathcal{S}_\psi \Lambda_\psi \mathcal{S}_\psi^\dagger) U_j^\dagger \mathcal{S}_\varphi \\ &= \sum_j p_j (\mathcal{S}_\varphi^\dagger U_j \mathcal{S}_\psi) \Lambda_\psi (\mathcal{S}_\psi^\dagger U_j^\dagger \mathcal{S}_\varphi) \equiv \sum_j p_j V_j \Lambda_\psi V_j^\dagger\end{aligned}$$

where we define $V_j := \mathcal{S}_\varphi^\dagger U_j \mathcal{S}_\psi$ and subsequently, we have $V_j^\dagger = \mathcal{S}_\psi^\dagger U_j^\dagger \mathcal{S}_\varphi$, which are unitaries. Now, the matrix components can be identified for V_j as $V_{j,kl}$ such that we have

$$(\boldsymbol{\lambda}_\varphi)_k = \sum_{jl} p_j V_{j,kl} (\boldsymbol{\lambda}_\psi)_l V_{j,lk}^\dagger = \sum_{jl} p_j |V_{j,kl}|^2 (\boldsymbol{\lambda}_\psi)_l$$

We define a matrix $\mathcal{D}$ with entries

$$\mathcal{D}_{kl} = \sum_j p_j |V_{j,kl}|^2$$

such that we have

$$\boldsymbol{\lambda}_\varphi = \mathcal{D} \boldsymbol{\lambda}_\psi$$

The entires of $\mathcal{D}$ are non-negative by definition, and we have rows and columns summing up to unitary, thereby, the matrix $\mathcal{D}$ is doubly stochastic and we have

$$\rho_\psi \succ \rho_\varphi$$

■

We can now proceed to characterising bipartite entanglement through the notion of majorization.

Theorem 11.4.2. *A bipartite pure state $|\varphi\rangle$ can be transformed to another pure state $|\psi\rangle$ by LOCC if and only if $\rho_\psi \succ \rho_\varphi$.*

Proof. ($\implies$): Suppose $|\varphi\rangle$ is transformed to state $|\psi\rangle$ by virtue of local operations and classical communication. By Theorem 11.2.1, we can assume that a bipartite system represented by A and B, with A performing a measurement with generalised measurement operators $\{M_i^A\}$, then sending the result to B, who performs an unitary transformation U_i. From the post-measurement theorem, we have A with density matrix ρ_φ transforming to state ρ_ψ, such that

$$\rho_\psi = \frac{M_j^A \rho_\varphi M_j^{A\dagger}}{\mathrm{Tr}(\rho_\varphi M_j^{A\dagger} M_j^A)}$$

Further, to express ρ_φ as a convex combinations of elements of ρ_ψ, note that we could polar decompose the matrix $M_j^A \sqrt{\rho_\varphi}$, such that there exists an unitary V_j that

$$M_i^A \sqrt{\rho_\varphi} := \sqrt{M_i^A \rho_\varphi M_i^{A\dagger}} V_i = \sqrt{\mathrm{Tr}(\rho_\varphi M_j^{A\dagger} M_j^A)\rho_\psi} V_i = \sqrt{p_i \rho_\psi} V_i$$

where p_i is the probability of outcome i. Premultiplying this equation by its adjoint, we thus realise,

$$\begin{aligned} (M_i^A \sqrt{\rho_\varphi})^\dagger M_i^A \sqrt{\rho_\varphi} &= (\sqrt{p_i \rho_\psi} V_i)^\dagger (\sqrt{p_i \rho_\psi} V_i) \\ \implies \sqrt{\rho_\varphi} M_i^{A\dagger} M_i^A \sqrt{\rho_\varphi} &= p_i V_i^\dagger \rho_\psi V_i \end{aligned}$$

Further, the completeness relation can be implemented for $\sum_i M_i^{A\dagger} M_i^A = \mathbb{I}$, such that

$$\begin{aligned} \sum_i p_i V_i^\dagger \rho_\varphi V_i &= \sum_i \sqrt{\rho_\varphi} M_i^{A\dagger} M_i^A \sqrt{\rho_\varphi} \\ &= \sqrt{\rho_\varphi} \left(\sum_i M_i^{A\dagger} M_i^A \right) \sqrt{\rho_\varphi} = \rho_\varphi \end{aligned}$$

Hence, we have

$$\rho_\varphi = \sum_i p_i V_i^\dagger \rho_\psi V_i$$

and by Thereorem 11.4.1, we can conclude $\rho_\psi \succ \rho_\varphi$

($\impliedby$): Let us assume $\rho_\psi \succ \rho_\varphi$, then we can pose Theorem 11.4.1, for the existence of a probability distribution p_j and unitary matrices U_j such that

$$\rho_\varphi = \sum_i p_i U_i \rho_\psi U_i^\dagger$$

Motivated by the previous instance, we construct operators M_j^A through the action

$$M_i^A \sqrt{\rho_\varphi} := \sqrt{\mathrm{Tr}(\rho_\varphi M_i^{A\dagger} M_i^A)\rho_\psi} U_i^\dagger = \sqrt{p_i \rho_\psi} U_i^\dagger$$

These define a set of measurement operators $\{M_i^A\}$ as seen from the completeness relation

$$\begin{aligned}\sum_i \sqrt{\rho_\varphi} M_i^{A\dagger} M_i^A \sqrt{\rho_\varphi} &= \sum_i (M_i^A \sqrt{\rho_\varphi})^\dagger (M_i^A \sqrt{\rho_\varphi}) \\ &= \sum_i (\sqrt{p_i \rho_\psi} U_i^\dagger)^\dagger (\sqrt{p_i \rho_\psi} U_i^\dagger) \\ &= \sum_i p_i U_i \rho_\psi U_i^\dagger = \rho_\varphi\end{aligned}$$

Hence, inverting the matrix $\sqrt{\rho_\varphi}$, we have

$$\sum_i M_i^{A\dagger} M_i^A = \rho_\varphi^{-\frac{1}{2}} \rho_\varphi \rho_\varphi^{-\frac{1}{2}} = \mathbb{I}$$

which proves the completeness relation. Thereby, A performs the measurement described by operators $\{M_i^A\}$, obtaining outcome i and corresponding state $|\psi_i^A\rangle \propto M_i^A|\varphi\rangle$. The reduced density matrix corresponding to the state $|\psi_i^A\rangle$ is $\rho_{\psi,i}^A = \mathrm{Tr}_B\{|\psi_i^A\rangle\langle\psi_i^A|\}$, thereby

$$\begin{aligned}\rho_{\psi,i} &\propto \mathrm{Tr}_B\{M_i^A|\varphi\rangle\langle\varphi|M_i^{A\dagger}\} \\ &= M_i^A \rho_\varphi M_i^{A\dagger} = (M_i^A\sqrt{\rho_\varphi})(M_i^A\sqrt{\rho_\varphi})^\dagger \\ &= (\sqrt{p_i\rho_\psi}U_i^\dagger)(\sqrt{p_i\rho_\psi}U_i^\dagger)^\dagger \\ &= p_i\sqrt{\rho_\psi}U_i^\dagger U_i\sqrt{\rho_\psi} = p_i\rho_\psi\end{aligned}$$

Hence, up to normalization $\rho_{\psi,i} \equiv \rho_\psi$. Now consider the state $|\psi_i^A\rangle$, which we can convert to $|\psi\rangle$ through the action of an unitary V_i such that the density matrices are equivalent. Thus, we can convert state $|\varphi\rangle$ to state $|\psi\rangle$ by virtue of LOCC. ■

Chapter 12

Quantum Error Correction

"To err is human, to forgive divine."

– Alexander Pope, *An Essay on Criticism*

12.1 Introduction

Any physical system is prone to errors. You may think that perfect isolation of a quantum system can prevent decoherence, hence no errors. This is not true! As quantum gates are unitary transformations chosen from a continuum of possible values, they cannot be implemented with perfect accuracy. Effects of tiny imperfections in the gate can accumulate and cause fatal errors.

All the algorithms and the usefulness of quantum computers we saw in this text so far are not of practical utility if the errors cannot be corrected. So our next task is to identify and correct errors.

Where to start? Drawing inspiration from classical error correction, we can try to devise quantum error correction. The act of error correction seems to be extremely non-trivial in the first place. For noting if there are errors, we need to inherently measure the state, which would imply the collapse of the state, and we are restricted by the no-cloning nature of quantum mechanics to not have copies to be redundant. The result of measuring a qubit can also destroy its quantum correlations with other qubits with which it might be entangled. Such disruptions are stochastic and unpredictable and introduce major errors of their own. We must turn to less obvious forms of monitoring. Further, our classical idea of errors is primarily flipping the bits. But here we could have further sources of error, like the phase flip or other issues, which are purely quantum in nature.

Despite all these hurdles, there is an enormous literature on quantum error correction that is raising hopes for practically useful quantum computers in the near future.

12.2 Bit Flip Code

Consider a case where the error channel flips $|0\rangle$ to $|1\rangle$ or the other way with probability p. Let $\rho_{\text{initial}} = |\phi\rangle\langle\phi|$ be the initial density matrix of the system. Then, after passing through the error channel, it becomes

$$\rho_{\text{final}} = (1-p)\,|\phi\rangle\langle\phi| + pX\,|\phi\rangle\langle\phi|\,X$$

If we do not take any additional caution to correct the error, then the probability of failure is

$$P_{\text{error}} = 1 - \langle\phi|\,\rho_{\text{final}}\,|\phi\rangle$$

If we consider the initial state to be a pure state of the form $|\phi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, then

$$P_{\text{error}} = 1 - \left(1 - p + p|\,\langle\phi|\,X\,|\phi\rangle\,|^2\right) = p\left(1 - |\alpha^*\beta + \beta^*\alpha|^2\right)$$

Thus, the probability of failure is of order p. Is there a way to reduce this error?

Having redundancy is a good old method to reduce errors. Even if an error occurs, it is less likely to affect all our redundant qubits. By majority rule, we can determine and correct the errors. We know that the no-cloning theorem says we can not copy a quantum state, but instead we can copy the basis state. Consider making three copies of the basis.

$$|\bar{0}\rangle \to |000\rangle \text{ and } |\bar{1}\rangle \to |111\rangle$$

These $|\bar{0}\rangle$ and $|\bar{1}\rangle$ are called *logical qubits* and the ones without overline, $|0\rangle$ and $|1\rangle$ are *physical qubits*. Thus, we are *encoding* a single qubit state to a three qubit Hilbert space.

