Using Graph States for Quantum Computation and Communication

Thesis by

Kovid Goyal

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California

2009
(Submitted May 27, 2009)

To Dr. Ajay Patwardhan, for inspiring a whole generation of physicists.

Acknowledgements

I would like to start by acknowledging my thesis adviser, John Preskill, for giving me the freedom to pursue my interests and for setting a very high standard for me to aspire to. John has also provided the direction I needed at critical points in my career.

I would like to thank Robert Raussendorf for sharing a lot of his ideas with me and patiently explaining them when needed. Most of the great ideas in Chapter 2 were originated by him. Robert has been a mentor and a guide.

I would like to thank Jim Harrington for introducing me to the numerical techniques needed for the analysis of the measurement based quantum computer and for writing relatively clear and easy to follow code.

I would like to thank Austin Fowler for asking me a lot of questions and thereby greatly improving the clarity and depth of my understanding. Austin also inspired me to make this work as clear and easy to follow as possible. The magic state distillation circuit diagrams and the CNOT gate construction in Chapter 2 are his.

I would like to thank Ben Toner, Panos Aliferis, Greg Ver Steeg and Prabha Mandayam Dodamanne for many stimulating discussions.

I would like to thank my parents, Ashima and Niraj, for supporting my desire to become a physicist and for providing me with a stimulating and enjoyable childhood.

I would like to thank my wife, Krittika, for being there and for sharing my love of physics.

Finally, I'd like to acknowledge Caltech, and the people that make up its community, for being such a great place to have gone to gradschool.

Abstract

In this work, we describe a method to achieve fault tolerant measurement based quantum computation in two and three dimensions. The proposed scheme has an threshold of 7.8×10^{-3} and poly-logarithmic overhead scaling. The overhead scaling below the threshold is also studied. The scheme uses a combination of topological error correction and magic state distillation to construct a universal quantum computer on a qubit lattice. The chapters on measurement based quantum computation are written in review form with extensive discussion and illustrative examples.

In addition, we describe and analyze a family of entanglement purification protocols that provide a flexible trade-off between overhead, threshold and output quality. The protocols are studied analytically, with closed form expressions for their threshold.

Contents

List of Figures

List of Tables

List of Asides

Chapter 1

Introduction

As the components of computers have gotten smaller and smaller, quantum mechanical effects have begun to play an increasingly important role in the process. Traditionally, these quantum effects are classified as noise and efforts are made to minimize their effects on the computation. However, in recent years it has been realized that exploiting these quantum effects can allow us to do a number of surprising and useful things, both in computation and communication.

1.1 Computation Powered By Quantum Mechanics

A classical computer (defined loosely as a computer that regards quantum effects as noise) represents data as bits, that is, as two level systems whose two levels are labeled by 0 and 1. Classical data consists of *bit strings*, sequences of 0 and 1. The fundamental (and only) operation that a classical computer can perform on a single bit is to flip it. If there are two or more bits, the classical computer can in addition calculate their product or their sum. It is on this simple foundation that classical computation is built.

A quantum computer, on the other hand, represents data as *qubits*. A qubit is also a two level system, except that it is a *quantum* system. This means, in particular, that it can be in a *superposition* of 0 and 1. For a given set of n bits, a classical computer can put them into only a single state out of the 2^n possible states, whereas in a quantum computer they can be in a superposition of *all* 2^n states at once. This is what gives quantum computers their power. Examples of qubits include: the spin of an electron, where the 0 and 1 states correspond to the up and down states of the electron spin along some predefined axis; photon polarization, where the 0 and 1 states correspond to left and right circularly polarized light; etc.

A qubit, which in general is in a superposition of 0 and 1 is represented by a state vector: $\alpha|0\rangle + \beta|1\rangle$; where α, β are complex numbers satisfying the relation $|\alpha|^2 + |\beta|^2 = 1$. A quantum computer has a much broader range of single qubit operations available to it than a classical computer. It can perform arbitrary rotations on qubits. Since the coefficients of the state vector are complex, the rotation operators are *unitary* (members of the group $SU(2)$). In addition, when more than one qubit is present, the quantum computer can create *entanglement* between them. While a precise definition of entanglement is difficult to achieve, roughly speaking, two quantum systems are said to be entangled when the state of the combined system cannot be described in terms of only the states of its two parts. Entanglement leads to correlations between physically measurable quantities of the two systems that are stronger than anything that can be achieved classically [Bel64]. Entanglement and the strong correlations it generates turn out to be essential to the functioning of quantum computers.

So what can these quantum computers do? One large category of problems where quantum computers achieve a significant speed up over their classical counterparts is *unstructured search*. This involves searching for a particular item in an unstructured collection of items. The collection of items could be the solution space for some problem, or a list of unrelated items like telephone numbers or passwords. As long as the items are not related to each other in any way and it takes the same amount of time to test if a given item is the desired item, Grover's algorithm [Gro01], or slight modifications of it, can achieve a quadratic speedup over the best classical algorithms. If the collection has N items, the best classical algorithm requires on average $N/2$ tries to find the desired item. Grover's algorithm can do it in $\sqrt{N}$ tries on average. This technique can provide a dramatic speedup when trying to solve NP-complete problems using brute force search.

Another famous quantum algorithm is Shor's algorithm, for factoring large numbers or calculating discrete logarithms [Sho97]. It is a polynomial time algorithm, which means that for a composite number of size N, it requires a number of steps that grows only polynomially with the size of N, $O(\log^3 N)$, to find its factors. In contrast, the best known classical algorithm, the general number field sieve [Pom96], requires a time close to exponential in the size of the number, $O\left(e^{\log^{1/3} N}\right)$. This implies that most modern cryptography schemes like RSA [RSA78] whose security is ultimately based on the difficulty of factoring large numbers are vulnerable to attack by quantum computers.

Since the simulation of natural quantum systems is prohibitively expensive on classical computers (because of superposition, one has to keep track of 2^N variables when simulating a system with N degrees of freedom), a working quantum computer would prove to be an efficient means of simulating the quantum dynamics of such systems, allowing for the numerical investigation of their properties. Since most such systems have very many degrees of freedom, analyzing them analytically often proves intractable and numerical analysis plays a very important role in our understanding of such systems.

1.1.1 A Working Quantum Computer?

The quantum algorithms described above all assume that all quantum operations are perfect. This is obviously not true in the real world. In fact, because qubit implementations are often based on microscopic systems, it is particularly difficult to isolate the components of a quantum computer from noise. In addition, the state of a qubit is specified by continuous variables, as opposed to the binary 0 or 1 in a classical computer. Thus, noise in a quantum computer can have subtle effects that have no classical analogue.

Nonetheless, one of the seminal achievements of quantum information science is the development of fault tolerant methods for quantum computation. The keystone of fault tolerance is the Threshold Theorem (see Aside 2.3), which basically states that as long as the noise is "bounded," it is always possible to perform a quantum computation with arbitrary accuracy and reasonable overhead.

In most discussions of quantum fault tolerance, a stochastic error model is used, as is the case for this work. For such an error model, the requirement that the noise be bounded, translates to an error rate threshold, that is, a maximum error rate affecting all noisy operations/qubits. Errors can typically affect all parts of the computation, such as storage, gates, measurement and preparation. Threshold theorems exist for more general error models as well, including adversarial and non-Markovian noise [AP09; TB05a]. In this work, we assume a stochastic noise model and use it to arrive at estimates for the fault tolerance threshold.

The technique that most proposals use to achieve fault tolerance is the use of a *quantum error correcting code*. Recall that in a classical computer, you only have bit flip errors. In a quantum computer, states are in superpositions of classical states and you can have general unitary errors. In addition, any measurement will collapse the superposition into a single classical state. Most interactions of qubits with their environment act like partial measurements of the qubit, destroying these superpositions. This phenomenon is known as *decoherence*.

Decoherence is what makes the world classical at large scales. It is also a big problem for quantum computers, since if it is not controlled, they will behave just like classical computers. It turns out that correcting decoherence is a special case of correcting general unitary errors. Any single qubit unitary can be expanded as

$$U = \alpha I + \beta X + \gamma Y + \delta Z,$$

where X, Y, Z are the Pauli matrices, I is the identity and $\alpha, \dots, \delta$ are complex numbers that are constrained by the relation $U^{-1} = U^{\dagger}$ (unitarity condition). Since $Y = iXZ$, an error correcting code that corrects only X and Z errors is sufficient. The conceptually simplest quantum error correcting code is the repetition code. The classical repetition code simply encodes $\overline{0} \rightarrow 000$ and $\overline{1} \rightarrow 111$. This code can correct any single error by a majority vote. For example, if a single error affects 000 making it 010, a majority vote would conclude that the logical bit should be still be 0. The quantum repetition code however has to protect against both bit flip X and phase flip Z errors. Phase flips affect the signs (phases) in a superposition. The encoding for the quantum code uses nine qubits for a single logical qubit

$$\begin{aligned}|\overline{0}\rangle &\rightarrow (|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)\\ |\overline{1}\rangle &\rightarrow (|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle).\end{aligned}$$

By comparing qubits within blocks of three, we can correct X errors, just as for the classical code. By comparing the signs of the three blocks and using majority voting again we can correct single Z errors.

The repetition code is the simplest to understand but far from the most effective. Many other codes and fault tolerance schemes based on them exist. For example, the "Fibonacci scheme" of [AP09] uses a four qubit code that is capable of only detecting the presence of an error, but not correcting it to achieve a fault tolerant quantum computer with threshold 6.7×10^{-4}.

In addition to the threshold, another important consideration when designing a quantum computer is its overhead, that is, the amount of extra resources needed to perform the computation fault tolerantly. For example, one of the benefits of a quantum computer is that, using Shor's algorithm, it can factor large integers in polynomial time. Now if we were to build a fault tolerant version of the computer that required exponentially many resources to perform Shor's algorithm, this benefit would be lost. For instance, the Fibonacci scheme mentioned above uses an error-detection code, which means that it must post-select (i.e., throw away all instances that have detected errors). This typically leads to an exponential scaling of the overhead requirements. You will see an example of post-selection and the four qubit code at work, albeit in a different context, in Chapter 4.

In this work, we present a novel scheme for fault tolerance based on two ideas: measurement based quantum computation and topological error correcting codes, described in the following sections. This scheme has a very high estimated threshold (for the stochastic error model) and also has a poly-logarithmic scaling of the overhead with the size of the computation.

1.1.2 Measurement Based Quantum Computation

The motivation for measurement based computing is that in the traditional circuit based model, you need to perform two qubit interactions between arbitrary qubits at arbitrary times during the computation. This is typically worked around by performing successive SWAP gates to bring the target qubits next to each other before performing each two qubit interaction. By contrast, a measurement based computation has all the two qubit gates performed in a translationally invariant manner, in a single time step, right at the start of the computation. For certain architectures, that have a natural translational invariance, such as optical lattices, this is a big advantage. Once the two qubit gates are performed, the actual computation is carried out by only single qubit measurements. Two qubit gates are tricky to perform experimentally, because they suffer from a basic contradiction. In a quantum computer, you want to isolate your qubits from the environment and each other as much as possible so as to lessen decoherence. But, in order to perform a two qubit gate, qubits must interact with each other. As a result, two qubit gates are often the most difficult element to perform in a physical implementation of a quantum computer. This makes the measurement based model particularly well suited to experimental implementation.

Measurement based computation is based on the properties of certain multi-qubit quantum entangled states. When a part of such a state is measured, the state of the remainder depends both on the basis chosen for the measurement as well as the random measurement result. Given an entangled multi-qubit state, by choosing the right basis and compensating for the randomness of the measurement results, it is possible to drive an arbitrarily complex quantum computation entirely by single qubit measurements. Measurement based computation is described in more detail in the first part of

Chapter 2.

1.1.3 Topological Error Correcting Codes

Topological error correction is motivated by the observation that in many physical systems, noise processes are local. That is, the noise that affects one part of the system is uncorrelated with noise that affects another part. Typically, noise correlations fall off exponentially with distance. This means, that if we can store information in the large scale structure of a system, this information should have a high degree of natural robustness against local noise processes. Mathematically, "the large scale structure" of a system is described by its *topology* [All02].

This idea has been explored for quantum computers by designing so called, *topological quantum computers* [Nay+08]. In these systems, qubits are represented by *anyons* and quantum logic is carried out by braiding these anyons around each other. Anyons are a special class of quasiparticle in a two dimensional space. For example, anyons are formed by the excitations in an electron gas in a very strong magnetic field, and carry fractional units of magnetic flux in a particle like manner. This phenomenon is called the fractional quantum Hall effect [TSG82; Lau83].

Anyons, like fermions cannot occupy the same quantum state. If you consider the space time diagram of anyons, then their world lines never intersect. Braiding refers to winding the anyon world lines around each other. The combined state of the anyons (i.e. the state of the quantum computer) depends only on how the anyons have been wound around each other, i.e., on the topology of the braid pattern. Two qubit quantum gates are performed by winding a pair of anyons about each other and measurements can be performed by annihilating a pair of anyons.

Fraction Hall effect computation schemes have their own difficulties, which center around the problems of isolating and controlling individual anyons. An alternative way to introduce topology into quantum computation is to use a topological error correcting code [SA98; Den+02a]. Introduced by Kitaev, these codes live on lattices of ordinary qubits. The anyons are once again quasiparticles formed from the excitations of the qubits (strictly speaking from the excitation of the check operators of the topological code). In such a system, controlling anyons is no more difficult than controlling ordinary qubits. Indeed, when combined with measurement based computation, all that's needed to control the anyons is single qubit measurements.

When used as the basis of a quantum computer, the anyons of a topological error correcting code suffer from a limitation. The anyons are Abelian, and so they cannot be used to perform the full set of required gates to make the quantum computer universal. This can be worked around in two ways. One way is to use *magic state distillation* [BK05a] to prepare ancilla states that can be used to complete the set of gates. This is the approach used in this work. Another technique is to use a more complicated lattice that supports non-Abelian anyonic excitations [LW05]. The disadvantage of this approach is that such lattices typically have very complex Hamiltonians, making experimental implementation difficult.

Toplogical error correcting codes and their use in a fault tolerant quantum computer are discussed in detail in Chapter 2.

The fault tolerance scheme we present using these concepts has an estimated threshold of $\mathbf{7.8 \times 10^{-3}}$ and an overhead that scales **polylogarithmically** as $\log^3 N_G$, where N_G is the number of gates in the ideal circuit.

1.2 Quantum Mechanics in Communication

Quantum mechanics, via Shor's algorithm, can be used to breach the security of modern data encryption and transport protocols. However, what it takes with one hand, it gives back with the other. The phenomenon of quantum entanglement can be used to establish secret, shared randomness between parties that hold parts of an entangled state [Stu+02]. This shared randomness can then be used in encryption schemes like the one-time pad for secure, encrypted communication. Shared entanglement has many other uses as well, such as quantum teleportation [Ben+93] and super dense coding [BW92].

All these applications depend on the creation of high quality entanglement between (possibly widely) separated parties. One way to create high quality entanglement between remote parties is *purification*. This is applicable in a scenario where the parties share a number of low quality entangled states and need to create a single high quality state from them. Also since each party holds only one part of each copy, the operations they can perform are limited to local operations between the same part of all the copies. We allow each party to make partial measurements on their parts and communicate the results to each other via classical channels. It is important to note that whatever purification operations are performed, they will themselves be noisy and therefore it is impossible to use this technique to create absolutely noise free states.

In Chapter 4, we present a new family of protocols for the purification of multi-party entangled states. This family of protocols allows for tunable amounts of involved post-selection. This allows for designing protocols that have a desired tradeoff between threshold, output quality and overhead. Here overhead is the number of noisy states needed, on average, to produce a single pure state. Output quality is the "pureness" of the output state and threshold refers to the minimum quality of the input noisy states.

1.3 Structure

This work is structured as follows: Chapter 2 contains an introduction to measurement based computation in general and a description of our fault tolerant measurement based quantum computer. Chapter 3 has a detailed analysis of the threshold and overhead requirements of the fault tolerant quantum computer. Chapter 4 contains the description of several new families of purification protocols that can be used to purify large multi-qubit states.

The next section has a brief note on the notation used throughout this work.

1.4 Notation

Qubit operators are represented by upper case Roman and Greek characters. For multipartite states, the individual part being operated upon is indicated by a subscript. Thus, the expression $O_i M_j$ denotes the operators O and M acting on the i and j parts of the state, respectively. Some of the more commonly used operators are listed below.

Table 1.1 – Notation for common qubit operators

Notation	Meaning
X	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ (Pauli-X matrix)
Z	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ (Pauli-Z matrix)
Y	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ (Pauli-Y matrix)
H	$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ (Hadamard matrix)
$\overline{O}$	Logical (encoded) version of operator O
$\Lambda(X)$	Control-NOT gate
$\Lambda(Z)$	Control-PHASE gate
$\Lambda(U)$	Control-U gate, where U is any unitary
K_j	the graph operator acting on qubit j and its neighbors
$\lvert\pm\rangle$	The ±1 eigenstates of the X operator
$\lvert 0\rangle, \lvert 1\rangle$	The ±1 eigenstates of the Z operator

Chapter 2

Measurement Based Quantum Computing

2.1 Introduction

In this chapter, we will describe a scheme for fault tolerant measurement based quantum computation. The scheme is implemented on a translation invariant qubit lattice using only single qubit measurements. In Chapter 3 we will see that this scheme has a very high error threshold of $\sim 7.8 \times 10^{-3}$.

In this section, we introduce measurement based quantum computation in general, and in the next, we describe our specific proposal to make measurement based quantum computation fault tolerant. In Section 2.3, there is a brief note on how to implement the scheme in two physical dimensions. In the final section, we present an overview of the proposed scheme. It may be a good idea to read this first and keep it in mind as you read the rest of the chapter.

2.1.1 Computation as correlation

Normally, we think of computation as a process that takes some input, does some processing using the input, and outputs some results. Typically, the input consists of the actual input data as well as some ancillary bits that are "erased," that is, set to some standard state. Once the computation completes, the output is stored in the ancillary bits. Ideally, at the end of the computation, the input bits have not been disturbed. This is represented schematically in Figure 2.1.

Computation can be re-interpreted as the process of creating correlations between the input and output. As an illustration, consider a very simple computation, with input $a \in \{0, 1\}$ that simply copies the input to the output. Equivalently, this computation can be described as exactly correlating the output and input bits. Classical and quantum circuits that implement this computation are shown in Figure 2.2. For more general computation, the output bits will be exactly correlated with some *function* of the input bits. In the above, trivial example, the function is the identity function. In the circuit model of computation, both classical and quantum, (for example, Figure 2.2), this func-

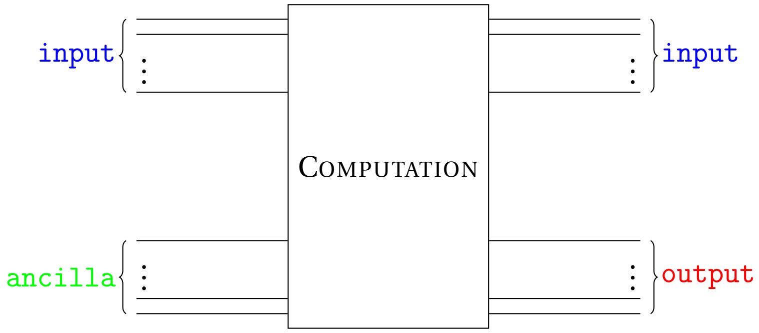


Figure 2.1 – Birds eye view of computation

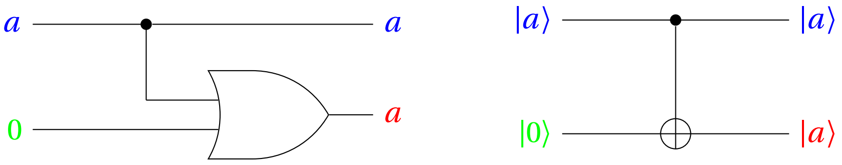


Figure 2.2 – The COPY operation using classical and quantum circuits. In the quantum circuit, the output qubit has to be measured in the computational basis to get the value of the a bit.

tion on the input bits is defined by the pattern of gates in the circuit. In other models of computation the function may be defined differently. For example, in the *Turing model*, the function being computed is encoded as a sequence of instructions that have to be followed by the *Turing machine*.

Here, we are interested in the measurement based model of quantum computing. In this model, the correlation function is described by a sequence of measurements on individual qubits.

2.1.2 Measurement Based Quantum Computation

When a part of an entangled quantum system is measured, the state of the unmeasured portion depends on the initial state of the whole system, as well as the basis chosen for measurement (it also depends on the random measurement outcome, but this can be compensated for). Since a computer works by manipulating the state of bits, we can exploit this property of measurement to design a "measurement based quantum computer." Such a computer consists of two parts; a pre-entangled "resource" state that starts out in some standard, highly entangled state, and a measurement pattern. The measurement pattern describes the algorithm (circuit) being implemented. In order to make our computer as simple to implement as possible, we would like to restrict ourselves to "local measurements," i.e., to the measurement of only individual qubits.

Such a scheme was first described by Raussendorf and Briegel in [RB01]. The particular resource state they chose, is the $2D$ cluster state, which is a state consisting of a number of qubits arranged in a simple $2D$ square lattice. The qubits are all entangled

Aside 2.1: *The Stabilizer Formalism*

The stabilizer formalism [Dan97] can be used to represent multi-qubit states in terms of the operators under which they are invariant. Since the description of the space of such operators is exponentially smaller than the description of the space of states they represent, the stabilizer formalism is a powerful tool for analyzing such spaces.
An operator K *stabilizes* a subspace $\mathscr{S}$, when $\forall\,|\psi\rangle \in \mathscr{S}$,

$$K\,|\psi\rangle = \psi.$$

The stabilizer formalism is particularly useful when the space of operators considered is a Hermitian subset of the Pauli group.
Given a subspace of states, if a unique group of stabilizer operators can be found that stabilize the subspace such that there is no state outside the subspace that is stabilized by the group, then the group is known as the *stabilizer* of the subspace. The stabilizer group can be represented fully by its *generators* ($\mathscr{G}$). For a N element stabilizer, the generator has $\log_2 N$ elements, yielding a compact description of the stabilizer group.

The action of unitaries in the stabilizer formalism can be easily represented by noting that $UKU^\dagger$ stabilizes $U|\psi\rangle$,

$$UKU^\dagger U\,|\psi\rangle = UK\,|\psi\rangle = U\,|\psi\rangle\,.$$

Thus, the action of unitary U on $|\psi\rangle$ is equivalent to the action of $U\cdot U^\dagger$ on the generators $\mathscr{G}$.

Measurement in the Pauli basis can also be treated easily in the stabilizer formalism. Pauli measurements map stabilizers to new stabilizers. The effect of the measurement of Pauli operator P on the stabilizer generator $\mathscr{G}$ can be calculated as follows:

- If P commutes with all generators $g \in \mathscr{G}$, the stabilizer is unchanged.
- Suppose P anti-commutes with the generators $g_1,\ldots,g_r$. Then, the new generator set $\mathscr{G}'$ is obtained by replacing g_1 by cP and $g_2,\ldots,g_r$ by $g_1g_2,\ldots,g_1g_r$, where $c = \pm 1$ is the measurement outcome.

It can be easily checked that $\mathscr{G}'$ generates the stabilizer of the post-measurement state.

Example: Consider the Bell state $|\phi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. It has the stabilizer generator $\mathscr{G} = \langle X_1X_2,\ Z_1Z_2\rangle$. Suppose we apply the Hadamard operator (H) to the first qubit, then the new generator is $\mathscr{G}' = \langle X_1Z_2,\ Z_1X_2\rangle$. If we now measure the X_1X_2 operator on our transformed state, we get a new generator $\mathscr{G}'' = \langle cX_1X_2,\ Y_1Y_2\rangle$.

