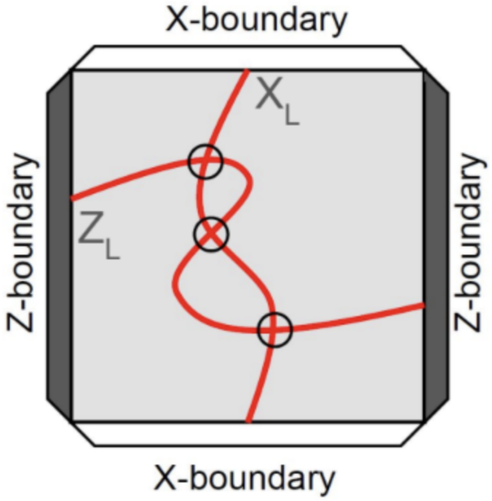
A survey of topological quantum error correction

Sam Burdick

Abstract

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



Version 0.1 [March 29, 2024]

The musical notation and quantum circuit example in the Appendix were taken respectively from the tutorials [1] and [2]. The image in the preface was generated by AI.

The content of this work, including this disclaimer, is subject to change at any time without prior notice. The author makes no guarantees regarding the accuracy, reliability, or completeness of the information provided. Except for fair use and educational purposes, no part of this work may be reproduced or distributed without the owner's express written permission. The owner assumes no liability for the code examples and screenshots provided in this work. Any examples are intended for demonstration purposes and may not be suitable for all applications. TQECoin™ is an experimental cryptocurrency created by the owner. Its value is not guaranteed and may fluctuate significantly or even decrease to zero. You may earn TQECoin rewards for accurately identifying and correcting errors in this work. The amount of cryptocurrency rewarded will monotonically increase over time at the owner's discretion. This work and any information about TQECoin does not constitute financial, investment, tax, or legal advice. Consult your own professional advisor(s) before making any decisions regarding cryptocurrencies. Since TQECoin is a new project, carefully research its potential risks and uncertainties before accepting it. You are responsible for securing your TQECoin. Use reputable wallets, enable strong security measures, and never share your private keys. TQECoin rewards may have tax implications. Consult a tax advisor to understand your obligations in your jurisdiction. Any offers may be restricted based on your location or other factors. Ensure you are eligible to participate before accepting any TQECoin rewards. There is no guarantee of the future value of TQECoin. The owner makes no representations or warranties about its potential appreciation or depreciation.

1 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 update this work to the best of their ability.

If you've ever written a for loop that adds up a list of numbers of length n, for $1 \leq n \leq m$, then at some point you may have been shown the mathematical way of explaining it:

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

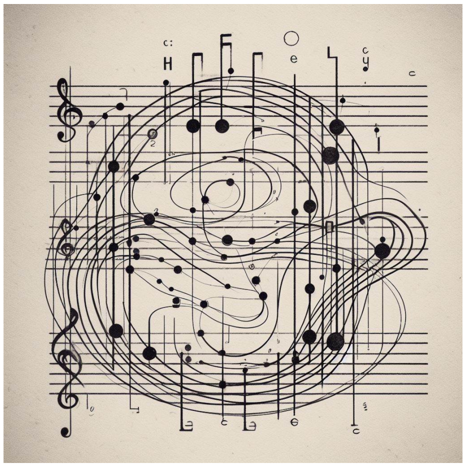
But, the math you learned beforehand was about getting you to learn *calculus*, which doesn't really use such machinery for the most part. I'd like to assure you that the complexity of the mathematical statements herein are no more complex than equation and have a similar flavor.

The only way to truly add a tool to your kit is to practice using it. For that, the appendices offer exercises to help you check your understanding. Difficulty rankings denote each exercise, making this work suitable for self-study.

In conclusion, this is not a quantum mechanics text, it is a computer science text.

2 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.



An author cannot of course remain unaffected by the sum of his experiences...
—J. R. R. Tolkien (1954) [3]

3 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 is 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, based in western Washington state, 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.

The precise details of this system are still in the works; I intend to publish the entire source code I used to mine the blocks, as well as the initial proof-of-work of the blockchain; additional hashes used in the blockchain can be salted with the text of the contribution. The system shall be fully up by December 31, 2024. As a caveat, if there are legal reasons why this cannot be done, an alternative bounty system shall be implemented. Regardless, any error corrections will be acknowledged; I permit anonymity as well.

Happy hunting!

—S. M. B.

Harambee ("Let us all pull together")
—National motto of Kenya

Everything is difficult in the beginning.
—Chinese proverb

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

Contents

4 Introduction

Intense hardware error rates severely limit the practicality of quantum computers [4]. 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 [5]. 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 *surface 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 surface code.