In other words, out initial state $|\phi\rangle$ becomes,

$$|\phi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \longrightarrow |\psi\rangle = \alpha\,|000\rangle + \beta\,|111\rangle$$

The circuit in the figure 12.1 does exactly this.

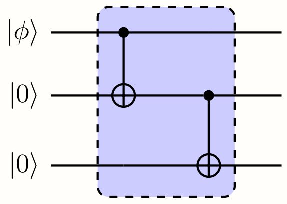


Figure 12.1: Copying the basis state

Algebraically the action of this circuit can be seen as,

$$\begin{aligned}(P_0 \otimes \mathbb{I} + P_1 \otimes X)\left((a|10\rangle + b|11\rangle) \otimes |0\rangle\right) &= (P_0 \otimes \mathbb{I} + P_1 \otimes X)\left(a|100\rangle + b|110\rangle\right) \\ &= (P_0 \otimes \mathbb{I})(a|100\rangle) + (P_0 \otimes \mathbb{I})(b|110\rangle) \\ &+ (P_1 \otimes X)(a|100\rangle) + (P_1 \otimes X)(b|110\rangle) \\ &= a|100\rangle + b|111\rangle \\ (P_0 \otimes \mathbb{I} + P_1 \otimes X)((a|100\rangle + b|111\rangle) \otimes |0\rangle) &= (P_0 \otimes \mathbb{I})(a|1000\rangle) + (P_0 \otimes \mathbb{I})(b|1110\rangle) \\ &+ (P_1 \otimes X)(a|1000\rangle) + (P_1 \otimes X)(b|1110\rangle) \\ &= a|1000\rangle + b|1111\rangle = a|1000\rangle + b|1111\rangle\end{aligned}$$

Now with three qubits, there are many possibilities. There may be no error, an error at just one qubit, an error on two qubits, or even errors on all three qubits. So the after passing through the error channel the density matrix becomes,

$$\begin{aligned}\sigma' &= (1-p)^3 \left|\psi\right\rangle \left\langle\psi\right| \\ &+ p(1-p)^2 \left(X_1 \left|\psi\right\rangle \left\langle\psi\right| X_1 + X_2 \left|\psi\right\rangle \left\langle\psi\right| X_2 + X_3 \left|\psi\right\rangle \left\langle\psi\right| X_3\right) \\ &+ p^2(1-p) \left(X_1X_2 \left|\psi\right\rangle \left\langle\psi\right| X_1X_2 + X_2X_3 \left|\psi\right\rangle \left\langle\psi\right| X_2X_3 + X_3X_1 \left|\psi\right\rangle \left\langle\psi\right| X_3X_1\right) \\ &+ p^3 \left(X_1X_2X_3 \left|\psi\right\rangle \left\langle\psi\right| X_1X_2X_3\right)\end{aligned}$$

Even after having copies, the quantum information will be lost if we measure any one qubit. Instead, we can do some clever collective measurements on the qubits such that the state does not change after measurement.

Consider the operators Z_1Z_2 and Z_2Z_3 and their action on the following states. The table 12.1 shows some possible states the tree qubits may be in after going through the error channel. Notice that both Z_1Z_2 and Z_2Z_3 do not change the state of any of these. In other words, the tabulated states are eigenstates of Z_1Z_2 and Z_2Z_3 whose eigenvalues are helping in locating the errors. Such measurements is called a *syndrome measurement*.

State	Z_1Z_2	Z_2Z_3	Action to correct the error
$\alpha \left|000\right\rangle + \beta \left|111\right\rangle$	+1	+1	$\mathbb{I}$
$\alpha \left|100\right\rangle + \beta \left|011\right\rangle$	-1	+1	X_1
$\alpha \left|010\right\rangle + \beta \left|101\right\rangle$	-1	-1	X_2
$\alpha \left|001\right\rangle + \beta \left|110\right\rangle$	+1	-1	X_3
$\alpha \left|110\right\rangle + \beta \left|001\right\rangle$	+1	-1	X_3

Table 12.1: Action of Z_1Z_2 and Z_2Z_3 on various three-qubit states

It is important that our measurement to diagnose the bit flip is a collective measurement on two qubits at once. We infer the value of Z_1Z_2 and Z_2Z_3 but get to learn nothing about the separate values of Z_1, Z_2 or Z_3, doing so would damage the encoded state.

We can perform the above pair of Z-measurements using the circuit 12.2 with the help of two additional *ancilla* qubits. Note that the 2 cNOT outside the box is the one we saw earlier that copies the basis state.

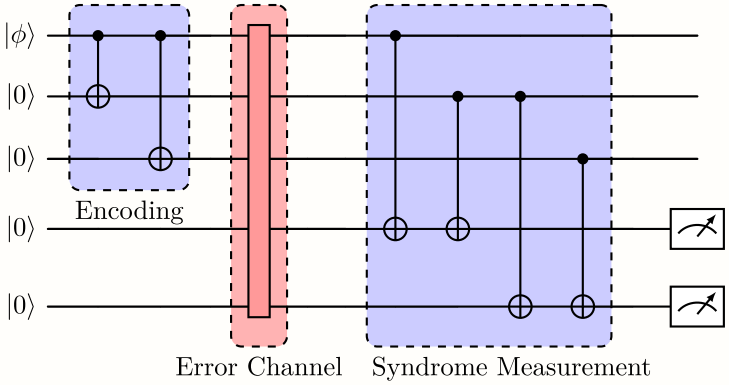


Figure 12.2: Syndrome measurements for bit flip error

After performing Z_1Z_2 and Z_2Z_3 we get 4 possible final states for all possible bit flip errors (i.e not only single bit flip.)
Notice both Z_1Z_2 and Z_2Z_3 have $+1$ eigen value for $\alpha\,|000\rangle + \beta\,|111\rangle$ and $\alpha\,|110\rangle + \beta\,|001\rangle$.

$$\begin{aligned}
\sigma_0 &= (1-p)^3|\psi\rangle\langle\psi| + p^3 X_1X_2X_3|\psi\rangle\langle\psi|X_1X_2X_3 \\
\sigma_1 &= (1-p)^2 pX_1|\psi\rangle\langle\psi|X_1 + (1-p)p^2 X_2X_3|\psi\rangle\langle\psi|X_2X_3 \\
\sigma_2 &= (1-p)^2 pX_2|\psi\rangle\langle\psi|X_2 + (1-p)p^2 X_1X_3|\psi\rangle\langle\psi|X_1X_3 \\
\sigma_3 &= (1-p)^2 pX_3|\psi\rangle\langle\psi|X_3 + (1-p)p^2 X_1X_2|\psi\rangle\alpha\psi \mid X_1X_2
\end{aligned}$$

Now depending on the Z_1Z_2, Z_2Z_3 values we operate with X_1, X_2 ar X_3
After which the state becomes:

$$\begin{aligned}
\sigma_0' &= (1-p)^3|\psi\rangle\langle\psi 1 + p^3 X_1X_2X_3 \mid \psi\rangle\langle\psi|X_1X_2X_3 \\
\sigma_1' &= (1-p)^2 p|\psi\rangle\langle\psi| + (1-p)p^2 X_1X_2X_3|\psi\rangle\langle\psi|X_1X_2X_3 \\
\sigma_2' &= (1-p)^2 p|\psi\rangle\langle\psi| + (1-p\rangle p^2 X_1X_2X_3|\psi\rangle\langle\psi|X_3X_2X_3 \\
\sigma_3' &= (1-p)^2 p|\psi\rangle\langle\psi| + (1-p)p^2 X_1X_2X_3|\psi\rangle\langle\psi|X_1X_2X_3
\end{aligned}$$

The final density matrix is $\sigma' = \sum_{k=0}^{3} \sigma_k'$
Does this redundancy actually help? Calculating the P_{error}, we find that the error probability has indeed reduced to p^2 from p (note $p < 1$).

$$\begin{aligned}
P_{\text{error}} &= 1 - \langle\psi|\,\sigma'\,|\psi\rangle \\
&= p^2(3-2p)\left(1 - |\,\langle\psi|\,X_1X_2X_3\,|\psi\rangle\,|^2\right) \\
&= p^2(3-2p)\left(1 - |\alpha^*\beta + \beta^*\alpha|^2\right)
\end{aligned}$$

Remarks. *Why just three copies? We can make multiple copies of the basis state and use appropriate syndrome measurements to reduce the error further. By encoding* $|0\rangle \rightarrow |0\rangle^{\otimes(2r-1)}$ *and* $|1\rangle \rightarrow |1\rangle^{\otimes(2r-1)}$, P_{error} *vanishes as* r *tends to infinity.*

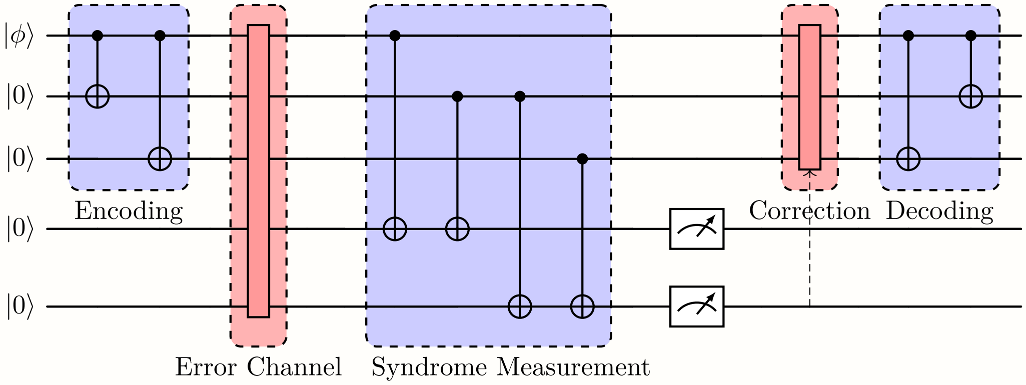


Figure 12.3: Bit Flip Error Correction Circuit

12.2.1 Effect of bit flip on Bloch sphere

Recall that an arbitrary density matrix can be written as $\frac{1}{2}(\mathbb{I} + \vec{r} \cdot \vec{\sigma})$, where $\vec{r}$ is the Bloch vector and $\vec{\sigma}$ is Pauli vector and $\text{Tr}(\rho^2) = \frac{1+|\vec{r}|^2}{2}$. Thus, there is a vector $\vec{r} = (r_1, r_2, r_3)$ in the Bloch sphere correcponding to every density matrix.