For a more leisurely discussion of the stabilizer formalism, see [MI00].

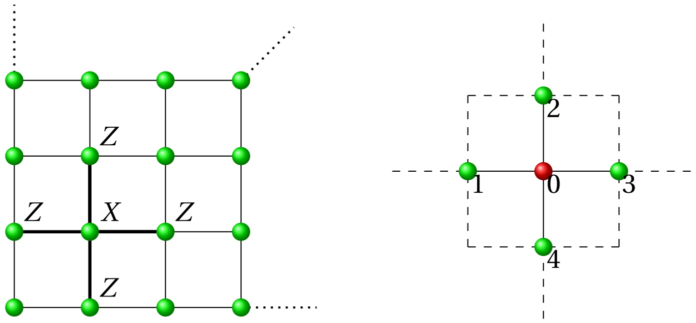


Figure 2.3 – a)The $2D$ cluster state consists of qubits at the vertices of a two dimensional grid. It is the $+1$ eigenstate of the $K_j = X_j \prod_i Z_i$ operators, one at each vertex. It can be created by starting with qubits in the $|+\rangle$ state and performing $\Lambda(Z)$ gates along the edges of the lattice. b) A single qubit embedded in a larger cluster. The cluster is one example of a graph of degree four.

with their neighbors. The cluster state is an example of a *graph state* (see Aside 2.2) of degree 4. See Figure 2.3 for a schematic visualization of a $2D$ cluster state. The cluster state can be constructed by starting with all qubits in the $|+\rangle$ state and performing a $\Lambda(Z)$ gate along every edge in the cluster. The $\Lambda(Z)$ gates all commute, so their order is unimportant.

For the purpose of performing measurement based quantum computing alone, a $2D$ graph state while sufficient, is not necessary. A simpler state could be used. Its choice is dictated by the fact that it is highly regular and easy to create experimentally, in a single time step, for example, in an optical lattice [Mar+02b; Mar+02a; Man+03a; Man+03b].

Let's look at the effects of measuring qubits in cluster states. Since we want to keep things simple for the sake of experimental implementation, we will only consider only single qubit measurements.

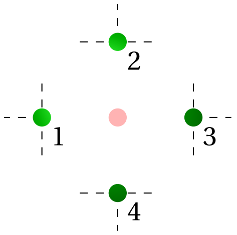
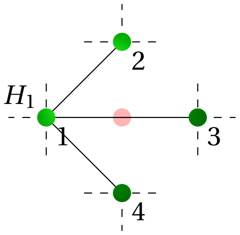
Measurement in the Pauli basis. Since the cluster state in Figure 2.3a is a graph state with stabilizer $\{K_j\}_{j=1}^{N}$, we can use the stabilizer formalism (see Aside 2.1) to easily deduce the effect of measuring qubit 0 (from Figure 2.3b) in the Pauli basis. In particular, the measurement in the X and Z bases has a nice interpretation, shown in Table 2.1.

Measurement in a rotated basis. Measuring cluster qubits in a rotated basis allows us to create arbitrary correlations between input and output qubits. Recall that a cluster state is created by performing $\Lambda(Z)$ gates between qubits prepared in the $|+\rangle$ state. Now consider a slightly modified two qubit cluster state, in which the first qubit starts out in the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. The cluster state is then,

$$\frac{1}{\sqrt{2}}(\alpha|0\rangle|+\rangle + \beta|1\rangle|-\rangle). \tag{2.1}$$

Now suppose that the first qubit is measured in the rotated basis $\frac{1}{\sqrt{2}}\{|0\rangle \pm e^{i\phi}|1\rangle\}$. The measurement has two possible outcomes denoted by $-1^m;\ m \in \{0,1\}$. The measure-

Table 2.1 – Measuring a cluster qubit in the Pauli basis

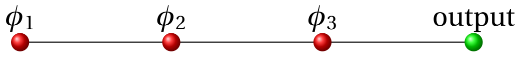
Measurement basis	Result	Interpretation
Z		The measured qubit is removed from the cluster with all its connections
X		The measured qubit is removed from the cluster, but its connections are "bridged." The actual state is the shown graph up to a Hadamard gate on qubit 1.

ment leaves the second qubit in the state

$$\alpha\,|+\rangle + (-1)^m e^{i\phi}\beta\,|-\rangle = X^m H U_z(\phi)\,|\psi\rangle\,. \tag{2.2}$$

Here, $U_z(\phi) = \exp(i\phi Z)$ is a rotation about the z-axis of the Bloch sphere. We see that the second qubit is left in a state equal to the first qubit, but rotated by the angle ϕ about the z-axis. In other words, the state of the second (output) qubit is correlated to the state of the first (input) qubit. Note that the angle of rotation ϕ is set by the choice of measurement basis. However, because of the randomness of quantum measurement, we are left with an extra factor of X^m, that depends on the measurement outcome, m. These extra factors, called by-product operators are an unavoidable side effect of trying to perform gates by measurement, and have to be compensated for when performing measurement based computation. Fortunately, as we will see, this is not difficult.

Performing arbitrary single qubit gates. By chaining measurements in different rotated bases, it becomes possible to implement arbitrary single qubit gates by measurements alone. Consider the following sequence of qubits each measured in a different rotated basis



Using Eq. (2.2) we see that this sequence of measurements performs the following unitary on the output qubit

$$U = HZ^{m_3} U_z(\phi_3)\, HZ^{m_2} U_z(\phi_2)\, HZ^{m_1} U_z(\phi_1)\,. \tag{2.3}$$

Where $m_{1,2,3}$ are the three measurement outcomes. Using the identities

$$
\begin{aligned}
HZH &= X \\
HU_z(\phi)H &= U_x(\phi) \\
XU_z(\phi) &= U_z(-\phi)X \\
ZU_x(\phi) &= U_x(-\phi)Z,
\end{aligned}
\tag{2.4}
$$

we can rewrite U as,

$$U = X^{m_3} Z^{m_2} X^{m_1} H U_z(\gamma) U_x(\beta) U_z(\alpha). \tag{2.5}$$

Where $\alpha = \phi_1,\ \beta = (-1)^{m_1}\phi_2,\ \gamma = (-1)^{m_2}\phi_3$. First note that this expression has exactly the same form as Eq. (2.2). It is a rotation followed by a by-product operator that depends on the measurement outcomes. Thanks to Euler's rotation theorem, we know that an arbitrary $3D$ rotation can be decomposed as a rotation about the z-axis, followed by a rotation about the x-axis followed by another rotation about the z-axis. Thus, U is an arbitrary single qubit unitary operator. Note that the choice of $\phi_2,\ \phi_3$ depends on the measurement outcomes of measuring qubits 1 and 2 respectively. Thus, a time-ordering is imposed on our measurement pattern.

The by-product operators that remain at the end of the measurements are unimportant. They need never be physically applied as they can always be accounted for when interpreting the measurement of the output qubits. For example, if the output qubit has to be measured in the computational basis, any extra Z by-product operators have no effect on the measurement outcome and any X by-product operators simply flip the measurement outcome. When multiple single qubit operations are applied, all the by-product operators can be commuted through using Eq. (2.4) to appear on the left where they can be absorbed into the interpretation of the final measurement outcome.

Two qubit gates. While we now know how to perform arbitrary single qubit operations by measurement (and thereby create correlations between the input, and their corresponding output, qubits), we still need to be able to create correlations between different input qubits. This can be achieved by implementing a simple two qubit gate, the controlled-NOT ($\Lambda(X)$) gate, defined, in the computational basis, as

$$\Lambda(X)|c\rangle|t\rangle = |c\rangle|c+t \mod 2\rangle. \tag{2.9}$$

A $\Lambda(X)$ gate can be implemented by the construction shown below. Suppose qubits 1 and 4 were in the states $|t\rangle$ and $|c\rangle$ respectively, before the qubits were entangled. After the entangling operation, qubits 1 and 2 are measured in the X basis, leaving qubits 3 and 4 in the state $X_3^{m_1+m_2+1} Z_4^{m_1} |c+t \mod 2\rangle|c\rangle$; where m_1, m_2 are the measurement results. Thus, we have performed a $\Lambda(X)$ between qubits 1 and 4, up to the by-product operator $X_3^{m_1+m_2+1} Z_4^{m_1}$.

Aside 2.2: *Graph States*

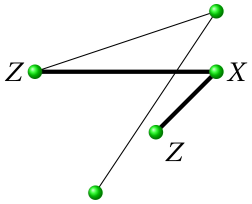
Graph states are a class of multiparticle entangled states that can be represented by mathematical graphs (collections of vertices connected by edges). The vertices represent qubits, and the edges, entanglement between the qubits. A N qubit graph state $|\psi\rangle$ is defined as the state that obeys the N eigenequations

$$K_j\,|\psi\rangle = |\psi\rangle\,; \qquad j = 1\ldots N. \tag{2.6}$$

Where the operators K_j are defined as

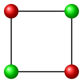
$$K_j := X_j \prod_{i\in N(j)} Z_i. \tag{2.7}$$

$N(j)$ is the *neighborhood* of the j qubit, i.e., the neighborhood of the j vertex in the graph of the state. This is shown in the example below.

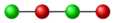


A graph state, showing one stabilizer operator K_j.

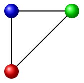
Bi-colorable graph states are a special class of graph states that we will encounter repeatedly. A bi-colorable graph is a graph whose vertices can be "colored" with two colors so that every vertex is connected only to vertices of a different color. Some examples:



An even cycle



A $1D$ cluster state



An odd cycle. This is **not** bi-colorable.

Errors in a graph state. Graph states have the nice property that any error can be represented in terms of chains of Z operators. Recall that any single qubit operator can be expanded as sum in terms of Pauli operators and the identity. Now consider a X error on qubit j. It anti-commutes only with the stabilizer generators $\{K_i\}$, where i runs over the neighbors of j. Thus, it will affect only these operators. However, the error chain consisting of Z operators acting on every neighbor of qubit j also anti-commutes with only these operators and thus is exactly equivalent to a single X_j error. This gives us the identities,

$$X_j = \bigotimes_i Z_i \qquad Y_j = Z_j \bigotimes_i Z_i \qquad Z_j = Z_j. \tag{2.8}$$

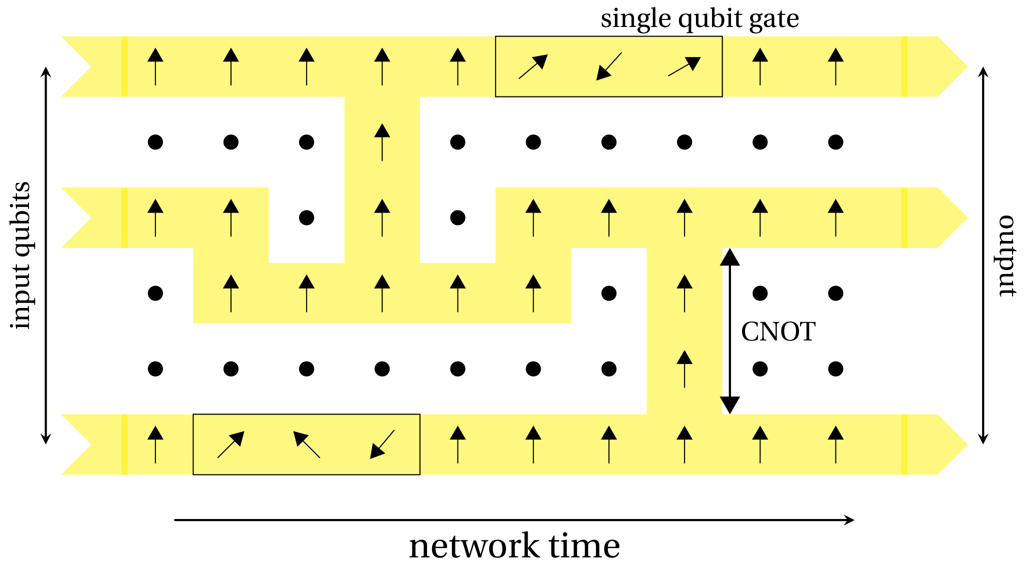
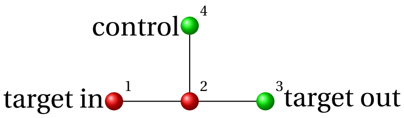


Figure 2.4 – Schematic representation of a measurement based computer. Vertical arrows represent measurement in the X basis, dots represent measurement in the Z basis and slanting arrows represent measurement in a rotated basis. Computation proceeds from left to right, with the output being obtained from the measurement results of the right most qubits. Each yellow stream represents the path of a "logical" qubit, as it moves from left to right. This computer operates on three logical qubits.



With the CNOT gate and arbitrary single qubit unitaries, we have a universal set of gates and can perform any desired quantum computation [Bar+95]. We can combine these elements (single qubit operations and $\Lambda(X)$ gates) into a full fledged measurement based quantum computer. For a schematic representation of a measurement based computer, see Figure 2.4.

While the quantum computer we have described will work very well in the absence of noise, in the real world, we must deal with noise and the impossibility of performing perfect quantum operations. Schematically, a measurement based computer works as follows:

- A $2D$ cluster state is prepared
- The qubits of the cluster state are measured column-by-column from left to right. The basis of measurement depends on the measurement outcomes from previous measurements. (Strictly speaking, there is no need to proceed column-by-column, as long as the time ordering imposed by the single qubit gates is preserved.)

- The output is obtained by correctly interpreting the measurement results of the rightmost column, keeping in mind the by-product operators from previous measurements.

- Both the input data as well as the algorithm to be performed are encoded in the measurement pattern. We could instead have put the input data into the left most column of qubits before creating the cluster state, but encoding it in the measurement pattern gives us a consistent interpretation of a computer as something that takes classical data (a measurement pattern) as input and outputs classical data, in the form of measurement results, as well.

- We can think of the computer as operating on a register of "logical" qubits. At each time step (column), every logical qubit is represented by a single physical qubit. At the final time step the logical qubit is measured by measuring the physical qubit that encodes it.

The last item above points to a straightforward technique for making measurement based computers tolerant to noise. Rather than representing logical qubits by single physical qubits, we can instead encode a logical qubit into several physical qubits. However, the naive approach of doing this directly, by using an error correcting code, such as the Steane code [Ste96], has certain drawbacks. The principal ones being that it will no longer be possible to implement the computation via single qubit measurements and creating a $2D$ cluster state between logical qubits is much harder to do experimentally.

An alternative approach, described in [AL06], is to implement the fault tolerant version of the ideal circuit on the measurement based computer. Fault tolerance is thus achieved automatically, by the circuit construction itself. While this approach undoubtedly works, it is abstract, and does not care what type of computer it is being implemented on. By using a fault tolerance scheme that directly leverages the architecture of the measurement based computation model, we can hopefully achieve better performance. Such a scheme is described in the next section. If you would like a more in depth treatment of the measurement based computation paradigm than presented above, see [DH06].

2.2 Fault Tolerant Measurement Based Quantum Computation

In the previous section, we saw that in the measurement based quantum computer, each logical qubit is represented by a single physical qubit at every timestep. In order to modify the measurement based computer to make it *fault tolerant* (see Aside 2.3), the first step is to use a quantum error correcting code to encode logical qubits. We must chose this code carefully to be compatible with the measurement based paradigm (i.e., an easy to prepare initial state with only single qubit operations to follow).

A good choice for such an error correcting code turns out to be the surface code (see Aside 2.4). Looking at the schematic of the measurement based computer (Figure 2.4), we see that the input qubits require one dimension and computation proceeds

Aside 2.3: *Fault tolerance*

In general terms, when we say that we want a computer to be fault tolerant, we mean that it should be somehow "resistant to noise." In other words as long as the noise affecting the computation is somehow bounded, we should be able to design a computer that computes with arbitrary accuracy, without too much overhead.

This requirement is formalized in the context of quantum computation via the *Threshold Theorem*.

Theorem 1. *If the noise per elementary operation is below a constant non-zero threshold then an arbitrarily long quantum computation can be performed with arbitrary accuracy and small operational overhead.*

This theorem has been proven for the circuit model of computation [AGP06] and it applies, as well, to the measurement based model, via the mapping in [AL06].

While the threshold theorem shows that there must exist an implementation of the computer that is fault tolerant, it is silent on how to construct one. In this chapter and the next, we present a design of a fault tolerant measurement based quantum computer that has a very high threshold and whose overhead requirements scale well with increasing circuit size.

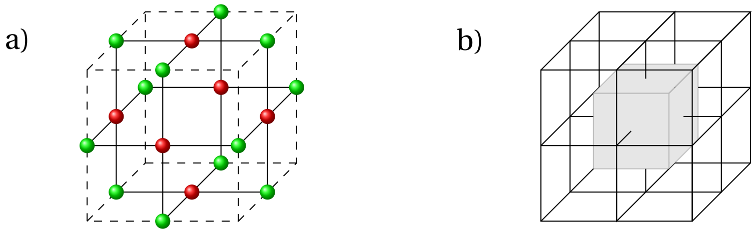


Figure 2.5 – The cubic lattice used for fault tolerant measurement based computation. a) The elementary cell of the lattice. It has 18 qubits and is tiled in $3D$ to build the lattice. The solid lines indicate entanglement between the qubits. The lattice is a bi-colorable graph state of degree 4. b) The dual cell (qubits suppressed for clarity), shown inside a section of the lattice with eight primal cells. The dual lattice is obtained by translating from the primal lattice by the vector $\frac{1}{2}(a, a, a)$, where a is the side of the primal cell.

Table 2.2 – The duality transforms in two and three dimensions. The transforms are *symmetric* ($(x^*)^* = x$).

dimension	primal	dual	maps to
$3D$	vertex	cube	The dual cube centered at the primal vertex
	edge	face	The dual face through which the primal edge passes
	face	edge	The dual edge that passes through the primal face
	cube	vertex	The dual vertex at the center of the primal cube
$2D$	vertex	face	The dual face with the primal vertex at its center
	edge	edge	The dual edge intersecting the primal edge
	face	vertex	The dual vertex at the center of the primal face

along the other dimension, from left to right. Surface codes are two dimensional, so a measurement based computer with its qubits encoded in a surface code, will be three dimensional. The $3D$ structure we use is a cubic lattice ($\mathscr{L}$) with qubits on the faces and edges of the elementary cell (see Figure 2.5). Hereafter, the computer based on this lattice will be referred to as the $\mathfrak{C}_{3D}$.

If we define the elementary cell to have a side of length 2 units, then, we can identify two sub-lattices within the full lattice. Call these sublattices primal ($\mathscr{L}_p$) and dual ($\mathscr{L}_d$). Each sub-lattice is a cubic lattice, but now only with qubits on its edges. The vertices of these lattices are located at the co-ordinates:

$$\mathscr{L}_p = (e[ven], e, e)$$
$$\mathscr{L}_d = (o[dd], o, o).$$

$\mathscr{L}_d$ is *dual* to $\mathscr{L}_p$ under the symmetric mapping $()^*$, defined in Table 2.2.

The sub-lattices are important as all the structures we eventually introduce for quantum computation belong on one or the other of these sub-lattices, and the way they interact depends on which sub-lattice they belong to. Furthermore, throughout this work, we adopt the convention that logical qubits are encoded on the primal sub-lattice while the dual sub-lattice is used to encode correlations between the logical qubits. The opposite convention of using the dual lattice to encode logical qubits is also possible. Indeed, the two conventions can even be mixed to an extent. However, for the sake of clarity and standardization, we will stick to the first convention.

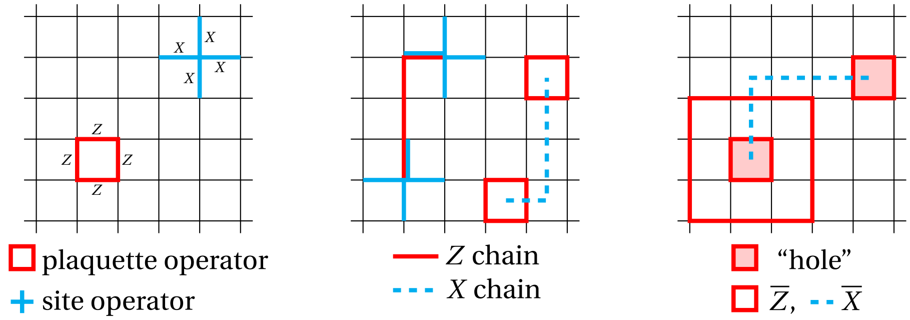
The first stage in defining $\mathfrak{C}_{3D}$ is to define the encoding of the logical qubits, which we will address in the next section.

2.2.1 The Logical Qubit

In the measurement based computer (Figure 2.4), the computation could be divided into timesteps with each logical qubit being encoded into a single physical qubit at every time step. Similarly, for $\mathfrak{C}_{3D}$, the computation can be divided into timesteps. At each timestep, the logical qubits are encoded in the surface code on a single $2D$ plane. When

Aside 2.4: *Surface codes*

Surface codes [SA98; Den+02a], are a way of encoding quantum information (states) into the topology of a lattice of qubits. The simplest (and most directly relevant) example is to use a $2D$ lattice with the qubits living on the edges of the lattice. Such a state is a stabilizer state (see Aside 2.1), with two types of stabilizer generators (plaquette and site). If all the stabilizers are enforced, then the stabilized subspace consists of a single state, the $+1$ eigenstate of all the stabilizers. By itself, this state is not very interesting, but if we relax one of the constraints (i.e., no longer require a particular generator to have eigenvalue $+1$), then the stabilized subspace has dimension two. We can interpret the new state as carrying a localized excitation at the site of the relaxed generator, or a *quasi particle*. In the diagram below, we see an example of such quasi-particles, called "holes."



In this diagram, we see the various structures defined on the lattice. First, the stabilizer generators, of two types, plaquette and site. It is easy to see that they commute with each other. Errors on the lattice can be either Z chains, or X chains. A X chain consists of the X operator acting on all edges that intersect the dashed blue line, while Z chain consists of Z operators acting along the thick red line. Error chains anti-commute only with plaquette and site operators at their ends and thus leave syndrome only at their ends (syndrome consists of the results from measuring the plaquette and site operators at all locations on the lattice).

Encoding logical qubits. In the third diagram, we see an example of using a pair of quasi particles to encode a logical qubit. The logical $\overline{Z}$ operator is defined as any closed loop of Z operators that encircles one of the holes, while the logical $\overline{X}$ operator is defined as any chain of X operators stretching between the two holes. A similar set of definitions exists for a pair of holes made up of site operators.

Error correction. First note that any closed loop that does not enclose a hole is equivalent to the logical identity operation, since it is in the stabilizer of the state and commutes with the $\overline{X}$ and $\overline{Z}$ operators. Since error chains leave a syndrome only at their ends, we just have to match all positive syndrome locations into pairs using a chain matching algorithm. We then apply the appropriate chain of operators between each pair to convert the error chains into trivial closed loops. If we make sure our holes are well separated and that each hole is "thick" (i.e., made by tiling several adjacent plaquettes), then the only error chains that will be misinterpreted by our error correction procedure and thus become logical errors, will have to be long. For a local error model, the probability of such chains is exponentially suppressed in their length, leading to a robust encoding of logical qubits.

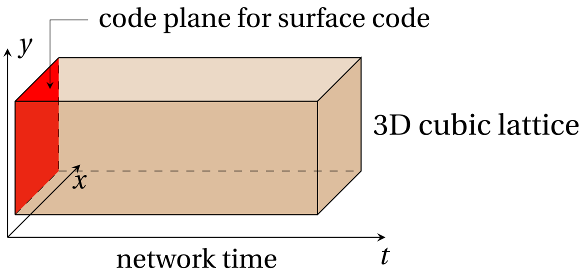


Figure 2.6 – The schematic $\mathfrak{C}_{3D}$. Logical qubits are encoded in the surface code on a $2D$ plane. At each timestep, the left most plane is measured, and the logical qubits move onto the next plane. The output is obtained when the rightmost plane is measured.

