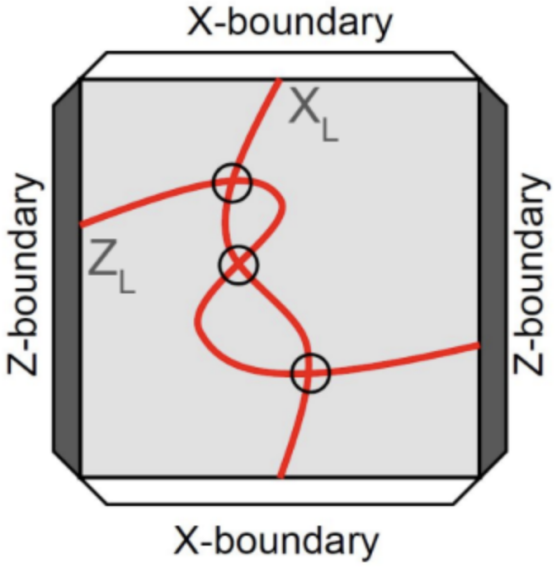
Topological quantum error correction: a friendly introduction

Sam Burdick



α	alpha
β	beta
δ	delta
Γ, γ	gamma
Θ, θ	theta
ϕ, φ	phi
ψ	psi
σ	sigma
Ω, ω	omega
$\|\psi\rangle$	physical qubit
$\|\psi\phi\rangle = \|\psi\rangle \|\phi\rangle$	qubit product
$\|\psi_L\rangle$	logical qubit
$\binom{\alpha}{\beta}$	two-dimensional state vector
$C(n, k)$	binomial coefficient: "n choose k"
$\in$	set membership
$\otimes$	Kronecker product
$\{\}$	empty set
$\mathbb{Z}$	integers
$\mathbb{R}$	real numbers
$\mathbb{C}$	complex numbers
i	imaginary unit $\sqrt{-1}$
Σ	discrete summation
$\int_a^b, \int_{\mathcal{R}}$	Riemann integral
I	identity matrix
X, Y, Z	Pauli matrices
H	Hadamard gate
$\|\cdot\|$	set cardinality; complex modulus
$\times$	Cartesian product; dimensionality
∂A	boundary of the point set A
$\bigcup, \bigcap$	union, intersection of sets

Version 0.3

The musical notation and quantum circuit example were taken respectively from the tutorials [1] and [2].
Cover art by Austin G. Fowler.
The topographical map in the preface was created by MapBliss.
The pencil sketches were partially generated by the author using AI.
The fault tolerance illustration in section 12 was drawn by David Chiu.

Printed in the United States of America.

For the computer science community.

I graduated from the library when I was twenty-seven. I discovered that the library is the real school.
—Ray Bradbury

Contents

1 Preface

THIS work began its life as a series of notes, intended to be composed into a survey paper. It became clear early on that an entire *volume* was in order, due to the sheer amount of prerequisites. Since the bulk of quantum computing (and thus, quantum error correction) literature targets physicists, there is little that accommodates either the academic computer scientist or a practicing software engineer; this work aims to provide a solid grasp of the field for both camps.

Contemporary quantum computer hardware is notorious for being error-prone. To enable its scalability and general accessibility, error correction techniques must be applied; one such method, known as *topological quantum error correction* (TQEC), reliably detects minute hardware errors. This work offers an informal theoretical introduction to TQEC, followed by a discussion of classical software that supports it.

A note on the term *topological*: the prefix *topo-* appears throughout mathematics, and frequently convokes the term *topographical*, used in conjunction with cartography. Our usage of the term is more subtle: the Greek prefix *topos* ($\tau o \pi o \sigma$) means *place*, or locality. In that sense, we can think as a map in the abstract sense (such as a hash table), whereby we visit one "place" (key-value pair) at a time.

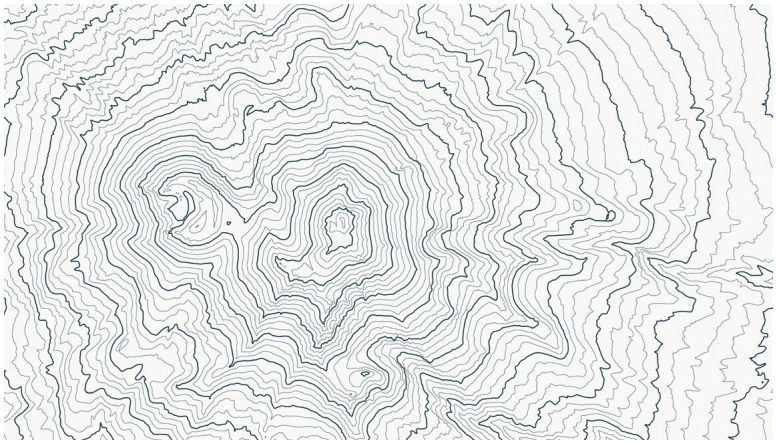
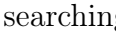


Figure 1: A topographical map of Mount Shasta.

In short, this work targets readers with an upper-undergraduate or graduate rank in computer science, and assumes background familiarity with high

school algebra, geometry and trigonometry, summation notation, asymptotic analysis (big-O), and graph algorithms. You may think this work as a toolbox that you can reopen at any point in your quantum error correction work; reading it all the way through in one sitting would be rather tedious.

The only way to truly add a tool to your kit is to practice using it. You will find various exercises strewn about this text, with approximate degrees of *spiciness* indicated:

-  : Immediately clear to a careful reader.
-  : Should take a few minutes of writing on scratch paper, or searching through other parts of the text.
-  : A good candidate for a take-home exam problem.
-  : A research endeavor of indeterminate scope.

Major results are demarcated as *theorems*; intermediate results that are important in their own right are called *lemmas*. We do not attempt to approve any such propositions borrowed from other texts.

> *To eat well, I always disagree with critics who say that all restaurants should be fine dining. You can get a Michelin star if you serve the best hamburger in the world.*
> —David Chang (2020)

2 To the student

This work is written for you: a computer science student interested in quantum error correction, and willing to learn a little math along the way. Due to the current job landscape, it's a somewhat common misconception that a physics degree is a prerequisite to quantum computing. The reality is that you do not need one to contribute to this burgeoning field.

The goal of this work is to enable such contributions. Think of it as a map: mathematically defined, and aiming to be useful; since the terrain is not the map; the author intends to regularly update this work to the best of their ability.

If you've ever written two nested for loops that together add up lists of numbers of length k, for $1 \leq k \leq n$, then at some point you may have been shown the mathematical way of explaining it:

$$\sum_{k=0}^{n} k = O(n^2) \tag{1}$$

Most likely, the math you learned beforehand was meant to teach you the fundamentals of *calculus*, which only uses such machinery in more advanced contexts. I'd like to assure you that the complexity of the mathematical statements herein are no more complex than equation and have a similar flavor. In conclusion, this is *not* a quantum mechanics text; it is a computer science text through and through.

3 To the instructor

It is the author's hope that this work provides a "boots on the ground" perspective into the realm of topological quantum error correction, and quantum computing in general. The upper-rank undergraduate or graduate student new to quantum computing should benefit from the clear and concise language used in this work. The subtitle *user manual* hints at one of the key motivators in producing this text, which is to assist users and developers of `TQEC.app`.

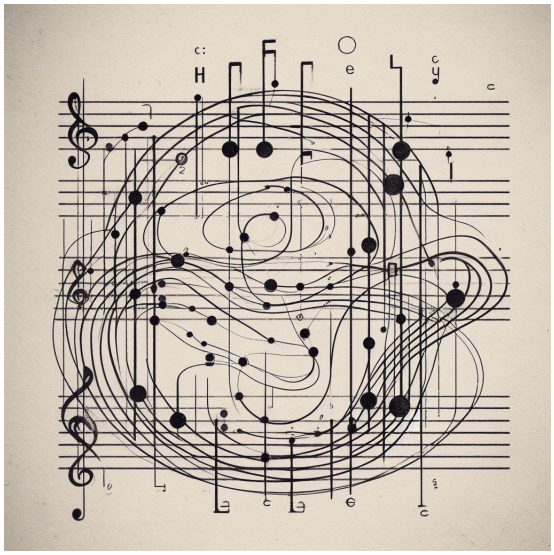
4 Acknowledgements

The author would like to thank Austin Fowler, for his community organization and mentorship; Unitary Fund, for their generous support of the `TQEC` open–source project; and those who have contributed to `TQEC.app`.

> *This is only possible when there are no global topological obstructions to this identification.*
> —Mina Aganagić, Costin Popescu, and John H. Schwarz [3]

> *An author cannot of course remain unaffected by the sum of his experiences...*
> —J. R. R. Tolkien (1954) [4]

> *It is our chief aim to entertain, then inform...*
> —Card Walker (1982)



> *Another good debugging practice is to keep a record of every mistake made. Even though this will probably be quite embarrassing, such information is invaluable to anyone doing research on the debugging problem, and it will also help you learn how to reduce the number of future errors.*
> —Donald Knuth (1997) [5]

5 Contribution guidelines

This is a work by one person, but could not have been possible without many; to list them all would simply be impossible. Moreover, as time goes on, I suppose members of the community will offer error corrections of their own. Maintaining this work as an open-source project is simply not possible, due to copyright concerns; however, to acknowledge any errors found in the work, one `TQECent` will be issued as payment for correctly identifying an error. Technical authors frequently offer bounties in the form of handwritten checks, or an equivalent exchange. To be more environmentally friendly, secure, and open to all, this work proposes a new cryptocurrency, `TQECoin`; a single block of which shall be more than ten times less carbon intensive than mailing a handwritten check via USPS.

Back-of-the napkin calculations reveal that the marginal carbon footprint of writing and mailing a check in the United States is 20g of CO_2. Assume that an Intel i7 8th gen/Nvidia RTX 2060 Super based computer can mine a single `TQECoin` block in 24 hours and performs no other tasks. This process uses roughly 2g of CO_2, making it viable to produce 100 `TQECent`s at the prescribed rate. The blockchain or any other data needn't be stored on a third-party service. Out of fairness, I cannot offer any other form of reward; if you would like to contribute without holding any cryptocurrency yourself; I will keep your cent and not sell it. All contributions will be listed on my website[1] for quick reference.

Happy hunting!

—S. B.

> *Harambee* ("Let us all pull together")
> —National motto of Kenya
>
> *Everything is difficult in the beginning.*
> —Chinese proverb

[1] `https://smburdick.github.io`

6 Introduction

Intense hardware error rates severely limit the practicality of quantum computers [6]. Research labs such as Google Quantum AI have made significant strides towards error tolerance in recent years, having produced a logical qubit prototype in 2023. However, Google does not expect to have a single *long-lived* logical qubit, comprising on the order of 1000 physical qubits, until 2025 at the earliest [7]. While these figures may sound discouraging, several promising avenues for correcting qubit errors remain; this work discusses improved noise resilience through *topological quantum error correction* (TQEC).

This work is organized as follows. We begin by introducing the key mathematical concepts for understanding TQEC, such as Pauli errors, stabilizers, and *planar codes*.[2] Next, we discuss syndrome analysis, a set of classical algorithms used to fully implement the error model. To the author's knowledge, this work is the first survey of TQEC written with the chief aim of assisting those with a computer science background interested in the domain of quantum error correction. It is our intention to develop a practical and *ground-up* understanding of the planar code.

The initial sections of this work provide some background for readers interested in quantum error correction, but are perhaps unfamiliar with the quantum computing notation, or in need of a mathematical refresher. For a deeper look into the basics, a free Coursera course is under development [8], and two helpful textbooks are [9] and [10]. For a deeper dive into the field of quantum error correction, see [11]. It is the author's hope that this work will serve as a learning tool for those interested in contributing to TQEC through further research and development.

[2] Also known as a *surface code* in the literature. Confusingly, other kinds of codes are referred to as such; think in terms of *polymorphism*.

7 Mathematical preliminaries

I strongly recommend reviewing the table of contents and circling the names of sections that are unfamiliar to you, and focusing on those.

7.1 Set theory

The most basic mathematical object is called a *set*, which is simply a collection of distinct objects. For example,

$$\{\triangle, 42, e\} \tag{2}$$

is a set. It's not a terribly useful one. The size (or *cardinality*) of a set is denoted $|\cdot|$.

The most widely-referenced set of all is possibly $\mathbb{Z}$, which are commonly known as "whole numbers" but are better known within the world of coding as *integers*:

$$\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\} \tag{3}$$

Another surprisingly useful set is the empty set $\{\}$ (sometimes denoted $\varnothing$), which contains no elements; that is, $|\{\}| = 0$.