The supplements at the end 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 [6], and two helpful textbooks are [7] and [8]. For a deeper dive into the field of quantum error correction, see [9]. 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.

5 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 [7]. Fortunately, every quantum error can be modeled as a linear combination of *Pauli errors*

$$P = \{I, X, Z, XZ\}.^{3} \tag{2}$$

Lemma 1. *The set P is an orthonormal basis for all 2x2 matrices.*

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 $XX = ZZ = I$ means that, if we apply an X gate in the presence of an X error, it's as if nothing happened to the qubit (and likewise for Z).

Theorem 1. *If we can correct X and Z errors, we can correct any local quantum error.*

[2] Also known as the *toric code* in the literature.

[3] We include I for completeness; XZ can be written as $e^{-i\pi/2}Y$, where $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ is the *other* kind of Pauli error.

To see why, consider the following matrix decomposition:

$$B = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \implies B = \frac{a+d}{2} I + \frac{b+c}{2} X + \frac{a-d}{2} Z + \frac{b-c}{2} XZ. \tag{3}$$

Therefore, we refer to B as an *error 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:

$$X|0\rangle = |1\rangle \qquad Z|0\rangle = |0\rangle \tag{4}$$

$$X|1\rangle = |0\rangle \qquad Z|1\rangle = -|1\rangle \tag{5}$$

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

We disregard non-Pauli quantum errors, particularly *global* errors, in this discussion.[4] Stabilizer measurements can correct local errors, which is the focus of *topological* quantum error correction.

> *In fact, if one proceeds on this basis it hardly appears possible to avoid the empirically untenable conclusion...*
> —Wolfgang Pauli (1955) [10]

6 Stabilizers

6.1 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

$$A|\psi\rangle = |\psi\rangle. \tag{6}$$

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.

6.1.1 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 $|\psi\rangle$.

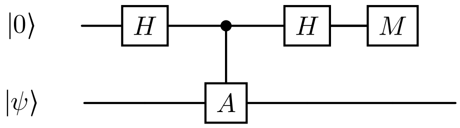


Figure 1: The A circuit, which is not a stabilizer.

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

[4] Global errors affect *global phase*, and represent a factor of $e^{i\gamma}$ which can be ignored during any modularization of probability amplitude, since $|e^{i\gamma}z| = |z|$; global phase is *unobservable*.

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

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:

$$|\psi\rangle \pm A\,|\psi\rangle \tag{7}$$

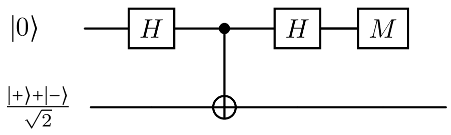
We'll want to be a bit more specific when we build actual circuits.

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.

6.2 Implementation

6.2.1 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 7:

$$0(|\psi\rangle + X\,|\psi\rangle) + 1(|\psi\rangle - X\,|\psi\rangle) \tag{8}$$

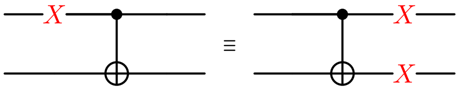
Suppose we were to build a quantum circuit with a state $|\phi\rangle$ representing the data we want to preserve, and assume an X error were to act on the state $|\phi\rangle$, which we write as

$$|\psi\rangle = X\,|\phi\rangle = \frac{|+\rangle + |-\rangle}{\sqrt{2}}. \tag{9}$$

To summarize, this stabilizer has a dual purpose:

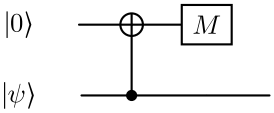
- To *detect* quantum errors, which are indicated by a 1 measurement;
- To *correct* quantum errors; this circuit can remove a single X error that occurred before the circuit ran.

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

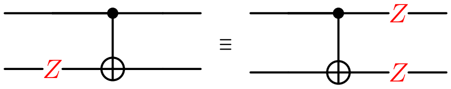


In this case, the X error is propagated into $|\psi\rangle$, however, we're able to *detect* the fact that an X error occurred at this point!

6.2.2 The Z stabilizer



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



> *It was these we had in mind when explaining* stability...
> —Erwin Schröedinger (1944) [11]

7 Surface codes

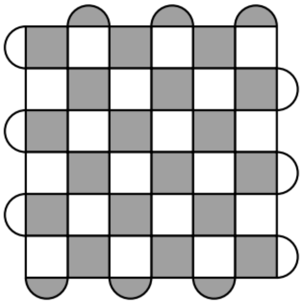


Figure 2: A distance 7 surface code. The gray and white tiles represent Z and X stabilizers, respectively. The corner of each plaquette contains a single data qubit, and each plaquette contains one measure qubit in its center [12].