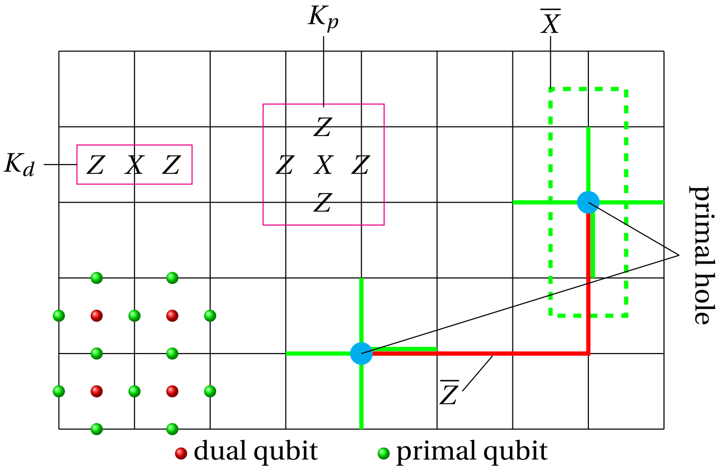


Figure 2.7 – A $2D$ slice of $\mathscr{L}_p$ showing a logical qubit. The stabilizer generators of the full $3D$ lattice when projected onto the slice are of two types, K_p and K_d. The logical $\overline{X}$ operator consists of X acting on every edge intersected by the dashed line. $\overline{Z}$ consists of Z operators acting on all the edges marked by the thick red line.

the plane is measured, the logical qubits move onto the next plane, and so on, until they are finally measured on the output plane at the right. By correctly interpreting this measurement result (keeping in mind the results from measuring previous planes), we get the result of the computation. This idea is represented schematically in Figure 2.6.

Just as was the case for the surface code (Aside 2.4), logical qubits in the $\mathfrak{C}_{3D}$ are made up of pairs of "holes." By convention, the holes live on $2D$ slices of $\mathscr{L}_p$. We saw in the case of the surface code that a hole introduces a degree of freedom, raising the dimension of the stabilized subspace from one to two. A hole has exactly the same effect here. Consider Figure 2.7, which shows a $2D$ slice of $\mathscr{L}_p$. The stabilizer generators of the full $3D$ lattice, when projected onto the slice, are of the form K_p and K_d. We can define a site operator (analogous to the site operator of the traditional surface code in Aside 2.4), as shown in the figure. By relaxing the requirement that the plaquette operator have eigenvalue $+1$, we introduce a degree of freedom, or a hole.

At first sight, the slice does not look much like a surface code. It has qubits on both edges *and* faces and the stabilizer generators K_p, K_d are not directly compatible. How-

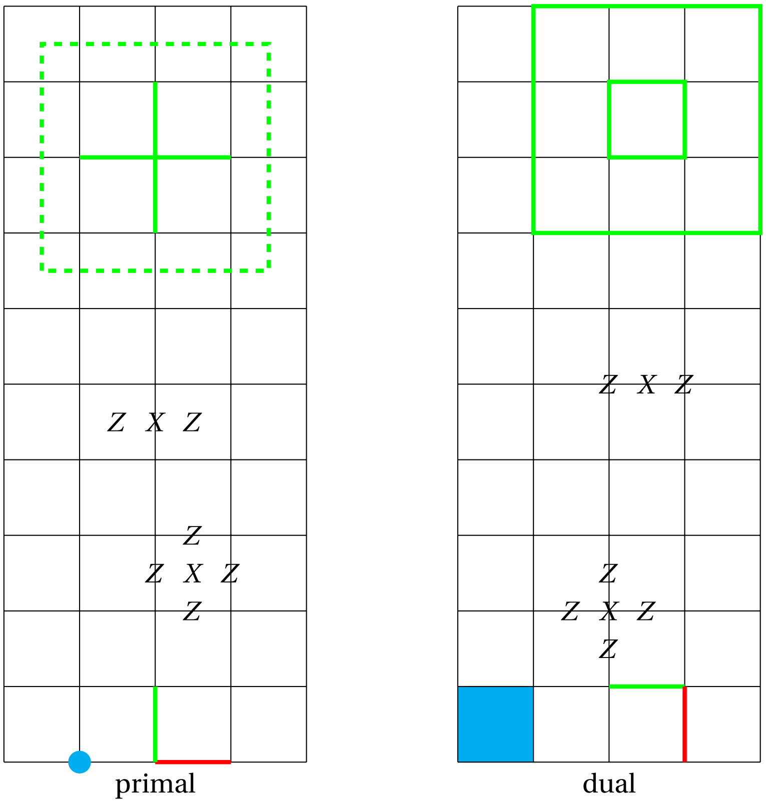


Figure 2.8 – The mapping from slice to surface code. See the discussion in the text.

ever, by measuring the face qubits in the X basis, we can convert the slice to a surface code. When the measurement pattern for $\mathscr{L}$ is described, we will see that the face qubits are indeed measured in the X basis. To understand the mapping from slice ⟷ surface code, we consider the $2D$ lattice dual to the slice (see Table 2.2 and Figure 2.8). The dual lattice has qubits on edges and vertices. The K_d stabilizers are associated with the edges of the dual lattice and the K_p stabilizers with the vertices. By multiplying four K_d stabilizers, we get a plaquette operator consisting of the X operator acting on the edges of the plaquette. Each edge is shared by two faces, thus there are two edges per face and therefore two K_d operators per face. One of these can be replaced by the plaquette operator. When the vertex qubits are measured in X, the other K_d is replaced by a single X operator, associated with the corresponding vertex. By multiplying this operator with K_p we are left with a site operator consisting of Z acting on the arms of the site. Thus, the stabilizer after measurement consists of plaquette and site operators, just as for the surface code in Aside 2.4 (except that the roles of the X and Z operators are reversed). Since the dual lattice supports a surface code, the primal lattice also supports the dual of the same code.

A pair of primal holes supports a single qubit. The logical $\overline{X}$ operator is a loop of X operators around either one of the primal holes in the pair, just as for the surface code. The logical $\overline{Z}$ operator is a chain of Z operation connecting the two primal holes. The choice of a particular loop or chain is immaterial as they all have the same eigenvalue.

Table 2.3 – Notation for the components of the sub-lattices $\mathscr{L}_p, \mathscr{L}_d$

Feature	$\mathscr{L}_p$	$\mathscr{L}_d$
vertex	v	$\overline{v} = q^*$
edge	e	$\overline{e} = f^*$
face	f	$\overline{f} = e^*$
cube	q	$\overline{q} = v^*$

This is a property of the surface code, and is easy to check. Since the loop and chain must intersect in an odd number of edges, it is easy to see that $[\overline{X}, \overline{Z}] = -1$. Thus, by adding pairs of holes to the slice, we can encode as many logical qubits as needed.

Remember that a hole is really just a location on the lattice where the corresponding stabilizer is not enforced. In the context of a measurement based computer, where slices are measured destructively (rather than measuring stabilizers directly, individual qubits are measured and the value of the stabilizers is computed by combining the measurement results), this means that when interpreting the measurement results, the value of the missing stabilizers must not be computed. For $\mathfrak{C}_{3D}$, this is automatically ensured by the construction of the lattice and the measurement pattern. What this means will become clear in the next section, where we define in detail the various components of $\mathfrak{C}_{3D}$.

2.2.2 The Topology of It

In order to understand precisely how the computation proceeds in $\mathfrak{C}_{3D}$, we will first have to identify and define the various structures that make it up. We have already defined the cubic lattice $\mathscr{L}$, along with the two sub-lattices $\mathscr{L}_p, \mathscr{L}_d$ that are dual to one another (see Table 2.2).

The sub-lattices $\mathscr{L}_p, \mathscr{L}_d$ are made up of vertices, edges, faces and cubes. These will be referred to by the notation defined in Table 2.3. The various components are related to each other, by the *boundary operator* (∂). The boundary of a cube is the set of six faces, the boundary of a face is the set of four edges and the boundary of an edge is two vertices.

Intuitively, the boundary operator is quite clear. For example the boundary of four adjacent faces would be the eight edges on the "outside" of the faces. However, the boundary operator is really a topological operator and to make its definition precise, we need to set up a *chain complex* [All02] on $\mathscr{L}$. This is done in Aside 2.5. With the precise definitions out of the way, we can enumerate the various objects that live in $\mathfrak{C}_{3D}$.

Qubits. Qubits in $\mathscr{L}$ come in two flavors, primal or dual. Primal qubits are located on edges of $\mathscr{L}_p$ or conversely, faces of $\mathscr{L}_d$. Dual qubits are located on the edges of $\mathscr{L}_d$ or conversely the faces of $\mathscr{L}_p$.

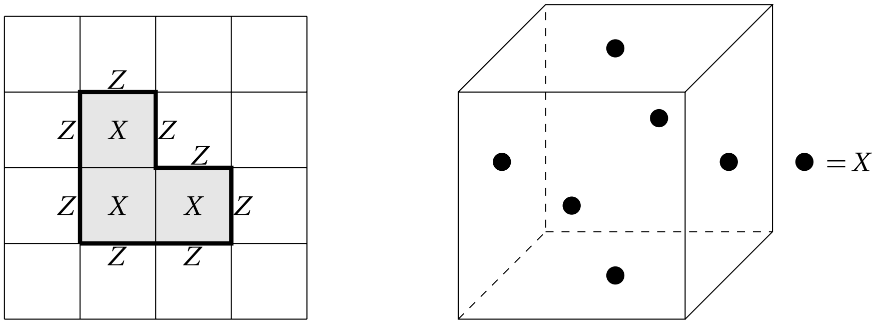


Figure 2.9 – Examples of correlation operators. On the left is a correlation operator in $2D$ with X acting on the faces and Z on the boundary. On the right is a special type of correlation operator, the syndrome operator. It consists of X acting on the six faces of a cube.

Correlations. A correlation is defined as a collection of faces, or formally, a 2-chain c_2 (see Aside 2.5). It is not strictly necessary for a correlation to be made up of contiguous faces, but all the correlations we will encounter in this work will be contiguous. Associated with every correlation c_2, is a *correlation operator*, defined as

$$K(c_2) := \bigotimes_{f \in c_2} X(f)\, Z(\partial f)\,. \tag{2.10}$$

In words, a correlation operator consists of the X operator acting on the face qubits and the Z operator acting on the qubits in the edges that form the boundary of the faces. Correlations can be either primal or dual. A primal correlation has X acting on dual qubits, with Z acting on primal qubits, while a dual correlation is the converse. Some examples are in Figure 2.9.

Correlations are used to move logical qubits forward with time and also to mediate interactions between logical qubits.

Errors. Recall in Aside 2.2, we saw that for a graph state, any error can be represented as a chain of Z operators. So it is natural to define errors as sets of edges. The edges need not be connected. Formally, an error is a 1-chain c_1 (see Aside 2.5). Associated with every error is an *error operator*, defined as

$$E(c_1) := \bigotimes_{e \in c_1} Z(e)\,. \tag{2.11}$$

In words, an error operators consists of the Z operator acting on the qubits in a set of edges. Errors can also be either primal or dual. A primal error operator is Z acting on primal qubits, while a dual error operator is Z acting on dual qubits.

Syndrome. Syndrome is extracted from the measurement results. It is used to detect the presence and type of errors that are affecting the computation. In the case of $\mathfrak{C}_{3D}$ that means locating the error chains. Syndrome is located at the vertices of $\mathscr{L}_p$ and $\mathscr{L}_d$.

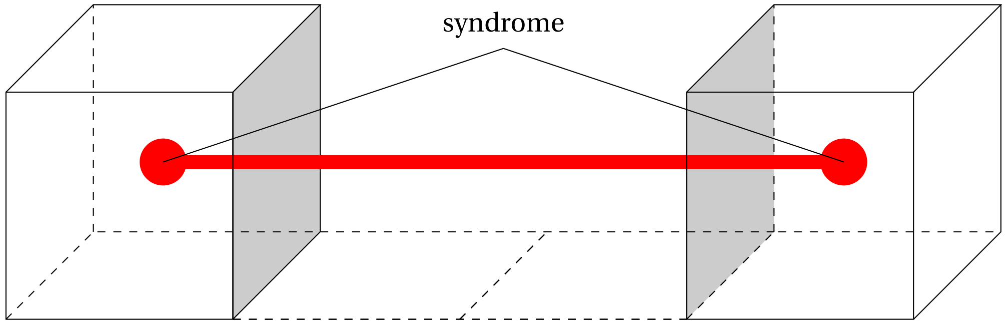


Figure 2.10 – Error chains leave syndrome at their ends.

Recall from Table 2.2 that every vertex is dual to a cube. This tells us what the operator that has to be measured to yield the syndrome at a vertex should be. It is the correlation associated with the faces of the cube dual to that vertex. The *syndrome operator* is

$$S(v) := \bigotimes_{f \in \partial v^*} X(f). \tag{2.12}$$

This follows from Eq. (2.10) and the observation that $\partial^2 q = 0$ (see Aside 2.5). In words, a syndrome operator is the X operator acting on the six faces of a cube. See Figure 2.9 for an example. Note that when restricted to a $2D$ slice, the syndrome operator becomes a site operator, as expected.

Syndrome and errors. Primal syndrome detects primal error chains while dual syndrome detects dual error chains. Let c_1 be an error chain, $\overline{q}$ be a dual cube corresponding to primal vertex $v = (\overline{q})^*$. The error operator $E(c_1)$ anti-commutes with the syndrome operator $S(v)$ only if $v \in \partial c_1$. Thus, error chains show syndrome only at their ends, just as for the surface code. This is shown in Figure 2.10.

Correlations and errors. Primal correlations are affected by dual error chains and dual correlations are affected by primal error chains. This is easy to see when you remember that error operators are made up of Z operators on edges, while correlation operators have X acting on face qubits. A primal correlation has X acting on primal faces, which are also dual edges. Thus, only a dual error operator can anti-commute with a primal correlation operator.

Furthermore, an error chain will only affect a correlation if it intersects the correlation an odd number of times. This follows from the the observation, $E(c_1)K(\overline{c_2}) = (-1)^{|\{c_1\}\cap\{\overline{c_2}^*\}|}K(\overline{c_2})E(c_1)$. If the number of intersections, $|\{c_1\}\cap\{\overline{c_2}^*\}|$, is odd, the correlation is conjugated to $-K(c_2)$ by the error, otherwise it is left unchanged.

Defects. Defects are line like objects that are used to give $\mathscr{L}$ an internal structure. When associated with logical qubits, defects can be thought of as extensions of holes into the network time direction. The logic of the quantum circuit being implemented is encoded in the topology of the defects (i.e., how they wind around each other). This will

Aside 2.5: *Homology on $\mathscr{L}$*

Define the chain groups as follows (where $z_k \in \mathbb{Z}_2$)

object	Chain group	Basis
vertices	$C_0 = \{c_0 : c_0 = \sum_k z_k v_k\}$	$\mathscr{B}(C_0) := \{v_k : \text{vertices in } \mathscr{L}_p\}$
edges	$C_1 = \{c_1 : c_1 = \sum_k z_k e_k\}$	$\mathscr{B}(C_1) := \{e_k : \text{edges in } \mathscr{L}_p\}$
faces	$C_2 = \{c_2 : c_2 = \sum_k z_k f_k\}$	$\mathscr{B}(C_2) := \{f_k : \text{faces in } \mathscr{L}_p\}$
cubes	$C_3 = \{c_3 : c_3 = \sum_k z_k q_k\}$	$\mathscr{B}(C_3) := \{q_k : \text{cubes in } \mathscr{L}_p\}$

The groups $\{C_i\}_{i=0}^{3}$ are Abelian under componentwise addition. For example, $c_1 + c'_1 = \sum_k z_k e_k + \sum_k z'_k e_k = \sum_k (z_k + z'_k) e_k$. For each $i = 1 \ldots 3$, there exists a homomorphism ∂_i mapping C_i to C_{i-1}, with the composition $\partial_{i-1} \circ \partial_i = 0$. With these definitions,

$$\mathscr{C} := \{C_0, C_1, C_2, C_3\},$$

is a chain complex [All02] and ∂ is a boundary operator that maps the chain c_i to its boundary which is a chain c_{i-i}. Similarly, the dual chain complex, $\mathscr{C}^* := \{C_0^*, C_1^*, C_2^*, C_3^*\}$ can be defined on $\mathscr{L}_d$.

We are now ready to define homology. Two chains $c_n, c'_n \in C_n$ are said to be homologically equivalent if there exists some $c_{n+1} \in C_{n+1}$, such that $c'_n = c_{n+1} + \partial c_{n+1}$. In other words, $c'_n =_h c_n$ if they differ by the boundary of a chain of the next higher dimension. The concept of homology encapsulates the property of surface codes that an error chain has syndrome only at its end-points (i.e., only at its boundary), and that we can correct such a chain by applying any chain that has the same end points.

Of particular interest to us, is the concept of *relative homology*. Suppose a chain complex is defined on a space $\mathscr{X}$ and there exists a sub-space $\mathscr{D} \subset \mathscr{X}$. Two chains are equal w.r.t relative homology, $c'_n =_r c_n$, if there exists some $c_{n+1} \in C_{n+1}(\mathscr{X})$ and $d_n \in C_n(\mathscr{D})$ such that $c'_n = c_n + \partial c_{n+1} + d_n$. This formalizes the idea that in a surface code, an error chain can start and end on a pair of defects (holes), and that the effect of the chain on the logical qubit is the same irrespective of the actual shape of the chain.

become clear as we progress. Formally, a defect d is defined as a connected set of edges and the faces they “contain,”

$$d := \{e : e \in d\} \cup \{f : \{\partial f\} \cap d = \{\partial f\}\}. \tag{2.13}$$

In words, a defect is a set of edges alongwith the faces the set of edges contains. A face is contained by a set of edges if its boundary is in the set of edges. Defects are either primal or dual, depending on whether they contain primal or dual edges and faces.

Singular qubits. Singular qubits (S-qubits) are single qubits located at widely separated locations in $\mathscr{L}$. They are used to initialize ancillary logical qubits in states that are used to perform non-Clifford gates.

The measurement pattern. Since $\mathfrak{C}_{3D}$ follows the measurement based quantum computer paradigm, all measurements are single qubit measurements. The basis in which a particular qubit is measured depends on what part of $\mathfrak{C}_{3D}$ it belongs to. The different parts that have different measurement patterns are listed below.

- *Singular qubits.* These qubits are measured in a basis rotated about the z-axis $\big(|0\rangle + \exp(i\phi)|1\rangle\big)$. The value of ϕ depends on the type of ancilla we are trying to create with the singular qubit.

- *Defect qubits.* Recall that defects are made up of two types of qubits, edge qubits and face qubits. Edge qubits are measured in the Z basis, while face qubits are measured in the X basis. This is true for both primal and dual defects.

- *Bulk qubits.* All the remaining qubits are measured in the X basis.

Now that we have specified the measurement pattern, we can see how it interacts with the surface code on a $2D$ slice of $\mathscr{L}_p$. Recall that "holes" on the surface code are simply locations where the corresponding site operator is not enforced. A hole is really the projection of a defect that extends in the time direction onto the slice. Therefore, the edges of L_p that extend in the time direction from the vertex at which a hole is located are defect edges. But primal edges are dual faces, so these two edges are also two faces of a dual cube centered at the vertex. Normally, the syndrome associated with the vertex would have been obtained by measuring the six faces of the dual cube in the X basis. Now however, because two of the faces are defect edges and therefore measured in the Z basis, the syndrome bit associated with the vertex is lost, and thus the stabilizer associated with it is not enforced. Note that since $\mathscr{L}$ is a graph state and all errors on it are Z chains, the plaquette operators of the surface code (that consist of Z acting on the edges of a plaquette) are automatically enforced. The site operators are enforced using the syndrome obtained from the measurement of the dual cubes centered at each primal vertex.

Correlations and defects. In a primal defect, edges in the boundary of a primal 2-chain are measured in the Z basis. Recall that a primal correlation consists of X operators on primal face (dual edge) qubits and Z operators on primal edge qubits. Therefore, a primal correlation can "end on" a primal defect. By "end on," we mean that it can have a primal defect as its boundary. Similarly, dual correlations can only end on dual defects. Similarly, primal correlations can wrap around dual defects and dual correlations can wrap around primal defects. This is analogous to having chains encircling holes in the surface code. In this manner, defects can be used to "guide" correlation surfaces.

Syndrome and defects. In the discussion of the measurement pattern, we saw that syndrome at holes is lost. This is a specific example of a more general phenomenon. For a given primal defect d, define the set $G := \{v \in \partial e : e \in d\}$. G is the set of vertices associated with the defect. Then, the syndrome operators $\{S(v^*) : v \in G\}$ do not commute with the measurement pattern. Thus the syndrome at all the vertices associated with a defect is lost.

Table 2.4 – Relationships between correlations, syndrome, defects and errors. "A bounds B" is synonymous with "B ends in A." The displayed objects do not interact with themselves. The relations are symmetric.

to this one ↓	**This object does ...** dual corr.	 primal defect	 dual defect	 primal err. cyc.	 dual err. cyc .
primal corr.	nothing	bound	repel	nothing	affect
dual corr.		repel	bound	affect	nothing
primal defect			encircle	pairwise end	encircle
dual defect				encircle	pairwise end
primal err. cyc.					nothing

However, for each defect d, there will be one syndrome bit associated with the defect as a whole. This is because there is a correlation surface (2-chain) that wraps around the defect as a whole. Since it wraps around the defect, it is closed, has no boundary and is of the form $\bigotimes_i X_i$. When the qubits in the bulk region are measured in the X basis, this correlation yields an extra syndrome bit.

Errors and defects. A primal error chain can end in a primal defect. This is because, primal syndrome is lost at a primal defect, as discussed above. Similarly, dual error chains can end in dual defects. However, there is a syndrome bit associated with the defect as a whole and this detects an error chain ending in the defect if the number of intersection between the correlation surface and the chain is odd. Thus error chains can *pairwise* end in defects.

The relationships between correlations, syndrome, defects and errors is summarized in Table 2.4.

2.2.3 Initialization and Measurement of Logical Qubits

The first step in performing a computation is preparing logical qubits in a known state. In the following discussion, the preparation and measurement of states in the X,Z bases is shown using defects made up of single edges. This is for clarity and ease of presentation. In an actual fault tolerant construction, the defects would be thick, as will be explained subsequently. The initialization in the rotated states however, requires the use of single qubits and cannot be topologically protected. These logical qubits must therefore be distilled before being used. The distillation will also be discussed later.