7.1.1 Unions and intersections

The union of two sets is defined as:

$$A \bigcup B = \{x : x \in A \text{ or } x \in B\}. \tag{4}$$

The intersection is

$$A \bigcap B = \{x : x \in A \text{ and } x \in B\}. \tag{5}$$

The *difference* of two sets is denoted $\setminus$:

$$A \setminus B : \{x : x \in A \text{ and } x \notin B\}. \tag{6}$$

Exercise 1. *Show that the natural numbers $\mathbb{N} = \{1, 2, 3\ldots\}$ is equivalent to $\{|a| : a \in \mathbb{Z} \setminus \{0\}\}$.*

Exercise 2. *(Addition principle) Show that if A and B are disjoint, that is, $A \bigcap B = \{\}$, then*

$$|A \bigcup B| = |A| + |B|. \tag{7}$$

7.1.2 Cartesian products

The Cartesian product of two sets A and B is defined as

$$A \times B = \{(a, b) : a \in A, b \in B\}. \tag{8}$$

Sometimes, we denote $A \times A$ as A^2.

Exercise 3. 🌶🌶🌶 *(Rule of product) Show that* $|A \times B| = |A||B|$.

7.1.3 Functions between two sets

We can say that the function $f(x) = x^2$ has a *signature*, written as

$$f : \mathbb{R} \to \mathbb{R}_{\geq 0}; \tag{9}$$

the *domain* appears on the left-hand side of the $\to$, followed by the *range* on the right.

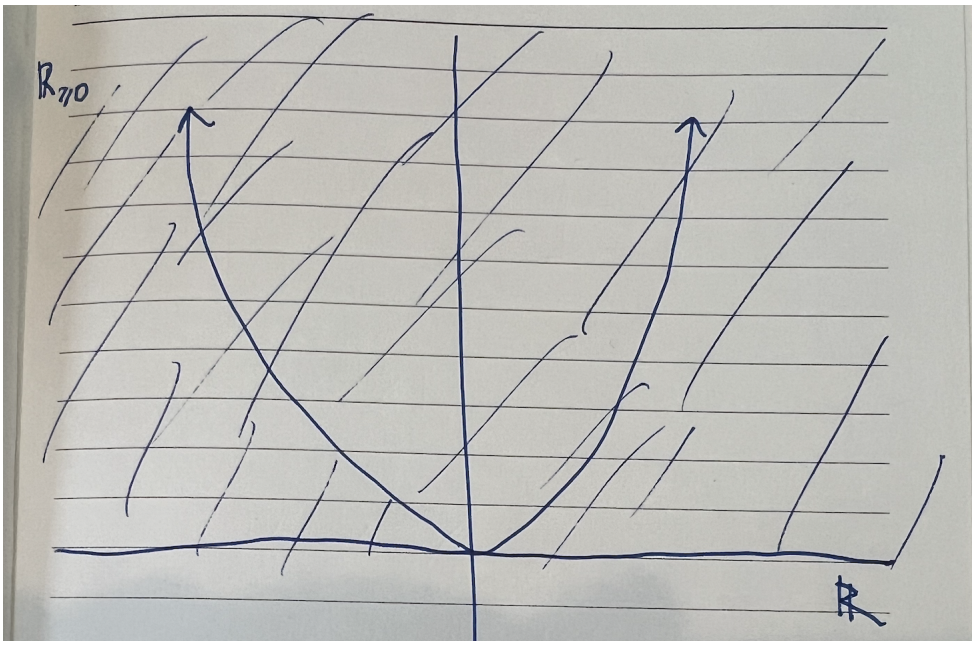


Figure 2: The mapping $f : \mathbb{R} \to \mathbb{R}^2$.

7.2 Probability theory

Rather than rigorously defining the axioms of probability, let's set down a few definitions. A probability is a real number p, restricted to the range $[0, 1]$, that represents the outcome of a particular event E in a controlled setting where Ω is the set of all possible events. For example, when a fair coin is tossed, we have $\Omega = \{\texttt{HEADS}, \texttt{TAILS}\}$, and $p(\texttt{HEADS}) = p(\texttt{TAILS}) = 1/2$. If we wanted to get heads all the time, we might add a little more weight to the underside; experimental data might show us that $p(\texttt{HEADS})$ has increased to $2/3$; we can deduce that $p(\texttt{TAILS})$ has decreased to $1/3$ by this formula:

$$\sum_{E \in \Omega} p(E) = 1 \tag{10}$$

If two events A and B are *independent*, then the probability of both happening is $p(A)p(B)$, and the probability of either of them happening is $p(A) + p(B)$.

Exercise 4. 🌶🌶🌶 *What is the probability of a monkey correctly typing a syntactically correct Hello, World program on its first try?*

Exercise 5. 🌶🌶🌶 *Prove the law of total probability: that for any discrete set of events comprising A and B_n $(n \geq 0)$, we have $P(A) = \sum_n P(A \cap B_n)$.*

Exercise 6. 🌶🌶🌶 *The binomial theorem states that, for any $n \geq 0$,*

$$(x + y)^n = \sum_{k \geq 0} C(n, k) x^{n-k} y^k, \tag{11}$$

where $C(n, k) = \frac{n!}{(n-k)!k!}$

- *For which values of x and y can we compute p^n?*
- *If p represents a probability, what is the meaning of the binomial theorem? (Hint: it has to do with how $C(n, k)$ is interpreted.)*
- *What happens if we let n go to infinity?*

Exercise 7. 🌶🌶🌶 *The expectation value of a real-valued function $f(x)$ is denoted*

$$\mu(f) = \int_{-\infty}^{\infty} x f(x) dx. \tag{12}$$

- *What is the expectation value of e^{-x^2}?*

- *Suppose we are given a complex-valued function $f(z)$. What function would you need to multiply the it by to produce a real-valued function $\hat{f}(x)$? Use your answer to compute $\mu(\hat{f})$.*

  Since all models are wrong the scientist must be alert to what is importantly wrong.
  —George E. P. Box (1976) [12]

7.3 The field of numbers

With those exciting definitions out of the way, it's time for a brief an-historic interlude into the realm of *number fields*. Recall that the symbol $\subset$ is a binary operator pronounced "subset". Ready?

$$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C} \tag{13}$$

The American high school mathematics curriculum requires you to learn about *all* of these; a working mathematician would tell you, though, that all of these sets have the same structure. So how do we make sense out of Equation 13? Working from left to right, we have $\mathbb{N}$, the "natural" numbers, $1, 2, 3 \ldots$. The integers are a little more interesting, since they give us 0 and negative naturals. Then we get the "rational" numbers $\mathbb{Q}$, for "quotient": these are analogous to `IEEE-754` floating point numbers. So what is $\mathbb{R}$?

The reality is that there is no good definition for $\mathbb{R}$: the "real" numbers. At this point, we leave behind the comfortable chair of software development and approach the chalkboard of a physics laboratory. A real number can be most succinctly defined as the result of a measurement. For example, let's say you wanted to measure the weight of a basketball, without bothering to order one (and possibly a bathroom scale) from Amazon. Your gym class experience would tell you that the radius is about one foot (or one-third of a meter). A quick search would tell you the volume of a sphere is $4\pi r^2$, giving you an exact answer of $4\pi/9$ m^3.

But there's that finicky number in there: π. What is its nature? Is it a rational number? Somebody in ancient Greece didn't think so, and for that they paid the ultimate price. (Yes, this really happened. Kind of.) After the funeral, the Athenian academicians convened to introduce a new set of numbers to reconcile themselves to their confounding reality: there are numbers that transcend human intuition. Since "transcendent" hadn't been coined yet, they ended up going with "real," since they came up with it by solving a geometry problem (the ratio of a circle's circumference to a diameter).

In fact, the ancient Greeks only knew π as an integer. Only in 1400 was it known to ten places, and today it can be calculated to an arbitrary precision with fancy formulae such as

$$\frac{1}{\pi} = \frac{\sqrt{2}}{2} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{k!^4 (396)^{4k}}, \tag{14}$$

discovered by Ramanujan just over a century ago. What a time to be alive!

Exercise 8.  *Show that* $\mathbb{Z} = \{\pm n : n \in \mathbb{N}\} \bigcup \{0\}$

Exercise 9.  *Show that* $\mathbb{R} = \mathbb{C} \setminus \{ai : a \in \mathbb{R}\}$.

> The Moon is handed down by memory to be eleven thousand yojanas in diameter. Its peripheral circle happens to be thirty three thousand yojanas when calculated. (Bhishma Parva of the Mahabharata, 6.12.40)

7.4 Complex variables

You may remember being asked to find the solutions to polynomial equations, such as $x^2 + 1 = 0$. To solve for x, simply subtract 1 from both sides and... take the square root of both sides. We're left with $x = \pm\sqrt{-1}$. (The $\pm$ is added since squaring a negative number is positive.) Mathematicians struggled at this juncture for centuries, but in the early 1800s settled on a new symbol to indicate the "imaginary" unit: $i = \sqrt{-1}$ [13].

Complex numbers are thus compactly defined as:

$$\mathbb{C} = \{a + bi : a, b \in \mathbb{R}, i = \sqrt{-1}\}. \tag{15}$$

Just like real numbers, complex numbers can be added, multiplied, and so forth to produce new complex numbers.

Sometimes, we care about the distance between a complex number and the origin $(0, 0)$. The *modulus* of a complex number $z = a + bi$ is equal to

$$|z| = \sqrt{a^2 + b^2}. \tag{16}$$

Geometrically, it is the distance from the origin $(0, 0)$ to the point (a, b) in Cartesian coordinates, as illustrated in Figure 3.

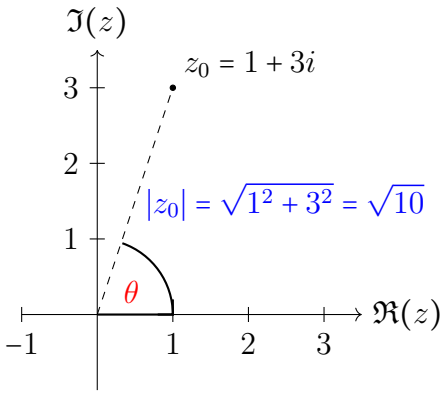


Figure 3: An illustration of the complex modulus (denoted $|z_0|$) against the *complex plane*; the horizontal axis represents the "real part" of a complex number z, denoted $\mathfrak{R}$, while the horizontal axis represents the imaginary part, denoted $\mathfrak{I}$.

The *complex conjugate* of $z = a + bi$ is $z^* = a - bi$; simply flip the $+$ or $-$ sign in front of the imaginary part. The *argument* of z is the angle subtended by

the line of length $r = |z|$ and the positive $\Re$ axis; in figure 3, it's denoted as θ. We may now define a complex number in its *polar form*:

$$z = re^{i\theta} = r(\cos\theta + i\sin\theta) \tag{17}$$

where $r = |z|$, and $e = 2.71828\ldots$ is a very special number. Letting $r = 1$ and $\theta = \pi$, we're left with *Euler's identity*

$$e^{i\pi} + 1 = 0, \tag{18}$$

onto which many have ascribed an aesthetic quality; our goal is to put it to use in describing qubits and gate actions.

Exercise 10. 🌶🌶🌶 *Show that $zz^* = |z|$ for any $z \in \mathbb{C}$.*

Exercise 11. 🌶🌶🌶 *Prove de Moivre's theorem: for any $z \in \mathbb{C}$ and $n \in \mathbb{N}$,*

$$z^n = r^n(\cos n\theta + i\sin n\theta) \tag{19}$$

Exercise 12. 🌶🌶🌶 *Prove the well-known identity*

$$\int_{-\infty}^{\infty} e^{-x^2}dx = \sqrt{\pi}.$$

Hint 1: Multiply the left-hand-side by itself. Hint 2 (for habanero lovers): use Cauchy's integral formula: $\oint_C f(z)dz = 0$.

Exercise 13. 🌶🌶🌶 *Prove that if*

$$\Gamma(z) = \int_0^{\infty} t^{z-1}e^{-t}dt \qquad \Re(z) > 0$$

then $\Gamma(z) = z\Gamma(z-1)$.