When ρ goes through a bit flip channel, it becomes $p\rho + (1-p)X\rho X$. As

$$\rho = \frac{\mathbb{I} + r_1 X + r_2 Y + r_3 Z}{2} \text{ and } X\rho X = \frac{\mathbb{I} + r_1 X - r_2 Y - r_3 Z}{2}$$

The Bloch vector changes after going through the bit flip channel.

$$\rho \longrightarrow p\rho + (1-p)X\rho X$$

$$(r_1, r_2, r_3) \mapsto p(r_1, r_2, r_3) + (1-p)(r_1, r_2, r_3)$$

The new coordinates are,

$$\begin{aligned} r_1' &= r_1 \\ r_2' &= (2p-1)r_2 \\ r_3' &= (2p-1)r_3 \end{aligned}$$

Thus, the x-coordinate remains unchanged, and the y and z coordinates get squeezed by a factor of $2p-1$. This is depicted in the figure 12.4. As the norm of the Bloch vector $|\vec{r}|$ can only decrease in this process, the trace, $\text{Tr}(\rho^2)$, can only decrease or stay the same.
An interesting thing happens at $p = 0.5$. Both y and z coordinate vanishes and the Bloch sphere becomes a projection onto the x axis.

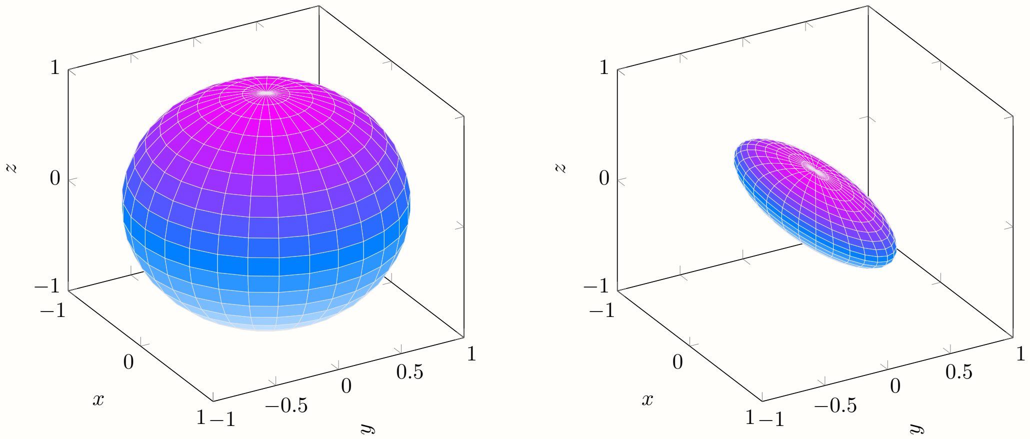


Figure 12.4: Effect of bit flip channel on the Bloch sphere, with $p = 0.2$

12.3 Phase Flip Code

Here the error channel flips the phase of the qubit with probability p. In other words the initial density matrix $\rho_{\text{initial}} = |\phi\rangle\langle\phi|$ becomes, $\rho_{\text{final}} = p\rho + (1-p)Z\rho Z$.

Can one modify the bit flip circuit to account for phase flips? Recall that $X = HZH$. So, if we apply Hadamard gates across the error channel, then any phase flip will appear like bit flips, and the same circuit used for bit flips can be used for phase flips as well.

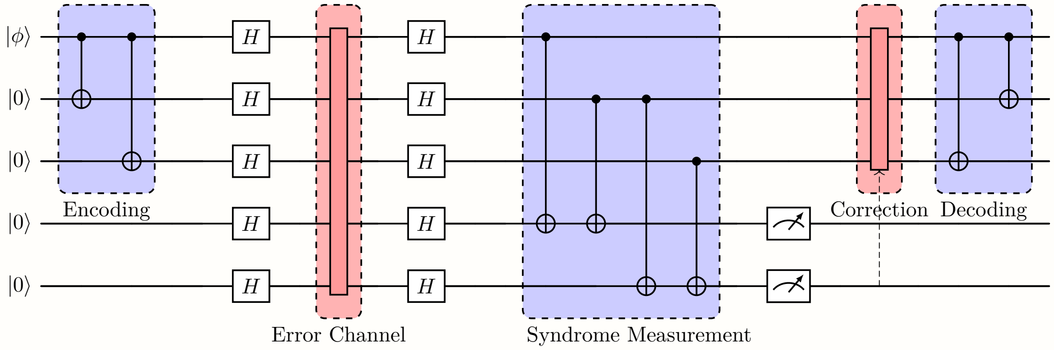


Figure 12.5: Phase Flip Error Correction Circuit

One example of encoding, correcting and decoding is,

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{\text{encoding}} \alpha|000\rangle + \beta|111\rangle \xrightarrow{H} \alpha|+++\rangle + \beta|---\rangle \xrightarrow{\text{error}} \alpha|+-+\rangle + \beta|-+-\rangle$$

$$\alpha|+-+\rangle + \beta|-+-\rangle \xrightarrow{H} \alpha|010\rangle + \beta|101\rangle \xrightarrow{\text{correction}} \alpha|000\rangle + \beta|111\rangle \xrightarrow{\text{decoding}} \alpha|0\rangle + \beta|1\rangle$$

12.3.1 Effect of Phase Flip Channel on the Bloch Sphere

In the case of a phase flip channel, the following is the effect on the Bloch sphere,

$$\rho \longrightarrow p\rho + (1-p)X\rho X$$

$$(r_1, r_2, r_3) \mapsto p(r_1, r_2, r_3) + (1-p)(-r_1, -r_2, r_3)$$

$$= ((2p-1)r_1, (2p-1)r_2, r_3))$$

as $ZXZ = -X$ and $ZYZ = -Y$, thus $Z\rho Z$ flips the sign of both r_1 and r_2. Overall, the z-coordinate remains the same, and x and y coordinates get squeezed. Similar to what we saw in the case of bit flip code, at $p = 0.5$. Both x and y coordinate vanishes, and the Bloch sphere becomes a projection onto the z axis.

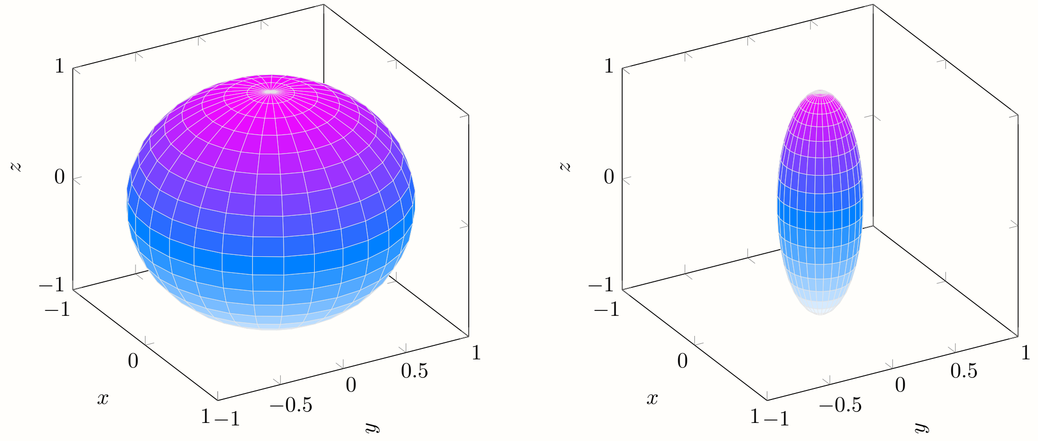


Figure 12.6: Effect of phase flip channel on the Bloch sphere, with $p = 0.2$

12.4 Bit-Phase Flip Code

A combination of bit and phase flip gives $XZ = -iY$. Its action on the Bloch sphere is shown in figure 12.7.

$$\rho \longrightarrow p\rho + (1-p)X\rho X$$

$$(r_1, r_2, r_3) \mapsto p(r_1, r_2, r_3) + (1-p)(-r_1, r_2, -r_3)$$

$$= ((2p-1)r_1, r_2, (2p-1)r_3))$$

To handle such and more general types of errors, where both bit and phase flips can occur, we need a more involved error correction code. One such code is Shor's code, which we will see in the subsequent sections.

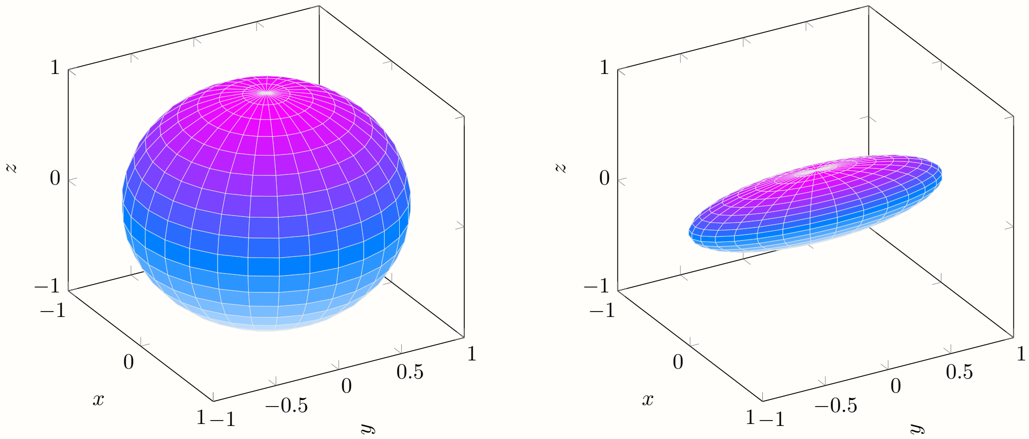


Figure 12.7: Effect of bit-phase flip channel on the Bloch sphere, with $p = 0.2$

12.5 Operator Formalism for Error Correction

12.5.1 Kraus Operator in Error Correction

The concept of Kraus operator was introduced in the section 9.2. Here, let us recall the Kraus operator and see how it comes into play in error correction.