A surface code is a mathematical object used to describe a *scalable* (error-tolerant) quantum circuit. A 2D surface code is represented as a checkerboard, where dark *plaquettes* (spaces) are Z stabilizers, and white plaquettes are X stabilizers. The defining parameter of the surface code is its distance d, which determines the number of plaquettes in the horizontal and vertical directions. Surface codes can be manipulated and scaled into larger codes through lattice surgery; we'll cover the steps needed to do that and provide some examples.

7.1 Logical states and operators

A surface code can be prepared in the logical $|0\rangle$ or $|1\rangle$ states, denoted $|0_L\rangle$ and $|1_L\rangle$; thus a surface code represents a single *logical qubit*. Likewise, it can be prepared in $|+_L\rangle$ and $|-_L\rangle$.

7.2 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.

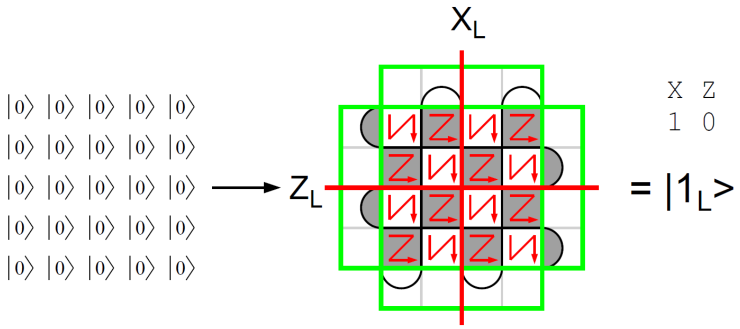


Figure 3: Preparing a $|0_L\rangle$ state, then applying a logical X operator. This transformation occurs from left to right, since it's touching the dark plaquettes; when an X error has been tracked, we note that in the logical Pauli frame, and update $|0_L\rangle$ to $X\,|0_L\rangle = |1_L\rangle$. 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. [13]

7.3 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 4 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 control qubit

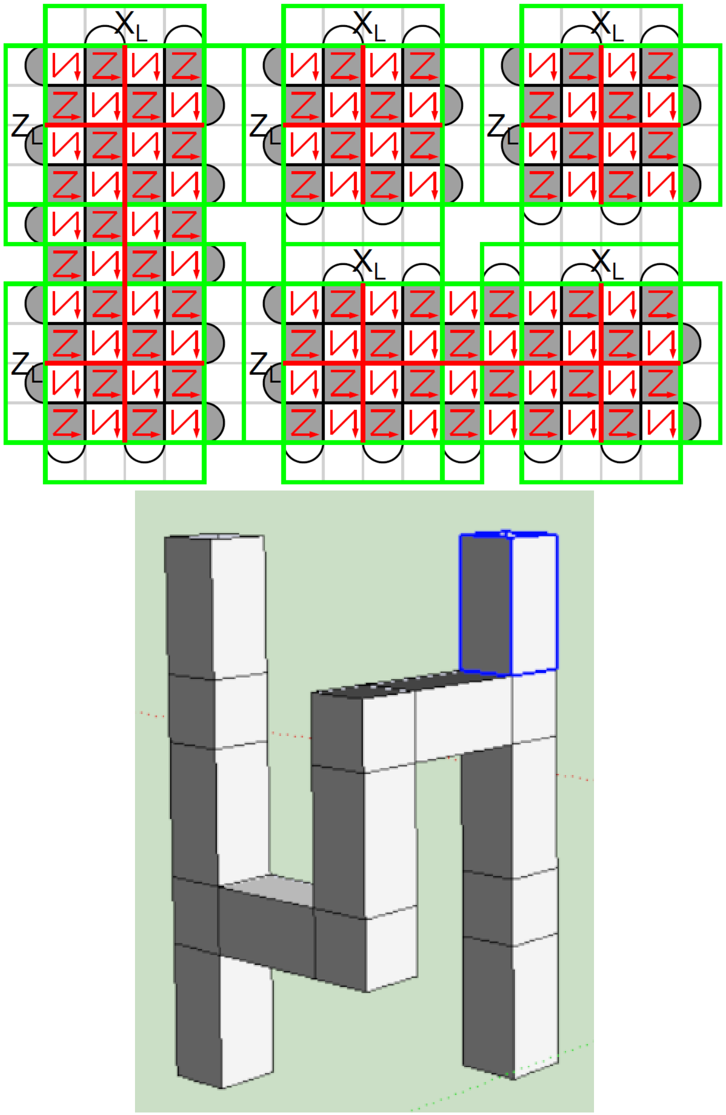


Figure 4: Representing a set of five merged surface codes as a 3D model [13].