Exercise 14. 🌶🌶🌶 *The Taylor series of the real-valued function e^x is given as*

$$e^x = \lim_{n\to\infty}\sum_{k=0}^{n}\frac{x^k}{k!} \tag{20}$$

- *If $x, n \in \mathbb{N}$, how can you interpret the right-hand side?*
- *Is the series still valid for $x \in \mathbb{C}$? How do you know?*

We must therefore conclude that the square root of a negative number cannot be either a positive number or a negative number, since the squares of negative numbers also take the sign plus: consequently, the root in question must belong to a entirely distinct species of numbers, since it cannot be ranked either among positive or negative numbers.
—Leonhard Euler (1770) [14]

Indeed, if a mathematician is asked to justify his interest in complex numbers, he will point, with some indignation, to the many beautiful theorems...
—Eugene Wigner (1960) [15]

7.5 Linear algebra

> *We have tried to make this text as helpful as possible with this transition. Every definition is stated carefully, set apart from the text.*
> —Robert Beezer (2012) [16]

7.5.1 Matrices

Matrices are multi-dimensional arrays of numbers. For our purposes we shall focus on *square* matrices, which have the same number of rows and columns. Our first example of one, a 2x2 matrix, can be expressed as

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}. \tag{21}$$

You can multiply two matrices together, using the formula

$$A_1 A_2 = \begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix} \begin{pmatrix} a_2 & b_2 \\ c_2 & d_2 \end{pmatrix} = \begin{pmatrix} a_1 a_2 + b_1 c_2 & a_1 b_2 + b_1 d_2 \\ c_1 a_2 + d_1 c_2 & c_1 b_2 + d_1 d_2 \end{pmatrix}.$$

Importantly, for any matrices A and B, you should expect that $AB \neq BA$; they are *non-commutative* with respect to multiplication. Notably, A and B can also be *anti-commutative*, which would mean $AB = -BA$. We'll now defy your expectations by showing off a widely-used matrix known as the *identity matrix*:

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

Multiplying any matrix A by I produces A:

$$AI = IA = A$$

Exercise 15. 🌶🌶🌶 *The inverse of a 2x2 matrix is defined as*

$$A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \tag{22}$$

Show that $AA^{-1} = I$.

Exercise 16. 🌶🌶🌶 *A commonly used matrix operator in quantum mechanics is the trace, denoted* Tr *which can be defined in terms of a matrix's indices:*

$$\mathrm{Tr}(A) = \sum_{i=1}^{n} A_{i,i} = A_{1,1} + A_{2,2} + \cdots + A_{n,n} \tag{23}$$

Compute the trace of each example matrix in this section.

Exercise 17. 🌶🌶🌶 *Show that*

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

rotates a vector in $\mathbb{R}^2$ *by* θ *counterclockwise.*

Exercise 18. 🌶🌶🌶 *The Hadamard product, frequently used in machine learning applications, and denoted here as* $(\cdot\cdot)$*, is the component-wise product of two matrices, such that* $(AB)_{i,j} = (A)_{i,j}(B)_{i,j}$ *for all valid* i *and* j*. Write an equation that explicitly defines the elements of* (AB)*.*

> *It is the world that has been pulled over your eyes to blind you from the truth...*
> —Morpheus, *The Matrix* (1999)

7.5.2 The Kronecker product

The Kroenecker product (also referred to as the tensor product) of two matrices is different from multiplication in that it produces a matrix with different dimensions.

$$A \otimes B = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \otimes \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} = \begin{pmatrix} a_1b_1 & a_1b_2 & a_2b_1 & a_2b_2 \\ a_1b_3 & a_1b_4 & a_2b_3 & a_2b_4 \\ a_3b_1 & a_3b_2 & a_4b_1 & a_4b_2 \\ a_3b_3 & a_3b_4 & a_4b_3 & a_4b_4 \end{pmatrix}. \tag{24}$$

To describe things verbally, we're packing the second argument into the first argument.

> *God made the natural numbers; all else is the work of man.*
> —Leopold Kronecker (late 19th cent.) [17]

7.5.3 The commutator

The conditions under which matrices commute are so widely discussed in the realm of quantum computing, it's worth introducing a new symbol used to buttress the discussion. If X and Y are square matrices, the *commutator* of both is

$$[X, Y] = XY - YX. \tag{25}$$

In the event that X and Y commute, we have

$$[X, Y] = XY - YX = XY - XY = \mathcal{O}, \tag{26}$$

where $\mathcal{O}$ is a matrix with all zeroes; we will substitute it for 0 in this text for simplicity.

Exercise 19. 🌶🌶🌶 *Show that* $[A, I] = [I, A]$ *for any matrix* A.

> *Building a road might create temporary jobs, but does it really create wealth if it doesn't also shorten commute times or otherwise make society better off?*
> —Myron Scholes

7.5.4 Vectors

A vector is a one-dimensional array. We focus on vectors with two numeric entries α and β, denoted

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \tag{27}$$

The symbol $|\cdot\rangle$ is known as a *ket* in quantum computing literature. We can multiply a vector by a numeric value λ like so:

$$\lambda\,|\psi\rangle = \lambda \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \lambda\alpha \\ \lambda\beta \end{pmatrix} \tag{28}$$

Finally, we can multiply a matrix times a vector, which gives back a new vector:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} a\alpha + b\beta \\ c\alpha + d\beta \end{pmatrix} \tag{29}$$

Exercise 20. 🌶🌶🌶 *Compute*

$$\begin{pmatrix} 2 & 3 \\ 5 & 6 \end{pmatrix} \begin{pmatrix} 7 \\ 9 \end{pmatrix}.$$

7.5.5 Basis vectors

If any set of state vectors V can be written in terms of a linear summation of elements of a set of vectors $\mathcal{B}$, then we say that $\mathcal{B}$ is a *basis* of, or *spans*, V. For example, the Euclidean basis

$$\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \tag{30}$$

can construct any vector in $\mathbb{C}^2$. If $|\mathcal{B}|$ is minimal, we say $\mathcal{B}$ is *orthonormal*. We also say that all of these vectors are *linearly independent*.

Exercise 21. 🌶🌶🌶 *What is a basis for the vector space of all 2x2 matrices?*

> *The* anharmonic co-ordinates *and equations employed, for the plane and for space, were suggested to the writer by some of his own* vector forms*; but their* geometrical interpretations *are assigned.*
> —William Rowan Hamilton (1866) [18]

7.6 Proof writing

Whenever you are asked to *prove* something, you might get a bit nervous, which is perfectly natural. Due to time constraints, computer scientists–and even mathematicians–are frequently left in the dark as to the amount of rigor fit for a proof. Here are some useful tricks of the trade:

7.6.1 By induction

Imagine you are attempting to prove a statement (or, *proposition*) that depends on a natural number n; we will call it $P(n)$. Induction is all about proving $P(0)$ (the base case), followed by proving that $P(n)$ implies $P(n+1)$.

Example 1. *(Triangle numbers) Let $S(n) = S(n-1) + n$. The closed form expression of $S(n)$ is*

$$\sum_{k=0}^{n} k = \frac{n(n+1)}{2} \tag{31}$$

for $n \geq 0$; we define $S(0)$ to be 0.

Proof. Consider $S(1) = S(0) + 1 = 1(1+1)/2 = 1$; the base case checks out. Now we assume that $S(k)$ works as advertised for $k = 1, 2, \ldots, n$. We need to prove that $S(n+1)$ is really $S(n) + (n+1)$; let's unpack the summation:

$$\begin{aligned} S(n+1) &= \sum_{k=0}^{n+1} k \\ &= (n+1) + \sum_{k=0}^{n} k \\ &= (n+1) + S(n) \end{aligned}$$

□

The important part of the example is extracting information from the more complicated bits (in this case, the summation), and re-writing what's less in terms of simpler objects. If you've ever written a recursive program, you already know how to do this; in fact, it appears in the preface of this very manuscript.

Exercise 22. 🌶🌶🌶 *Prove by induction the core binomial identity:*

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n. \tag{32}$$

7.7 Further reading

A solid, free and online introduction to linear algebra is [16]. For summation notation, defined precisely, refer to [19], chapter 2. For the axioms of probability theory, refer to [20]. A fabulous introduction to combinatorics (also known as *finite* mathematics) is [21].

8 Qubits defined

In the quantum computing literature, $|\psi\rangle$ and sometimes $|\phi\rangle$ are often used to denote quantum bits, or *qubits*. Equation 27 is a standard (vector) representation of a qubit, where $\alpha, \beta \in \mathbb{C}$; therefore, $|\psi\rangle$ is a vector in the vector space $\mathbb{C}^2$.

On occasion, we must "multiply" two qubits together: we write this as

$$|\psi\rangle\,|\phi\rangle = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ v_1 \\ v_2 \end{pmatrix} \tag{33}$$

which can be thought of more as a *concatenation* per se, and we'll usually keep it as $|\phi\rangle\,|\psi\rangle$ or just $|\phi\psi\rangle$ when we work on actual circuits.

Exercise 23. 🌶🌶🌶 *Calculate* $CX\,|\phi\rangle\,|\psi\rangle$.

> This is how you are to build it: The ark is to be three hundred cubits long, fifty cubits wide and thirty cubits high. (Genesis 6:15)

8.1 Superposition and basis vectors

In quantum computing, complex numbers serve a special purpose: they allow us to write out a qubit $|\psi\rangle$ as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{34}$$

where $\alpha, \beta \in \mathbb{C}$, which are known as the *probability amplitudes* of $|\psi\rangle$, meaning that the probability of measuring $|\psi\rangle$ in the $|0\rangle$ state is $|\alpha|^2$ (and similarly for $|1\rangle$). By the definition of probabilities, we have

$$|\alpha|^2 + |\beta|^2 = 1. \tag{35}$$

A common question is: if qubits represent something physical, why do we use complex numbers? The reason has to do with the fact that real-valued probabilities p are limited to the range $0 \le p \le 1$. We may want to add probabilities together, but we can't allow the total probability to exceed one. Using complex numbers allows us to add and subtract probability *amplitudes* instead.

The reality is that many of the values of α and β we use initially within quantum computing are *real* numbers, but some of them are complex-valued, so we define it this way to cover all the cases. (This still works since the reals are a *subset* of $\mathbb{C}$.)

$|0\rangle$ and $|1\rangle$ are together known as the *Z basis* vectors; equation (34) describes the superposition of $|\psi\rangle$ in Z basis vectors.

Exercise 24. 🌶🌶🌶 *Show that the probability of measuring $|1\rangle$, when $|+\rangle$ is projected into the Z-basis, is 1/2.*

> *Mathematics is the foundation of all exact knowledge of natural phenomena.*
> —David Hilbert (1900) [22]

8.2 Eigenvalues and eigenstates

For a given matrix A and state vector $|\psi\rangle$, the *eigenvalue* $\lambda \in \mathbb{C}$ is defined as

$$A|\psi\rangle = \lambda A$$

By this definition, we can say that $|\psi\rangle$ is the λ-*eigenstate* of A.

Exercise 25. 🌶🌶🌶 *Find the eigenvalues and eigenstates of*

$$\begin{pmatrix} 2 & 8 \\ 1 & 4 \end{pmatrix}.$$

Exercise 26. 🌶🌶🌶 *Find the eigenvalues and eigenstates of X and Z.*

> *...daß sich das schwingende Elektron gegenüber Licht, das viel kurzwelliger ist als alle Eigenschwingungen des Systems, wie ein freies Elektron verhält.*[3]
> —Werner Heisenberg (1925) [23]

[3] *...that the oscillating electron behaves like a free electron compared to light, which has a much shorter wavelength than all the natural oscillations of the system.* (Google-translated)

8.3 Simultaneous eigenstates

A state vector $|\psi\rangle$ is a *simultaneous* λ–eigenstate of two operators A and B if $A|\psi\rangle = B|\psi\rangle = \lambda|\psi\rangle$. Of course, $|\psi\rangle$ can admit a simultaenous eigenstate for an arbitrary number of operators; in the realm of quantum computing, we often have operators that anti-commute, that is, $AB = -BA$. As a result, we find ourselves dealing with $+1$ or -1 eigenstates.

Exercise 27. 🌶🌶🌶 *Show that* $|00\rangle$ *is the simultaneous* $+1$ *eigenstate of the operators* $I \otimes Z$ *and* $Z \otimes I$.

> *Editing is unique to film. You can see something from different points of view almost simultaneously, and it creates a new experience.*
> —Stanley Kubrick

8.4 Observables

An observable is a physical property that can be measured. In the context of implementing TQEC via superconducting qubits, we treat each physical qubit as if they were isolated from one another. In that sense, two neighboring qubits cannot "observe" one another.

Mathematically speaking, we treat each observable as a Hermitian operator. For example, since the Hadamard matrix is hermitian, a gate action $H|\psi\rangle$ is observed by the qubit $|\psi\rangle$.

When looking at an individual qubit (without any error or gate actions), we can treat two states $|\phi\rangle$ and $|\psi\rangle$ equally if there exists a nonzero complex number λ such that $|\psi\rangle = \lambda|\phi\rangle$.

8.5 The Bloch sphere

We've seen how complex numbers have an easily-visualized polar form, that is, $z = a + bi = re^{i\theta}$. Since a qubit contains two complex numbers, we now have four real numbers that encodes a single qubit. Since four dimensions are hard to visualize on paper, it would be nice to have a three-dimensional representation to help us visualize a qubit.