We know that the dynamics of a closed quantum system is described by unitary evolution. The error channel can be seen as a system's environment that tampers with the system, which is no longer closed. How can we describe such an open quantum system? We can just consider our principal system and environment together as a giant quantum system, which is closed and whose evolution is unitary.

We can assume that our initial principal system and the environment are in a product state, $\rho \otimes \rho_{\text{env}}$. (This need not be always true, still the assumption is practically valid, as in many cases we prepare our initial setup and thus can make sure this happens.) Now let the overall evolution of the principal system and the environment be a unitary U, which also captures the system-environment interactions. Once the evolution is done, they no longer interact, but they can now be entangled. Thus, we can no longer write them in a product state. For this same reason, we may not be able to write the final state of the system only, say $\varepsilon(\rho)$ can be written as a unitary transform of the initial state ρ. That is,

$$\varepsilon(\rho) \neq \tilde{U}\rho\tilde{U}^\dagger$$

where $\tilde{U}$ is unitary evolution in the system's Hilbert space. But we can write $\varepsilon(\rho)$ as,

$$\varepsilon(\rho) = \text{Tr}_{\text{env}}(U\rho \otimes \rho_{\text{env}}U^\dagger)$$

What if the environment has infinite degree of freedom?

It turns out that for the above model to properly describe any transformation $\rho \to \varepsilon(\rho)$ where ρ is in d-dimensional Hilbert space, it is sufficient to model the environment as no more than d^2-dimensional Hilbert space.

Suppose $|e_k\rangle$ are the orthogonal basis of the environment (finite dimensional), and assuming the environment is initially in a pure state (if not using Schmidt decomposition we can purify as mentioned in section 9.1), we can write $\rho_{env} = |e_0\rangle\langle e_0|$ and,

$$\varepsilon(\rho) = \sum_k \langle e_k|U[\rho \otimes |e_0\rangle\langle e_0|]U^\dagger|e_k\rangle = \sum_k E_k \rho E_k^\dagger$$

where $E_k \equiv \langle e_k|U|e_0\rangle$ is an operator on the principal system Hilbert space, and we call this the *Kraus operator*, and the above representation of $\varepsilon(\rho)$ is called the *operator-sum representation.*

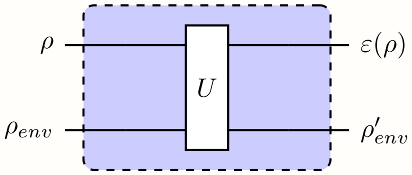


Figure 12.8: System and environment together as a closed quantum system

Is $\varepsilon(\rho)$ a valid density matrix too?

Given the transformation is trace preserving, we expect $\varepsilon(\rho)$ to also be a valid density matrix satisfying all properties of a density matrix, including the fact that it has unit trace. Thus,

$$\begin{aligned}\mathrm{Tr}\left(\varepsilon(\rho)\right) &= 1\\ \mathrm{Tr}\left(\sum_k E_k \rho E_k^\dagger\right) &= 1\\ \mathrm{Tr}\left(\sum_k E_k^\dagger E_k \rho\right) &= 1 \quad \text{(By linearity and cyclic property of trace)}\end{aligned}$$

As the above should hold for all ρ we can conclude that $\sum_k E_k^\dagger E_k = 1$.

Note that there are also non-trace preserving quantum operations for which $\sum_k E_k^\dagger E_k \leq 1$. These describe a process where the extra information about the process is obtained by measurements. Also, if a quantum process models a system interacting with the environment in a way that allows information or energy loss, then the resulting operation can decrease the total probability, thus $\mathrm{Tr}(\varepsilon(\rho)) < 1$.

Is there any physical motivation for expressing the evolved principal system in operation-sum representation? Imagine a situation where we measure the environment in $\{|e_k\rangle\}$ basis after letting the system and environment evolve together. Suppose outcome k occurred. Let ρ_k be the corresponding principal system given that the outcome k occurred. Formally, the post measurement state and ρ_k are,

$$\text{Post-measurement state} = \frac{M(U[\rho \otimes |e_0\rangle \langle e_0|]U^\dagger)M^\dagger}{\mathrm{Tr}(M(U[\rho \otimes |e_0\rangle \langle e_0|]U^\dagger)M^\dagger)}$$

$$\rho_k = \frac{\mathrm{Tr}_{env}(|e_k\rangle \langle e_k| \, U[\rho \otimes |e_0\rangle \langle e_0|]U^\dagger \, |e_k\rangle \langle e_k|)}{\mathrm{Tr}(|e_k\rangle \langle e_k| \, U[\rho \otimes |e_0\rangle \langle e_0|]U^\dagger \, |e_k\rangle \langle e_k|)} = \frac{E_k \rho E_k^\dagger}{\mathrm{Tr}(E_k \rho E_k^\dagger)}$$

Note that the probability of getting outcome k, $p(k) = \mathrm{Tr}(E_k \rho E_k^\dagger)$. Thus we can write,

$$\varepsilon(\rho) = \sum_k p(k)\rho_k = \sum_k E_k \rho E_k^\dagger$$

This gives the interpretation to view $\varepsilon(\rho)$ as a mixed state of all possible states of the system if a measurement was done on the environment in $\{|e_k\rangle\}$ basis.

Remarks. *Thus, the action of quantum operation is equivalent to taking the system from* ρ *and randomly placing it in* $\frac{E_k \rho E_k^\dagger}{\mathrm{Tr}(E_k \rho E_k^\dagger)}$ *with probability* $\mathrm{Tr}(E_k \rho E_k^\dagger)$*, which is similar to the effect of a noisy classical communication channel.*

The above shows that given an interacting quantum system, it gives rise to an operator-sum representation. But is the converse true? Given a set of operators $\{E_k\}$, can we associate it with a model environmental system and dynamics that gives the corresponding operator-sum representation, where the dynamics is either unitary or projective measurements. The answer turns out to be is yes![1]

We saw a few examples of errors and a way to correct those errors. Given the above operator language, one can now associate Kraus operators (as shown in table 12.2) with all the error correction codes seen so far.
A natural question to ask is, can any form of error be corrected? Here comes the power of operator representation, which can characterize correctable errors.

12.5.2 Quantum Error Correction Condition in Operator Formalism

Theorem 12.5.1. *Let* C *be a quantum code and* Π *projector onto* C*. Suppose* ε *(noise) is a quantum operator with operation elements* $\{E_i\}$*, then the necessary and sufficient condition for the existence of an error-correcting operation* R *correcting* ε *on* C *is that*

$$\Pi E_i^\dagger E_j \Pi = m_{ij} \Pi$$

[1] Proof for this can be found in the Quantum Computing and Quantum Information textbook by Isaac Chuang and Michael Nielsen

Error Type	Kraus Operator	Syndrome	Bloch Sphere: $(r_1, r_2, r_3) \to$
Bit Flip	$\{\sqrt{p}\mathbb{I}, \sqrt{1-p}X\}$	Z_1Z_2 and Z_2Z_3	$(r_1, (2p-1)r_2, (2p-1)r_3)$
Phase Flip	$\{\sqrt{p}\mathbb{I}, \sqrt{1-p}Z\}$	X_1X_2 and X_2X_3	$((2p-1)r_1, (2p-1)r_2, r_3)$
Bit-Phase Flip	$\{\sqrt{p}\mathbb{I}, \sqrt{1-p}Y\}$	Z_1Z_2 and Z_2Z_3	$((2p-1)r_1, r_2, (2p-1)r_3)$

Table 12.2: Summary of the error codes with corresponding Kraus operators

for some Hermition matrix $\boldsymbol{m} = [m_{ij}]$. The above condition is called quantum error correction condition or the Knill-Laflamme (KL) condition. The operator elements $\{E_i\}$ for the noise ε are called errors, and if such R exists, we say $\{E_i\}$ constitutes a correctable set of errors.

Proof. We will prove the sufficiency by constructing R given the KL condition.

($\Rightarrow$): Suppose $\{E_i\}$ satisfies KL condition, $\boldsymbol{m} = [m_{ij}]$ is Hermitian so it can be diagonalized $\Lambda = U^\dagger \boldsymbol{m} U$. Define $F_k \equiv \sum_i U_{ik} E_i$. As F_k is a sum of unitaries times operator elements, we know that $\{F_k\}$ is also a set of operator elements of ε.
Substituting F_k in KL condition we get,

$$\begin{aligned}
\Pi F_k^\dagger F_l \Pi =& \sum_{ij} U_{ki}^\dagger U_{jl} \Pi E_i^\dagger E_j \Pi \\
=& \sum_{ij} U_{ki}^\dagger U_{jl} m_{ij} \Pi = \sum_{ij} U_{ki}^\dagger mij U_{jl} \Pi \\
=& \Lambda_{kl} \Pi
\end{aligned}$$

Since $F_k\Pi$ is a linear map, we can find a polar decomposition

$$F_k\Pi = \mathcal{U}_k \sqrt{\Pi F_k^\dagger F_k \Pi} = \sqrt{\Lambda_{kk}} \mathcal{U}_k \Pi$$

since $\Pi^2 = \Pi \Rightarrow \sqrt{\Pi} = \Pi$, and for some unitary $\mathcal{U}_k$.
We have $F_k\Pi = \sqrt{\Lambda_{kk}}\mathcal{U}_k\Pi$, implying, $\mathcal{U}_k \Pi \mathcal{U}_k^\dagger = \frac{F_k \Pi \mathcal{U}_k^\dagger}{\sqrt{\Lambda_{kk}}}$ Now, define $\Pi_k \equiv U_k \Pi U_k^\dagger$, then,

$$\Pi_l \Pi_k = \Pi_l^\dagger \Pi_k = (\mathcal{U}_l^\dagger \Pi \mathcal{U}_l)(\mathcal{U}_k \Pi \mathcal{U}_k^\dagger) = \frac{\mathcal{U}_l^\dagger \Pi F_l^\dagger}{\sqrt{\Lambda_{ll}}} \frac{F_k \Pi \mathcal{U}_k^\dagger}{\sqrt{\Lambda_{kk}}} = \frac{\mathcal{U}_l^\dagger}{\sqrt{\lambda_{ll}}} \Pi F_l^\dagger F_k \Pi \frac{\mathcal{U}_k^\dagger}{\sqrt{\Lambda_{kk}}} = 0$$

Note that $\Pi F_l^\dagger F_k \Pi = \Lambda_{lk}\Pi$ and it is 0 when $l \neq k$. Thus, Π_k 's can be thought of as orthonormal projective measurements.