7.4 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 5 illustrates a basic example of operator movement.

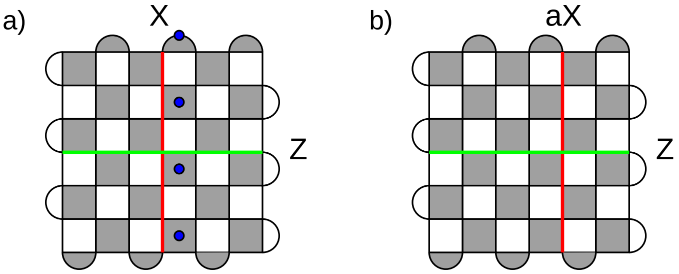


Figure 5: Performing operator movement. (a) The blue dots represent measurement qubits, and the green and red lines represent measurement along the Z and X axes. (b) The product aX represents the result of the product of X stabilizer measurements across the blue measure qubits [14].

7.5 Rationale

Why are plaquettes laid out in such a fashion? Because Pauli operators are *anticommutative*, that is,

$$XZ = -ZX. \tag{10}$$

If we have five consecutive alternating Z and X stabilizers:

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

Usually, we have an odd number of 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. [15].

> *This is a code because the real problem is the prevention of war.*
> —J. Robert Oppenheimer (1955) [16]

8 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.

It should be noted that the 2D grid construction is known as a *standard* (formerly, *rotated*) surface code; other architectures, such as "heavy hex," have been demonstrated. [17].

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

8.1 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.

8.1.1 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.

8.1.2 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

$$W = \sum_{e \in E'} w(e) \tag{11}$$

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

8.1.3 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.

8.1.4 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 O(1) execution of the algorithm in parallel [18]. The *sparse blossom* approach utilizes this pruning, as well as linear programming, to performing the algorithm in microsecond time on a single core. [19]

> *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) [20]

9 Stim

9.1 Overview

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 [21]. The library was designed with real-time performance in mind, making it useful for researchers and developers in the field [22].

9.2 How to use

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. It exports files to a special format as a pictorial representation of the circuit.

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

10 User interfaces for TQEC

10.1 TQEC.app

This website allows the user to specify a *unit cell*, which is a lattice of qubits Q laid out on a grid with integer lengths, and a set of grid spaces that contain each qubit in Q. 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 6 illustrates the creation of a circuit using the app.

> *Everyone seems to want user interface but they are not sure whether they should order it by the yard or by the ton.*
> —Alan Kay (1996) [24]

[6] The site is under active development, with new features are added on a regular basis [23].

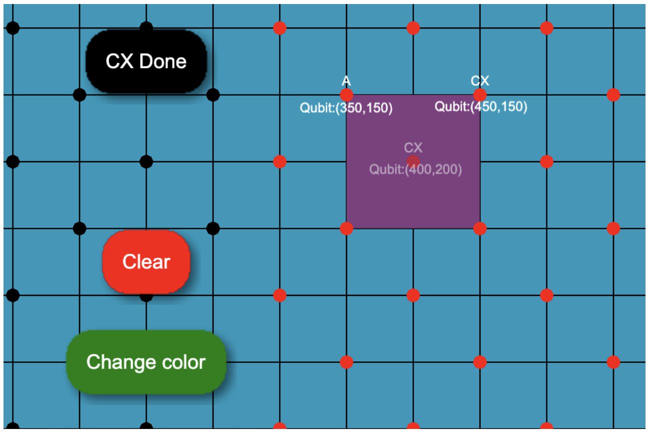


Figure 6: Editing a plaquette using `TQEC.app`. The red and black qubits are laid out according to the standard two-dimensional unit cell. Each plaquette contains an ancillary (measure) qubit, and some number of CX and CZ qubits, which comprise a canonical-form circuit [**TQEC.app**].

11 Conclusion

This work reviewed the core elements of TQEC, demonstrating its efficacy for realizing quantum computers. It reviews the core mathematical theory, such as Pauli errors, stabilizer measurements, and surface codes. It also discussed practical implementation concerns.