We're in luck: we can use an observability property of quantum mechanics to represent a qubit using *spherical* coordinates:

$$|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\varphi}\sin(\theta/2)|1\rangle \tag{36}$$

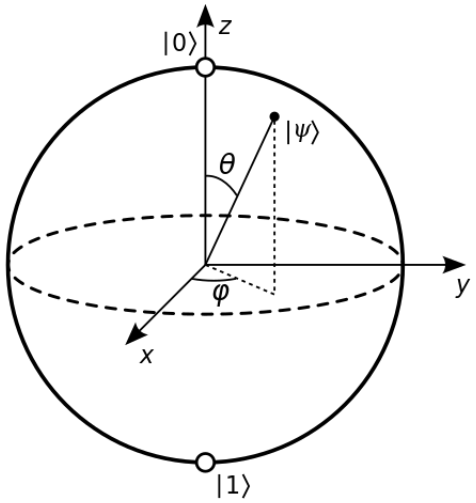


Figure 4: The Bloch sphere. Any point corresponds to a possible superposition of a qubit $|\psi\rangle$, assuming a radius of one. The angles θ and φ represent the superposition.

For a mathematical derivation of the Bloch sphere equation (using observability properties) refer to [24].

Exercise 28. 🌶🌶🌶 *To which point on the Bloch sphere does qubit* $|\psi\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ *correspond?*

> *In talking to you I feel very much more at ease than my colleagues who gave the speeches during the banquet.*
> —Felix Bloch

9 Combinatorial topology

Combinatorics is the study of finite objects. The most simple example, borrowed from the previous section, is the integers modulo 2, $\mathbb{Z}_2 = \{0, 1\}$. Since computers have a finite set of information that we can measure at any given time, we can model computer data (both values and references of variables) according to discrete structures. One way to think about a 32-bit *word* of information is that it's an element of the field ${\mathbb{Z}_2}^{32}$, which isn't terribly interesting in its own right, but gives us a taste of what's to come.

9.1 The binomial coefficients

One of the most useful symbols in combinatorics is the *binomial coefficient*, denoted $\binom{n}{k}$ or $C(n, k)$. The former representation is most common, while we also use the latter representation to avoid confusion with other symbols, such as, in our case, two-dimensional state vectors. Regardless, the binomial coefficient tells us how many ways we can select k objects from a set of n objects. This definition assumes that $n \leq k$, and n and k are natural numbers.

For example, let's say a company needed to replace two engineers, and the pool of applicants was 10 people. (Not very realistic, I know.) The recruiting team would have

$$\binom{10}{2} = \frac{10!}{(10-2)!2!} = 45$$

possible sets of candidates. There are countless identities that exploit binomial coefficients; a helpful resource for developing them is [25].

Exercise 29. *Show that* $n\binom{n}{k} = \binom{n+1}{k}$.

9.2 Graph theory

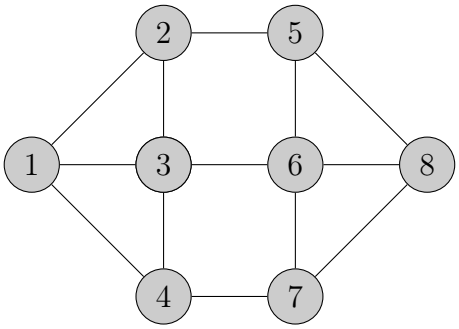
Graphs have a representation for being innocuous, perhaps due to their reputation of wide use throughout computer science. For our purposes, we'll set down a few of the many, many definitions used within this very rich field.

A graph G comprises a set of vertices and a pair of edges, denoted (V, E). Vertices are simply points on a 2-dimensional plane; each edge connects precisely two vertices.

A graph is named such since the quickest way to understand it is by etching it it on a piece of paper, or perhaps a stone tablet: the Ancient Greek

term "graph" ($\gamma\rho\alpha\phi\omega$) is a cognate to the modern English *carve*. Sometimes, they are referred to as *network* graphs due to their frequent application to routing problems.

There are many different kinds of graphs. A planar graph is one such that when drawn, there are no overlapping edges:



A widely-used graph is the complete graph, K_n. This graph contains exactly n nodes, and each node connects to the other. A simple example is K_6

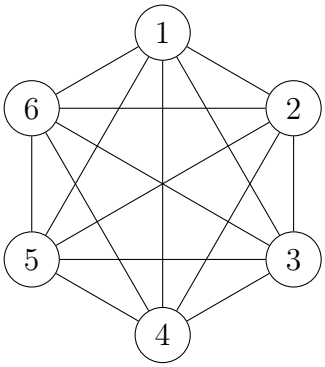


Figure 5: The complete graph with $n = 6$, K_6.

Exercise 30. 🌶🌶🌶 *Sketch K_5. How many times did you lift your pen?*

Exercise 31. 🌶🌶🌶 *Show that, given $K_n = (V, E)$, we have $|E| = \binom{n}{2}$.*

> *All men's miseries derive from not being able to sit in a quiet room alone.*
> —Blaise Pascal

9.3 Repetition codes

Building an error-correcting device requires the use of a mathematical structure known as a code. A classical code has 3 parameters $[n, k, d]$:

- n is the length of the code-words.
- $k < n$ is the length of encoded messages.
- d is the number of errors between code-words.

Imagine we wish to build an error correction circuit. Let's say we had a grid of 10 x 10 home-made transistors. Assume each transistor has a 1% probability at failing at any given time. We designate this code as a distance d code.

9.4 Further problems

Exercise 32. 🌶🌶🌶 *A bipartite graph has a vertex set $V = A \bigcup B$ such that each edge contains one vertex in A and another in B. Prove that no bipartite graph contains an odd cycle.*

10 Algebraic topology

The goal of this section is to introduce the mathematics used in advanced applications of TQEC, starting with defining the planar code precisely; my recommendation is to keep the key definitions of this section in your back pocket.

10.1 Groups

A *group* is a mathematical object that associates a set G with a binary operator. Typically we do not assign any particular symbol to this operator, akin to numeric multiplication. Formally, the operator maps G to $G \times G$ to G, and obeys these three properties:

- (Associativity) For any $a, b, c \in G$, we have $a(bc) = (ab)c$.
- (Identity) There exists an $e \in G$ such that, for any $a \in G$, we have $ae = ea = a$. e is referred to as the *identity* element of G.
- (Inverses) For all $a \in G$, there exists its inverse a^{-1} such that $aa^{-1} = e$.

Crucially, not every group is *commutative*: we can't assume that $ab = ba$.

A group that is commutative is known as an *Abelian* group. An example of an Abelian group is the integers, paired with addition, written as $(\mathbb{Z}, +)$, which we can write more compactly as $\mathbb{Z}$. For any $n \in \mathbb{N}$, the integers *modulo* n is defined as

$$\mathbb{Z}_n = \{a \pmod n : a \in \mathbb{Z}\} \tag{37}$$

Exercise 33. 🌶🌶🌶 *Prove that $\mathbb{Z}_n$ is an Abelian group.*

Exercise 34. 🌶🌶🌶 *Prove that $\mathbb{R}^2 = \mathbb{R} \times \mathbb{R}$ is a group.*

Exercise 35. 🌶🌶🌶 *(Heisenberg group) Prove that for all $a, b, c \in \mathbb{C}$, the set of upper-triangular 3x3 matrices, represented as*

$$\begin{pmatrix} 1 & a & c \\ 0 & 1 & b \\ 0 & 0 & 1 \end{pmatrix} \tag{38}$$

is a group under matrix multiplication.

Exercise 36. 🌶🌶🌶 *The "special" unitary group SU(2) contains the set of all 2x2 unitary matrices with determinant 1. Prove that SU(2) is non-Abelian.*

10.2 Generating subgroups

Given a group G, its subgroups H are denoted $H \subseteq G$ due to the set relationship. Importantly, H is a group in its own right, and has the same identity element as G. The trivial subgroups of G are itself and the empty set . If the elements of H can be put together such that you end up back with G, Remembering that sets cannot, by definition, contain duplicate elements, we say that H *generates* G, or $\langle H \rangle = G$.

Exercise 37. *Prove that* $\langle \mathbb{Z}_2 \rangle = \mathbb{Z}$.

10.3 Gaussian lattices

The *Gaussian integers* are defined as $\mathbb{Z}[i] = \{a + bi : a, b \in \mathbb{Z}, i = \sqrt{-1}\}$. If you were to plot the elements of $\mathbb{Z}[i]$ on the complex plane, you'll see a nice dotted structure; they form what is known as the Gaussian integers, or for our purposes a *lattice*:

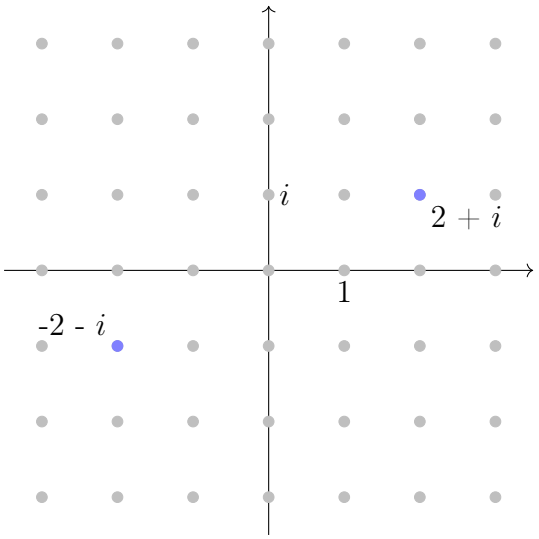


Figure 6: The Gaussian integers.

If we were to declare $F \subset \mathbb{Z}[i]$, we could systematically create an arbitrarily large lattice $\langle F \rangle$ that covers the entire plane.

Exercise 38. *Prove that* $\mathbb{Z}[i]$ *is a group.*

Exercise 39. 🌶🌶🌶 *For which φ is $e^{i\varphi} \in \mathbb{Z}[i]$?*

Exercise 40. 🌶🌶🌶 *Prove that F is a subgroup of $\mathbb{Z}[i]$.*

Exercise 41. 🌶🌶🌶 *Prove that if $|\mathbb{C}| = 2^{\aleph_0}$, then $|\mathbb{Z}[i]| = \aleph_0$. (Hint: have you ever studied Hebrew?)*

10.4 Equivalence classes

The structure of every possible $\mathbb{Z}_n$ admits what is known as an *equivalence class* of $\mathbb{Z}$. To show what an equivalence class is, consider the example $n = 3$. First, compute each element: that is easy enough, and we denote the sets as $A = \{[0], [1], [2]\}$. Next, associate each $a \in A$ with the set $\{a \pm nk : k \in \mathbb{Z}\}$:

$$[0] \to \{\ldots -3, 0, 3, 6, 9 \ldots\}$$
$$[1] \to \{\ldots -2, 1, 4, 7, 10 \ldots\}$$
$$[2] \to \{\ldots -1, 2, 5, 8, 11 \ldots\}$$

It is clear that the values of the association forms a *partition* of $\mathbb{Z}$. Thus, A is known as an *equivalence class* of $\mathbb{Z}$.

Exercise 42. 🌶🌶🌶 *Prove the above claim.*

> *...the importance of applications* [of groups] *such as coding theory and cryptography has grown significantly.*
> —Thomas W. Judson (2022) [26]

10.5 Topological braids

The group of n strands, denoted B_n, is a fundamental structure in topology, and enables advanced contributions to our field. If you've ever played cat's cradle, you already know what a braid looks like: imagine holding out both hands, and tying rubber bands between each of your four fingers; to illustrate B_4. Like it or not, each possible way you could have done that formulates a group.

Your first question should now be: what do my fingers represent? The identity is rather simple: your left pinkie connects to your right pinkie, and so on, up to your index fingers. But let's assume our cradle is a little more

interesting than that, and your baby brother comes over and tries to bend your bands out of shape. Since you're bigger than him, he only manages to change how the bands are interlaced, and they don't slip from your fingers at all. Since your brother has the same genes as you, his actions constitute an identity operation.

The next day, since it's show-and-tell day in math class, you do the same thing, but instead the professor comes over, and swaps two of the rubber bands that connect to your right hand. He asks, "Johnny, is this what you brought to class today?" Tim, who you could have sworn was just asleep, blurts out "no, you've composed what he brought in with... umm..." The prof replies, "that's right, Tim, but what did I *compose* it with?"

Being an astute student, you grab some chalk and gets to work, only you realized you forgot the book, so you begin deriving the composition rule yourself... but since tomorrow is the final day of classes, you figure that studying for the final exam will be a better use of your time. Now dissatisfied with the public education system, you search in vain for a good book on the subject of braid theory.

Okay, enough with the anecdote. The best way to fully understand the structure of a braid is in terms of what's known as a permutation group, which are denoted as S_n. Permute is just a fancy term for mapping: for

example, our identity braid maps every key in the domain to itself, which we can denote as

$$\mathrm{id} = \{1 \to 1, 2 \to 2, 3 \to 3, 4 \to 4\} \tag{39}$$

since it's our identity element. We can write this more succinctly as (1234).