Now the syndrome measurement can he defined as the projectors Π_k augmented by an additional n projector if necessary to satisfy the completeness relation. Let $\Pi' = \mathbb{I} - \sum_k \Pi_k$. Therefore, $\{F_k\}$ being equivalent to $\{E_k\}$, has the action on the code space, as

$$F_k \Pi = \sqrt{\Lambda}_{kk} \mathcal{U}_k \Pi$$

Note that we can get back to the code space by applying $\mathcal{U}_k^\dagger$.
Thus, the combined detection-recovery step corresponds to the quantum operator

$$R(\sigma) = \sum_k \mathcal{U}_k^\dagger \Pi_k \sigma \Pi_k \mathcal{U}_k$$

Note that, we have

$$\begin{aligned}\mathcal{U}_k^\dagger \Pi_k F_l \sqrt{\rho} &= \mathcal{U}_k^\dagger \Pi_k^\dagger F_l \Pi \sqrt{\rho} = \frac{\mathcal{U}_k^\dagger \mathcal{U}_k \Pi F_k^\dagger F_l \Pi}{\sqrt{\Lambda_{kk}}} \sqrt{\rho} \\ &= \delta_{kl} \sqrt{\Lambda}_{kk} \Pi \sqrt{\rho}\end{aligned}$$

Therefore, we have

$$R(\varepsilon(\rho)) = \sum_{kl} \mathcal{U}_k^\dagger \Pi_k F_l \rho F_l^\dagger \Pi_k \mathcal{U}_k = \sum_{kl} \delta_{kl} \Lambda_{kk} \rho \propto \rho$$

as required by the recovery operation.

($\Leftarrow$): Suppose $\{E_i\}$ is a set of errors which is perfectly correctable by an error-correction operation R with operation elements $\{R_j\}$. Define the quantum operator $\varepsilon_c(\rho) \equiv \varepsilon(\Pi\rho\Pi)$ where $\Pi\rho\Pi$ is in the code space.

We thereby have, $R\left(\varepsilon_c(\rho)\right) \propto \Pi\rho\Pi$, expanding this out $\sum_{ij} R_j E_i \Pi\rho\Pi E_i^\dagger R_j^\dagger = \mathcal{C}\Pi\rho\Pi$, where $\mathcal{C}$ is constant.
Thereby, $\{R_j E_i\}$ is identical to a single operator element $\sqrt{\mathcal{C}}\Pi$, such that $R_k E_i \Pi = \mathcal{C}_{ki}\Pi$, taking adjoint $\Pi E_i^\dagger R_k^\dagger = \mathcal{C}_{k,1}^\dagger \Pi$, therefore $\Pi E_i^\dagger R_k^\dagger R_k E_j \Pi = \mathcal{C}_{k_i}^* \mathcal{C}_{kj} \Pi$
As R is trace-preserving $\sum_k R_k^\dagger R_k = 1$, therefore,

$$\sum_k \Pi E_i^\dagger R_k^\dagger R_k E_j \Pi = \Pi E_i^\dagger E_j \Pi = \sum_k \mathcal{C}_{k_i}^* \mathcal{C}_{k_j} \Pi = m_{ij} \Pi$$

where m_{ij} is Hermitian. This matches with the KL condition. ■

12.6 Shor's Code

We saw the bit flip and the phase flip code. Building on this, can one design an error correction scheme that can correct any arbitrary error on a single qubit? Let us try to build one such code.

Like the earlier examples, we can start by encoding our physical qubits into a certain number of logical qubits. Suppose the state of the encoded qubit is $|\psi\rangle = \alpha |\bar{0}\rangle + \beta |\bar{1}\rangle$, and after the action of the noise channel it becomes $\varepsilon(|\psi\rangle\langle\psi|) = \sum_i E_i |\psi\rangle\langle\psi| E_i^\dagger$. We saw that physically we can interpret the action of the error channel as changing the initial state of the system to $E_i |\psi\rangle\langle\psi| E_i^\dagger$ with a certain probability. Thus, focusing on one term $E_i |\psi\rangle\langle\psi| E_i^\dagger$, notice that as the Pauli matrices, $\{\mathbb{I}, X, Y, Z\}$, forms a basis we can write $E_i = c_{i0}\mathbb{I} + c_{i1}X + c_{i2}XZ + c_{i3}Z$. Thus the un-normalized state $E_i|\psi\rangle$ can be written as superposition of I, $X_1 \left|\psi\right\rangle$, $X_1 Z_1 \left|\psi\right\rangle$ and $Z_1 \left|\psi\right\rangle$. Thus, if we have an error correction code that can correct just X and Z type errors in a single qubit, we can use it to correct E_i. As measuring the error syndrome will collapse the state to one of the above Pauli basis states, and recovery can be done appropriately.

Now let us see how to construct a code that can take care of both bit and phase flip of a single qubit. Naturally, let us just try to combine the codes we already saw, by first encoding using the phase flip code and then the bit flip code on each of the phase flip encoded qubits. This will give a 9-qubit encoding for a single qubit.

$$\left|0\right\rangle \xrightarrow{\text{phase flip}} \left|+++\right\rangle \xrightarrow{\text{bit flip}} \frac{\left|000\right\rangle + \left|111\right\rangle}{\sqrt{2}} \frac{\left|000\right\rangle + \left|111\right\rangle}{\sqrt{2}} \frac{\left|000\right\rangle + \left|111\right\rangle}{\sqrt{2}} \equiv \left|\overline{0}\right\rangle$$

$$\left|1\right\rangle \xrightarrow{\text{phase flip}} \left|---\right\rangle \xrightarrow{\text{bit flip}} \frac{\left|000\right\rangle - \left|111\right\rangle}{\sqrt{2}} \frac{\left|000\right\rangle - \left|111\right\rangle}{\sqrt{2}} \frac{\left|000\right\rangle - \left|111\right\rangle}{\sqrt{2}} \equiv \left|\overline{1}\right\rangle$$

Applying the KL criterion to the Shor code

For Shor's code, we have the correctable error set, described by $E = \{\mathbb{I}, X_i, Y_i, Z_i\}$ for $i = 1, 2, \ldots, 9$. We choose the code basis as $|\overline{0}\rangle$ and $|\overline{1}\rangle$. It is easy to note that $\langle\overline{0}|E_i^\dagger E_j|\overline{1}\rangle = 0$, since the basis kets are constructed orthogonally, and there is no transformation connecting the two.

The significant condition to check is for whether $\langle\overline{0}|E_i^\dagger E_j|\overline{0}\rangle = \langle\overline{1}|E_i^\dagger E_j|\overline{1}\rangle$, where usually both are not zero. For $E_i = Z_\alpha$, $E_j = Z_\beta$, we have both equaling one, and similarly for other combinations of Z_α's. For any other operator, both quantities are equal to zero. Thereby, the KL condition is satisfied for Shor's.

The circuit to construct the above encoding is again just a combination of the bit flip and phase flip circuit as shown in figure 12.9.

This method of stacking and constructing an encoding is called *concatenation* and is a useful trick to construct new codes from old ones.

After encoding is done, how to correct this error? Again, we can just extend the syndrome measurements of the bit flip code and phase flip code as shown in the table 12.3 and table 12.4.

The complete circuit for Shor's code is given in Figure 12.10. Note that, unlike bit flip and phase flip code, this circuit does not have a measurement operator, as measurements can be equivalently converted to control not gates 5.6.

It is important to note what type of errors Shor's code can correct and cannot. If we cluster each of the nine qubits into sets of three, then Shor's code can correct:

- Single qubit bit flip error in any 1 out of 3 qubits in a cluster.

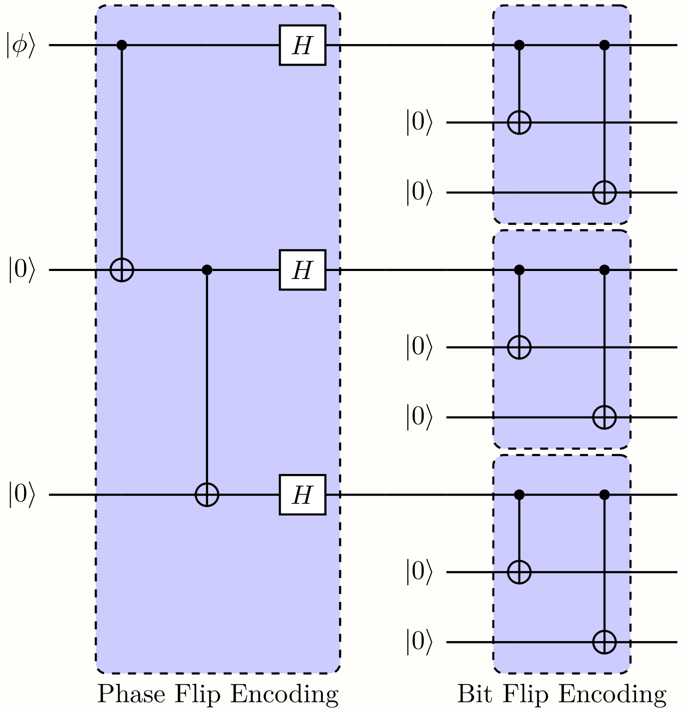


Figure 12.9: Concatenation of bit flip and phase flip encoding

Error	Z_1Z_2	Z_2Z_3	Z_1Z_2	Z_2Z_3	Z_1Z_2	Z_2Z_3	Action to correct the error
$\mathbb{I}$	+1	+1	+1	+1	+1	+1	$\mathbb{I}$
X_1	-1	+1	+1	+1	+1	+1	X_1
X_2	-1	-1	+1	+1	+1	+1	X_2
X_3	+1	-1	+1	+1	+1	+1	X_3
X_4	+1	+1	-1	+1	+1	+1	X_4
X_5	+1	+1	-1	-1	+1	+1	X_5
X_6	+1	+1	+1	-1	+1	+1	X_6
X_7	+1	+1	+1	+1	-1	+1	X_7
X_8	+1	+1	+1	+1	-1	-1	X_8
X_9	+1	+1	+1	+1	+1	-1	X_9