> *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) [25]

12 Acknowledgements

The author would like to thank Unitary Fund, for their generous support of the TQEC open–source project; Austin Fowler, for his mentorship, and for drawing many of the figures used in this work; and the individual contributors 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 (1997) [26]

13 Works cited

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. Tolkien, J. R. R. *The lord of the rings trilogy* (Allen & Unwin London, 1954).
4. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum***2,** 79. ISSN: 2521-327X(Aug. 2018).
5. Google. Suppressing quantum errors by scaling a surface code logical qubit. *Nature***614,** 676–681(2023).
6. Fowler, A. G. *TQEC design automation Coursera course* 2024. `https://bit.ly/3x0hIg7`.
7. Nielsen, M. A. Chuang, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2010).
8. Yanofsky, N. S. Mannucci, M. A. *Quantum Computing for Computer Scientists* (Cambridge University Press, 2008).
9. Lidar, D. A. Brun, T. A. *Quantum error correction* (Cambridge university press, 2013).
10. Pauli, W. *The Influence of Archetypal Ideas on Kepler's Theories* (Pantheon Books, 1955).
11. Schrödinger, E. *What Is Life?* (Cambridge University Press, 1944).
12. Fowler, A. G., Whiteside, A. C., McInnes, A. L. Rabbani, A. Topological Code Autotune. *Physical Review X***2.** ISSN: 2160-3308(Oct. 2012).
13. Fowler, A. G. *Logical CNOT* 2024.
14. Fowler, A. G. Gidney, C. Low overhead quantum computation using lattice surgery. arXiv: `1808.06709[quant-ph]`(2019).
15. 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).
16. Oppenheimer, J. R. *The Open Mind* (Simonand Schuster, 1955).
17. Benito, C., López, E., Peropadre, B. Bermudez, A. Comparative study of quantum error correction strategies for the heavy-hexagonal lattice. arXiv: `2402.02185[quant-ph]`(2024).
18. 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).

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

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

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

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

23. *Topological quantum error correction tools* `https://bit.ly/43sFEVn`.

24. Kay, A. C. *The early history of Smalltalk* chap. History of programming languages—II (TODO, 1996).

25. Knuth, D. E. *The Art of Computer Programming. Fundamental Algorithms* ISBN: 9780201896831 (Addison-Wesley, 1997).

26. Mina Aganagić Costin Popescu, J. H. S. Gauge-invariant and gauge-fixed D-brane actions. *Nuclear Physics B*(1997).

27. *The Lord of the Rings Special Extended DVD Edition: The Appendices* 2003.

28. Beezer, R. A. *A First Course in Linear Algebra* ISBN: 0984417559 (Congruent Press, 2012).

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

30. Box, G. E. P. Science and Statistics. *Journal of the American Statistical Association* **71,** 791–799.

31. Nahin, P. J. *An Imaginary Tale: The Story of* $\sqrt{-1}$ (Princeton University Press, 2008).

32. Euler, L. *Elements of Algebra* ISBN: 9781847286475 (Lulu Press, Incorporated, 2006).

33. Wigner, E. The Unreasonable Effectiveness of Mathematics in the Natural Sciences. *Communications on Pure and Applied Mathematics* **13**(1960).

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

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

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

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

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

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

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

41. 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).

42. Toffoli, T. Reversible Computing(1980).

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

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

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

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

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

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

49. Green, J. `https://www.ign.com/articles/quantum-error-review`(2023).

It's never going to be perfect; you simply run out of time.
—Peter Jackson (2003) [27]

14 Supplement: mathematical preliminaries for TQEC

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); graph theoretic algorithms. You may think of this section 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. Moreover, you will find various exercises strewn about, with approximate degrees of *spiciness* indicated:

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

> *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)

14.1 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) [28]

14.1.1 Matrices

Matrices are multi-dimensional arrays of numbers. In quantum error correction, when studying the particulars of matrices, we focus on 2x2 matrices, written as:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}. \tag{12}$$

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.[7] We'll now defy your expectations by showing off a widely-used matrix known as the *identity matrix*:

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

[7] As opposed to A and B being *anti-commutative*, which would mean $AB = -BA$.

Multiplying any matrix A by I produces A:

$$AI = IA = A$$

Exercise 1. ◌◌◌*Compute* $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 5 \\ 6 \end{pmatrix}$.