Exercise 43. 🌶🌶🌶 *What does the permutation* (4321) *represent?*

Exercise 44. 🌶🌶🌶 *Show that* S_n *is a group of order* $|S_n| = n!$.

The actual braid group is a little more complicated than that, since it involves equivalence classes and something called homotopy. Since other works treat these issues better than the author ever could, I'm going to stop the math bandwagon and switch over to computer science. I hope you enjoyed the ride.

Exercise 45. 🌶🌶🌶 *Explain what we mean when we say "topological" quantum error correction.*

> *Thus, although Abel shared with many mathematicians a complete lack of musical talent, I will not sound absurd if I compare his kind of productivity and his personality with Mozart's.*
> —Felix Klein [27]

10.6 Further reading

> *I suggested that topologically ordered states can serve as a physical analogue of error-correcting quantum codes.*
> —Alexei Kitaev (2006) [28]

11 Quantum gates

Quantum gates are the basic components of quantum circuits, which act on individual qubits. Each gate is represented by a *unitary matrix*.

11.1 Hermitian matrices

A hermitian matrix A is any square matrix that is *self-adjoint*. We denote the adjoint with the "dagger" symbol, †, which we can define as

$$U^\dagger = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^\dagger = \begin{pmatrix} a^* & c^* \\ b^* & d^* \end{pmatrix} \tag{40}$$

Thus, a matrix A is Hermitian if and only if $A^\dagger$ The adjoint of a matrix is also known as its *conjugate transpose*, which we find by swapping the rows and columns (which yields the transpose), and taking the conjugate of each element.

Exercise 46. 🌶🌶🌶 *Find a 2x2 matrix which is Hermitian.*

11.2 Unitary matrices

A unitary matrix U is any square matrix such that

$$U^\dagger U = UU^\dagger = I. \tag{41}$$

Theorem 1. *Every quantum computing operator (errors and gates) is represented by a unitary matrix.*

Exercise 47. 🌶🌶🌶 *Show that if U is unitary, then $U = U^{-1}$.*

> *Of special interest are the* unitary *representations, in which the linear transformations leave invariant a positive definite quadratic form in the co-ordinates of a vector.*
> —Paul Dirac (1944) [29]

11.3 Hadamard (H)

A Hadamard gate places a Z-basis qubit in superposition. Its matrix is defined as

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

The *plus* and *minus* states, which comprise the X basis, are defined as:

$$|+\rangle = H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$
$$|-\rangle = H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Both the plus and minus states demonstrate superposition: if we were to measure the qubit in either state, there'd be a $1/2$ probability of measuring a 0 or 1.

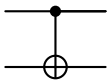
Exercise 48. 🌶🌶🌶 *Prove that H squares to the identity, meaning $HH = I$.*

Exercise 49. 🌶🌶🌶 *Prove that H is unitary.*

Exercise 50. 🌶🌶🌶 *Show that $X|+\rangle = |+\rangle$.*

> *It has been written that the shortest and best way between two truths of the real domain often passes through the imaginary one.*
> —Jacques Hadamard (1945) [30]

11.4 Controlled-X (CX)



CX is our first example of a two-qubit gate; the control bit remains unchanged, but depending on its value, will "flip" the second gate, which is to say, apply the X matrix:The row vectors are $\{|-\rangle, |+\rangle\}$.

$$X|\psi\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}. \tag{42}$$

A useful working definition of CX is by analogy to classical XOR, which conditionally applies a NOT gate to the input line; hence the term CNOT frequently used in the quantum computing to describe this circuit. It takes two qubits, control and target.Look closely at the diagram: it's a literal target! If the control qubit is $|0\rangle$, and the target qubit remains the same; otherwise, it is set to $|1\rangle$.

The CX gate's matrix representation is

$$CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

As with H, the matrix representation of CX is involutary and unitary. The matrix representation of CNOT is $I \otimes X$. The full computation of CNOT by its matrix definition is left as an optional exercise to the reader.

Exercise 51. *Describe the difference between AB and $A \otimes B$ in terms of gates and state vectors.*

> *No matter how correct a mathematical theorem may appear to be, one ought never to be satisfied that there was not something imperfect about it until it also gives the impression of being beautiful.*
> —George Boole

11.5 Controlled-Z (CZ)

This is a similar gate to CX, but there's no *target* qubit. The action is rather simple to understand: if the input qubits are equal to $a|11\rangle$, the result state is $-a|11\rangle$. For any $a|\psi\rangle$, where $\psi \in \{[00],[10],[01]\}$, $a|\psi\rangle$ is left alone. The matrix representation of CZ can be determined by $I \otimes Z$, where

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{43}$$

The row vectors are obtained from the Z basis, where $Z = \{|0\rangle, -|1\rangle\}$.

Exercise 52. 🌶🌶🌶 *Describe H in terms of X and Z.*

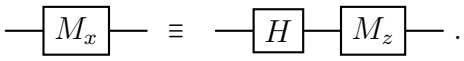
> *Competent means we will never take anything for granted. We will never be found short in our knowledge and in our skills. Mission Control will be perfect.*
> —Gene Kranz (2001) [31]

11.6 Measurement and reset

A measurement gate projects a qubit into a particular basis. For example, M_z projects a single qubit into the Z basis, meaning that it returns either a (classical) 0 or 1. An M_x gate measures in the X *basis*, or

$$\{|+\rangle, |-\rangle\} = \{H|0\rangle, H|1\rangle\}. \tag{44}$$

An M_z gate can therefore be written as



We don't need a matrix representation for M, since its operation is defined by the hardware implementation.

Exercise 53. 🌶🌶🌶 *Prove the Z basis measurement identity.*

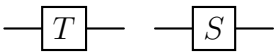
> *We are talking about a predictive theory, not just measurements after the fact.*
> —Richard Feynman (1965) [32]

By contrast, the reset gate forces a qubit into the $|0\rangle$ state.Physically, this is similar to a measurement operation, but we wait around for the qubit to relax into the $|0\rangle$ state.

Exercise 54. 🌶🌶🌶 *Explain why M and R are the least time-performant quantum gates.*

> *This workspace gets reset to* 0 *after each subroutine of our algorithm, so we will not include it when we write down the state of our machine.*
> —Peter W. Shor (1997) [33]

11.7 T and S



The T gate is defined as

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \tag{45}$$

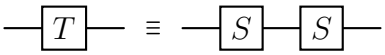
This is also referred to as the $\pi/8$ *gate*. This is due to it corresponding with a $\pi/8$ rotation about the vertical axis of the Bloch sphere. A similarly defined gate is S, also commonly denoted P (for "phase"):

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \tag{46}$$

Note that $i = e^{i\pi/2}$; consider $\left(e^{i\pi/4}\right)^2 = e^{i\pi/2}$, meaning that $e^{i\pi/4}$ is the square root of i; hence, we can write

$$T = \sqrt{S}, \tag{47}$$

which tells us how we can build a T gate using S gates:



S and T are collectively referred to as *phase shift* gates, admitting a general form

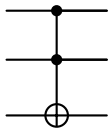
$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix} \tag{48}$$

where φ, called the *phase angle*, is of period 2π, constricting it to a single rotation about the unit circle.

Exercise 55. *For which value of φ does the phase shift matrix square to the identity?*

> *I Tiresias, though blind, throbbing between two lives...*
> —T. S. Eliot, *The Waste Land* (1922)

11.8 Toffoli



This gate is also referred to as CCNOT, since it takes two control bits; the value of the target bit is flipped if both control bits are equal to $|1\rangle$. Since we're considering three qubits and a new output state, we'll need a tensor product with four matrices to derive the matrix representation, which is $I \otimes I \otimes I \otimes X$.

Exercise 56. 🌶🌶🌶 *Show how to construct an S gate using Toffoli gates.*

11.8.1 Clifford gates and universality

The *Clifford gates* comprise the set $\{CX, H, S\}$; combined with T, they're known as *Clifford-T*.

Theorem 2. *Clifford-T gates build a universal quantum computer.*

The proof of the theorem is beyond the scope of this work, but it's useful to have the definition and statement handy. The Toffoli gate will frequently be included in this set due to its efficiency. One of the core justifications for this claim is that each of these gates is *reversible*, since $U^{-1} = U$ for any unitary matrix U.This is impossible to achieve with a classical computer since heat loss within the circuit destroys information. For example, an XOR gate cannot be reversed; it combines two wires into a single one.

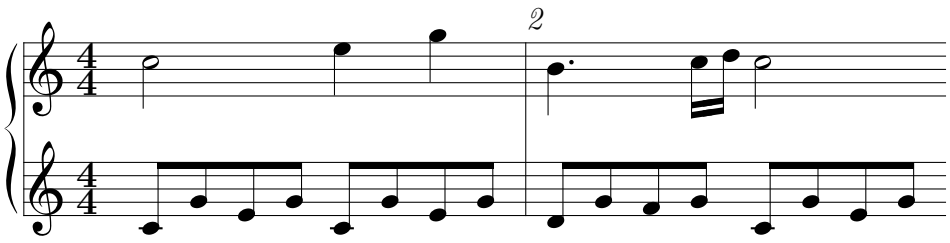
> *It contains, next, a generalization of them, applicable to any number of dimensions; and a demonstration that the algebra thus obtained is always a compound...*
> —William K. Clifford (1878) [34]

11.9 Further reading

Success is a lousy teacher. It seduces smart people into thinking they can't lose.
—Bill Gates

12 Quantum circuits

What does a quantum circuit look like?



Well, that's not a *bad* answer.

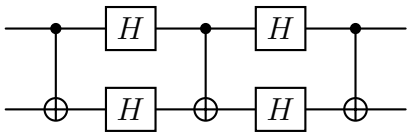


Figure 7: A quantum circuit.

That's better. But those who've had the fortune (or perhaps *mis*fortune) of studying music theory may draw a few key connections:

1. Time moves from left to right.
2. Each horizontal line has a relationship to the other.
3. The lines are logically and temporally connected.
4. The connections along the lines can be differently shaped.

Let's explain the meaning for quantum computing, by examining the top horizontal line of Figure 7.

1. *Time moves from left to right*: The action occurs in the order top-left dot, followed by the "H," another dot, and so on.
2. *Each horizontal line has a relationship to the other.* Each line represents a qubit.

3. *The lines are logically and temporally connected.* The lines between two qubits represent actions that depend on both of them.

4. *The connections along the lines can be differently shaped.* In this picture, we have two kinds of *gates*, some of which act on a single qubit (H), and others act on two (another kind).

A person of any mental quality has ideas of his own. This is common sense.
—Franz Liszt

12.1 The Bell pair

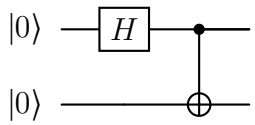


Figure 8: A circuit that entangles two qubits (prepared in the state $|00\rangle$) into a Bell pair

Figure 8 depicts our first example, which prepares a quantum state known as a *Bell pair* (also known as an EPR[4] pair): on the left, both qubits are prepared in the $|0\rangle$ state; at this point, the qubits are prepared as $|\psi_0\rangle = |00\rangle$. Next, a Hadamard gate is applied to the first qubit, producing the state

$$|\psi_1\rangle = |+\rangle\,|0\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)|0\rangle = \frac{|00\rangle + |10\rangle}{\sqrt{2}}$$

(Here, we're treating kets as algebraic objects we can add and multiply with complex numbers.) It's important to write $|\psi_1\rangle$ in this manner, since we're about to apply a CX gate, which requires us to inspect both qubits simultaneously. After applying CX, we're left the Bell pair

[4]Named for Einstein, Podolsky, and Rosen, for pointing out that this state suggests that "hidden information" leads to quantum entanglement; decades later, Bell pointed out that quantum theory is incompatible with such a notion [35].

$$|\psi_2\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

(Note that we don't need to use the matrix representation of CX, which is rather cumbersome.) The Bell pair is the simplest example of *quantum entanglement*: it is only possible to measure 00 or 11, making the qubits "entangled."

12.2 Teleportation

We can use the (entangled) output of a Bell pair circuit to perform our next trick, teleportation. This is one of those things that sounds impossible but can be demonstrated rather simply:

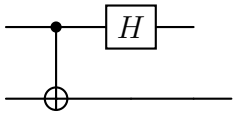


Figure 9: A circuit demonstrating quantum teleportation.

12.3 Exercises

Exercise 57. 🌶🌶🌶 *Prove that the circuit in Figure 15.3 corrects* $X\left|(\alpha|0\rangle + \beta|1\rangle)\right\rangle$.