Table 12.3: Syndrome measurements to detect single bit flip errors

Error	$X_1X_2X_3X_4X_5X_6$	$X_4X_5X_6X_7X_8X_9$	Action
$\mathbb{I}$	+1	+1	$\mathbb{I}$
$Z_1/Z_2/Z_3$	-1	+1	$Z_1/Z_2/Z_3$
$Z_4/Z_5/Z_6$	-1	-1	$Z_4/Z_5/Z_6$
$Z_7/Z_8/Z_9$	+1	-1	$Z_7/Z_8/Z_9$

Table 12.4: Syndrome measurements to detect single phase flip errors

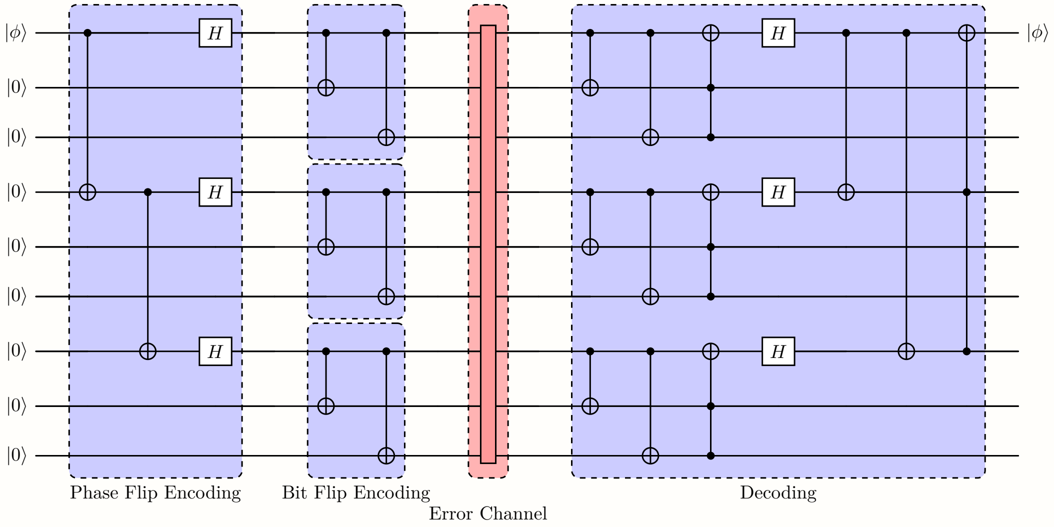


Figure 12.10: Shor's Error Correction Code

- Phase flip in 1 cluster.

The combination of the above two covers all types of single-qubit errors. Shor's code cannot correct for:

- Two bit flips in a single cluster.
- Phase error in two different clusters.

The keen reader would have wondered and probably noticed the connection between the logical qubits and the syndrome measurements in each of the above error correction codes. The logical qubits always turn out to be the $+1$ eigenstates of the syndrome measurement operators. Should this always be the case? We have seen the operator formalism of error correction code; there is yet another elegant formalism of the error correction code stabilizer formalism. This gives the answer to our above question and leverages the underlying group structure of the Pauli operators.

Remarks. *It is worth noting the power of the above code. Regardless of the nature of E_i, it can be a small rotation of 1 degree in the Bloch sphere about the X axis or a complete replacement of the qubit with some garbage. We can correct this continuum of errors by only correcting for a discrete subset of errors.*

12.7 Stabilizer Formalism for Error Correction

12.7.1 Pauli Group

As seen before, the four Pauli operators, $\mathbb{I}, X, Z, Y$ allow us to express the four possible effects of the environment on a qubit. These operators form a group $G = \mathcal{P}$ and they exhibit several nice properties:

- Anticommuting
$$\{X, Z\} = \{Y, Z\} = \{Z, X\} = 0$$
- $P^2 = \mathbb{I}$, for all $P \in \mathcal{P}$
- Span the space of 2×2 matrices, describing the transformation of a single qubit.

Further, two Pauli matrices are equivalent if $\sigma_j = c\sigma_i$, where $c = \pm 1, \pm i$, which lets us define the set of equivalence classes of Pauli operators $[\mathcal{P}]$. Note that the set of Pauli operators, $\mathcal{P}$, is not an Abelian group. However, the set Π of equivalence classes, $[\mathcal{P}]$, of Pauli operators, also called the projective Pauli group, forms an Abelian group.

We further define the 1-qubit Pauli group, $\mathcal{P}_1$, which consists of the Pauli operators, $\mathbb{I}, X, Y, Z$, together with the multiplicative factors, $\pm 1, \pm i$, as

$$\mathcal{P}_1 := \{\pm\mathbb{I}, \pm i\mathbb{I}, \pm X, \pm iX, \pm Z, \pm iZ, \pm Y, \pm iY\}$$

whose cardinality is $|\mathcal{P}_1| = 2^4 = 16$. The members of the 1-qubit Pauli group are unitary, either commute or anticommute, and are either Hermitian or anti-Hermitian. Note that the generators of $\mathcal{P}_1$ are $\mathcal{G}_1 = \{X, Z, i\mathbb{I}\}$.

Generalizing our ideas, the n-qubit *Pauli group* $\mathcal{P}_n$ consists of the r^n tensor products $\mathbb{I}, X, Y, Z$ and an overall phase of $\pm 1, \pm i$, thereby, the group has 4^{n+1} elements. One element of the group is an n-tuple, the tensor product of n one-qubit Pauli operators, and can be used to describe the error operator applied to an n-qubit register. We define the *weight* of such an operator in $\mathcal{P}_n$ to be the number of tensor factors that are not equal to $\mathbb{I}$.

12.7.2 Stabilizer Subgroup

The stabilizer formalism is a concise way to describe a quantum error-correcting code using a set of quantum operators. Assume that the m codewords of a code are represented by the vectors $|\psi\rangle$, which are n-qubit registers. We have identified a set of q operators M_j that enable us to detect errors that may affect any of the codewords. In this context, the term *detect* refers to a measurement process that does not disclose any information about the actual state, but rather indicates whether the codeword has been affected by errors.

The stabilizer $\mathcal{S}$ of a quantum code is an Abelian subgroup of the n-qubit Pauli group, with generators $\{M_1, M_2, \dots\}$ where M_i stabilises the code words with positive eigenvalue. The codewords are thereby the eigenvectors satisfying $M_i|\psi_i\rangle = (+1)|\psi_i\rangle$. For an error $|\varphi_i\rangle = E_\alpha|\psi_i\rangle$ due to the error operator E_α, the stabilizers act as *syndrome* measurements with $M_i|\varphi_i\rangle = (-1)|\varphi_i\rangle$.

The *normalizer* $\mathcal{N}(\mathcal{S})$ is the set of elements that fix the stabilizer code under conjugation. Further, $\mathcal{S} \subset \mathcal{N}(\mathcal{S})$ is a normal subgroup. Since the elements of the normalizer $\mathcal{N}(S)$ move codewords around in the code space, they are the *encoding* operators. Only the elements $E \in \mathcal{N}(\mathcal{S})\backslash\mathcal{S}$ act on the codewords nontrivially.

The *centralizer* $\mathcal{C}(\mathcal{S})$ is the set of elements in $\mathcal{P}_n$ that commute with all the elements of the stabilizer $\mathcal{S}$. Since $\mathcal{S}$ can be shown to be an Abelian subgroup, it can be shown that the normalizer equals the centralizer, $\mathcal{N}(\mathcal{S}) = \mathcal{C}(\mathcal{S})$.

12.7.3 Quantum Error Correction Condition in stabilizer Formalism

Consider $|\psi_i\rangle$ as a codeword, with a series of operators M, that stabilize code words, such that

$$M|\psi_i\rangle = (+1)|\psi\rangle$$

The errors, $E = \{E_1, E_2, \dots\}$, affecting a codeword are also a subgroup of the n-qubit Pauli group, $E \in \mathcal{P}_n$, with each error operator E_i being a tensor product of n Pauli matrices. The *weight* of an error operator is equal to the number of errors affecting a quantum word, thus, the number of Pauli operators other than $\mathbb{I}$ in this n-dimensional tensor product. Note that the error operators anticommute with the generators of the stabilizer group $\mathcal{S}$. That is,

$$M(E|\psi_i\rangle) = (-1)E(M|\psi_i\rangle)$$

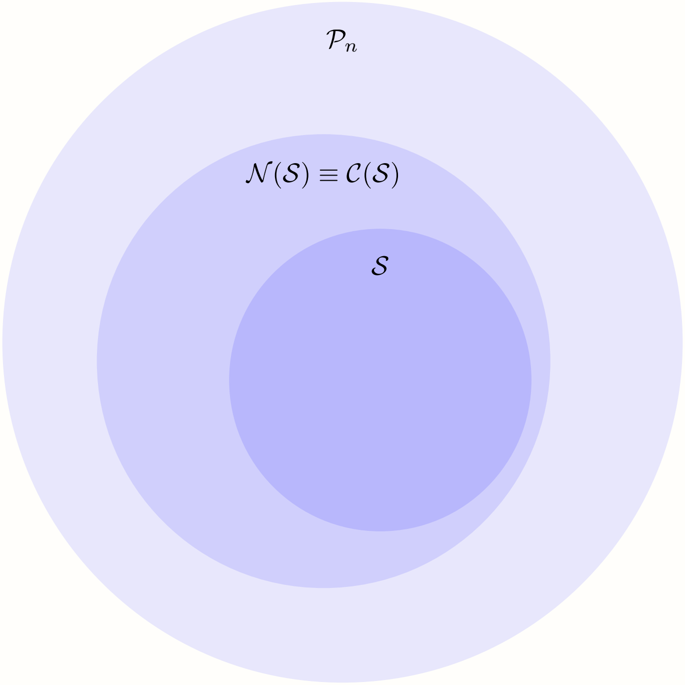


Figure 12.11: Error pattern classification for stabilizer codes. Correctable Errors: $E \in \mathcal{P}_n \backslash \mathcal{N}(\mathcal{S}) \equiv \mathcal{P}_n \backslash \mathcal{C}(\mathcal{S})$. Non Detectable Errors: $E \in \mathcal{C}(\mathcal{S}) \backslash \mathcal{S}$.

since $\{M, E\} = 0$. Therefore, to detect errors, we have to compute the eigenvectors of the generators and identify those with an eigenvalue of -1.