Initialization in an eigenstate of $\overline{Z}$. In order to create a logical qubit that is in an eigenstate of the $\overline{Z}$ operator, we use the construction shown in Figure 2.11. The qubits on the blue edges are measured in the Z basis, while all other qubits are measured in the X basis. The logical qubit is created on the right side of the lattice, with the shown logical operators. To see that the logical qubit is in an eigenstate of the $\overline{Z}$ operator, focus on the correlation operator $K_Z = \bigotimes_{i=i}^{4} X_i \bigotimes_{j=5}^{12} Z_j$. K_Z is compatible with (commutes with) the

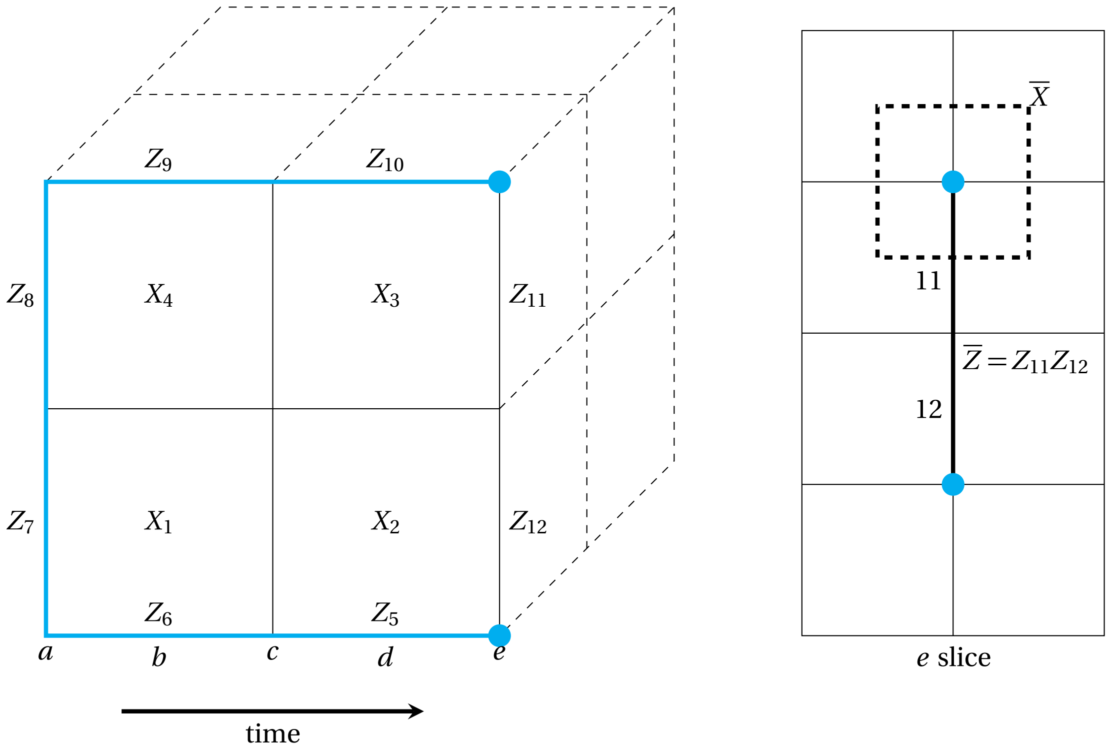


Figure 2.11 – Initializing a logical qubit in the Z eigenbasis on $\mathscr{L}_p$. The a, c, e slices belong to $\mathscr{L}_p$, while the b, d slices are in $\mathscr{L}_d$. See discussion in text for how the initialization works.

measurement pattern. After slices a, b, c and d are measured, we can multiply it by the resulting stabilizers of qubits $1 \ldots 10$ to reduce it to $(-1)^{\sum_{i=1}^{10} m_i} Z_{11}Z_{12}$, where $m_i \in \{0, 1\}$ are the results of measuring qubits $1 \ldots 10$. But, $\overline{Z} = Z_{11}Z_{12}$. Thus, the logical qubit on slice e is initialized to either $|\overline{0}\rangle$ or $|\overline{1}\rangle$, depending on the known measurement results from measuring slices a, b, c and d. In the language of measurement based computing, we have created the state $|\overline{0}\rangle$ up to the by-product operator $X^{\sum_i m_i}$.

Initialization in an eigenstate of $\overline{X}$. In order to create a logical qubit that is in an eigenstate of the $\overline{X}$ operator, we use the construction shown in Figure 2.11. The qubits on the blue edges are measured in the Z basis, while all other qubits are measured in the X basis. The logical qubit is created on the right side of $\mathscr{L}_p$, as before. To see that the logical qubit is in an eigenstate of the $\overline{X}$ operator, the important correlation operator is carried on the eight shaded faces of the two dual cells. It has the form $K_X = \bigotimes_{i=1}^{8} X_i$. This operator commutes with the measurement pattern. After slices a, b are measured, K_X reduces to the site operator on slice c. But the site operator is also the $\overline{X}$ operator, thus, we have prepared an eigenstate of $\overline{X}$ on slice c. Whether the state is the ±1 eigenstate depends on the known measurement outcomes from the previous slices.

Measurement of the $\overline{X}$, $\overline{Z}$ operators. Measurement of these operators can be performed by simply time-reversing the preparation procedure, as shown schematically in

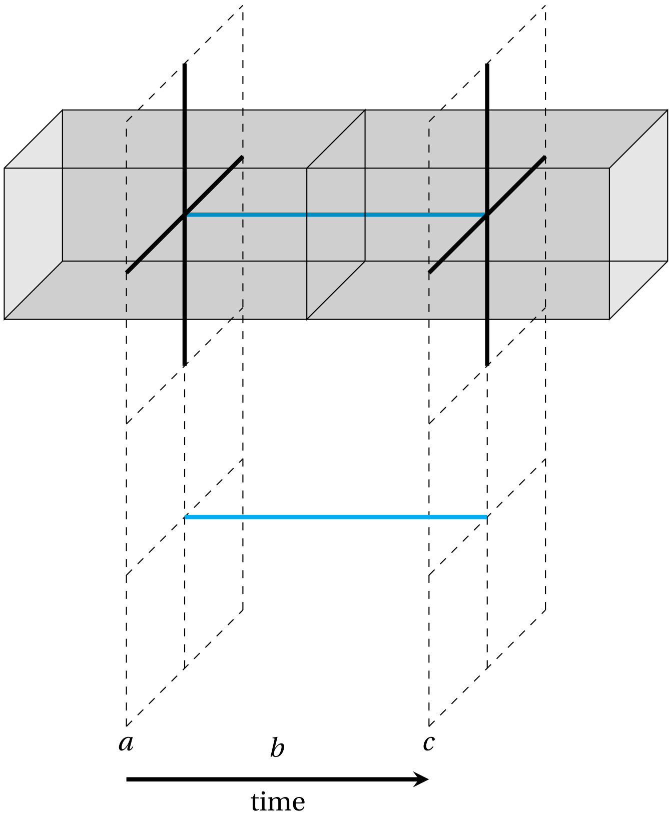


Figure 2.12 – Initializing a logical qubit in the X eigenbasis on $\mathscr{L}_p$. The dashed grids show slices of $\mathscr{L}_p$. The time axis is scaled by a factor of two. The two shaded cells belong to $\mathscr{L}_d$. The thick black edges of the site operators intersect eight faces of the shaded cells. See discussion in text for how the initialization works.

Table 2.5 – Preparation and measurement of $\overline{X}$, $\overline{Z}$. Time flows from left to right.

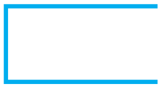
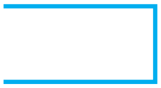
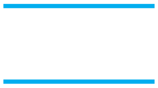
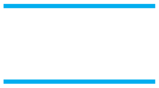
Basis	Preparation	Measurement
Z		
X		

Table 2.5. It is easy to see that the measurement pattern imposed by the shape of the defects, allows for the measurement of the logical operators.

Initialization in a rotated state. A rotated state is a state of the form $|\pm_\theta\rangle := |0\rangle \pm e^{i\theta}|1\rangle$. If we define the rotation operator $U_Z(\theta) := e^{-i\frac{\theta}{2}Z}$, then, up to a global phase, $|\pm_\theta\rangle = U_Z(\theta)|\pm\rangle$. To prepare a logical qubit in such a rotated state, the construction shown in Figure 2.13 is used. The red qubit is a singular qubit and is measured in the the rotated basis $\{|\pm_\theta\rangle\}$. This results in a logical qubit encoded in the $|\overline{\pm_\theta}\rangle$ state on the right most slice.

To understand how this works, first define R_θ as the operator whose eigenstates are $|\pm_\theta\rangle$. Now suppose that we choose a particular ring of X operators as the logical operator $\overline{X} = X_1 \ldots X_n$ and a chain of Z operators as the logical operator $\overline{Z} = Z_n \ldots Z_{n+m}$. The ring and chain intersect at the n^{th} qubit (i.e., at the singular qubit). Remember that this is a surface code, so the particular choice of the ring, chain and singular qubit doesn't matter as long as the topology is preserved. In this case that means that the ring must encircle one hole and the chain must start and end at the two holes. Also they must intersect an odd number of times and one of those intersections must be the singular qubit. Now it is easy to see that $\overline{R_\theta} = X_1 \ldots X_{n-1} R_{\theta\,n} Z_{n+1} \ldots Z_{n+m}$.

The next step is to remember that we can inject arbitrary states into a graph state. Ordinarily, a graph state is created by performing the $\Lambda(Z)$ gate between qubits initialized to the $|+\rangle$ state. However, if one of the qubits is initially in the arbitrary state $|\psi\rangle$ before the $\Lambda(Z)$ gates are performed and if all but one of the qubits in the resulting graph state are measured in the X basis, the remaining qubit will be in the $|\psi\rangle$ state (up to some by-product operators). This follows from Eq. (2.2), by setting $\phi = 0$.

Now realize that measuring the singular qubit in the $|\pm_\theta\rangle$ basis is the same as rotating it by $U_z(-\theta)$ and then measuring it in the X basis. But, $U_z(-\theta)$ commutes with $\Lambda(Z)$, so this is equivalent to rotating the qubit before the entangling operations. In other words, by measuring R_θ, we are effectively injecting the state $|\pm\rangle_\theta$ into the larger lattice. Whether the + or − state is injected depends on the measurement outcome. In particular, if after the preceding slices have been measured, we were to measure $\overline{R_\theta}$ on the right most slice, we would always get the eigenstate indicated by the preceding measurement results.

This technique can be used to initialize logical qubits in any state of the form $|\pm_\theta\rangle$. However, two such states are of particular interest to us; the eigenstates of the $A = \frac{X+Y}{\sqrt{2}}$ and Y operators. These are used as ancillas to perform certain quantum gates. It should

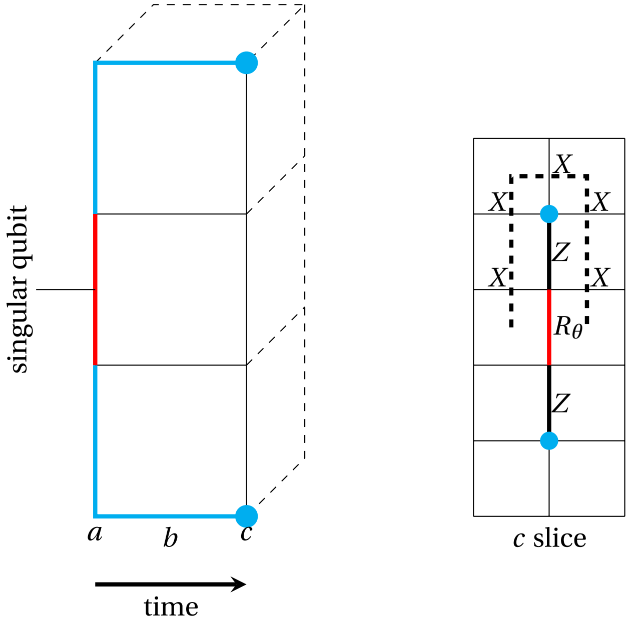


Figure 2.13 – Initialization of logical qubits in the $|\pm_\theta\rangle$ state. Slices a, c belong to $\mathscr{L}_p$. The slice c shows the logical operator $\overline{R_\theta}$. See discussion in text.

be emphasized that while the preparation of the eigenstates of the $\overline{X}$ and $\overline{Z}$ operators can be made fault tolerant simply by thickening the defects and making sure they are far apart, this does not work for initializing logical qubits in a rotated state, because of the need to use a singular qubit. As a result, these states must be distilled before being used in the computation.

2.2.4 Error Correction

Now that we are familiar with the various parts of $\mathfrak{C}_{3D}$, let's discuss how errors are corrected. Error correction proceeds similarly to error correction in the surface code, except for two differences:

- In the surface code measurements are non destructive. Measurement results are used to determine what correction operators to apply. In $\mathfrak{C}_{3D}$ measurements are destructive and error correction involves processing the measurement results to determine the measurement pattern on future slices.

- In $\mathfrak{C}_{3D}$ there are two independent topological error correcting codes. A primal code that lives on $\mathscr{L}_p$ and a dual code that lives on $\mathscr{L}_d$. They are independent in the sense that primal errors leave only primal syndrome and can end only in primal defects and vice versa for dual errors.

Just as for the surface code, error chains leave syndrome only at their ends. The task of the error correction algorithm is to correctly pair up all the locations that have nontrivial syndrome. See Figure 2.14 for an example. Once the pairing up is performed, the error represented by each pair is corrected. This is done by choosing a chain (typically

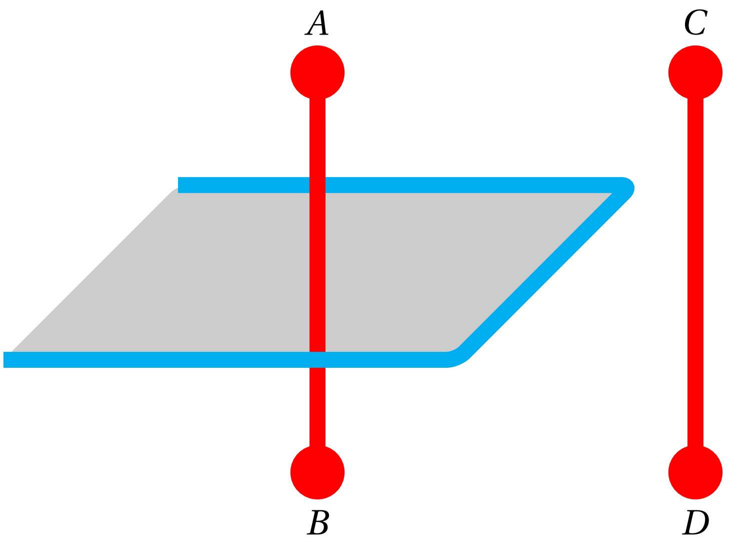


Figure 2.14 – Two error chains that leave syndrome at their ends. If the error correction algorithm correctly pairs up (A, B) and (C, D), the error will be corrected. However, if it pairs up (A, C) and (B, D), there will be an error anti-commuting with the shown correlation surface, leading to a logical error in the computation.

the shortest chain) that runs between the syndrome locations and flipping the X measurement results on all qubits along that chain. If the pairing up of syndrome locations was done correctly, the updated measurement results will give a computation result that is the same as if no errors had occurred.

Note that in $\mathfrak{C}_{3D}$ measurement results are used *twice*. The first time, they are processed to calculate the syndrome. This is done by multiplying the measurement results on the faces of every cube in the primal and dual lattices. Once that is done, the syndrome is used to update the original measurement results. These updated measurement results are then interpreted to get the result of the computation. The details of the error model and algorithm used for pairing up syndrome locations as well as a corresponding threshold estimate are in Chapter 3.

2.2.5 Gates

In order to implement universal quantum computation, we need a set of gates that can be implemented fault tolerantly and that constitute a universal set. The gates we choose are: Z-preparation, X-preparation, Z-measurement, X-measurement, $\Lambda(X)$, $\exp(i\frac{\pi}{8}Z)$, $\exp(i\frac{\pi}{4}X)$ and I. We have already seen how to perform X,Z preparation and measurement. We now discuss the remaining gates.

The identity gate. The identity gate is mediated by the two correlation operators K_X and K_Z, discussed previously for the preparation of states in the X and Z bases. They are shown schematically in Figure 2.15. Label the left most and right most slices I and O. Then, $K_X|_{I\cup O} = \overline{X}_I \otimes \overline{X}_O$ and $K_Z|_{I\cup O} = \overline{Z}_I \otimes \overline{Z}_O$. These correlation operators are compatible with the measurement pattern between the input and output slices. Thus, after

Aside 2.6: *Quantum correlations ≡ gate action*

In $\mathfrak{C}_{3D}$, logical gates (gates on logical qubits) are mediated by correlation operators. To see precisely what this means, consider a lattice $\mathscr{L}$ that supports logical qubits on its left and right most slices. The left most slice is the input slice and the right most slice is the output slice. Divide the lattice into three regions, I, M, O. I, O refer to the input and output slices respectively and M is the lattice between the two slices ($I \cap M = O \cap M = I \cap O = \varnothing$). Let $|\psi_{\mathscr{L}}\rangle$ be the quantum state of the lattice and $|\psi_{\mathscr{L}}\rangle_{I\cup O}$ be the state restricted to the input and output slices. By convention, the input and output slices are primal slices. For simplicity, we assume there is only a single logical qubit whose state on the input and output slices is denoted by $|\overline{\psi}\rangle_I$ and $|\overline{\psi}\rangle_O$, respectively. Note that $|\overline{\psi}\rangle_I$ refers to the state of the logical qubit on the input slice *before* measurement.

Now suppose that all qubits in the region M are individually measured in the X or Z basis as specified by the pattern of defects in M. Further suppose that after the measurement, the state $|\psi_{\mathscr{L}}\rangle_{I\cup O}$ obeys the eigenequations,

$$\overline{X}_I \otimes \left(\overline{U}\,\overline{X}_O\,\overline{U}^{\dagger}\right) |\psi_{\mathscr{L}}\rangle_{I\cup O} = (-1)^{\lambda_X} |\psi_{\mathscr{L}}\rangle_{I\cup O} \tag{2.14}$$

$$\overline{Z}_I \otimes \left(\overline{U}\,\overline{Z}_O\,\overline{U}^{\dagger}\right) |\psi_{\mathscr{L}}\rangle_{I\cup O} = (-1)^{\lambda_Z} |\psi_{\mathscr{L}}\rangle_{I\cup O}\,. \tag{2.15}$$

Where $\lambda_{X,Z} \in \{0,1\}$ depend, in general, on the measurement results from measuring M. Now, if the qubits on the input slice are measured, it follows from Theorem 1 of [RBB03] that the state of the logical qubit on the output slice is,

$$|\overline{\psi}\rangle_O = \overline{U} U_{\Sigma} |\overline{\psi}\rangle_I\,.$$

Where U_{Σ} is a by-product operator given by

$$U_{\Sigma} = \overline{Z}^{x+\lambda_X}\,\overline{X}^{\lambda_Z}\,.$$

Here $x \in \{0,1\}$ is the result of measuring $\overline{X}_I$, which is automatically measured when the qubits on the input slice are measured.

Thus, the correlations of Equations (2.14, 2.15) are equivalent to performing the gate $\overline{U}$ on the logical qubit. In $\mathfrak{C}_{3D}$ these correlations are simply the projections of the correlation operators $K(c_2)$ onto slices of the primal lattice.

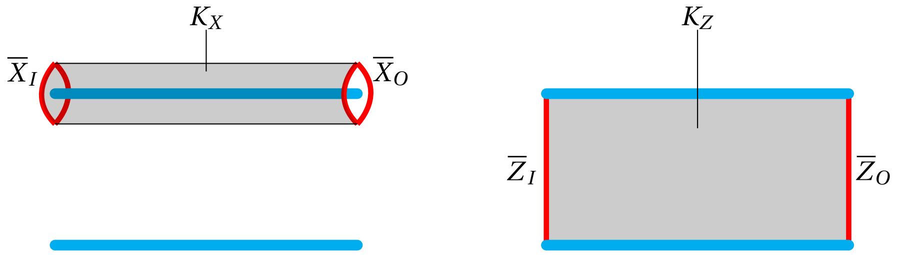


Figure 2.15 – Correlation surfaces for the identity gate.

all qubits between the input and output slices are measured, the remaining state $|\psi\rangle_{I\cup O}$ obeys the eigenequations

$$\overline{X}_I \otimes \overline{X}_O |\psi\rangle_{I\cup O} = \pm|\psi\rangle_{I\cup O}$$
$$\overline{Z}_I \otimes \overline{Z}_O |\psi\rangle_{I\cup O} = \pm|\psi\rangle_{I\cup O}.$$

Comparing with Equations (2.14, 2.15), we see that this construction implements the identity gate. The ± on the right hand side of these equations depends on whether there are any error chains affecting the correlation operators or not.

The $\Lambda(X)$ gate. The $\Lambda(X)$ gate maps control and target operators as

$$X_c \rightarrow X_c X_t \qquad X_t \rightarrow X_t$$
$$Z_c \rightarrow Z_c \qquad Z_t \rightarrow Z_c Z_t.$$

Then the correlator eigenequations for the $\Lambda(X)$ gate must be

$$\overline{X}_{c_I} \otimes \overline{X}_{c_O} \overline{X}_{t_O} |\psi\rangle_{I\cup O} = \pm|\psi\rangle_{I\cup O} \qquad \overline{X}_{t_I} \otimes \overline{X}_{t_O} |\psi\rangle_{I\cup O} = \pm|\psi\rangle_{I\cup O}$$
$$\overline{Z}_{c_I} \otimes \overline{Z}_{c_O} |\psi\rangle_{I\cup O} = \pm|\psi\rangle_{I\cup O} \qquad \overline{Z}_{t_I} \otimes \overline{Z}_{c_O} \overline{Z}_{t_O} |\psi\rangle_{I\cup O} = \pm|\psi\rangle_{I\cup O}.$$

The correlation operators that enforce these eigenequations are shown in Figure 2.16.

The non-Clifford gates. The gates $\exp(i\frac{\pi}{8}Z)$, $\exp(i\frac{\pi}{4}Z)$ and $\exp(i\frac{\pi}{4}X)$ are implemented with the help of the ancilla states $|A\rangle := \left(|0\rangle + e^{i\frac{\pi}{4}}|1\rangle\right)/\sqrt{2}$ and $|Y\rangle := (|0\rangle + i|1\rangle)/\sqrt{2}$, eigenstates of the $A = \frac{X+Y}{\sqrt{2}}$ and Y operators respectively. Recall that we discussed how to initialize a logical qubit in one of these states. The initialization procedure is not fault tolerant, due to the use of singular qubits. As a result, the logical qubit must be distilled. This is discussed in the next section. Once we have nice, distilled ancilla states, we can use the circuits shown in Figure 2.17 to perform the gates. The defect configurations are the simplest possible that still yield the correct correlations. Checking this is left as an exercise to the reader.

The circuit for $\exp(i\frac{\pi}{8}Z)$ is probabilistic and succeeds with probability 1/2. On failure, the gate $\exp(-i\frac{\pi}{8}Z)$ is applied instead. This can be corrected for by a subsequent application of the $\exp(i\frac{\pi}{4}Z)$, which is deterministic. Thus, on average, one logical $\exp(i\frac{\pi}{8}Z)$ and 1/2 a logical $\exp(i\frac{\pi}{4}Z)$ have to be performed for every $\exp(i\frac{\pi}{8}Z)$ in the ideal circuit.