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

$$A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \tag{13}$$

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

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

14.1.2 Vectors

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

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \tag{14}$$

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:

$$\lambda |\psi\rangle = \lambda \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \lambda\alpha \\ \lambda\beta \end{pmatrix} \tag{15}$$

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

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} a\alpha + b\beta \\ c\alpha + d\beta \end{pmatrix} \tag{16}$$

> *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) [29]

14.1.3 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 set of vectors

$$\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}^{8} \tag{17}$$

is a basis of $\mathbb{C}^2$, since any two-dimensional vector can be written in terms of those. If $|\mathcal{B}|$ is minimal, we say $\mathcal{B}$ is *orthonormal*.

Exercise 3. 🌶🌶◌*Prove Lemma 1.*

[8] Known as the *Euclidean* basis vectors.

14.2 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:

$$\sum_{E \in \Omega} p(E) = 1 \tag{18}$$

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

> *Since all models are wrong the scientist must be alert to what is importantly wrong.*
> —George E. P. Box (1976) [30]

14.3 Complex variables

Before we can define a complex number, we need to review *real* numbers, which in the physical sciences can be thought of as the result of a measurement. They include $17/4, 2.71828, \pi$, and so on; the set containing all of them is written as $\mathbb{R}$.

You may remember being asked to find the solutions to polynomials, 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 ± 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: $i = \sqrt{-1}$ [31].

Complex numbers are thus compactly defined as:

$$\mathbb{C} = \{a + bi : a, b \in \mathbb{R}, i = \sqrt{-1}\}. \tag{19}$$

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

The *modulus* of a complex number $z = a + bi$ is equal to

$$|z| = \sqrt{a^2 + b^2}. \tag{20}$$

Geometrically, it is the distance from the origin $(0,0)$ to the point (a,b) in Cartesian space, as illustrated in Figure 7.

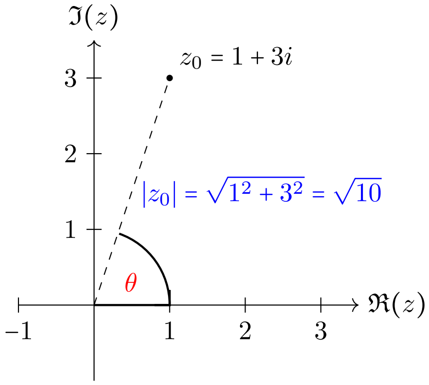


Figure 7: An illustration of the complex modulus (denoted $|z_0|$) against the *complex plane*; the horizontal axis represents the "real part" of a complex number z, denoted $\Re$, while the horizontal axis represents the imaginary part, denoted $\Im$.

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 7, it's denoted as θ. We may now define a complex number in its *polar form*:

$$z = re^{i\theta} = r(\cos\theta + i\sin\theta) \tag{21}$$

where $r = |z|$, and $e = 2.71828\ldots$ is a special number.[9] Letting $r = 1$ and $\theta = \pi$, we're left with *Euler's identity*

$$e^{i\pi} + 1 = 0, \tag{22}$$

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

Exercise 5. 🌶🌶🌶*Show that $zz^* = |z|$ for any $z \in \mathbb{C}$.*

Exercise 6. 🌶🌶🌶*Prove de Moivre's theorem: for any $z \in \mathbb{C}$ and $n \in \mathbb{N}$, $z^n = r^n(\cos n\theta + i\sin n\theta)$*

Exercise 7. 🌶🌶🌶*Prove the well-known identity*

$$\int_{\mathbb{R}} e^{-x^2} dx = \frac{\sqrt{\pi}}{2}. \tag{23}$$

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

> *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...*
> —Leonhard Euler (1770) [32]

[9] One of the most special in all of mathematics and science indeed.

> *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) [33]

14.3.1 Eigenvalues

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 8. 🌶🌶*Find the eigenvalues and eigenstates of X and Z.*

> *...dais sich das schwingende Elektron gegeniiber Licht, das viel kurzwelliger ist sis alle Eigenschwingungendes Systems, wie ein freies Elektron verhlt.*[10]
> —Werner Heisenberg (1925) [34]

14.4 Qubits defined

In the quantum computing literature, $|\psi\rangle$ and sometimes $|\phi\rangle$ are often used to denote quantum bits, or *qubits*. Equation 14 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

$$|\psi\rangle\,|\phi\rangle = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ v_1 \\ v_2 \end{pmatrix} \tag{24}$$

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

Exercise 9. 🌶*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)

14.5 Superposition and basis vectors

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

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{25}$$

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

[10] *...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)

$$|\alpha|^2 + |\beta|^2 = 1. \tag{26}$$

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 $0 \leq p \leq 1$. 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* together.

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 (25) describes the superposition of $|\psi\rangle$ in Z basis vectors.