The ability to correct errors can be quantified by the *code distance*, d, defined as the largest weight d such that we have

$$\langle\psi_i|O|\psi_j\rangle = c_E\delta_{ij}$$

as effectively, the smallest possible weight of an operator $O \in \mathcal{P}_n$, such that, the weight of E violates the above condition of orthonormality. For $i \neq j$, this corresponds to the smallest weight such that $|\psi_i\rangle$ and $|\psi_j\rangle$ are not distinguishable, that is, orthogonal. Intuitively, the code distance measures how far one basis state in the code space is away from another basis state, hence the name distance.

For the stabilizer $\mathcal{S}$, with $n-k$ generators, then it enocdes k qubits, with code distance d, where d is the smallest weight of a Pauli operator in $\mathcal{N}(\mathcal{S})\backslash\mathcal{S}$. We thereby, compactify, as a $[n, k, d]$ stabilizer code, with n being the length of a codeword, k the number of information symbols, and d being the distance of the code, such that the cardinality of the stabilizer is $|\mathcal{S}| = 2^{n-k}$, and the generator M, has $|M| = n-k$. Rather than specifying the entire group, we only need its $n-k$ independent generators. Each generator imposes a constraint that effectively halves the dimension of the available Hilbert space. Thus, starting with n physical qubits (a 2^n dimensional space), these $n-k$ constraints define a 2^{n-k} times smaller subspace of dimension 2^k, perfectly suited to encode k logical qubits.

The code's power is its distance d, which measures its resilience. An error E is detected if it anti-commutes with a stabilizer, producing a measurable syndrome. An error is undetectable if it commutes with all stabilizers. The set of all such commuting operators is the normalizer, $\mathcal{N}(\mathcal{S})$.

Undetectable errors fall into two categories: *Trivial Errors*: If the error E is an element of the stabilizer itself ($E \in \mathcal{S}$), it is harmless as it leaves the code states unchanged; *Logical Errors*: If the error E is in the normalizer but not the stabilizer ($E \in \mathcal{N}(\mathcal{S})\backslash\mathcal{S}$), it acts as a non-trivial operation on the encoded logical qubits (e.g., a logical bit-flip $\overline{X}$). This corrupts the information without being flagged. The code distance d is therefore the minimum weight (the number of qubits it acts on non-trivially) of an operator in this set of dangerous logical errors, $\mathcal{N}(\mathcal{S})\backslash\mathcal{S}$. This is the smallest error that can silently damage the encoded data.

12.8 Toric Code

Imagine a square lattice, but with its end edges identified and seen as a torus. This is extremely similar to periodic boundary conditions in other aspects of physics, where we exploit the periodicity of the square lattice with endpoints identified. We can imagine assigning operators along the individual points of the lattice. Fig. 12.12 represents the toric code setup, where solid lines gives the lattice, and on each edge of the lattice lies a blue dot which represents a qubit. For an $n \times n$ lattice, we have $2n^2$ qubits since we can imagine each square hosting $\frac{1}{2} \times 4$ qubits, summing on each of its edges.

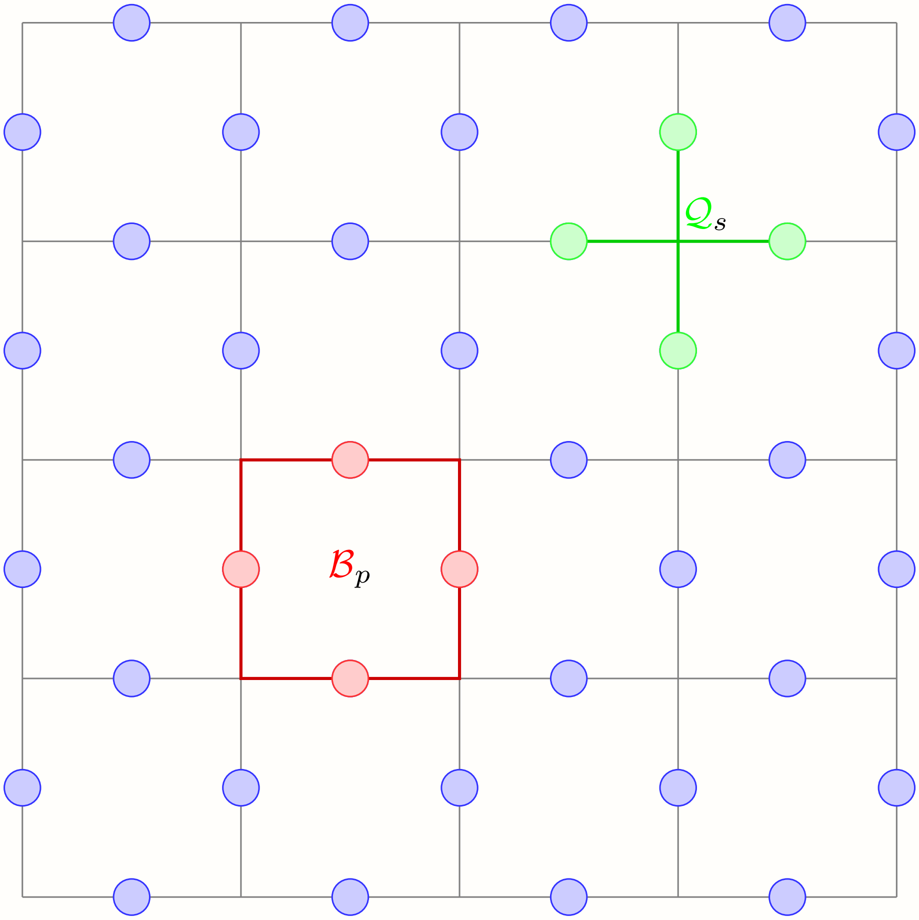


Figure 12.12: Toric Code

We define two types of stabilizer generators on the lattice as, Star operators $\mathcal{Q}_s$ and Plaquette operators $\mathcal{B}_p$ as

$$\mathcal{Q}_s = \prod_{j \in \text{Star}(s)} X_j, \quad \mathcal{B}_p = \prod_{j \in \text{Plaquette}(p)} Z_j,$$

where we note that $\mathcal{Q}_s$ and $\mathcal{B}_p$ individually commute, and also commute with each other for any pair of s, p. This can be explicitly seen through noting

$$\begin{aligned}\left[\mathcal{Q}_s, \mathcal{Q}_{s'}\right] &= \left[\prod_{j \in \text{Star}(s)} X_j, \prod_{k \in \text{Star}(s')} X_k\right] \\ &= \prod_{j \in \text{Star}(s), k \in \text{Star}(s')} \left[X_j, X_k\right] = 0,\end{aligned}$$

since the individual Z operators are independent on different locations s and s' given by Z_j and Z_k respectively, and the commutator product simplifies. Similarly,

$$\begin{aligned}\left[\mathcal{B}_p, \mathcal{B}_{p'}\right] &= \left[\prod_{j \in \text{Plaquette}(p)} Z_j, \prod_{k \in \text{Plaquette}(p')} Z_k\right] \\ &= \prod_{j \in \text{Plaquette}(p), k \in \text{Plaquette}(p')} \left[Z_j, Z_k\right] = 0.\end{aligned}$$

Further, we note the interesting relation,

$$\begin{aligned}\left[\mathcal{Q}_s, \mathcal{B}_p\right] &= \left[\prod_{j \in \text{Star}(s)} X_j, \prod_{k \in \text{Plaquette}(p)} Z_k\right] \\ &= \prod_{j \in \text{Star}(s), k \in \text{Plaquette}(p)} \left[X_j, Z_k\right]\end{aligned}$$

where we analyse the cases. When X_j and Z_k are widely separated as in Fig. 12.12, we note the independency and thereby, $\left[X_j, Z_k\right] = 0$. But, when we have overlapping cases for X_j and Z_k as in Fig. 12.13. Here, note that there always exist two overlaps, say i_1 and i_2. Hence, we have the product $\left[\mathcal{Q}_s, \mathcal{B}_p\right] \sim \left[\ldots X_{i_1} X_{i_2} \ldots, \ldots Z_{i_1} Z_{i_2} \ldots\right]$, where these overlaps cancel each other out through their commutation relations. Since the overlap between the star and plaquette operator always ends up in overlap on even number of qubits, it is easy to show that $\left[\mathcal{Q}_s, \mathcal{B}_p\right] = 0$.