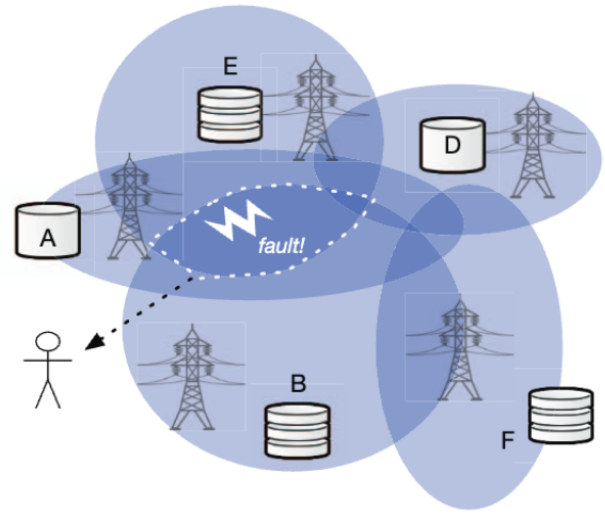
Exercise 58. 🌶🌶🌶 *Combine the Bell pair and quantum teleportation circuits, and prove that it preserves the state* $|00\rangle$.

> *Consider a pair of spin one-half particles formed somehow in the singlet spin state and moving freely in opposite directions. Measurements can be made, say by Stern-Gerlach magnets, on selected components of the spins...*
> —John W. Bell (1964) [35]

This circuit is interesting because it has inclines and declines. Not just up, but down as well.
—Murray Walker

13 Fault-tolerant computing

The best-performing quantum computers available today are made of *superconducting* circuits, which can only operate at near-absolute zero temperatures. Due to construction and operating costs in the hundreds of millions of dollars, it's likely that quantum computers will only exist in the cloud for the foreseeable future; quantum computers will likely be incorporated into distributed systems. By that token, we can treat quantum computers as *nodes* in a geographically distributed computer system.



13.1 Formalism

Distributed systems are commonly modeled as a *network graph* $G = (V, E)$; since the representations of V and E are arbitrary, we can think of G as a *topological* object.

13.2 The CAP theorem

Theorem 3. *(Brewer) Any distributed data store cannot guarantee more than two out of three of: consistency, availability, and performance.*

13.3 Exercises

Exercise 59. 🌶🌶🌶 *Show that a complete graph K_n represents a fault-tolerant peer-to-peer network.*

13.4 Further reading

An excellent reference in the area of topological distributed computing is [36].

14 Pauli errors

Classical hardware errors are limited to bit flips and data loss, which can be mitigated via parity checks and replication. Quantum computers do not allow such replication, as the no-cloning theorem prohibits copying of an arbitrary quantum state, and measurement destroys quantum information. Furthermore, qubit information is continuous, and thus qubit *errors* are extremely likely [9]. Fortunately, every quantum error can be modeled as a linear combination of *Pauli errors*, starting with the usual suspects

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

14.1 Locality

It's important to restrict our study to topologically trivial errors. Technically, that means those that are restricted to simple loops, and braids. Suppose we had a quantum computer comprising four qubits, arranged in a two-dimensional square. Since our qubit is on the order of 10^{-10} meters (or one Angstrom, if you like) in diameter, it can be affected by any observable (Pauli) error, denoted $\sigma_{X|Z}$. With the assistance of advanced hardware, we can ignore errors that are unobservable, which are denoted $e^{i\gamma}$ in Figure 10.

14.2 Formalism

To recap, we have defined X and Z according to their respective quantum operators. Refer to the quantum gate supplements. A guy named Wolfgang Pauli discovered that these specific matrices came up pretty often in his studies of quantum mechanics, so he decided to name them after himself. For our purposes, we can define the *Pauli error* matrices as such:

$$\mathcal{P} = \{I, X, Z, XZ\}. \tag{49}$$

We include the identity matrix I for completeness; the product XZ can be written as $e^{-i\pi/2}Y$, where

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{50}$$

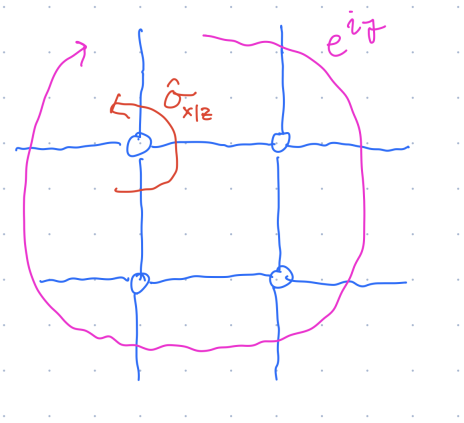


Figure 10: A set of four error-prone qubits. A topologically trivial error $\sigma_{X|Z}$ encircles a qubit, since it's observable by the qubit; the non-local error term $e^{i\gamma}$ is unobservable, and can thus be "factored out."

is the *other* kind of Pauli error, which we include here out of completeness; since it's hard enough to remember a handful of matrices, we will hopefully never need to use it again.

Lemma 1. *The set $\mathcal{P}$ is an orthonormal basis for all 2x2 matrices.*

Exercise 60. 🌶🌶🌶 *Prove Lemma 1.*

Notably, these matrices are known more specifically as Pauli *operators*, since X and Z gates operate on qubits in an identical manner to errors. X and Z errors occur in quantum circuits "in the wild"; the fact that

$$XX = ZZ = I \tag{51}$$

means that, if we apply an X gate in the presence of an X error that "bit-flips" the qubit, the X error is now corrected (and likewise for Z).

Theorem 4. *If we can correct X and Z errors, we can correct any local quantum error.*

To see why, consider the following matrix decomposition:

$$B = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \implies B = \frac{a+d}{2} I + \frac{b+c}{2} X + \frac{a-d}{2} Z + \frac{b-c}{2} XZ. \tag{52}$$

Therefore, we refer to B as an *error model* (or *noise* model) and galvanize our focus on the Pauli errors X and Z. When programming error correction circuits, to shortcut the effort of manipulating matrices, it is helpful to remember a few rules:

$$X|0\rangle = |1\rangle \qquad Z|0\rangle = |0\rangle \tag{53}$$
$$X|1\rangle = |0\rangle \qquad Z|1\rangle = -|1\rangle \tag{54}$$

since we frequently represent $|\psi\rangle$ in the Z basis.

We disregard what are known as *topologically non-trivial* quantum errors, which only affect what is referred to as global phase; they represent a factor of $e^{i\gamma}$ which can be ignored during any modularization of probability amplitude, since $|e^{i\gamma}z| = |z|$; hence, global phase is *unobservable*. Stabilizer measurements can correct so-called topologically trivial (or, local) errors, hence why we call our work *topological* quantum error correction.

14.3 Exercises

Exercise 61. *Prove that $e^{i\pi/2}XZ = Y$.*

Exercise 62. *Show that $\sigma^2 = I$ for all $\sigma \in \mathcal{P}$.*

Exercise 63. *Prove the shortcut rules for X and Z.*

Exercise 64. *Show that $e^{i\gamma}$ is "unobservable" with respect to any quantum system $|\psi\rangle$. (Hint: take the modulus of $|\psi\rangle$.)*

Exercise 65. *Create an analogous set of shortcut rules with respect to the X basis.*

Exercise 66. *For which A does $AX = Y$? Decompose A in terms of the elements of $\mathcal{P}$.*

Exercise 67. *Show that $iXYZ = -I$.*

Exercise 68. *The Kronecker delta $\delta_{i,j}$ is equal to 1 if $i = j$, and 0 otherwise. Prove that any Pauli matrix can be written as*

$$\begin{pmatrix} \delta_{j,3} & \delta_{j,1} - i\delta_{j,2} \\ \delta_{j,1} + i\delta_{j,2} & -\delta_{j,3} \end{pmatrix} \tag{55}$$

for $1 \leq j \leq 3$.

Exercise 69. 🌶🌶🌶 *The Hamiltonian $H_\sigma(t)$, for each $\sigma \in \mathcal{P}$, denotes the total energy of a quantum system at time t. The Schrodinger equation tells us that*

$$H_\sigma(t)\,|\psi\rangle = i\hbar \frac{\partial}{\partial t}\,|\psi\rangle\,. \tag{56}$$

What does this tell us about a quantum system's dynamics?

> *In fact, if one proceeds on this basis it hardly appears possible to avoid the empirically untenable conclusion...*
> —Wolfgang Pauli (1955) [37]

15 Stabilizers

15.1 Formalism

A stabilizer A of a state $|\psi\rangle$ is an operator such that $|\psi\rangle$ is the +1 eigenstate of A. In the absence of a Pauli error, we have

$$A|\psi\rangle = |\psi\rangle . \tag{57}$$

This isn't to say that A has no impact on $|\psi\rangle$; its action is a projection onto a particular eigenstate. Let's define its action.

15.2 The controlled-A gate (CA)

To illustrate the stabilizer formalism, let's introduce a new gate, which we'll call A, which takes an arbitrary input $|\psi\rangle$, and reports the measurement of an *ancilla*, which we initialize as $|0\rangle$ and name q_0. This bit controls whether the A operator is applied to $|\psi\rangle$.

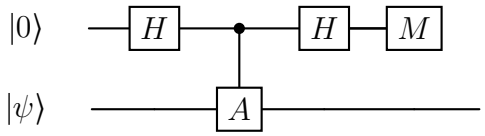


Figure 11: The A circuit, which is not a stabilizer.

Let's examine the action of this circuit on the state $|0\rangle|\psi\rangle$:

$$\begin{aligned}
&\xrightarrow{Hq_0} |+\rangle|\psi\rangle \\
&\xrightarrow{\mathrm{CA}|\psi\rangle} |0\rangle|\psi\rangle + |1\rangle A|\psi\rangle \\
&\xrightarrow{Hq_0} |+\rangle|\psi\rangle + |-\rangle A|\psi\rangle \\
&\xrightarrow{Mq_0} 0(|\psi\rangle + A|\psi\rangle) + 1(|\psi\rangle - A|\psi\rangle)
\end{aligned}$$

The final line of the above indicates that, in the case that $A|\psi\rangle = |\psi\rangle$, it is only possible to measure a 0 in this circuit.

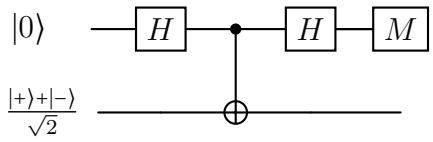
But now, the input state $|\psi\rangle$ has been transformed into a superposition of $|\psi\rangle$ and $A|\psi\rangle$, which we may denote as:

$$|\psi\rangle \pm A|\psi\rangle \tag{58}$$

We'll want to be a bit more specific when we build actual circuits. In general, to construct a circuit that measures a stabilizer, an ancillary qubit is prepared in the $|0\rangle$ state, which is read by a measure gate as 1 in the presence of an error. We implement this circuit in two different ways to measure X and Z stabilizers.

15.3 The X stabilizer

Since we only care about correcting X and Z errors, we can implement the A stabilizer measurement circuit as X and Z stabilizer measurement circuits. Let's start with the X stabilizer:



This circuit is the same as the controlled-A, but instead a CX circuit is in is place. Let's replace A with X. From equation 58:

$$0(|\psi\rangle + X|\psi\rangle) + 1(|\psi\rangle - X|\psi\rangle) \tag{59}$$

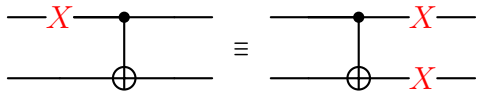
Suppose we were to build a quantum circuit with a state $|\phi\rangle$ representing the data we want to preserve, and assume an X error were to act on the state $|\phi\rangle$, which we write as

$$|\psi\rangle = X|\phi\rangle = \frac{|+\rangle + |-\rangle}{\sqrt{2}}. \tag{60}$$

To summarize, this stabilizer has a dual purpose:

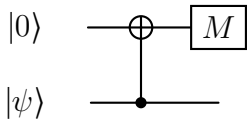
- To *detect* quantum errors, which are indicated by a 1 measurement;
- To *correct* quantum errors; this circuit can remove a single X error that occurred before the circuit ran.

A careful reader may ask what would happen if an X error occurs immediately before or after the control point; consider the following identity:



In this case, the X error is propagated into $|\psi\rangle$, however, we're able to *detect* the fact that an X error occurred at this point!

15.4 The Z stabilizer



The Z stabilizer operates in a similar fashion to the X stabilizer, and can similarly detect and correct a single Z error by exploiting the identity

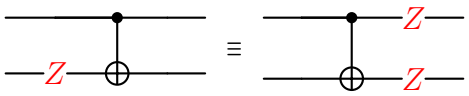


Figure 12 provides an example of measuring three consecutive stabilizers.