Exercise 10. *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) [35]

15 Supplement: quantum gates

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

15.1 Unitary matrices

A unitary matrix U is any matrix that is *self-adjoint.* The adjoint of a matrix is equal to the *conjugate transpose*, which we find by swapping the rows and columns (which yields the transpose), and taking the conjugate of each element. We denote the adjoint with the “dagger" symbol, †:

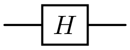
$$U^\dagger = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^\dagger = \begin{pmatrix} a^* & c^* \\ b^* & d^* \end{pmatrix} \tag{27}$$

Every quantum computing operator, both as an error and a gate, is represented by a unitary matrix.

Exercise 11. *Prove the self-adjoint property of any unitary matrix: $UU^\dagger = I$.*

> *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) [36]

15.2 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:

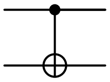
$$\begin{aligned} |+\rangle &= H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\ |-\rangle &= H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \end{aligned}$$

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 12. 🌶🌶🌶*Prove that H squares to the identity,*[11] *meaning $HH = I$.*

> *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) [37]

15.3 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:[12]

$$X|\psi\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}. \tag{28}$$

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.[13] 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

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

[11] H is also known as an *involutary* matrix.

[12] The row vectors are $\{|-\rangle, |+\rangle\}$.

[13] Look closely at the diagram: it's a literal target!

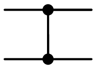
As with H, the matrix representation of CX is involutary and unitary. The matrix representation CNOT *tensor product* (or *Kroenecker* product) $I \otimes X$, which we can define more precisely as[14]

$$A \otimes B = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \otimes \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} = \begin{pmatrix} a_1b_1 & a_1b_2 & a_2b_1 & a_2b_2 \\ a_1b_3 & a_1b_4 & a_2b_3 & a_2b_4 \\ a_3b_1 & a_3b_2 & a_4b_1 & a_4b_2 \\ a_3b_3 & a_3b_4 & a_4b_3 & a_4b_4 \end{pmatrix}. \tag{29}$$

Exercise 13. 🌶○○*Describe the difference between AB and $A \otimes B$ in terms of gates and state vectors.*

> *God made the natural numbers; all else is the work of man.*
> —Leopold Kronecker (late 19th cent.) [38]

15.4 Controlled-Z (CZ)



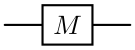
This is a similar gate to CX, but there's no *target* qubit. The action is rather simple to understand: if the input qubits are equal to $a\,|11\rangle$, the result state is $-a\,|11\rangle$. For any $a\,|\psi\rangle$, where $\psi \in \{[00],[10],[01]\}$, $a\,|\psi\rangle$ is left alone. The matrix representation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

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.^{15} \tag{30}$$

Exercise 14. ○○○*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) [39]

15.5 Measurement

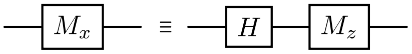


A measurement gate projects a qubit into a particular basis. For example, M_z projects a single qubit into the Z basis, meaning that it returns either a (classical) 0 or 1. An

[14] Confusingly, the ⊗ symbol is used to represent many kinds of "tensor products." The full computation of CNOT = $I \otimes X$ by this definition is left as an optional exercise to the reader; the author used a chatbot to produce the typesetting of this formula. ☺

[15] The row vectors are obtained from the Z basis: $Z = \{|0\rangle, -|1\rangle\}$

M_x gate measures in the X *basis*, or $\{|+\rangle, |-\rangle\} = \{H|0\rangle, H|1\rangle\}$. An M_z gate can therefore be written as



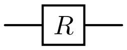
.

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

Exercise 15. 🌶*Prove identity*

> *We are talking about a predictive theory, not just measurements after the fact.*
> —Richard Feynman (1965) [40]

15.6 Reset

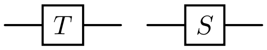


This gate forces a qubit into the $|0\rangle$ state.[16]

Exercise 16. 🌶🌶🌶*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) [41]

15.7 T and S



The T gate is defined as

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \tag{31}$$

This is also referred to as the $\pi/8$ *gate*[17]. A similarly defined gate is S, also commonly denoted P (for "phase"):

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \tag{32}$$

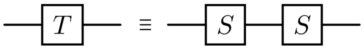
[16] Physically, this is similar to a measurement operation, but we wait around for the qubit to relax into the $|0\rangle$ state.

[17] This is due to it corresponding with a $\pi/8$ rotation about the vertical axis of the Bloch sphere **??**.