Thereby, it is interesting to note that there are n^2 Star operators defined at each $\mathcal{Q}_s$ for the X operator, and similarly, n^2 Plaquette operators at each $\mathcal{B}_p$ for the Z operator. But these aren't all independent and are connected by a simple relation. Note that, in the product of all Star and Plaquette operators, we encounter the X and Z respectively twice since adjacent stars and neighbouring Plaquettes share a common qubit. This results in products of $X^2 = \mathbb{I}$ and $Z^2 = \mathbb{I}$, across all qubits, resulting in

$$\prod_s \mathcal{Q}_s = \prod_s \left(\prod_{j \in \text{Star}(s)} X_j\right) = \mathbb{I}, \quad \prod_p \mathcal{B}_p = \prod_p \left(\prod_{j \in \text{Plaquette}(p)} Z_j\right) = \mathbb{I}.$$

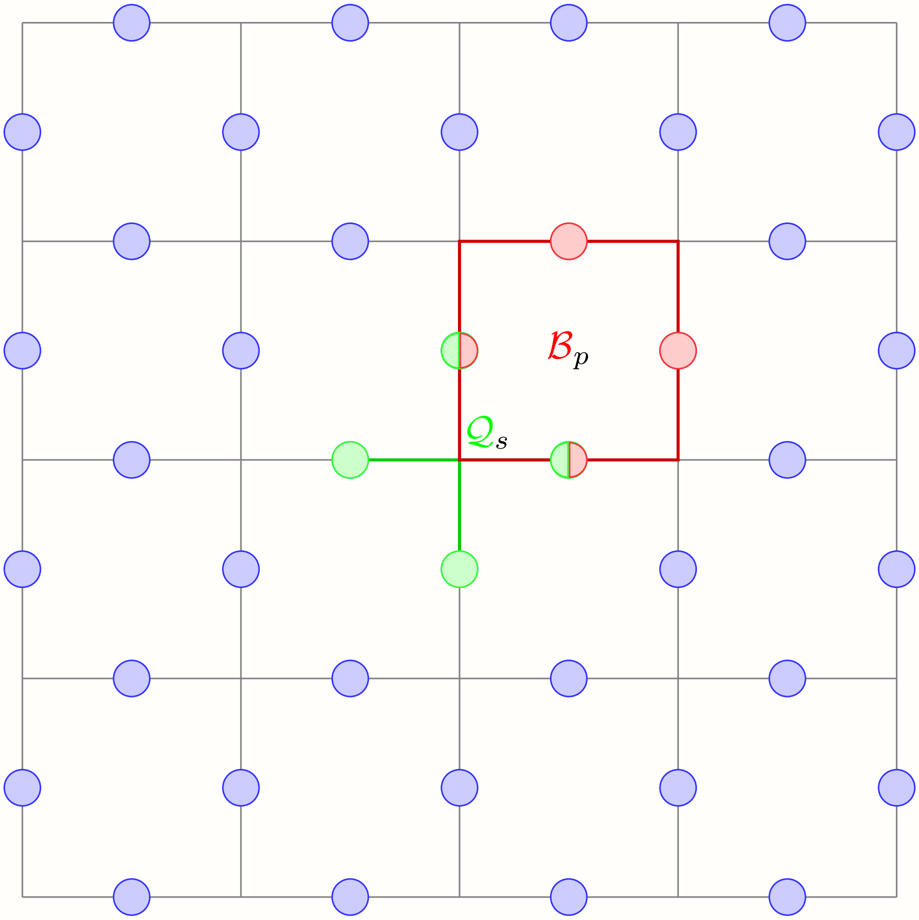


Figure 12.13: Overlapping Toric Code

Thereby, we have one of the Star and Plaqquete operators being determined through the others, leading to $2(n^2 - 1) = 2n^2 - 2$ independent stabilizer operators in total. Thereby, we encode in dimension $2^{2n^2-(2n^2-2)} = 2^2$, that is, we encode 2 qubits into $2n^2$ qubits.

We further define the set of logical operators as cycles on the torus, as in Fig. 12.14, defined as cycles since the edges are identified. Precisely, we have

$$\mathcal{X}_i = \prod_{s\in\text{Vertical}(i)} \mathcal{Q}_s, \quad \mathcal{Z} = \prod_{p\in\text{Horizontal}(i)} \mathcal{B}_p,$$

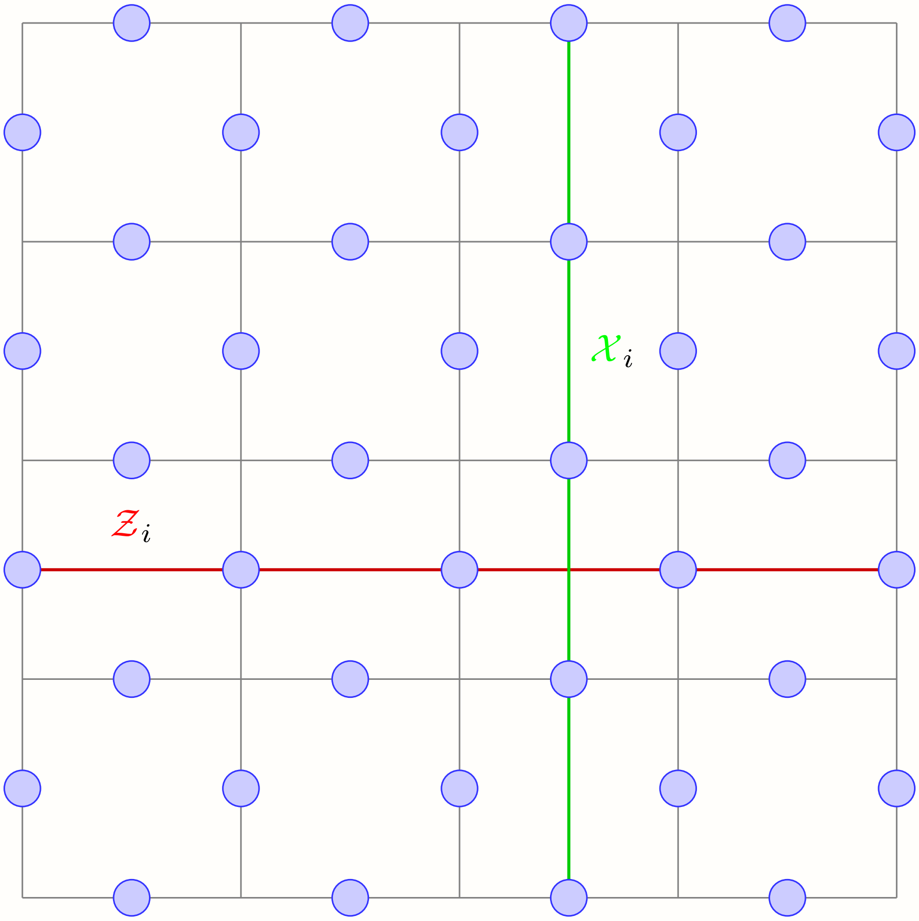


Figure 12.14: Operators on the Torus

The size of the logical operators encodes the property of the code distance, which is seen to be n since there are n independent qubits in each of the definitions of the cyclic operators.

We see that the maximum weight of the cyclic operators described above is along the length of the torus, such that we have code distance n, which scales with the square root of the number of qubits encoded.

12.9 Quantum Hamming Bound

A natural question to ask is how efficiently we can make the error correction code in terms of the number of physical qubits used. Can we have an arbitrarily small number of physical qubits? If not, what is the smallest number of physical qubits needed?

To answer the above questions, let's suppose a non-degenerate code is used to encode k-qubits in n-qubits such that it can correct errors on any subset of t or fewer qubits. Suppose $j \leq t$ errors occur, then the number of possible locations where this can happen is $\binom{n}{j}$. If in each of these locations, the errors can be one of X, Y, or Z, then the total number of errors that may occur on t or fewer qubits is $\sum_{j=0}^{t} \binom{n}{j} 3^j$.

As the code is non-degenerate, to encode k qubits we need a 2^k-dimensional space. Thus, each of the above errors must correspond to an orthogonal 2^k-dimensional subspace. As we are encoding with n qubits, this number should be less than 2^n, the dimension of the physical qubit states. Therefore, we have,

$$\sum_{j=0}^{t} \binom{n}{j} 3^j 2^k \leq 2^n$$

The above condition is known as the *quantum Hamming bound.*

For correcting a single qubit error, we have $k = 1$ and $t = 1$. On substituting this, the quantum Hamming bound gives us $n \geq 5$. Thus, to answer the question asked at the start of this section, to correct an arbitrary single-qubit error, we need at least 5 physical qubits. Do we have an error correction code that works on exactly 5 physical qubits? The answer turns out to be yes![2]

[2] To know more about the 5-qubit code, refer to the textbook: Quantum Computing and Quantum Information by Isaac Chuang and Michael Nielsen

References

[1] Scott Aaronson. *Quantum computing since Democritus.* Cambridge University Press, 2013.

[2] Giuliano Benenti, Giulio Casati, and Giuliano Strini. *Principles of quantum computation and information: Basic tools and special topics*, volume 2. World Scientific, 2004.

[3] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Phys. Rev. Lett.*, 70:1895–1899, Mar 1993.

[4] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

[5] Emmanuel Desurvire. *Classical and quantum information theory: an introduction for the telecom scientist.* Cambridge university press, 2009.

[6] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[7] Richard P Feynman. Simulating physics with computers. In *Feynman and computation*, pages 133–153. cRc Press, 2018.

[8] Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation.* American Mathematical Soc., 2002.

[9] Fang Xi Lin. Shor's Algorithm and the Quantum Fourier Transform. *Lecture notes*, 2013.

[10] Manindra, Agrawal and Neeraj, Kayal and Nitin, Saxena. Primes is in p. *Ann. of Math*, 2(781–793), 2002.

[11] Dan C Marinescu. *Classical and quantum information.* Academic Press, 2011.

[12] N David Mermin. *Quantum computer science: an introduction.* Cambridge University Press, 2007.

[13] Gary L. Miller. Riemann's Hypothesis and Tests for Primality. *Journal of Computer and System Sciences*, 1976.

[14] Rajat Mittal. *Lectures on Quantum Computing.* IIT Kanpur, 2023.

[15] M. A. Nielsen. Conditions for a Class of Entanglement Transformations. *Phys. Rev. Lett.*, 83:436–439, Jul 1999.

[16] M.A. Nielsen and I.L. Chuang. Quantum Computation and Quantum Information. *Cambridge University Press*, 10th Anniversary Edition, 2011.

[17] John Preskill. Lecture notes for physics 229: Quantum information and computation. *California institute of technology*, 16(1):1–8, 1998.

[18] Thomas E Roth, Ruichao Ma, and Weng C Chew. An introduction to the transmon qubit for electromagnetic engineers. *arXiv preprint arXiv:2106.11352*, 8:18–72, 2021.

[19] Schönhage and V. Strassen. Concentration inequalities. *Schnelle Multiplikation grosser Zahlen, Computing*, 7(281–292), 1971.

[20] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, 26(5):1484–1509, October 1997.

[21] Urbasi Sinha, Christophe Couteau, Thomas Jennewein, Raymond Laflamme, and Gregor Weihs. Ruling out multi-order interference in quantum mechanics. *Science*, 329(5990):418–421, 2010.

[22] Robert Sutor. *Dancing with qubits.* Packt Publishing Birmingham, UK, 2019.

[23] Mahesh T S. *PH4323 / PH 6543 Quantum Information.* IISER Pune, 2024.

[24] Bei Zeng, Xie Chen, Duan-Lu Zhou, Xiao-Gang Wen, et al. *Quantum information meets quantum matter.* Springer, 2019.