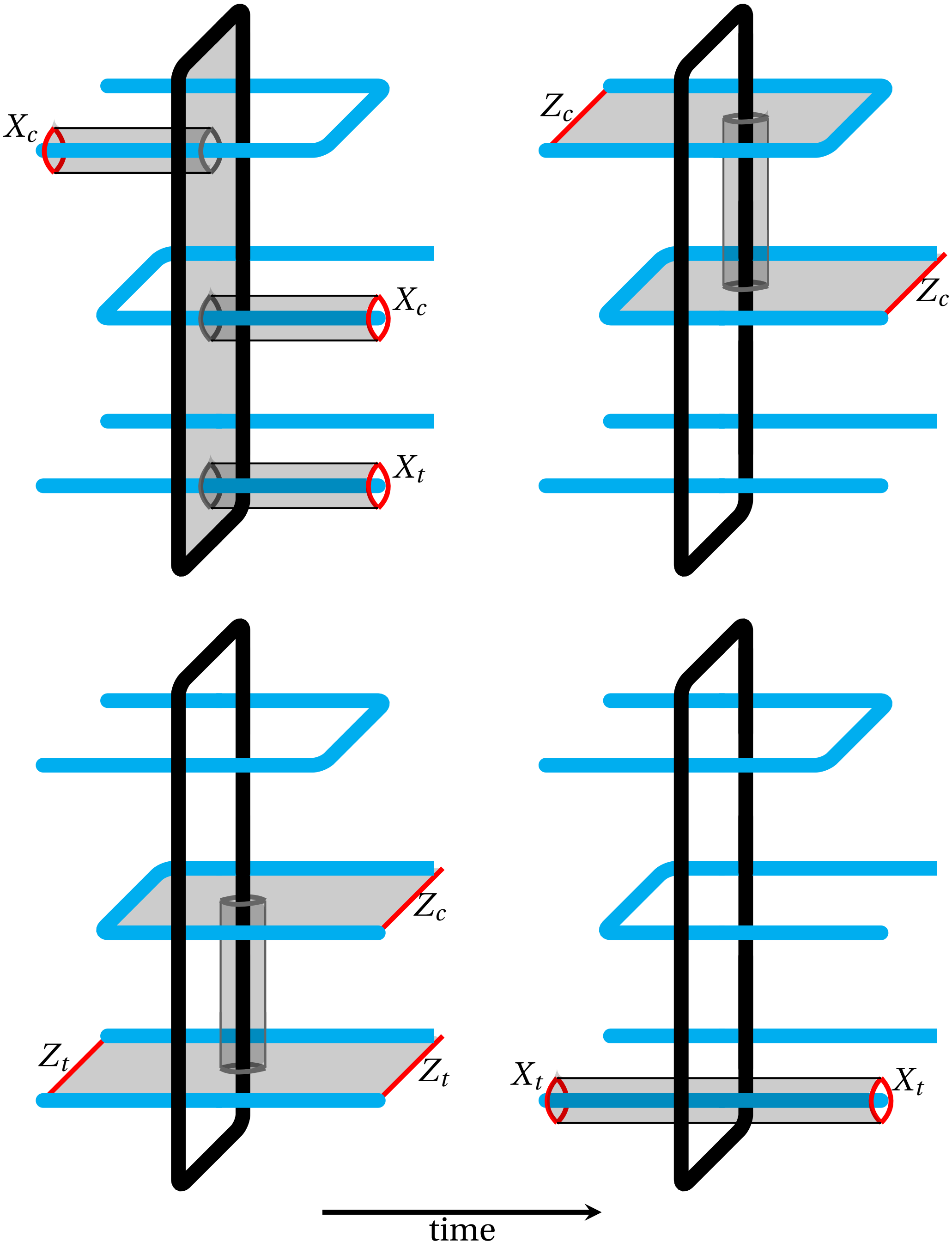


Figure 2.16 – The correlation surfaces for the $\Lambda(X)$ gate. The upper two primal defects constitute the control qubit. The lower primal defect is the target qubit. The dual defect mediates correlations between the control and target qubits, as shown.

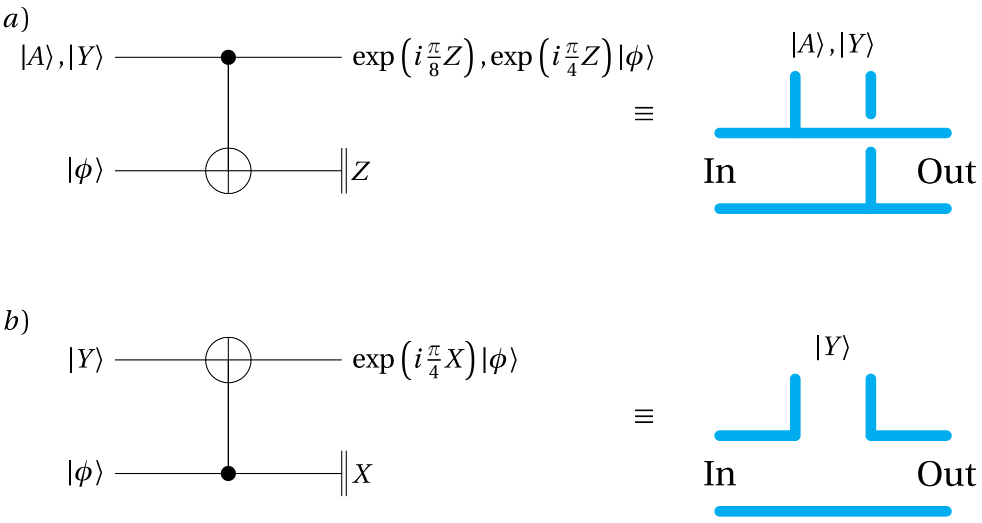


Figure 2.17 – Circuits to perform non-Clifford gates. *a*) Circuit for $\exp(i\frac{\pi}{8}Z)$ and $\exp(i\frac{\pi}{4}Z)$. *b*) Circuit for $\exp(i\frac{\pi}{4}X)$.

State distillation The ancilla states used to perform non-Clifford gates are noisy and need to be distilled. We use a variant of the magic state distillation circuit described in [BK05a], adapted to $\mathfrak{C}_{3D}$. The circuits for the distillation of $|A\rangle$ and $|Y\rangle$ states are shown in Figure 2.18. The circuits use only topologically protected gates (X,Z measurement and $\Lambda(X)$). At each level of distillation, $|A\rangle$ distillation requires 15 $|A\rangle$ states at the lower level and $|Y\rangle$ distillation requires 7 $|Y\rangle$ states at the lower level.

2.2.6 Making It Fault Tolerant

So far the discussion has been restricted to describing the logical structure of $\mathfrak{C}_{3D}$ and explaining how the quantum gates work. We have not really talked about fault tolerance. Indeed, the structures defined in previous sections are *not* fault tolerant. This is because the involved defects are thin and close together. We saw in Section 2.2.4 that error chains leave syndrome at their ends. However, an error chain that ends in a pair of defects will leave no syndrome. Similarly, an error chain that forms a closed loop around a defect has no ends and thus leaves no syndrome. Both these types of error chains lead to logical errors, as illustrated in Figure 2.20.

To make $\mathfrak{C}_{3D}$ fault tolerant, we rescale the elementary cell to a logical cell made up of $\lambda \times \lambda \times \lambda$ elementary cells. The cross-section of a defect with the perpendicular slice becomes an area of $d \times d$ elementary cells (see Figure 2.19). The building block of our computer is now this logical cell. Each gate is composed of a number of logical cells. For example, in Figure 2.21, we see a CNOT gate made up of logical cells. This construction guarantees a distance of $2(\lambda - d)$ between neighboring defects as well as a minimum length of $2(d+1)$ for chains that circle defects (this is because the error correction procedure can turn a chain of length $2(d+1)$ into a loop encircling a defect). For near-local error models, this means an exponential suppression of logical error probability in these lengths. A detailed analysis of the overhead imposed by this fault tolerance scheme can

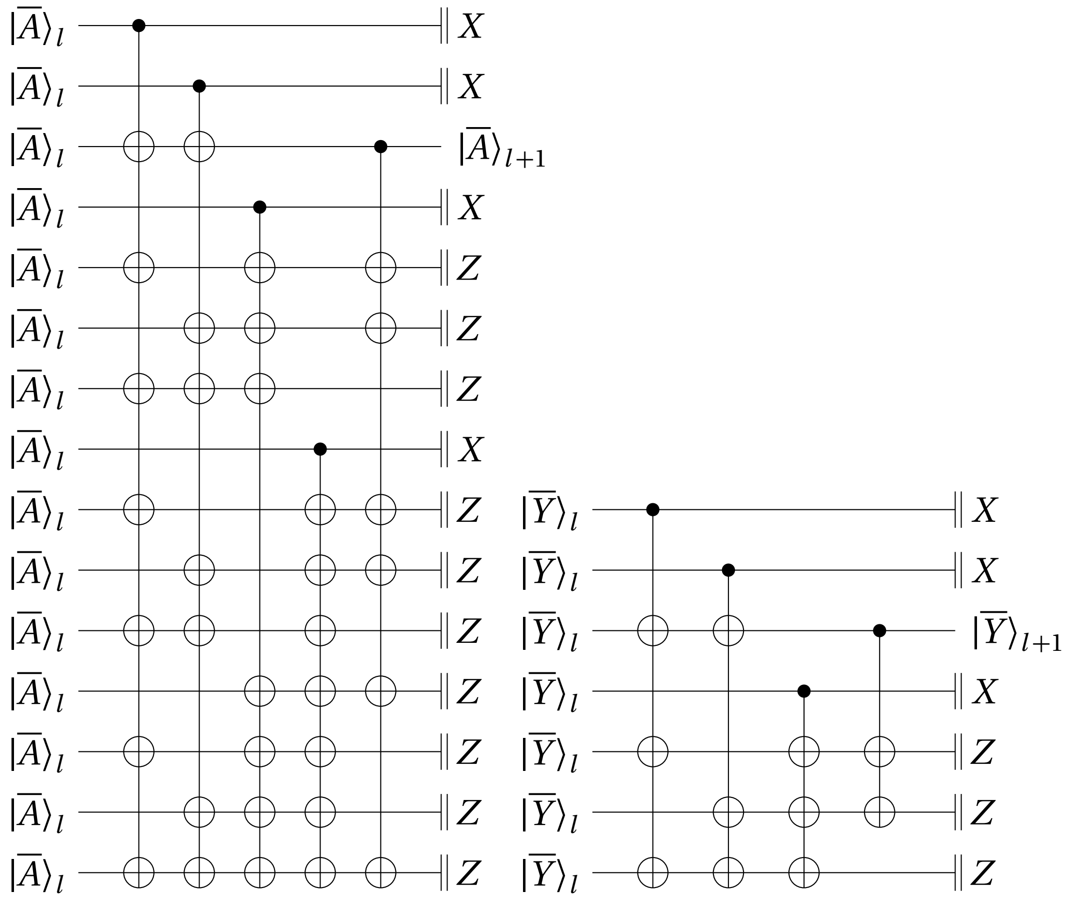


Figure 2.18 – Distillation circuits for $|A\rangle$ and $|Y\rangle$. The circuits use only topologically protected operations.

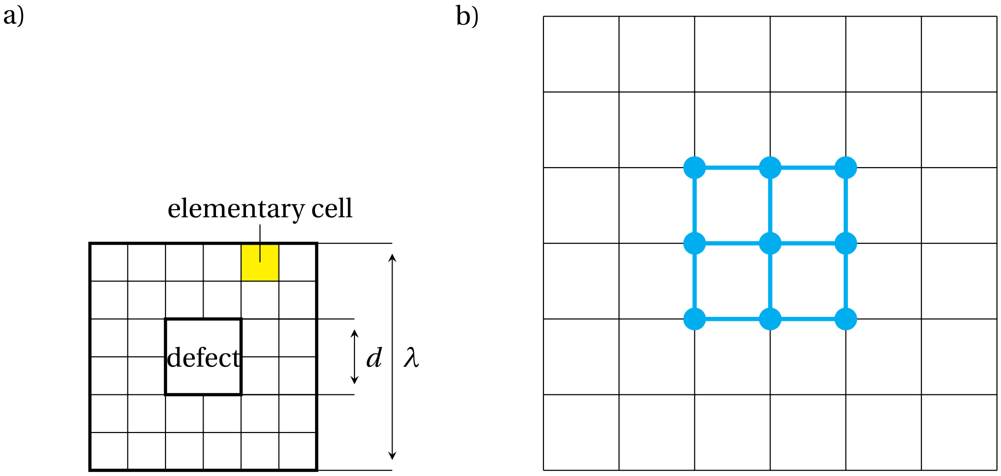


Figure 2.19 – a) The logical cell. It is rescaled from the elementary cell of the lattice by a factor of λ in each direction. The defects have cross-section $d \times d$. The elementary cell can be either a primal or a dual cell, corresponding to the type of defect being considered. b) The measurement pattern associated with a defect. Blue edges are measured in the Z basis. The blue dots indicate that the edges perpendicular to the slice at those locations are also measured in the Z basis, creating “holes.”

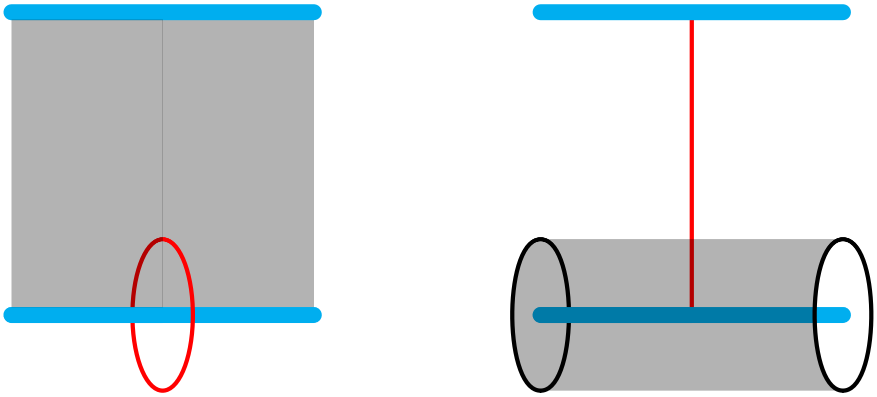


Figure 2.20 – Error chains leading to logical errors, affecting an identity gate. The first chain leads to a logical $\overline{Z}$ error, while the second leads to a logical $\overline{X}$ error. Because these chains have no endings (the second chain ends in a defect), they leave no syndrome and cannot be corrected.

be found in Chapter 3.

2.3 Mapping To A Two Dimensional System

So far, we have discussed computation on a three dimensional lattice. The $3D$ nature of $\mathfrak{C}_{3D}$ is dictated by the need to use two dimensions to encode the logical qubit. While this cannot be got around, it is possible to implement the three dimensional computation on a $2D$ physical lattice. The $2D$ lattice is not a slice of the full $3D$ lattice. Instead it has qubits on faces, edges *and* vertices. It can be thought of as a body-centered-cubic lattice half a cell thick.

In the $3D$ version, we use $|+\rangle$ preparation and $\Lambda(Z)$ gates for the creation of $\mathscr{L}$ and subsequently perform local measurements in the X, Y, Z and $X+Y$ bases. These operations can be mapped to the $2D$ lattice as follows. Note that this mapping does not affect information processing. In particular, error correction and gate logic are still the same as in $\mathfrak{C}_{3D}$. The temporal order of operations is shown in Figure 2.22.

1. *Space-like edges (primal and dual).* For each such edge, group together the $|+\rangle$ preparation, measurement and trailing time like oriented $\Lambda(Z)$ gate, and denote the combination by $\{|+\rangle, \Lambda(Z), P\}$. P denotes the measurement of the space-like edge. If the measurement on the trailing end of the trailing time-like $\Lambda(Z)$ gate is in the Z basis, then

   $$\{|+\rangle, \Lambda(Z), P\} \longmapsto P, \tag{2.16a}$$

   otherwise,

   $$\{|+\rangle, \Lambda(Z), P_X\} \longmapsto H, \tag{2.16b}$$

   $$\{|+\rangle, \Lambda(Z), P_{X+Y}\} \longmapsto He^{i\frac{\pi}{8}Z}, \tag{2.16c}$$

   $$\{|+\rangle, \Lambda(Z), P_Y\} \longmapsto He^{i\frac{\pi}{4}Z}, \tag{2.16d}$$

   $$\{|+\rangle, \Lambda(Z), P_Z\} \longmapsto P_X. \tag{2.16e}$$

2. *Time-like edges (primal and dual).* For each such edge, group together the preparation and measurement, and denote the combination by $\{|+\rangle, P\}$. P denotes the measurement of the time-like edge. Then,

   $$\{|+\rangle, P_Z\} \longmapsto I, \tag{2.17a}$$

   $$\{|+\rangle, P\} \longmapsto \{|+\rangle, P\}, \text{ for } P \neq P_Z. \tag{2.17b}$$

3. *Space-like oriented* $\Lambda(Z)$ *gates.*

   $$\Lambda(Z) \longmapsto \Lambda(Z) \tag{2.18}$$

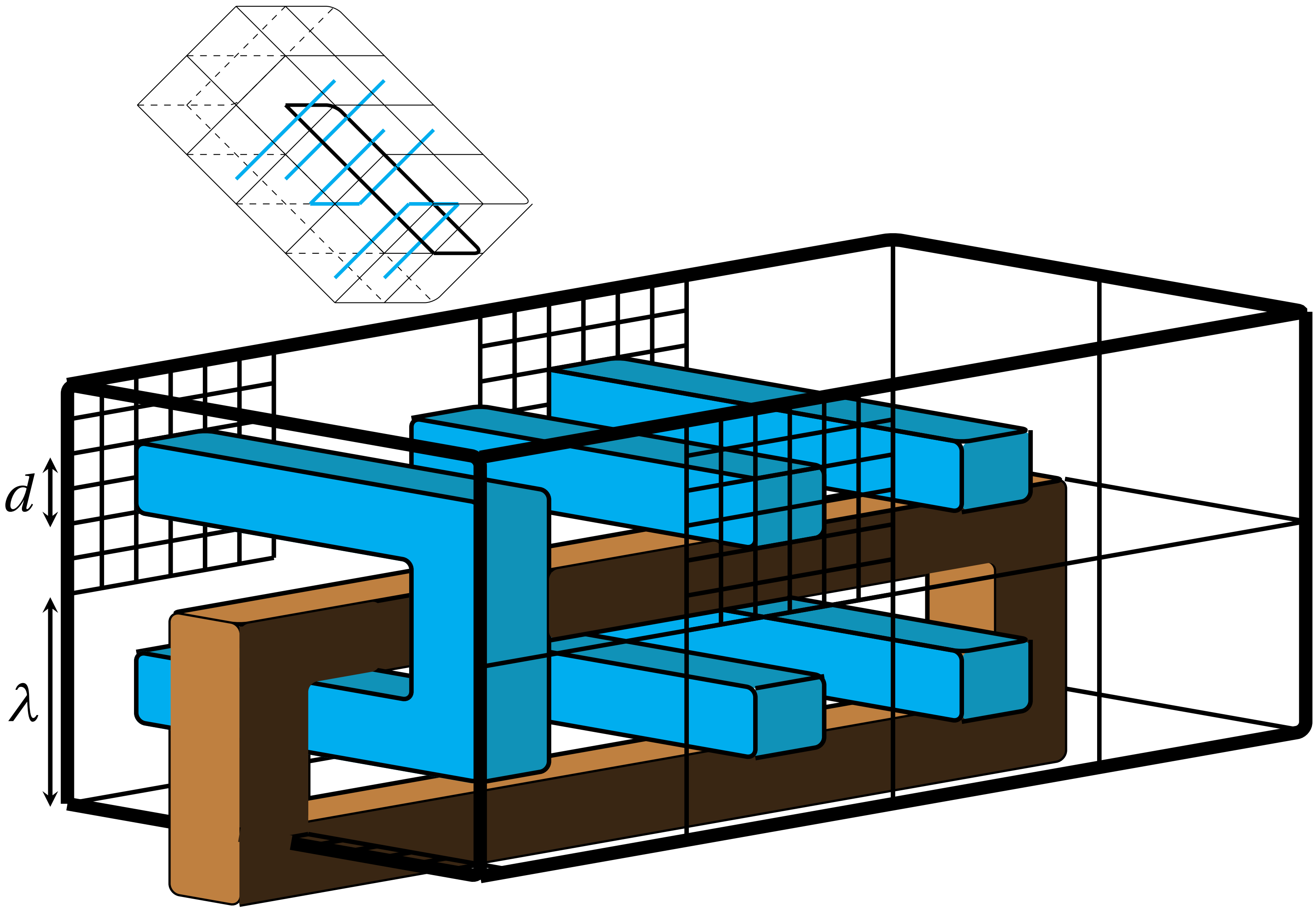


Figure 2.21 – Example of a fault-tolerant $\Lambda(X)$ with $d = 2$ and $\lambda = 6$. Note that all cells shown are primal cells. The dual defect (brown) is made up of dual cells. The dual defect overlaps two primal logical cells in every plane. It does not matter that the dual defect is close the to the primal defect, as error chains cannot run between defects of different types. The line diagram offers an alternate perspective on the defects.

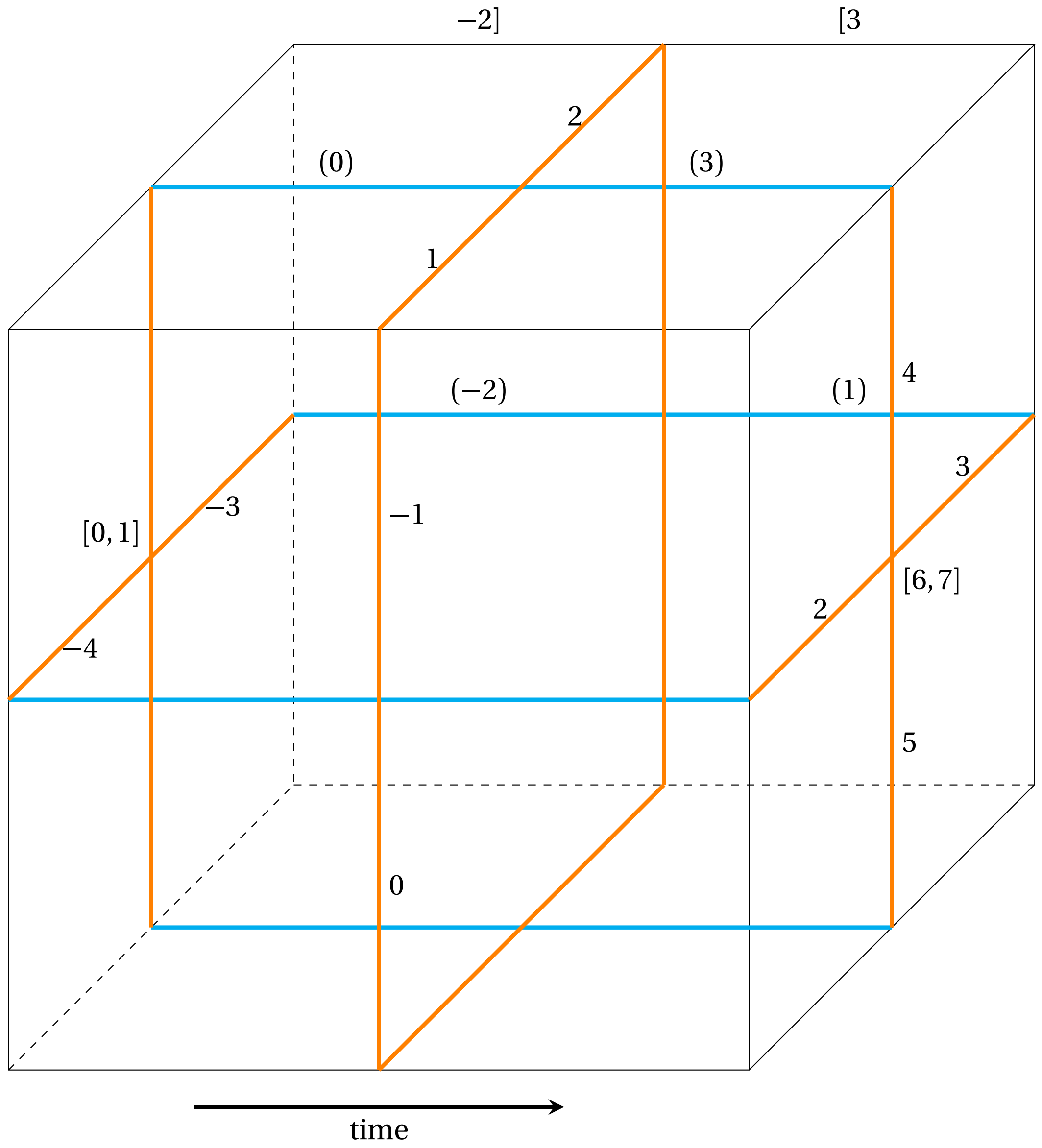


Figure 2.22 – Temporal order of operations in the bulk of the lattice (away from any defects or singular qubits), for the $2D$ version of the $\mathfrak{C}_{3D}$. Shown is the elementary cell of $\mathscr{L}$ with one axis converted to time. The labels on the edges denote the time steps at which the corresponding $\Lambda(Z)$ gate is performed. The labels at the syndrome vertices denote measurement and (re-)preparation times $[t_M, t_P]$, and the labels (t_H) denote times for Hadamard gates. The pattern is periodic in space, and in time with period six.