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

$$T = \sqrt{S}, \tag{33}$$

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

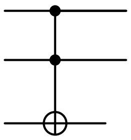
$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix} \tag{34}$$

where φ, called the *phase angle*, is of period 2π.[18]

Exercise 17. 🌶🌶🌶*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)

15.8 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 18. 🌶🌶🌶*Show how to construct an S gate using Toffoli gates.*

> *With these constraints, one can* satisfactorily *deal with both functional and structural aspects of computing processes; at the same time, one attains a closer correspondence between the behavior or abstract computing systems and the microscopic physical laws (which are presumed to be strictly reversible) that underly any concrete implementation of systems.*
> —Tommaso Toffoli (1980) [42]

[18] Meaning that $0 \le \varphi < 2\pi$, constricting it to a single rotation about the unit circle.

15.9 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.[19] 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.[20]

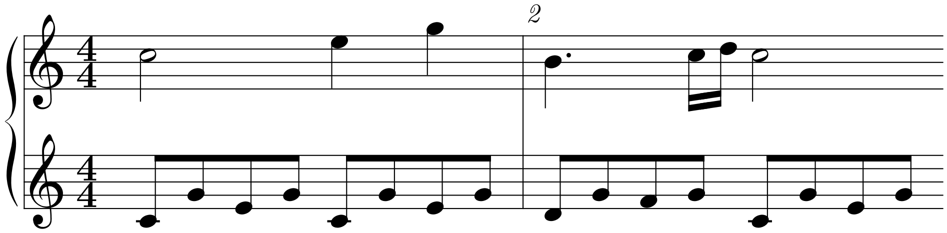
> *It contains, next, a generalization of them, applicable to any number of dimensions; and a demonstration that the algebra thus obtained is always a compound...*
> —William K. Clifford (1878) [43]

[19] The Toffoli gate will frequently be included in this set due to its efficiency.

[20] 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.

16 Supplement: quantum circuits

What does a quantum circuit look like?



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

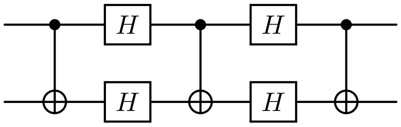


Figure 8: A quantum circuit.

That's better. But those who've had the fortune (or perhaps *mis*fortune) of studying music theory may draw some important[21] connections:

- Time moves from left to right.
- Each horizontal line has a relationship to the other.
- The lines are logically and temporally connected.
- 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 8.

- *Time moves from left to right*: The action occurs in the order top-left dot, followed by the "H," another dot, and so on.
- *Each horizontal line has a relationship to the other.* Each line represents a qubit.
- *The lines are logically and temporally connected.* The lines between two qubits represent actions that depend on both of them.
- *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

[21] To this work, not to the analogy.

16.1 The Bell pair

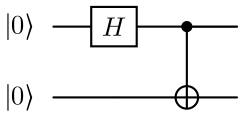


Figure 9: A circuit that entangles two qubits (prepared in the state $|00\rangle$) into a Bell pair

Figure 9 depicts our first example, which prepares a quantum state known as a *Bell pair* (also known as an EPR[22] 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 $|\psi\rangle_1$ 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

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

(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."

Exercise 19. 🌶🌶🌶*Prove that the circuit in Figure 6.2.1 corrects* $X\,|(\alpha\,|0\rangle + \beta\,|1\rangle)\rangle$.

> *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) [44]

17 Advanced supplement: algebraic topology

17.1 Groups

A *group* is a mathematical object that associates a set G with a binary operator[23] that maps the Cartesian product[24] $G \times G$ to G, and has these three properties:

[22] 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 [44].

[23] Typically we do not assign any particular symbol to this operator, akin to multiplication.

[24] $A \times B = \{(a, b) : a \in A, b \in B\}$.

- (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

$$\mathbb{Z}_n = \{a \pmod n : a \in \mathbb{Z}\} \tag{35}$$

Exercise 20. 🌶*Prove that $\mathbb{Z}_n$ is an Abelian group.*

Exercise 21. 🌶🌶*The "special" unitary group SU(2) contains the set of all 2x2 unitary matrices with determinant 1. Prove that SU(2) is non-Abelian.*

> *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 (late 19th or early 20th cent) [45]

17.2 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 $\{a \pm nk : k \in \mathbb{Z}\}$:

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

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

> *...the importance of applications* [of groups] *such as coding theory and cryptography has grown significantly.*
> —Thomas W. Judson (2022) [46]

18 Further reading

For summation notation, defined precisely, refer to [47], chapter 2. For probability theory, refer to [48].

> *I don't think these are the errors that the title was referring to.*
> —Jarrett Green (2023) [49]