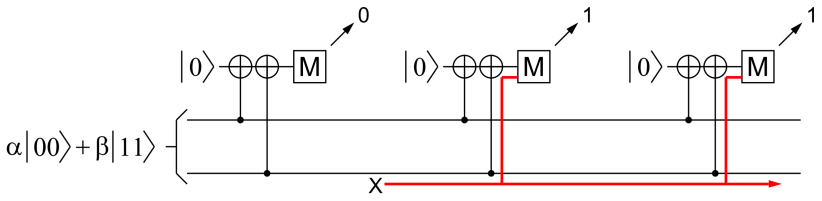


Figure 12: Three 2-qubit Z stabilizers in action. The red lines represent the presence of an X error, and the arrows represent the results of stabilizer measurement. The (classical) 0 and 1 measurements form a stream, which can fed into a classical algorithm to determine which qubits must be flipped, forming the basis of error-tolerant quantum computation [38].

The ancillary qubit will be in the $|0\rangle$ state if no error has occurred. If an error has occurred, the ancillary qubit will be in the $|1\rangle$ state. Figure 12 demonstrates examples of Z stabilizer measurement. In this example, each measurement requires an ancillary qubit (initialized to $|0\rangle$ at the start of each round of measurements).

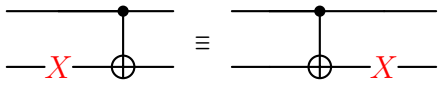
15.5 The canonical circuit

While the "musical" notation works well for visualizing circuits, the *algebra* we perform doesn't depend on such a visualization. The information (X/Z stab.) diagram needs to scale such that we can represent arbitrarily many qubits that connect to the single ancillary qubit. To do so, each qubit within a particular circuit component is denoted X_t or Y_t, such that each value of t is unique. This is known as the *canonical circuit*.

15.6 Exercises

Exercise 70. 🌶🌶🌶 *Prove that the circuit in figure 12 works as advertised.*

Exercise 71. 🌶🌶🌶 *Prove the "controlled X" identity:*



> *It was these we had in mind when explaining* stability...
> —Erwin Schröedinger (1944) [39]

16 Lattice surgery

16.1 The planar code

Given a Gaussian lattice $\mathcal{F} \subset \mathbb{Z}[i]$, the generated group $\mathcal{G} = \langle \mathcal{F} \rangle$ describes the *planar code*, (a kind of *surface* code) which we can use to build a two-dimensional quantum computer. If you were to take a classical integrated circuit out of its plastic packaging and hold it underneath an electron tunneling microscope, you'd most likely find a bunch of `NAND` gates cobbled together (which suffices to build a universal classical computer). Since we have to work a little harder than that to build a quantum computer, we begin our work here with a more abstract structure, where each point in the lattice corresponds to a physical qubit.

In that sense, the lattice works as a framework to hang things on. Suppose $C \subset \mathcal{G}$. A *plaquette* (or *tile*) P can be defined as any convex hull ∂C. We can now define the planar code as

$$\mathcal{S} = \bigcup_{i=1}^{\infty} P_i. \tag{61}$$

In most examples, ∂P_i is a square (such that $|\partial P_i| = 4$, or triangular, such that it's drawn as a half-moon shape. The collection of all the plaquettes is what we call the planar code: the fundamental parameter is the code distance d, which is its horizontal or vertical length, making each code square-shaped.

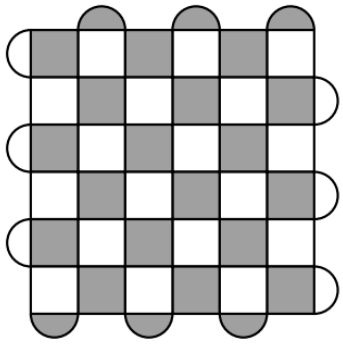


Figure 13: A distance 7 planar code. The gray and white tiles represent Z and X stabilizers, respectively. The corner of each plaquette contains a single data qubit, and each plaquette contains one measure qubit in its center [38].

If $P_i \bigcap P_j \neq \{\}$, we say that P_i and P_j "touch" (or, are *nearest neighbors*).

Lemma 2. *A planar code whose plaquettes satisfy the condition that, each nearest neighbor P_i and P_j are opposite colors, and have an even number or qubits in common, is regarded as* ***proper*** *(or standard) planar code.*

Theorem 5. *The planar code can correct $\left\lfloor \frac{d-1}{2} \right\rfloor$ topologically trivial errors per circuit run.*

16.2 Logical qubits

Due to the short-term longevity of physical qubits, we use $O(d^2)$ physical qubits (comprising plaquettes), to represent a single logical qubit, denoted $|\psi_L\rangle$. Importantly, the logical qubit is not a quantum mechanical object per se; it is a *combinatorial* object, defined entirely in terms of plaquettes.

The meaning of $|0_L\rangle$ is also rather subtle, because it doesn't correspond to a particular set of qubits, rather, we say that it is the simultaneous $+1$ eigenstates of all stabilizers contained in $\mathcal{S}$.

Example 2.

16.3 Logical operators

The standard planar code is a $d \times d$ array of logical qubits. If we know how to encode a (long-lived) logical qubit using one, the code can represent a single logical qubit. The key mathematical reason why all of this works is that Pauli operators are *anticommutative*, that is:

$$XZ = -ZX. \tag{62}$$

If a planar code comprises consecutive alternating Z and X stabilizers, we have

$$XZXZX = XZ(-ZX)X = -X$$

Since we have an odd number of plaquettes (or stabilizers) in one direction, so we end up with $-X$ or $-Z$ for any odd d. This means that in the presence of an X error, we end up with $-I$, which indicates the presence of an an X error [40].

16.4 Operator movement

Returning to the surface code, let's look more specifically at the effect of merging and splitting surface codes together. at what Figure 14 illustrates a basic example of operator movement.

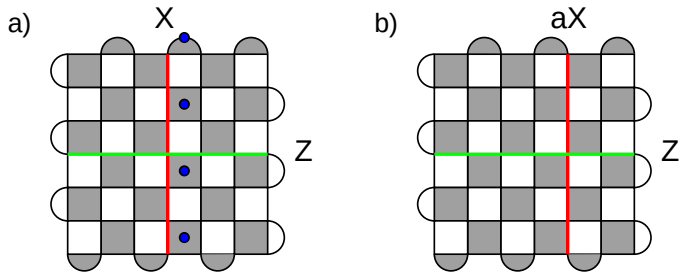


Figure 14: Performing operator movement. (a) The blue dots represent measurement qubits, and the green and red lines represent measurement along the Z and X axes. (b) The product aX represents the result of the Kronecker product of X stabilizer measurements across the blue measure qubits [41].

16.5 Logical CX

Now that we've built logical qubits and X and Z operators, we can now build more interesting logical gates, starting with CX. Figure 15 illustrates how we can represent a merged set of plaquettes into a single model. For each prism, the bottom represents the surface code; the upward dimension represents the arrow of time pointing upward; the X and Z stabilizers are traced. Starting from the bottom left of the figure and working out way up, the branch represents the possible results of the control qubit.

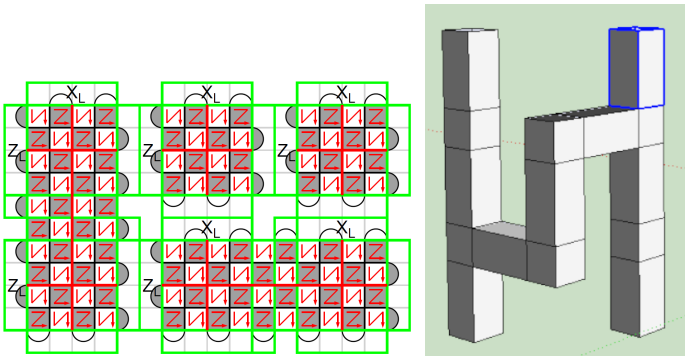


Figure 15: Representing CNOT as a set of five merged surface codes as a 3D model. [42]

16.6 Exercises

Exercise 72. 🌶🌶☝ *Write out the explicit simultaneous +1 eigenstates of a distance 3 surface code.*

> *This is a code because the real problem is the prevention of war.*
> —J. Robert Oppenheimer (1955) [43]

17 Syndrome analysis

Once we've implemented the surface code on physical hardware,[5] we can perform computations in an error-tolerant fashion. To do this, we must perform sequential measurements of the stabilizers: each measurement is known as a *round*, denoted as r. The data collected from each round is used to compute *syndrome measurements*, which tell use where errors occurred on the circuit; these data are streamed into a classical computer for further syndrome analysis.

17.1 Pauli frames

Imagine a quantum computer with four qubits, $[q_0, q_1, q_2, q_3]$, and a single round of stabilizer measurements are taken, wherein a Z error is detected on q_1, and an X error is detected on q_4. The corresponding *Pauli frame* for this set of detection events is $IXIZ$. This information is used in each round to update the *logical Pauli frame*. When an odd number of X or Z errors have been detected in a given physical Pauli frame, the logical Pauli frame is updated to reflect the presence of a logical X error.

As demonstrated in Figure 16 a $|0_L\rangle$ state, then applying a logical X operator. This transformation occurs from left to right, since it's touching the dark plaquettes; when an X error has been tracked, we note that in the logical Pauli frame, and update $|0_L\rangle$ to $X|0_L\rangle = |1_L\rangle$. The green boxes are known as *templates*: a single 4x4 template in the center containing the square plaquettes, as well as 1x4 templates containing the "triangular" half-moon plaquettes. [42]

[5] Not a trivial task, of course, but we'll later see how we can simulate one.

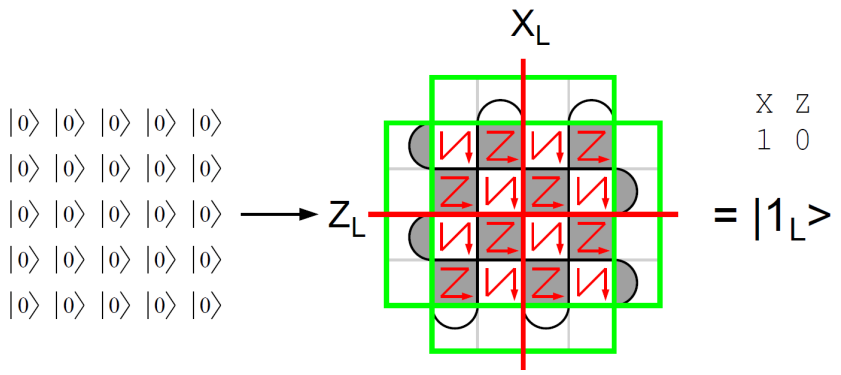


Figure 16: Representing a $|1_L\rangle$ state via a Pauli frame $[1, 0]$; measuring in the Z basis can be represented as multiplying by -1, since $Z|1\rangle = -|1\rangle$.

17.2 Minimum weight perfect matching (MWPM)

A quantum computer can't do its job alone; it needs a classical co-processor to analyze its measurement results. MWPM is one such classical algorithm, which analyzes detection events to correct errors.

17.2.1 Detection events

A *detector* is a parity of syndrome measurement bits in a quantum error correction circuit. That means we can measure a handful of syndrome outcomes, add them up to find parity of the result modulo 2. A *detection event* is a detector measurement of outcome 1.

17.2.2 The algorithm

MWPM is a graph algorithm that maps a graph $G = (V, E)$ to an edge subset $E' \subseteq E$, whereby E' is a *perfect matching* of E, such that

$$W = \sum_{e \in E'} w(e) \tag{63}$$

is minimized. E' is a perfect matching if each vertex in V is connected to exactly one edge in E'.

17.2.3 Putting it all together

MWPM for TQEC decomposes the error model by X and Z errors. Each detection event is modeled as a vertex in G, and for each edge $e = (u, v)$, $w(e)$ indicates the prior probability of detection event u flipping v. Thus, E' tells us which chain of detection events we can trust most when correcting errors in a circuit execution.

17.2.4 Implementation concerns

Minimizing W is a challenging problem; an intuitive approach is to use Dijkstra's algorithm on G repeatedly. More involved approaches utilize physical data to prune away unlikely candidates for the matching, creating a more focused search space, paving the way for amortized $\mathrm{O}(1)$ execution of the algorithm in parallel [44]. The *sparse blossom* approach utilizes this pruning, as well as linear programming, to performing the algorithm in microsecond time on a single core. [45]

17.3 Exercises

Exercise 73. 🌶🌶🌶 *Show that the naive implementation of MWPM performs in* $\Omega(|V|\log|V|)$ *time.*

Exercise 74. 🌶🌶🌶 *Prove that, given a reasonable set of physical assumptions, MWPM can be performed in* $\Theta(1)$ *time on average.*

> *As regards the specification of the conditions for any well-defined application of the formalism, it is moreover essential that the whole experimental arrangement be taken into account.*
> —Niels Bohr (1949) [46]

18 Software applications

The goal of this section is to detail the open-source software that implements the theory of TQEC to enable the editing and evaluation of error-tolerant quantum circuits.