Note that under the $2D$ mapping, no qubit in the scheme is ever idle between preparation and measurement. The identity in Eq. (2.17a) can be replaced by the single qubit completely depolarizing channel without affecting the computation, as it will be re-initialized prior to its next use. For a physical realization, it may be preferable to use a double layer $2D$ structure instead of a single layer. The advantage then is that all qubits within one layer, including the singular qubits, can be read out simultaneously. One 'clock cycle' consists of the following steps: 1) Ising interaction/$\Lambda(Z)$ gates between all pairs of nearest neighboring qubits in the lattice; 2) Simultaneous measurement of all qubits in layer a, re-preparation of all qubits in layer a; 3) Same as 1); 4) Same as 2), with $a \longleftrightarrow b$.

2.4 Summary

We can summarize the structure of $\mathfrak{C}_{3D}$ as:

- A $3D$ cubic lattice with qubits on edges and faces.
- Logical qubits are encoded on $2D$ slices of this lattice.
- Logical qubits are encoded by "holes." These holes extend in the third dimension to become "defects."
- Primal defects encode the logical qubits, while dual defects encode correlations between them (by winding around the primal defects).
- Most of the qubits in the lattice are measured in the X basis. These measurements teleport logical qubits from one slice to the next.
- Qubits in the defects are measured in the Z basis. These measurements relax site operators, creating quasi-particles or "holes."
- Defects also serve as boundaries for correlation surfaces and error chains, thereby giving the lattice a non-trivial topology.
- Logical qubits can be fault tolerantly initialized in the $|\bar{0}\rangle$ and $|\bar{+}\rangle$ states. They can also be initialized in rotated states ($|\bar{0}\rangle + e^{i\phi}|\bar{1}\rangle$) via measurements in rotated bases, but they need to be subsequently distilled.
- The following gates can be performed fault tolerantly on logical qubits, just by using the appropriate defect configurations: I, (X, Z)-preparation/measurement and $\Lambda(X)$.
- This set of gates is sufficient to perform magic state distillation, to create pure logical ancilla qubits in rotated states.
- These rotated states can, in turn, be used to perform the non-Clifford gates, completing a universal set.

- In order for the gates to be fault tolerant, it is necessary for the defects to be thick.
- The scheme can be physically implemented on a $3D$ lattice, two $2D$ layers or even a single $2D$ layer.

Chapter 3

Analyzing the Performance of the Fault Tolerant Computer

We analyze the performance of the fault tolerant computer by comparing its operational overhead with the ideal computer (i.e., the computer that operates in the absence of noise). The are two parameters of interest; the *threshold* and the *overhead*. The threshold is a noise rate below which the fault tolerant computation succeeds with high probability. In this work, we consider only stochastic error models, so the noise rate is simply the probability of an elementary (physical) operation failing. For an ideal computer, the threshold is 0. For $\mathfrak{C}_{3D}$, the threshold is 7.8×10^{-3}.

The overhead is a measure of the extra operations that need to be performed, on average, per operation in the corresponding ideal computation. Another, less precise measure, of overhead is the number of qubits the fault tolerant computer requires to fault tolerantly implement a given ideal circuit. For $\mathfrak{C}_{3D}$, the overhead scales poly-logarithmically with the size (number of gates) of the ideal circuit.

Section 3.1 has a discussion of the error model and its corresponding threshold estimate. Section 3.2 discusses the overhead of our scheme, and finally, Section 3.3 presents a more detailed description of some of the numerical techniques used to arrive at the threshold results.

3.1 Threshold

Error Model In order to arrive at an estimate of the threshold, we must first specify an error model. We assume the following:

1. Erroneous operations are modeled by perfect operations preceded/followed by a partially depolarizing single- or two-qubit error channel

$$T_1 = (1-p_1)[I] + \frac{p_1}{3}([X]+[Y]+[Z])$$
$$T_2 = (1-p_2)[I] + \frac{p_2}{15}([X_aX_b]+\cdots+[Z_aZ_b]).$$

2. The error sources are

   - Faulty preparation of individual qubit states $|+\rangle$, with error probability p_P,
   - Noisy Hadamard gates with error probability p_1 (Hadamard gates are used in the $2D$ modification in place of $\Lambda(Z)$ gates in the time-like direction),
   - Noisy $\Lambda(Z)$ gates with error probability p_2, and
   - Noisy measurement with error probability p_M.

   Note that in the modified scheme there is no storage error as no qubit is idle between preparation and measurement (see Figure 2.22).

3. Classical processing is instantaneous.

When calculating a threshold, we assume all error sources to be equally strong, $p_1 = p_2 = p_M = p_P := p$, so that the noise strength is described by a single parameter p.

Error correction. When estimating the threshold, two key facts about error correction in the cluster state must be noted:

- The error correction in the bulk region of the cluster state is topological. It can be mapped to the *random plaquette* $\mathbb{Z}_2$*-gauge model* (RPGM) in three dimensions [Den+02b]. If there are non-trivial error chains of finite smallest length l then, below the error threshold, the probability of error ε_{top} is

  $$\varepsilon_{top} \sim \exp(-\kappa(p)l). \tag{3.1}$$

  $\kappa(p)$ is some function of the error rate p that depends on the details of the error model. For a simple model of purely local, independent errors on each qubit of the lattice, we can calculate $\kappa(p)$. Consider a defect of circumference u and length l. An error chain winding around the defect has weight at least $u+4 \approx u$, and there are N such minimum weight chains. Thus, the probability $\varepsilon_{top}(u,N)$ for a logical error due to these types of chains is, to lowest contributing order,

  $$\varepsilon_{top}(u,l) = N\frac{u!}{\frac{u}{2}!^2}q^{\frac{u}{2}} \approx N\exp\left(\frac{\ln 4q}{2}u\right)\frac{1}{\sqrt{\frac{\pi}{2}u}}. \tag{3.2}$$

  Where q is the local error probability. Here, $\kappa(p) = -\frac{\ln 4q}{2}$. For the mildly non-local error model described above, we can estimate $\kappa(p)$ numerically (see Eq. (3.16)).

- Topological error correction breaks down near the singular qubits. This results in an effective error on the singular qubits that needs to be taken care of by an additional correction method, namely magic state distillation. This effective error is *local* because the singular qubits are well separated from one another, as we saw in Section 2.2.2.

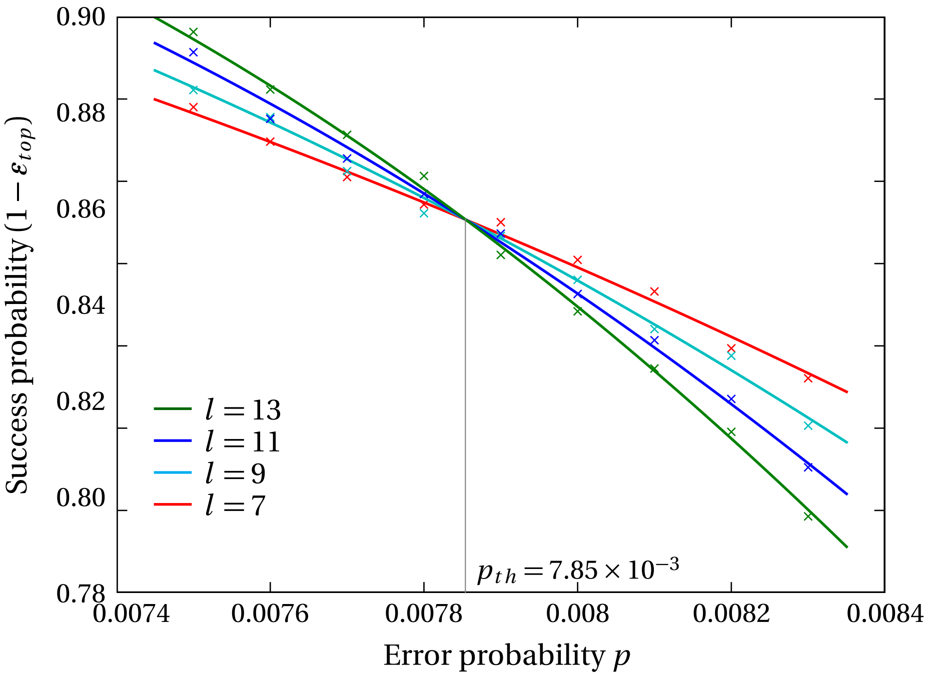


Figure 3.1 – Numerical simulation for the topological threshold. The curves are best fits taking into account finite size effects of the lattice size l. Beyond the smallest lattices, these finite size effects quickly vanish, and the curves intersect in a single point to a very good degree of accuracy. The value of p at the intersection gives the threshold.

The fault tolerance threshold associated with the RPGM is about 3.2×10^{-2} [Ohn+04; TN04], for a strictly local error model with one source. The threshold estimates given in this section are based on the *minimum weight chain matching algorithm* [Edm65] for error correction. This algorithm yields a slightly smaller threshold of 2.9% [WHP03] but is computationally efficient. For details see Section 3.3.

We now consider the singular qubits (S-qubits). They are used to create the noisy ancilla states $\rho^A \approx |A\rangle\langle A|, \rho^Y \approx |Y\rangle\langle Y|$ encoded via the construction displayed in Figure 2.17. $|A\rangle$, $|Y\rangle$ states are the eigenstates of the $\frac{X+Y}{2}$ and Y operators, respectively. See Section 2.2.3 for a discussion of the creation of ancilla states. Due to the effective error on the singular qubits, these ancilla states before distillation carry an error $\varepsilon_0^A := 1 - \langle A|\rho^A|A\rangle$, $\varepsilon_0^Y := 1 - \langle Y|\rho^Y|Y\rangle$ given by

$$\varepsilon_0^A = \varepsilon_0^Y = 6p. \tag{3.3}$$

The $6p$ term depends on the error model, and is calculated by counting the shortest chains in the vicinity of a singular qubit. It is accurate to lowest order in p. For a purely local error model consisting of only Z errors acting independently at each qubit, the lowest order term would be of order p^2, with a different coefficient.

Threshold. There are two types of threshold within the cluster, namely the topological one in the bulk of the lattice, far away from the singular qubits and thresholds from

$|A\rangle$ and $|Y\rangle$ state distillation. An estimate p^V_{th} for the topological threshold is found in numerical simulation of finite-size lattices to be

$$p^V_{th} = 7.8 \times 10^{-3}. \tag{3.4}$$

The results of the numerical simulation are shown in Figure 3.1 (see Section 3.3 for details).

The recursion relations for state distillation, in the limit of negligible topological error, are to lowest contributing order $\varepsilon^A_{l+1} = 35(\varepsilon^A_l)^3$ (c.f. [BK05b]) and $\varepsilon^Y_{l+1} = 7(\varepsilon^Y_l)^3$. The circuits for state distillation are shown in Figure 2.18. The corresponding distillation thresholds expressed in terms of the physical error rate p are

$$p^A_{th} = \frac{1}{6\sqrt{35}} \approx 2.8 \times 10^{-2}, \;\; p^Y_{th} = \frac{1}{6\sqrt{7}} \approx 6.3 \times 10^{-2}.$$

This threshold is the physical error rate below which the distillation would always succeed, provided the gates used in the distillation circuit were ideal. The physical error rate is calculated by solving the recursion relations above, and then calculating the physical error rate that would yield an effective error, on the logical qubits, at the lowest level, that is just below the distillation threshold. Since the topological threshold is much smaller than the distillation threshold, below the topological threshold, the (topologically protected) gates used in the distillation circuits are indeed ideal. Therefore the topological threshold sets the overall threshold for fault tolerant quantum computation.

3.2 Overhead

We would like to analyze the resource requirements for this scheme. For the analysis, the metric we use is the *operational cost per gate*, O_3. It is defined as the number of physical operations that need to be performed per logical gate in a circuit of size N_G (i.e., a circuit that has N gates of type G).

The fault tolerant construction is discussed in detail in Section 2.2.6. Logical operations are realized by twisting defects around each other. If there is no noise, it would be sufficient to have line like defects. Then, the elementary cells, Figure 2.5, would be the building blocks from which gates are constructed. However, in the presence of noise, there will be short error cycles wrapping around the cells, leading to logical errors with high probability. In order to achieve fault tolerance, we have to rescale the elementary cell to a cube of $\lambda \times \lambda \times \lambda$. The cross section of the defect with the plane perpendicular to the time direction becomes a square of area $d \times d$ (see Figure 2.19). Remember that as per Eq. (3.1) errors are suppressed exponentially in the length of the shortest chain.

The gate length L_G is the total length of defect within a gate in units of the rescaled cell. The gate volume V_G is the number of rescaled cells the gate G occupies. Each rescaled cell consists of λ^3 elementary cells and each elementary cell is built with 12 operations. The number of two qubit operations per elementary cell for a $3D$ lattice is $(6 \times 4)/2 = 12$. For the $2D$ modification, this number is 8, so we will use 12, as an upper bound.

Let $\varepsilon_{top}(G,\lambda,d)$ be the probability of failure of a topologically protected gate, such as the $\Lambda(X)$ gate. It is a function of the layout (i.e., defect structure) of the gate, defect thickness d and scale factor λ. The operational overhead $O_3(G)$ is then

$$O_3(G) = \min_{\lambda,d} 12\lambda^3 V_G \exp\left(\varepsilon_{top}(G,\lambda,d)N_G\right). \tag{3.5}$$

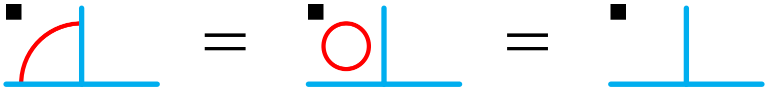
The exponential factor comes from the expected number of repetitions for a circuit composed of N_G gates G. For a given N_G, the overhead should be minimized by choosing $\lambda(N_G)$ and $d(N_G)$.

3.2.1 Clifford Gates

The overhead analysis is simplest for Clifford gates (i.e., gates that do not involve singular qubits and thus do not need distillation). To perform the minimization in Eq. (3.5), we need the gate error ε_{top} as a function of G,λ,d. The error chains leading to gate failure can be either cycles wrapping around defects or chains that start and end on defects (relative cycles). The probability of gate failure is exponential in the length of the shortest such chain and proportional to the number of such chains.

For a given defect thickness d, the minimal cycle length is $4(d+1)$ and the number of such cycles is λL_G; where L_G is the gate length. The minimal length of a relative cycle leading to an error is $\lambda - d$. It stretches between two neighboring defect segments one logical cell apart. The number of such relative cycles is at most $2L_G\lambda(d+1)$. There are shorter relative error cycles near junctions, but they are equivalent to the identity operation. This can be easily verified with the help of the appropriate correlation surfaces, as discussed in Chapter 2. Schematically,

(3.6)



Thus, by Eq. (3.1), the gate failure rate is,

$$\varepsilon_{top}(G,\lambda,d) = \lambda L_G\left(\exp(-4\kappa(d+1)) + 2(d+1)\exp(-\kappa(\lambda-d))\right). \tag{3.7}$$

In Eq. (3.2), $\kappa = \frac{\ln 4q}{2}$. However, for non-local error models, κ must be estimated numerically, by simulation on finite sized lattices (see Figure 3.2). In addition, Eq. (3.2) predicts a polynomial correction $l^{-1/2}$ to ε_{top}, for local noise. The numerical simulation finds a polynomial correction $l^{-0.85}$ for the non-local error model proposed above. We neglect the polynomial correction in our estimate of the overhead in Eq. (3.7). This is safe as including it would only reduce the overhead. We can now use this expression to perform the minimization in Eq. (3.5). For example, the overhead for the $\Lambda(X)$ gate is plotted in Figure 3.3.

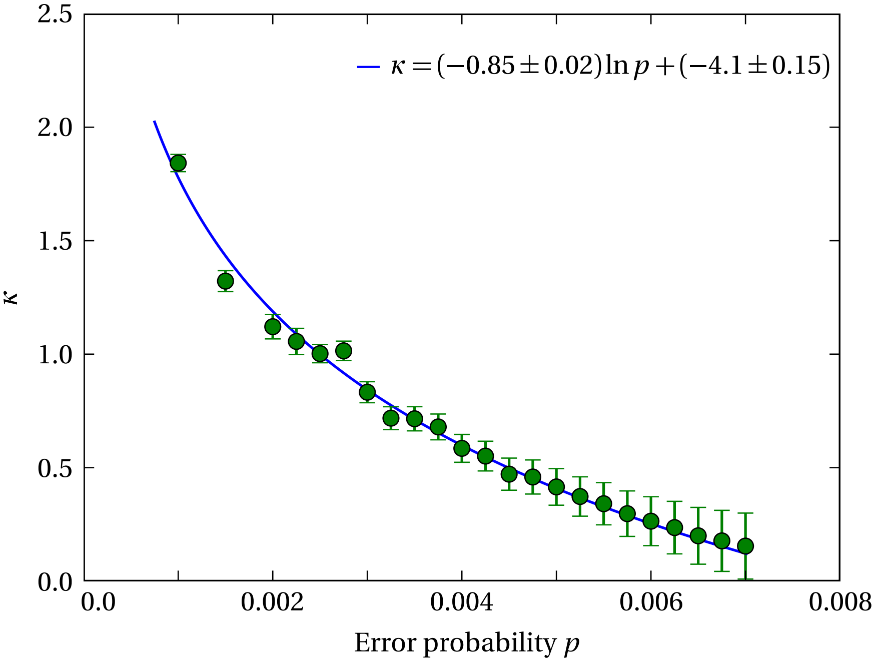


Figure 3.2 – The dependence of κ on the error rate p below the threshold. The curve is the best fit to the data points that come from numerical simulation on finite sized lattices.

3.2.2 Non-Clifford Gates

Non-Clifford gates require the use of singular qubits to prepare the ancilla states needed to perform the gates via gate teleportation. Remember that this preparation in not fault tolerant. So we must use magic state distillation to purify the ancilla. The gates used in the distillation procedure are all Clifford gates and thus are topologically protected as above. However, the distillation procedure is *concatenated*. At each level of concatenation, the optimal λ and d that minimize the overhead in Eq. (3.5) are different. Thus, the minimization is over the larger set of parameters, $\Lambda = \{\{\lambda_0, d_0, \lambda_1, d_1, \ldots, \lambda_{l_{max}}, d_{l_{max}}\}, l_{max}\}$.

In addition, there are now two types of error. First, the previously discussed error from non-trivial error cycles far away from the singular qubits. Second, the error associated with the singular qubits themselves, in whose neighborhood topological error correction breaks down, due to the presence of short error chains. The singular qubits are used for the distillation of $|A\rangle$ and $|Y\rangle$ states, the eigenstates of the $\frac{X+Y}{2}$ and Y operators, respectively.

Distillation of $|A\rangle$ and $|Y\rangle$ states is performed using the circuits shown in Figure 2.18, of volumes V_A, V_Y and length L_A, L_Y (see Table 3.2). Let $p_l^{A,Y}$ be the probability of success for the distillation at level l of $|A\rangle, |Y\rangle$. On success, there will be a residual error at level l denoted by $\epsilon_l^{A,Y}$ for $|A\rangle, |Y\rangle$, respectively. The performance of the distillation scheme is shown in Table 3.1. The expressions for success probability and residual errors hold to leading order in the contributing error probabilities $\epsilon_l^A, \epsilon_l^Y$. Further, a gate error cannot simultaneously lead to termination of the circuit and to a residual distillation error. Thus, we overestimate both error probabilities by adding the full weight

Table 3.1 – Performance of magic state distillation

state at level l	required states at level $l-1$	performance
$\|A\rangle$	$15\|A\rangle$	$p_l^A = 1 - 15\varepsilon_{l-1}^A - \varepsilon_{top}(L_A, \lambda_{l-1}, d_{l-1})$ $\varepsilon_l^A = 35(\varepsilon_{l-1}^A)^3 + \varepsilon_{top}(L_A, \lambda_{l-1}, d_{l-1})$
$\|Y\rangle$	$7\|Y\rangle$	$p_l^Y = 1 - 7\varepsilon_{l-1}^Y - \varepsilon_{top}(L_Y, \lambda_{l-1}, d_{l-1})$ $\varepsilon_l^Y = 7(\varepsilon_{l-1}^Y)^3 + \varepsilon_{top}(L_Y, \lambda_{l-1}, d_{l-1})$

Table 3.2 – The gates sizes for various gates and sub-circuits

gate	volume	length
$\Lambda(X)$ Figure 2.21b	$V_2 = 16$	$L_2 = 20$
$\exp(i\frac{\pi}{8}Z)$ Figure 2.17	$V_{1,z} = 2$	$L_{1,z} = 3$
$\exp(i\frac{\pi}{4}X)$ Figure 2.17	$V_{1,x} = 4$	$L_{1,x} = 4$
$\|A\rangle$ distillation circuit	$V_A = 168$	$L_A = 266$
$\|Y\rangle$ distillation circuit	$V_Y = 60$	$L_Y = 94$

$\varepsilon_{top}(L, \lambda_{l-1}, d_{l-1})$ to them.

The operational overheads for state distillation at level l, $O_{3,l}^A$ and $O_{3,l}^Y$ are given by the recursion relations

$$O_{3,l}^A = \frac{1}{p_l^A}\left(15 O_{3,l-1}^A + 12\lambda_{l-1}^3 V_A\right)$$

$$O_{3,l}^Y = \frac{1}{p_l^Y}\left(7 O_{3,l-1}^Y + 12\lambda_{l-1}^3 V_Y\right). \tag{3.8}$$

The recursion relations for $p_l^{A,Y}, \varepsilon_l^{A,Y}$ are given in Table 3.1. The initial conditions are $O_{3,0}^A = O_{3,0}^Y = 12$ and Eq. (3.3).

The distillation outputs states $|A\rangle$ and $|Y\rangle$ at level l_{max}. One such state $|A\rangle$ and, on average, 1/2 state $|Y\rangle$ is used to implement the $\exp(i\frac{\pi}{8}Z)$ gate via the construct displayed in Figure 2.17, of volume $V_{1,z}$ and length $L_{1,z}$. Its overhead is

$$O_3^{\pi/8} = \left(O_{3,l_{max}}^A + \frac{1}{2}O_{3,l_{max}}^Y + 18\lambda_{l_{max}}^3 V_{1,z}\right) \times \exp\left((\varepsilon_{l_{max}}^A + \varepsilon_{l_{max}}^Y + \varepsilon_{top}(L_{1,z}, \lambda_{l_{max}} d_{l_{max}}))N_G\right). \tag{3.9}$$

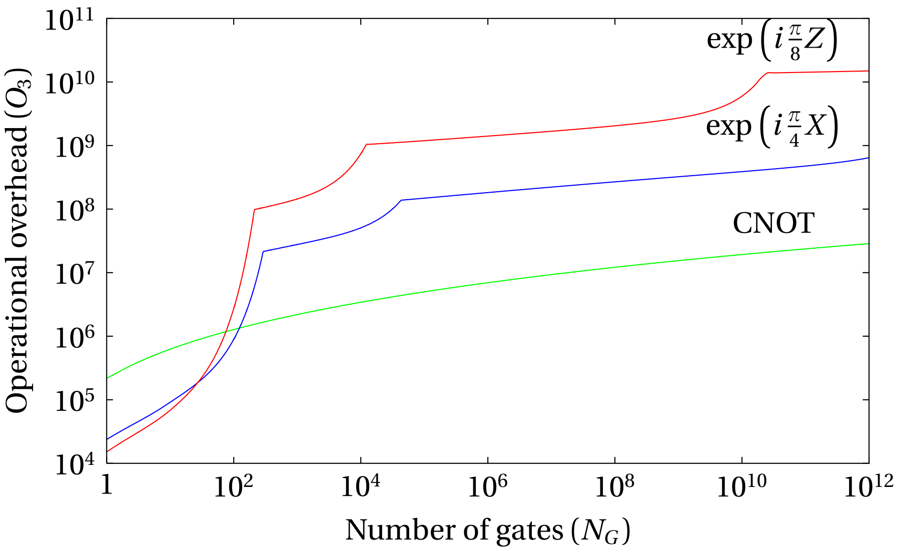


Figure 3.3 – Operational overhead as a function of the circuit size at $p = \frac{p_{th}}{3}$. The kinks in the curves correspond to increasing levels of concatenation l.