18.1 `TQEC.app`

This user interface allows the user to specify a *qubit unit cell* $\mathcal{F}$, which is a Gaussian lattice generator laid out on a grid with integer lengths, and a set of grid spaces that contain each qubit in $\langle \mathcal{F} \rangle$. The app then populates the viewport with qubits according to the unit cell; the user may then specify plaquettes, along with a canonical-form circuit within them. The backend service will utilize Stim to help build and validate the users' circuits. The author first envisioned this very work as an "instruction manual", but that will materialize as written and video walkthroughs of the app.[6] Figure 17 illustrates the creation of a circuit using the app.

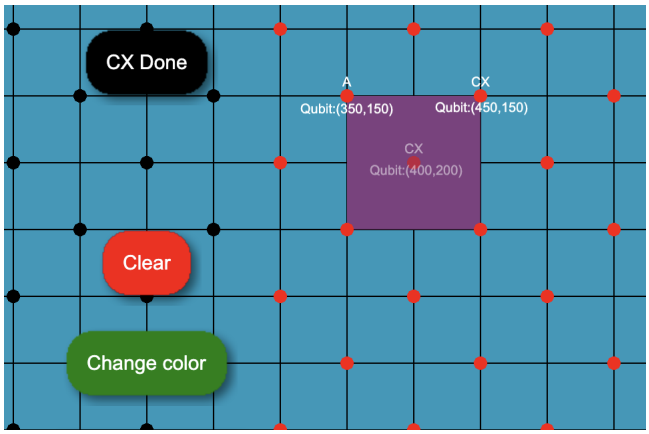


Figure 17: Editing a plaquette using `TQEC.app`. The red and black qubits are laid out according to the standard two-dimensional unit cell. Each plaquette contains an ancillary (measure) qubit, and some number of CX and CZ qubits, which comprise a canonical-form circuit.

The basic user journey for specifying canonical circuits via plaquettes is as follows:

[6] The site is under active development, with new features added on a regular basis [47].

1. Specify a unit cell using a constellation of qubits $C \subset \mathbb{Z}[i]$ and an $l \times w$ rectangular *footprint* (or, bounding box) $B(C)$.

2. Check each qubit in ∂C, which are granted coordinates relative to the footprint's top-left corner $(0, 0)$, like so: if there exists an $(x, 0) \in C$ such that $(x, l) \in C$, or a $(0, y)$ such that $(w, y) \in C$, then ask the user to eliminate some of the boundary qubits.

3. The app generates $\langle C \rangle$ generates a Gaussian lattice which tessellates the entire viewport with the qubits specified in the previous step. The footprint is left in place to designate the initial circuit workspace.

4. Each cell of the grid, denoted C_i, may contain any number of plaquettes $P \subset C_i$. Two plaquettes P_i and P_j may only overlap if and only if $P_i \bigcap P_j = \partial P_i \bigcap \partial P_j$.

5. A plaquette must contain at least 3 qubits (including one ancilla) in order to specify a canonical circuit within it.

6. Each qubit in C_i can be annotated with either X_n or Y_m, where n and m represent the time steps that the corresponding stabilizer runs at in the quantum circuit.

7. Each footprint C_i can be saved into the user's personal library for future use. Footprints can also be collated into *templates* of length k.

Any qubits left unlabeled will be compacted out of the final circuit. This procedure suffices to build any two-dimensional quantum computer using a quadrilateral unit cell. Figure 18 demonstrates the finished version.

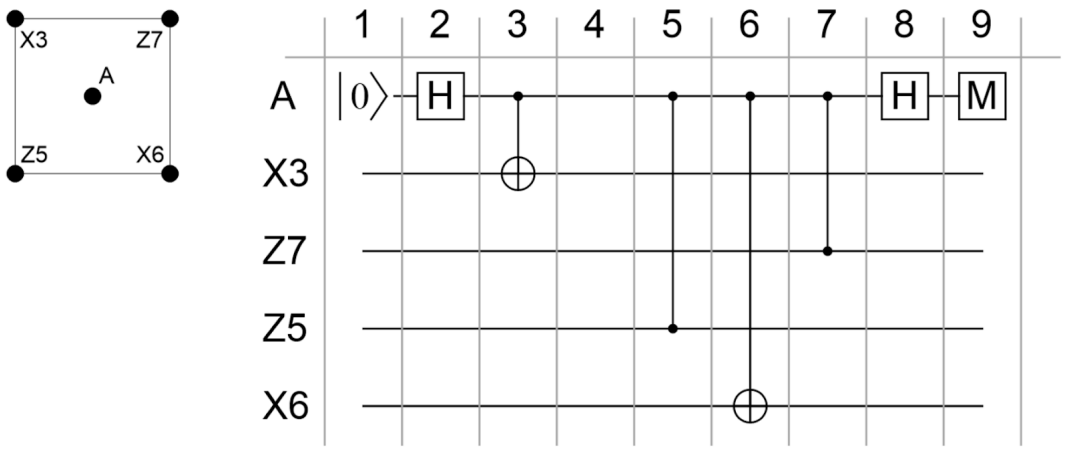


Figure 18: A canonical-form plaquette (left) and its corresponding circuit (right). Each column represents a time-step, and each row represents a qubit selected into the plaquette, with relative orderings indicated.

18.2 Stim

Stim is an open-source Python library that offers an API for simulating surface codes, wherein the user supplies parameters such as d, r, and error probability p. It generates circuits in a format per a custom language reminiscent of OpenQASM [48]. The library was designed with real-time performance in mind, making it useful for researchers and developers in the field [49]. The following code snippet illustrates a basic use case:

```
stim.Circuit
        .generated
                (
                "surface_code:rotated_memory_z",
                distance=d,
                rounds=r,
                after_clifford_depolarization=p,
                before_round_data_depolarization=p,
                before_measure_flip_probability=p,
                after_reset_flip_probability=p
                )
```

Stim is the highest-performing surface code simulator available; its core functionality is written in C++. It exports files to a special format as a pictorial

representation of the circuit. The developer documentation comes highly recommended as well.

> *You should really just use Stim.*
> —Unknown (2023)

18.3 Exercises

Exercise 75. 🌶🌶🌶 *What is the distance of a surface code in terms of k?*

Exercise 76. 🌶🌶🌶 *Suppose we made a surface code comprising connected square plaquettes which are made of five qubits: one for each corner, and one in the middle. Show that this surface code meets the user journey criteria.*

Exercise 77. 🌶🌶🌶 *Modify the user journey to permit hexagonal unit cells.*

Computers are made up of logic gates that stretch out to the horizon in a vast numerical irrigation system.
—Stan Augarten (1983) [50]

I don't think these are the errors that the title was referring to.
—Jarrett Green (2023)

References

1. Vogel, O. *MusiXTEX: Using TEX to write polyphonic or instrumental music* (2023).
2. Kay, A. *Tutorial on the Quantikz Package* 2023. arXiv: `1809.03842[quant-ph]`.
3. Mina Aganagić Costin Popescu, J. H. S. Gauge-invariant and gauge-fixed D-brane actions. *Nuclear Physics B*(1997).
4. Tolkien, J. R. R. *The Lord of the Rings* (Allen & Unwin London, 1954).
5. Knuth, D. E. *The Art of Computer Programming. Fundamental Algorithms* ISBN: 9780201896831 (Addison-Wesley, 1997).
6. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum***2,** 79. ISSN: 2521-327X(Aug. 2018).
7. Google. Suppressing quantum errors by scaling a surface code logical qubit. *Nature***614,** 676–681(2023).
8. Fowler, A. G. *TQEC design automation Coursera course* 2024. `https://bit.ly/3x0hIg7`.
9. Nielsen, M. A. Chuang, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2010).
10. Yanofsky, N. S. Mannucci, M. A. *Quantum Computing for Computer Scientists* (Cambridge University Press, 2008).
11. Lidar, D. A. Brun, T. A. *Quantum error correction* (Cambridge university press, 2013).
12. Box, G. E. P. Science and Statistics. *Journal of the American Statistical Association***71,** 791–799.
13. Nahin, P. J. *An Imaginary Tale: The Story of* $\sqrt{-1}$ (Princeton University Press, 2008).
14. Euler, L. *Elements of Algebra* ISBN: 9781847286475 (Lulu Press, Incorporated, 2006).
15. Wigner, E. The Unreasonable Effectiveness of Mathematics in the Natural Sciences. *Communications on Pure and Applied Mathematics***13**(1960).
16. Beezer, R. A. *A First Course in Linear Algebra* ISBN: 0984417559 (Congruent Press, 2012).

17. Bell, E. T. *Men of Mathematics* (1986).

18. Hamilton, W. *Elements of Quaternions* (Longmans, Green, & Company, 1866).

19. Graham, R. L., Knuth, D. E. Patashnik, O. *Concrete Mathematics: A Foundation for Computer Science*2nd (Addison-Wesley, Reading, MA, 1994).

20. Knuth, D. E. *The Art of Computer Programming, Volume 4B: Combinatorial Algorithms, Part 2* ISBN: 9780137926817 (Addison-Wesley, 2022).

21. Knuth, D. E. *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1* ISBN: 0201038048 (Addison-Wesley, 2011).

22. Wolrfram, S. *A New Kind of Science* (2002).

23. Heisenberg, W. Über quantentheoretische Umdeutung kinematischer und mechanischer Beziehungen. *Zeitschrift für Physik***33,** 879–893(1925).

24. Viswanath, P. *Quantum States And The Bloch Sphere* 2021. `https://medium.com/quantum-untangled/quantum-states-and-the-bloch-sphere-9f3c0c445ea3`.

25. Spivey, M. *The Art of Proving Binomial Identities* (CRC Press, 2019).

26. Judson, T. W. *Abstract Algebra: Theory and Applications* (Orthogonal Publishing L3C, 2022).

27. Klein, F. Ackerman, M. *Development of Mathematics in the 19th Century* (Math Sci Press, Brookline, Massachusetts, 1979).

28. Kitaev, A. Anyons in an exactly solved model and beyond. *Annals of Physics***321,** 2–111. ISSN: 0003-4916(Jan. 2006).

29. Dirac, P. A. M. *Unitary representations of the Lorentz group* 1944.

30. Hadamard, J. An Essay on the Psychology of Invention in the Mathematical Field(1945).

31. Kranz, G. *Failure is Not an Option. Mission Control from Mercury to Apollo 13 and Beyond* (Simon & Schuster, 2001).

32. Feynman, R. P., Leighton, R. B. Sands, M. *The Feynman Lectures on Physics. Quantum Mechanics* chap. 2 (Basic Books, 1965).

33. Shor, P. W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing***26,** 1484–1509. ISSN: 1095-7111(Oct. 1997).

34. Clifford, W. K. Applications of Grassmann's Extensive Algebra. *American Journal of Mathematics***1,** 350–358(1878).

35. Bell, J. S. On the einstein podolsky rosen paradox. *Physics***1,** 195(1964).

36. Maurice Herlihy Dmitry Kozlov, S. R. *Distributed Computing Through Combinatorial Topology* ISBN: 0124045782 (Morgan Kaufman, 2013).

37. Pauli, W. *The Influence of Archetypal Ideas on Kepler's Theories* (Pantheon Books, 1955).

38. Fowler, A. G., Whiteside, A. C., McInnes, A. L. Rabbani, A. Topological Code Autotune. *Physical Review X***2.** ISSN: 2160-3308(Oct. 2012).

39. Schrödinger, E. *What Is Life?* (Cambridge University Press, 1944).

40. Fowler, A. G., Mariantoni, M., Martinis, J. M. Cleland, A. N. Surface codes: Towards practical large-scale quantum computation. *Physical Review A***86.** ISSN: 1094-1622(Sept. 2012).

41. Fowler, A. G. Gidney, C. Low overhead quantum computation using lattice surgery. arXiv: `1808.06709[quant-ph]`(2019).

42. Fowler, A. G. *Logical CNOT* 2024.

43. Oppenheimer, J. R. *The Open Mind* (Simonand Schuster, 1955).

44. Fowler, A. G. Minimum weight perfect matching of fault-tolerant topological quantum error correction in average $O(1)$ parallel time. arXiv: `1307.1740[quant-ph]`(2014).

45. Higgott, O. Gidney, C. Sparse Blossom: correcting a million errors per core second with minimum-weight matching
. arXiv: `2303.15933[quant-ph]`(2023).

46. Bohr, N. *Atomic Physics and Human Knowledge* (1958).

47. *TQEC Visualizer App* 2024. `https://tqec.app`.

48. Gidney, C. *The Stim Circuit File Format (.stim)* `https://bit.ly/43sFEVn`.

49. Gidney, C. Stim: a fast stabilizer circuit simulator. *Quantum***5,** 497. ISSN: 2521-327X(July 2021).

50. Augarten, S. *State of the Art: A Photographic History of the Integrated Circuit* ISBN: 0899191959 (Ticknor & Fields, 1983).

It's never going to be perfect; you simply run out of time.
—Peter Jackson (2003)