Similarly, the overhead for the $\exp(i\frac{\pi}{4}X)$ gate is

$$O_3^{\pi/4} \quad = \quad \left(O_{3,l_{max}}^Y + 12\lambda_{l_{max}}^3 V_{1,x}\right) \times \exp\left((\varepsilon_{l_{max}}^Y + \varepsilon_{top}(L_{1,x}, \lambda_{l_{max}} d_{l_{max}}))N_G\right). \tag{3.10}$$

The operational overhead must be minimized over the parameter set Λ. This has been done numerically[BLN95], and the result is shown in Figure 3.3.

3.2.3 Overhead Scaling

The overhead for a given gate depends on two main parameters; the circuit size N_G and the underlying error probability p. In this section, we examine the behavior of the overhead with respect to each of these parameters.

3.2.3.1 The Large N_G Limit

For Clifford gates, first notice that the residual topological error ε_{top} in Eq. (3.7) is minimized when both exponentials fall off equally fast, i.e., $d_{min} = \lambda_{min}/5$ for large d, λ. Further, the overhead O_3 in Eq. (3.5) is minimized near

$$\varepsilon(\lambda(N_G)) = 1/N_G. \tag{3.11}$$

Then, $\lambda_{min} \sim \ln N_G/\kappa$, and

$$O_3 \sim \frac{\ln^3 N_G}{\kappa^3}. \tag{3.12}$$

For non-Clifford gates, we first compare the two contributions to ε_l^A (see Table 3.1),

$35(\varepsilon^A_{l-1})^3$ and ε_{top}. If ε_{top} is much larger than $35(\varepsilon^A_{l-1})^3$, it inhibits the convergence of ancilla distillation, forcing the use of additional distillation rounds, which are the most expensive component. On the other hand, if ε_{top} becomes much smaller than $35(\varepsilon^A_{l-1})^3$, it blows up the size of the logical cell. Therefore, for optimal operational resources, both contributions should be comparable. Then, in the large size limit, $\ln \varepsilon^A_l = 3 \ln \varepsilon^A_{l-1}$, $\lambda_l = 3\lambda_{l-1}$, $d_l = 3d_{l-1}$. Further, the success probabilities $p^{A,Y}_l$ for ancilla distillation quickly approach unity with increasing distillation level l. Therefore, in the large size limit, for the point of minimal operational resources, the recursion relations Eq. (3.8) can be replaced by

$$\left(\begin{array}{c} O_3^A \\ O_3^Y \\ \lambda^3 \\ d \\ \ln\varepsilon^A \\ \ln\varepsilon^Y \end{array}\right)_l = \begin{pmatrix} 15 & 0 & 12V_A & & & \\ 0 & 7 & 12V_Y & & & \\ 0 & 0 & 27 & & & \\ & & & 3 & & \\ & & & & 3 & \\ & & & & & 3 \end{pmatrix} \left(\begin{array}{c} O_3^A \\ O_3^Y \\ \lambda^3 \\ d \\ \ln\varepsilon^A \\ \ln\varepsilon^Y \end{array}\right)_{l-1} . \tag{3.13}$$

Thus, $O^{A,Y}_{3,l} \sim 27^l$, $\ln \varepsilon^{A,Y}_l \sim 3^l$. Then, with $\varepsilon \sim 1/N_G$ (Eq. (3.11)),

$$O_3^{A,Y} \sim \ln^3 N_G. \tag{3.14}$$

Note that the distillation operations, for the case of perfect Clifford gates, are associated with the more favorable scaling exponents $\log_3 15 \approx 2.46$ and $\log_3 7 \approx 1.77$, respectively. However, in our case the topological error protection of Clifford gates must keep step with the rapidly decreasing error of state distillation, by adjusting the scale factor λ. This leads to a scaling exponent of 3 for the Clifford operational resources (c.f. Eq. (3.12)), which dominates the resource scaling of the entire state distillation procedure, yielding Eq. (3.14).

Thus, we have poly-logarithmic scaling of the overhead with circuit size

$$O_3 \sim \ln^3 N_G. \tag{3.15}$$

3.2.3.2 Behavior Below Threshold

In the previous discussion, we saw that the large N_G scaling of the overhead is very good. However, the co-efficients are quite large and this leads to very high actual overheads near the threshold. For example, see Figure 3.3 for overhead numbers at an error rate of $p_{th}/3$.

Since the threshold Eq. (3.4) of the scheme is very high, it may make sense to run the computation at an error rate significantly below the threshold, thereby saving on overhead. In order to study the behavior of the overhead below the threshold, we need to find the behavior of κ as a function of p by numerical simulation on finite sized lattices (See Figure 3.2). The best fit for κ is

$$\kappa = (-0.85 \pm 0.02) \ln p + (-4.1 \pm 0.15). \tag{3.16}$$

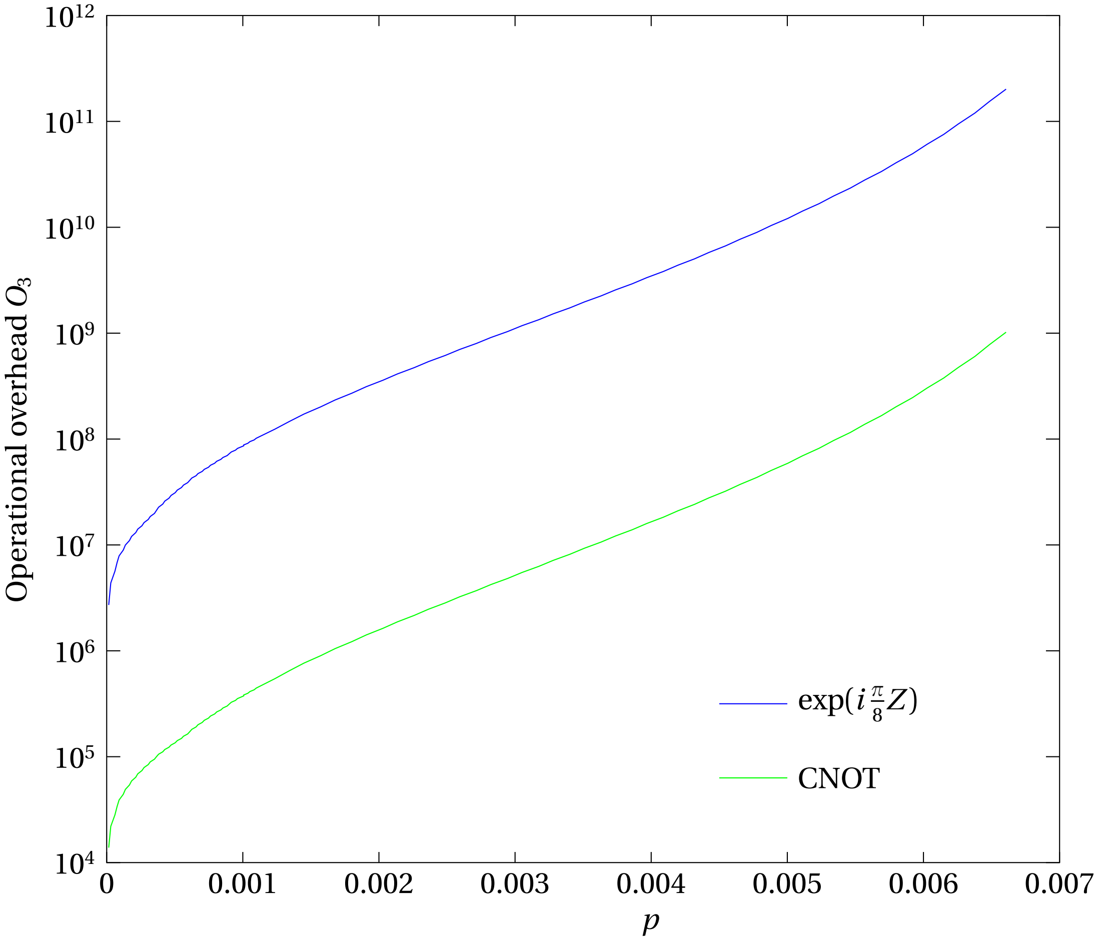


Figure 3.4 – The behavior of the operational overhead below threshold. The overhead decreases by four orders of magnitude at $\frac{p_{th}}{10}$.

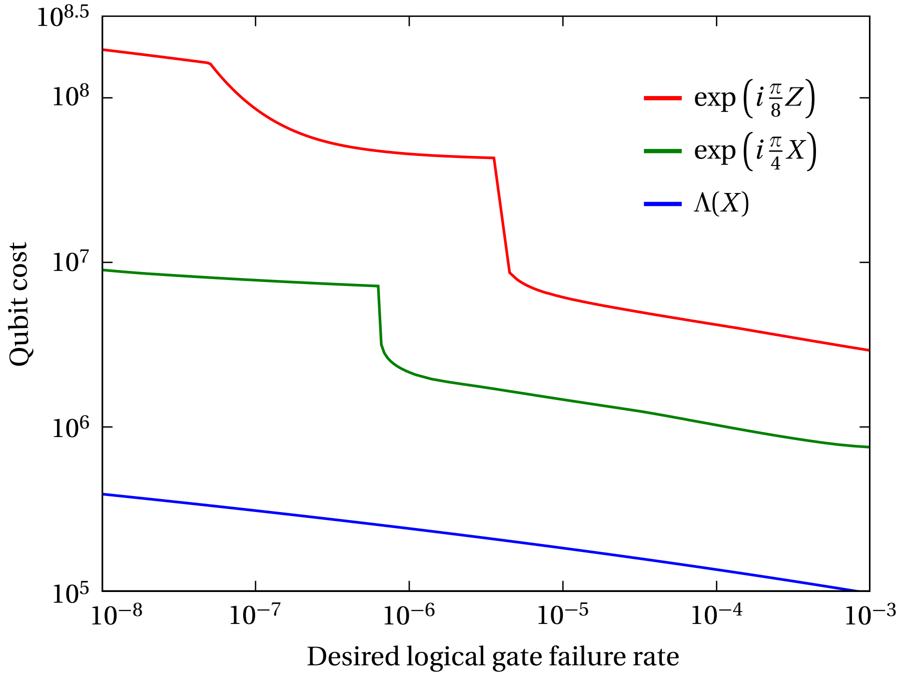


Figure 3.5 – Qubit cost to achieve a desired logical gate failure rate. Note that there are 6 qubits per elementary cell of $\mathscr{L}$. The kinks in the curves correspond to increasing levels of distillation l.

Using this result, we can estimate the behavior of the overhead below the threshold, as shown in Figure 3.4. The overhead is reduced by four orders of magnitude at $p_{th}/10$. Comparing this equation to Eq. (3.2), we see that for the non-local error model, chains of weight greater than the minimum weight also play a role in the threshold.

3.2.3.3 Overhead As A Function of Logical Gate Quality

Another interesting way to analyze the overhead is to calculate the overhead as a function of the desired logical gate failure rate. This is shown in Figure 3.5. Note that the y-axis shows the qubit cost rather than the operational overhead. There are 6 qubits per elementary cell (each face qubit is shared by two cells and each edge qubit by four) of $\mathscr{L}$, thus the qubit cost is half the operational overhead. We see that the rise in overhead is slow for increasing logical gate quality, except when an additional level of distillation is required.

3.3 Numerically Estimating the Threshold

The error model described in Section 3.1, including gate error for one and two qubit gates, preparation and measurement, effectively results in Z-errors on individual edges and correlated Z errors on two edges of $\mathscr{L}_p$ and $\mathscr{L}_d$. The location of correlated errors is shown in Figure 3.6.

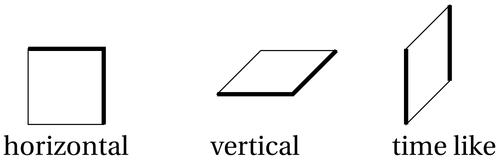


Figure 3.6 – The correlated errors (same for $\mathscr{L}_p$ and $\mathscr{L}_d$). Shown are horizontal, vertical and time like faces.

The effective error channels, after mapping all errors to Z chains (see Aside 2.2) can be derived easily. Every term in the error channel maps to a different Z chain and by using the graph stabilizer operators, they can be reduced to three independent effective error channels.

1. Time like edges

$$\mathscr{T}_{1,t} = \left(\left(1-\frac{8}{15}p_2\right)[I_t]+\frac{8}{15}p_2[Z_t]\right)^{\circ 2} \circ \left(\left(1-\frac{2}{3}p_P\right)[I_t]+\frac{2}{3}p_P[Z_t]\right) \circ \left(\left(1-\frac{2}{3}p_M\right)[I_t]+\frac{2}{3}p_M[Z_t]\right). \tag{3.17}$$

2. Space like edges (horizontal and vertical)

$$\mathscr{T}_{1,s} = \left(\left(1-\frac{8}{15}p_2\right)[I_s]+\frac{8}{15}p_2[Z_s]\right)^{\circ 3} \circ \left(\left(1-\frac{2}{3}p_1\right)[I_s]+\frac{2}{3}p_1[Z_s]\right)^{\circ 2}. \tag{3.18}$$

3. Correlated errors (see Figure 3.6)

$$\mathscr{T}_2 = \left(\left(1-\frac{8}{15}p_2\right)[I_{ab}]+\frac{8}{15}p_2[Z_{ab}]\right). \tag{3.19}$$

The threshold is estimated by performing a Monte-Carlo simulation on finite sized lattices of different size l. A size l lattice has $l \times l \times l$ elementary cells. We impose periodic boundary conditions on the lattice. The algorithm is as follows:

1. Since error correction proceeds independently on each sub-lattice, we can restrict attention to $\mathscr{L}_p$, without loss of generality.
2. Flip the qubits at every location in $\mathscr{L}_p$ in accordance with the above error model.
3. Now calculate the syndrome at the vertices of the lattice. Pair up the syndrome using minimum weight chain matching.
4. Flip the qubits along the chains identified in the previous step.
5. Consider the three boundary faces of the lattice in the x, y and t directions. If the number of error chains intersecting any of these faces is odd, the error correction

has failed. A chain intersects a given boundary face if the qubit on an edge perpendicular to the face is flipped. To count the number of intersections, we simply count the number of such flipped qubits.

This algorithm is repeated a large number of times ($\sim 10^5$) for a given lattice size l and error rate p to get an estimate of the failure rate for topological error correction $\varepsilon_{top}(l,p)$. The threshold is an intensive property. Plotting the failure curves for different lattice sizes (see Figure 3.1), the threshold is given by the point of intersection of the curves.

It is important to note that the weights used in the minimum weight matching algorithm depend on the error model. Since our error model is different for the time like and space like edges, different weights are used for them. In addition, the presence of correlated errors is handled by adding diagonal edges to the lattice.

3.3.1 Raising the Threshold

The above algorithm can be improved by using more intelligence when assigning weights to the edges of the lattice. While it is true that error correction on $\mathscr{L}_p$ and $\mathscr{L}_d$ proceed independently, error chains on the two sub-lattices are correlated. This is illustrated in Figure 3.7. The amount of correlation depends on the error model.

This correlation can be used to adjust the weights that are input into the minimum weight matching algorithm. First the above algorithm is performed on $\mathscr{L}_p$ and $\mathscr{L}_d$. Then the identified primal chains are used to update the weights of the corresponding dual edges and vice versa. The chain matching is then redone with the updated weights. This process can be concatenated infinitely, but the relative improvement in threshold after each round falls off very quickly.

The actual improvement in threshold that this procedure achieves is highly dependent on the error model. For the strongly correlated error model of a ZZ error following every ideal two-qubit gate, the improvement for a single round of concatenation is $\sim 25\%$. For the error model used above the improvement for a single round is $\sim 5\%$. The threshold result of 7.8×10^{-3} in this work is obtained by using the improved algorithm at one level of concatenation.

3.4 Summary

We have seen that $\mathfrak{C}_{3D}$ has a very high threshold of 7.8×10^{-3} and overhead requirements that scale poly-logarithmically with the size of the ideal circuit. The high threshold comes from the fact that topological error correction is very good at correcting local and near local errors, since the information is encoded in “global” structures. It’s also useful to remember that the overhead falls off very rapidly below the threshold, so when designing a physical implementation, it may make sense to trade some threshold for overhead.

The overhead requirements for the non-Clifford gates are much higher than for the $\Lambda(X)$ gate. This points to a direction for future work — using a topological measurement

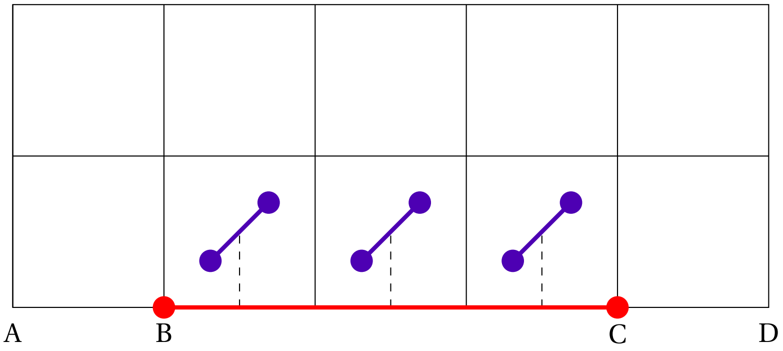


Figure 3.7 – Correlated primal and dual error chains on a lattice with periodic boundary conditions. These error chains can be caused by three Y errors on the edges between B and C. There are two bits of primal syndrome located at B and C. With uniformly weighted edges, the minimum weight matching algorithm would incorrectly identify the chain as B – A – D – C, leading to a logical error. However, there are 6 bits of syndrome available in the dual lattice and these can be used to correctly identify this error.

based scheme on a lattice that supports non-Abelian anyons. The lattice $\mathscr{L}$ supports only Abelian anyons and thus needs the non-Clifford ancillas to perform universal computation. A lattice supporting non-Abelian anyons would be able to perform these operations by simple braiding and therefore the whole operation would be topologically protected. A few such lattices are described in [LW05].

Chapter 4

Purification of Large Bi-colorable Graph States

The known protocols in quantum information processing require a certain degree of quantum-mechanical entanglement to achieve an advantage over their classical counterparts. Often, this quantum-mechanical "essence" is provided in terms of in-advance-prepared quantum states. For example, Bell states are used in a well-known protocol for quantum cryptography [Eke91], and schemes for multiparty cryptographic tasks using Greenberger-Horne-Zeilinger (GHZ) states and other Calderbank-Shor-Steane (CSS) states have been devised [CL04]. Further, in quantum computation, multiparticle entangled states can be used to streamline the execution of gates and subcircuits via gate teleportation [GC99], and cluster states represent a universal resource for quantum computation, as we saw in the previous chapters.

In most realistic scenarios the quality of entangled resource states is degraded by the effects of decoherence and methods of error detection or correction are required to counteract this process. One such method is state purification where a (close to) perfect copy of a quantum state is distilled out of many imperfect ones. Purification was first described for Bell states [Ben+96b; Ben+96a; LC99] and subsequently generalized to bi-colorable graph states and CSS states [DAB03; ADB05; HDM05]. Recently, a protocol for the purification of W states was presented in [MB05]. State purification is used, for example, to establish a perfect quantum channel between two parties [Ben+96b], to efficiently create long-range entanglement via quantum repeaters [Dür+99] or to render certain schemes for topological fault-tolerant quantum computation universal [Bra05].

Imperfect initial states are not the only sources of error for realistic state purification. With the exception of certain schemes of topological quantum computation such as [Bra05], errors in the gates for purification also need to be taken into account.

What can we expect to gain from an imperfect purification procedure? In the process of purification the errors of the initial state are replaced by the errors of the purifying gates. Thus, the amount of error may be reduced if the quality of the initial states is low compared to the quality of the gates for purification (but above threshold). Further, purification can be used to *condition* the error of a quantum state. For example, imperfect Bell-state purification can be used to establish a perfectly private if imperfect quantum channel [AB02]. In a multiparty scenario, for some protocols the purification gates act locally on each copy of the state to purify, resulting in a local or close to local error model for the final state. This feature attains relevance in the context of fault-tolerant quantum computation. Threshold theorems have been established for increasingly general types of error including coherent and long-range errors [TB05b; AGP06] but there are realistic scenarios in which standard error correction appears to fail [KF05]. In such a situation, state purification may be used to turn the error model into a more benign one.

The focus of this chapter is purification of bi-colorable graph states by imperfect means, a subject that has previously been studied in [ADB05; Dür+05; Kru+05]. We are interested in the interplay between threshold and overhead. Specifically, we seek protocols that, (I) work with erroneous purification gates, (II) have a high threshold and good quality of the output state, (III) scale efficiently, and (IV) are analytically tractable.

Hashing [CL04; Ben+96a; HDM05] protocols have a high threshold in the error of the initial state and require only a minimal resource overhead, but they break down as soon as the purification gates become slightly imperfect [1]. Recursive protocols such as [DAB03] also have a high threshold for error in the initial states and furthermore work with imperfect purification gates, but they are exponentially inefficient in the number of particles.

Our protocols are resistant to initial as well as purification errors and are computationally efficient. As a bonus, our protocols are analytically tractable for a wide class of errors. Specifically, our base protocol described in Section 4.2 can be analyzed for arbitrary input states and general probabilistic Pauli errors in the purification gates. This fact arises through a special locality property. So far, the exponential increase of parameters in the description of n-particle mixed states—even mixed stabilizer states—has been found to be an obstacle to analytic discussion, and only severely restricted error models have been treated in the literature.

In Section 4.1 we briefly review the protocol [DAB03] for purification of bi-colorable graph states. In Sections 4.2 and 4.3.3 we describe our purification protocols and characterize them in terms of purification threshold, output quality, and overhead. We conclude with a discussion of our results in Section 4.4.

4.1 Brief Review

Consider a graph $G(V,E)$ with vertex set V and edge set E. $G(V,E)$ is bi-colorable if V can be partitioned into two disjoint subsets A and B such that every edge in E connects a vertex in A with a vertex in B. E defines a *neighborhood* relation on elements of V; $\mathcal{N}(j) := \{i \in V : (i\, j) \in E\}$. Define the correlation operators

$$K_j := X_j \prod_{i \in \mathcal{N}(j)} Z_i \tag{4.1}$$

[1] For hashing, all N copies are included from the beginning. Each qubit of the state copies which are later measured is acted upon by a large number of noisy CNOT-gates. The error-correction procedure is applied only after the CNOTs have acted, such that their errors accumulate. Thus in the large N limit no matter how small the gate noise, the output state will be severely affected and the protocol will fail.

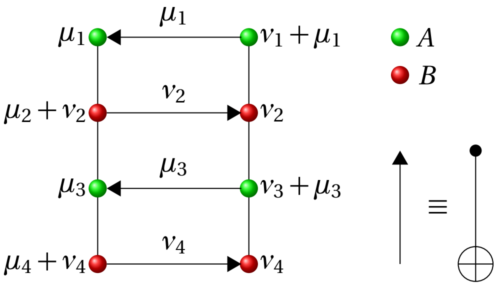


Figure 4.1 – Action of MCNOT in the graph basis. The arrows represent the direction of syndrome (or Z error) flow (i.e., the action of the MCNOT on the stabilizer)

where X, Y, and Z are the Pauli matrices. A graph state is a $|V|$-qubit state $|\mu\rangle$ ($\mu \in \{0,1\}^{|V|}$) that satisfies the eigenvalue equations

$$K_j\,|\mu\rangle = (-1)^{\mu_j}\,|\mu\rangle\,, \forall j = 1,\ldots,|V|. \tag{4.2}$$

The states $\{|\mu\rangle\}$ form a basis of the Hilbert space of $|V|$-qubit states called the *graph basis.*

We now briefly discuss the post-selection protocol of [DAB03]. The protocol works by taking two identical copies of a bi-colorable graph state and performing multiple CNOTs (MCNOT) between them, in a definite pattern as illustrated in Figure 4.1. Relabeling states in the graph basis to reflect the partition into colors A and B (i.e., $|\mu\rangle \equiv |\mu_A, \mu_B\rangle$), the effect of the MCNOT is [DAB03]

$$|\mu_A,\mu_B\rangle\,|\nu_A,\nu_B\rangle \mapsto |\mu_A,\mu_B+\nu_B\rangle\,|\nu_A+\mu_A,\nu_B\rangle\,, \tag{4.3}$$

where $+$ is elementwise addition modulo 2. Notice that information about μ_A has been copied into state 2 and information about ν_B has been copied into state 1. We then measure the local observables X and Z on copy 2, and reconstruct from the measurement outcomes the eigenvalues of all K_j with $j \in A$. Suppose we get -1 at the kth qubit. Then we know that either μ_k or ν_k was 1, but we do not have enough information to decide which one, so we throw away the states and start again. We keep doing this until all measurements are clear. By this procedure we correct, to lowest order, errors in the qubits of color A. In the next round we interchange the roles of colors A and B and so purify the B qubits. We can concatenate this procedure to achieve desired levels of purity. Because we are post-selecting states on the basis of a global measurement outcome, this protocol is inefficient for large states. This inefficiency can be addressed by using error correction instead of post-selection, to which we now turn.

4.2 Three-copy Protocol

The simplest way to get enough information to perform error correction is to do the MCNOT on three copies instead of two. The three-copy protocol consists of two subprotocols. We use three identical copies of the state in each subprotocol. The output of the first subprotocol is used as input for the next. Thus, we need nine copies to run a single round. Let the three identical copies be $\rho^{(0)}$, $\rho^{(1)}$, and $\rho^{(2)}$. Subprotocol 1 ($P1$):

i. Partition the graph into two colors A and B ($V = V_A \cup V_B$ and $V_A \cap V_B = \emptyset$).

ii. Perform the MCNOT between copies $\rho^{(0)}$ and $\rho^{(1)}$ and $\rho^{(0)}$ and $\rho^{(2)}$ such that information about qubits of color A flows from $\rho^{(0)} \to \rho^{(1)}$ and $\rho^{(0)} \to \rho^{(2)}$. As a side effect information about B will flow from $\rho^{(1)}, \rho^{(2)} \to \rho^{(0)}$. See Figure 4.3(a), below.

iii. Measure qubits of color A in the X basis and qubits of color B in the Z basis in states $\rho^{(1)}$ and $\rho^{(2)}$. This is a measurement of K_j for $j \in A$. If the measurement of K_j gives $+1$ (-1) we get a syndrome of 0 (1). Thus, for each $j \in A$ we have two bits of syndrome $\sigma_j^{(1)}$ and $\sigma_j^{(2)}$.

iv. Apply the correction $\prod_{j\in A} Z_j^{\sigma_j^{(1)} \cdot \sigma_j^{(2)}}$ to $\rho^{(0)}$.

For subprotocol $P2$ the roles colors A and B are interchanged.

First, we will analyze this protocol with ideal CNOT gates. This will allow us to derive simple closed-form recursion relations characterizing the behavior of the protocol, as well as analytical estimates of the threshold and efficiency. In Section 4.2.2 we generalize to noisy gates. The analysis is restricted to density matrices that are diagonal in the graph basis (i.e., probabilistic mixtures of graph states). At the end of Section 4.2.2, we will show that our results are valid for arbitrary density matrices.

4.2.1 Ideal Gates

Equation 4.3 implies that the effect of the MCNOT on $\rho^{(0)}$, $\rho^{(1)}$, and $\rho^{(2)}$ is

$$
\begin{aligned}
&|\mu_A^{(0)}, \mu_B^{(0)}\rangle \mapsto |\mu_A^{(0)}, \mu_B^{(0)} + \mu_B^{(1)} + \mu_B^{(2)}\rangle && (4.4)\\
&|\mu_A^{(1)}, \mu_B^{(1)}\rangle \mapsto |\mu_A^{(1)} + \mu_A^{(0)}, \mu_B^{(1)}\rangle \\
&|\mu_A^{(2)}, \mu_B^{(2)}\rangle \mapsto |\mu_A^{(2)} + \mu_A^{(0)}, \mu_B^{(2)}\rangle\,. && (4.5)
\end{aligned}
$$

Equation 4.2 implies that the effect of the correction is

$$
|\mu_A^{(0)}, \mu_B^{(0)}\rangle \mapsto |\mu_A^{(0)} + \sigma, \mu_B^{(0)}\rangle\,, \tag{4.6}
$$

where $\sigma_j := \sigma_j^{(1)} \cdot \sigma_j^{(2)}$. By measuring $\rho^{(1)}$ and $\rho^{(2)}$, we get two bits of syndrome for each qubit of color A in $\rho^{(0)}$. The syndrome is conclusive; it allows us to identify, to lowest order in the error probability on which state the error occurred. We can thus do error correction instead of post-selection. This will make the protocol scale efficiently in the size of the states. The price is a reduction of the threshold value.

We now derive a recursion relation for the expectation values $\langle K_j\rangle$, $j \in 1,\ldots,N$. They yield a necessary and sufficient condition for purification. For the moment we assume that the initial state ρ is diagonal in the graph basis–i.e., ρ is a probabilistic mixture. It is then safe to consider error probabilities. This assumption is not necessary, however.

It is removed in Section 4.2.2. Define $P_j(\rho)$ as the probability to find the eigenvalue -1 in the measurement of K_j on ρ as

$$P_j(\rho) := \mathrm{Tr}\left[\frac{1-K_j}{2}\rho\right] = \frac{1-\langle K_j\rangle}{2}. \tag{4.7}$$

Consider subprotocol $P1$. In order to analyze this protocol we make use of the fact that the error correction operation is local. It only uses information about $\langle K_j\rangle$ in each copy to apply a correction to the jth qubit in $\rho^{(0)}$. Thus, $\langle K_j\rangle$ should have nice decoupled recursion relations. We will later derive the recursion relations for the expectation value of arbitrary stabilizer elements, which in general are more complex.

First consider qubits of color B. From Eq. (4.4), $\mu_j^{(0)} \mapsto \mu_j^{(0)}+\mu_j^{(1)}+\mu_j^{(2)}$. Since our copies are identical, we have $P_j(\rho^{(0)}) = P_j(\rho^{(1)}) = P_j(\rho^{(2)}) = P_j$. Then, $P_j \mapsto P_j^3 + 3P_j(1-P_j)^2$. In terms of expectation values,

$$\langle K_j\rangle' = \langle K_j\rangle^3. \tag{4.8}$$

Under concatenation of $P1$ with itself, qubits of color B are *polluted* with $\langle K_j\rangle_{\rho^{(0)}} \longrightarrow \langle K_j\rangle_I = 0$.

Turning our attention to qubits of color A we note that error correction fails if $\mu_j = 1$ for more than one copy. Thus, $P_j \mapsto P_j^3 + 3P_j^2(1-P_j)$. In terms of expectation values

$$\langle K_j\rangle' = \frac{1}{2}\left(3 - \langle K_j\rangle^2\right)\langle K_j\rangle. \tag{4.9}$$

Under concatenation of $P1$ with itself, qubits of color A are *purified* with $\langle K_j\rangle_{\rho^{(0)}} \longrightarrow \langle K_j\rangle_{|\mathbf{0}\rangle\langle\mathbf{0}|} = 1$.

Subprotocol $P2$ is identical to $P1$ except that the roles of A and B are interchanged and the three copies are the output states from running $P1$ 3 times. The three-copy protocol is the composition of $P2$ with $P1$. Let $P = P2 \circ P1$; then Eqs. (4.8) and (4.9) imply that under the action of P

$$\langle K_j\rangle' = \begin{cases} \frac{1}{8}\left(3-\langle K_j\rangle^2\right)^3 \langle K_j\rangle^3 & \text{if } j \in A, \\ \frac{1}{2}\left(3-\langle K_j\rangle^6\right)\langle K_j\rangle^3 & \text{if } j \in B. \end{cases} \tag{4.10}$$

The recursion relations (4.10) have, for each color, a unique repulsive fixed point in the interval $(0,1)$ which separates the basins of attraction for the trivial fixed point at 0 and the nontrivial fixed point at 1 (See Figure 4.2). The upper fixed point corresponds to the perfect graph state. Thus, the stated protocol purifies a graph state if and only if

$$\begin{aligned} &\langle K_j\rangle > 0.7297 \text{ for all } j \text{ in } A \\ &\langle K_j\rangle > 0.9003 \text{ for all } j \text{ in } B. \end{aligned} \tag{4.11}$$

We can compare these thresholds to the thresholds for the post-selection protocol of [DAB03]. For this protocol, it is not known how to derive a threshold for general noise or even probabilistic Pauli noise. However, for the particular case where only indepen-

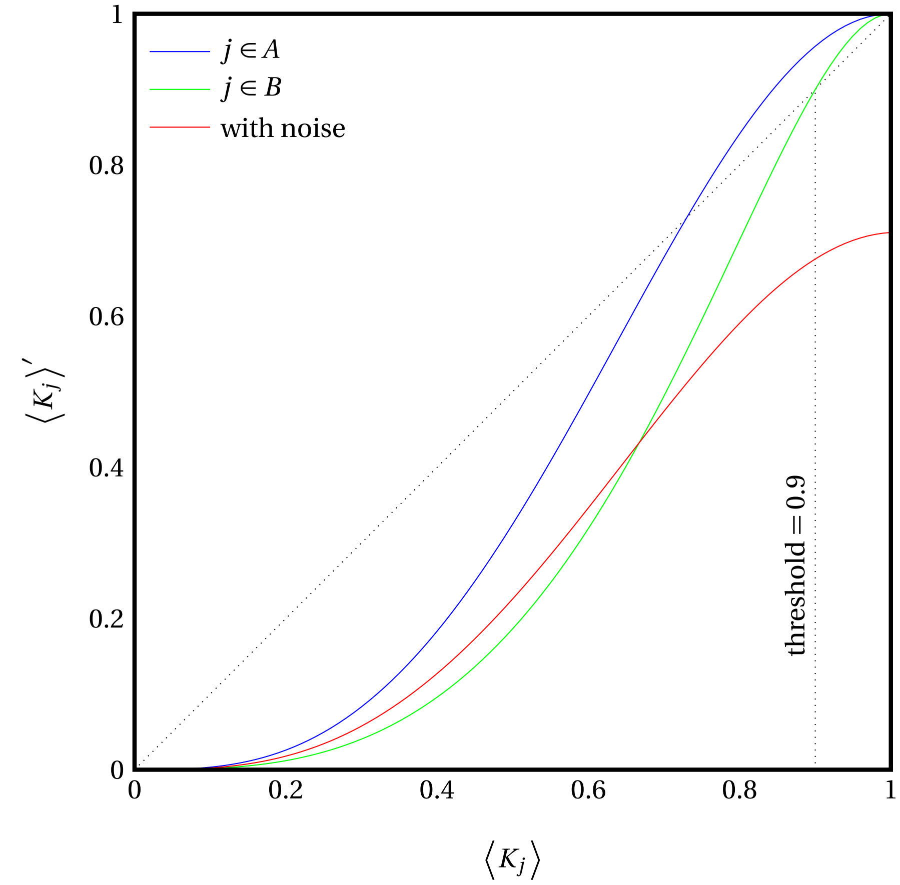


Figure 4.2 – Recurrence curves for the three-copy protocol. These simple curves fully encapsulate the behavior of the protocol with ideal gates. The point of intersection with $\langle K_j \rangle' = \langle K_j \rangle$ gives the threshold. If the gates are too noisy, the protocol breaks down, as indicated by the lowest curve.

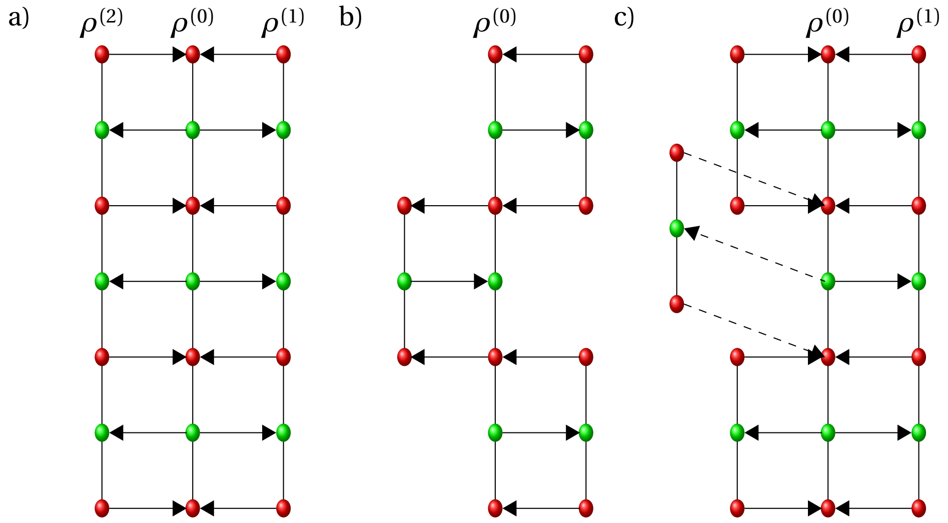


Figure 4.3 – The MCNOT for subprotocol P_1 in (a) The 3-copy protocol, (b) The band-aid protocol and (c) The conditional bandaid protocol. The dotted lines in c) indicate that the bandaids are applied only if there is an ambiguous syndrome at that location. Here we show graphs of degree 2, but these protocols can be applied to graphs of any degree.

dent local phase flip errors are assumed for the initial states, recursion relations can be derived even for post-selection. Then, the P1 (post-selection) recursion relation for $\langle K_j \rangle$ with $j \in B$ is $\langle K_j \rangle' = \langle K_j \rangle^2$ and for $j \in A$ is $\langle K_j \rangle' = \frac{2\langle K_j \rangle}{1+\langle K_j \rangle^2}$. The resulting threshold values are $\langle K_j \rangle_{\text{th}} = 0.2956$ for $j \in A$ and $\langle K_j \rangle_{\text{th}} = 0.5437$ for $j \in B$.

Returning to our protocol, it is possible to derive recursion relations for the expectation values of arbitrary stabilizer elements. They are not in general decoupled, but there is still a notion of locality. The generalized relation allows us to compute the recursion relations for stabilizers with small support efficiently. Define

$$K_{\mathbf{a},\mathbf{b}} := \prod_{i=1}^{|V_A|} K_i^{a_i} \prod_{j=1}^{|V_B|} K_j^{b_j}, \tag{4.12}$$

where $\mathbf{a} \in \{0,1\}^{|V_A|}$ and $\mathbf{b} \in \{0,1\}^{|V_B|}$. The factors in the first product are the stabilizer generators for qubits of color A, while those in the second product are for qubits of color B. Then (see Section 4.5.1) under the action of subprotocol $P1$,

$$\langle K_{\mathbf{a},\mathbf{b}} \rangle' = \frac{1}{2^{|\mathbf{a}|}} \sum_{\mathbf{a}_1,\mathbf{a}_2 \ll \mathbf{a}} (-1)^{\mathbf{a}_1 \cdot \mathbf{a}_2} \langle K_{\mathbf{a}+\mathbf{a}_1+\mathbf{a}_2,\mathbf{b}} \rangle \langle K_{\mathbf{a}_1,\mathbf{b}} \rangle \langle K_{\mathbf{a}_2,\mathbf{b}} \rangle, \tag{4.13}$$

where $\mathbf{f} \ll \mathbf{g}$ iff $f_j = 0$ whenever $g_j = 0$. Equations (4.9) and (4.8) are special cases for $\langle K_{\mathbf{a},\mathbf{b}} \rangle = \langle K_j \rangle$ with $j \in A, B$ respectively. An interesting feature of this equation is that it relates a correlator of weight $w = |\mathbf{a}| + |\mathbf{b}|$ to correlators of weight no more than w. This makes it feasible to calculate the recursion relations for correlators of small weight.

In order to discuss the behavior of this protocol under concatenation with itself, it is useful to switch back to probability variables. Then Eq. (4.10) implies that if the protocol is concatenated with itself k times,

$$P_j(\rho(k)) \leq \left(\frac{P_j(\rho(0))}{P_{\mathrm{th}}}\right)^{2^k} \tag{4.14}$$

where P_{th} is the threshold error probability. The k-concatenated protocol requires 3^{2k} identical copies, thus, the protocol is exponentially efficient under concatenation. The reduction of error, Eq. (4.14), is not conditioned on a particular post-selected syndrome. The overhead in number of required initial states is independent of the size N of the graph state. We conclude that under concatenation the protocol reaches the reference state $|\mathbf{0}\rangle$ with efficient use of resources. Contrarily, for the post-selection protocol [DAB03] the overhead acquires a dependence $\exp(\alpha N)$, with some $\alpha > 0$, due to post-selection of a particular syndrome.

4.2.2 Noisy Gates

Now we investigate what happens to this protocol when the CNOT gates themselves are noisy. In the three-copy protocol CNOT gates act on the same qubit in two states $\rho^{(m)}$ and $\rho^{(n)}$. We model a noisy two-qubit gate as an ideal gate followed by the two-qubit depolarizing channel [i.e., the $SU(4)$-invariant channel]

$$\mathrm{T}^{(k)} := (1-p_2)[I] + \frac{p_2}{16}\sum_{i,j=1}^{4}\left[D_i^{(k,m)} \otimes D_j^{(k,n)}\right], \tag{4.15}$$

where $D_{i,j} \in \{I, X, Y, Z\}$ and k is the qubit index. $D^{(k,m)}$ acts on the kth qubit of $\rho^{(m)}$. The Z gates applied in the error-correction steps and the measurement of the syndrome are assumed to be noiseless. This is natural since the Pauli phase flips Z may be omitted as physical operations and instead accounted for in the classical syndrome processing. We will include the effect of measurement errors in the analysis when we consider the more sophisticated protocols, which have higher thresholds than the three-copy protocol. If we consider the effect of $\mathrm{T}^{(k)}$ only on $\langle K_j\rangle$ in state $\rho^{(0)}$, then using Eq. (4.2) we can reduce the noise to an effective error. For every $k \in V : k \in \mathcal{N}(j) \cup \{j\}$

$$\mathrm{T}_{\mathrm{eff}}^{(k,j)}(\rho^{(0)}) = \left(1 - \frac{p_2}{2}\right)[I] + \frac{p_2}{2}[Z_j^{(0)}]. \tag{4.16}$$

If $k \notin \mathcal{N}(j) \cup \{j\}$, then $\mathrm{T}_{\mathrm{eff}}^{(k,j)}$ is just the identity map. Since every error channel commutes with every CNOT, we can model the noisy MCNOT as the ideal MCNOT followed by $|V|$ noise channels.

The error channel Eq. (4.15) is local (i.e., it acts only on qubit k in $\rho^{(m)}$ and $\rho^{(n)}$). Also the error operators are Pauli operators, which map graph states to graph states, keeping ρ diagonal in the graph basis. Thus we can expect the noisy recursion relations to have the same form as Eq. (4.10). Considering only subprotocol $P1$, the jth qubit in $\rho^{(0)}$ is affected by $2(d+1)$ error channels. For simplicity, we assume all vertices of the graph have the same degree d. If this is not the case, then there would be a different set of recursion relations for each degree. We can then choose d to be the maximum degree, in which case the recursion relations will be lower bounds for all other degrees. The total probability that the jth qubit is flipped by an error is $\frac{1-(1-p_2)^{2(d+1)}}{2}$. Thus, for qubits of color B,

$$\langle K_j \rangle' = \alpha^2 \langle K_j \rangle^3 , \tag{4.17}$$

where $\alpha = (1-p_2)^{(d+1)}$.

The situation is a little more complex for qubits of color A as the error in the MCNOT between $\rho^{(0)}$ and $\rho^{(1)}$ is propagated by the MCNOT between $\rho^{(0)}$ and $\rho^{(2)}$ (see Figure 4.3(a)). However, the form of the recursion relation remains the same. We get

$$\langle K_j \rangle' = \frac{\alpha^2}{2} \left(2 + \alpha^{-1} - \langle K_j \rangle^2 \right) \langle K_j \rangle . \tag{4.18}$$

For a derivation see Section 4.5.1.2. Composing subprotocols $P1$ and $P2$ we get the recursion relations for the three-copy protocol with noisy gates

$$\langle K_j \rangle' = \begin{cases} \frac{\alpha^8}{8} \left(2 + \alpha^{-1} - \langle K_j \rangle^2 \right)^3 \langle K_j \rangle^3 & \text{if } j \in A, \\ \frac{\alpha^4}{2} \left(2 + \alpha^{-1} - \alpha^4 \langle K_j \rangle^6 \right) \langle K_j \rangle^3 & \text{if } j \in B, \end{cases} \tag{4.19}$$

Here, qubits of color A behave worse. Solving the recursion relations for fixed points, we find that there are two non-trivial positive fixed points (see Section 4.5.2) for $\alpha > 0.9902$. Consider the interval $[0,1]$. It has at most three fixed points $0 = f_0 < f_1 \leq f_2 \leq 1$. f_0 and f_2 are attractive while f_1 is repulsive. Thus f_2 will be a stable fixed point for $\alpha > 0.9902$ and $\langle K_j \rangle_{\text{initial}} > f_1$. This gives a threshold for the noise affecting the gates that scales inversely proportional to the graph degree d,

$$p_{\text{th}} \approx \frac{9.8 \times 10^{-3}}{d+1} . \tag{4.20}$$

Specifically for degrees 2 and 4 we obtain

$$p_{\text{th}} = \begin{cases} 0.328\,\% \text{ for } d = 2, \\ 0.197\,\% \text{ for } d = 4. \end{cases} \tag{4.21}$$

This is a rather low value, but it will be substantially improved when we consider more sophisticated protocols.

We now show that the recursion relations Eq. (4.19) are valid regardless of whether or not the considered states are diagonal in the graph basis. To see this, let us define a depolarization operator $\mathscr{D}$ which converts an arbitrary n-qubit mixed state ρ into an n-qubit mixed state $\rho_D = \mathscr{D}\rho$ that is diagonal in the graph basis. $\mathscr{D}$ takes the form

$$\mathscr{D} = \left(\prod_{\mathbf{a}} \frac{[I] + [K_{\mathbf{a},\mathbf{0}}]}{2} \right) \left(\prod_{\mathbf{b}} \frac{[I] + [K_{\mathbf{0},\mathbf{b}}]}{2} \right) , \tag{4.22}$$

where **a** and **b** are vectors in a basis of $\{0,1\}^{|V_A|}$ and $\{0,1\}^{|V_A|}$ respectively.

We only consider $P1$, the first round of the protocol. It is associated with a transformation $P1: \rho \longrightarrow \rho' = R\left(\rho^{\otimes 3}\right)$. R and $\mathscr{D}$ commute–i.e.,

$$R\left((\mathscr{D}\rho)^{\otimes 3}\right) = \mathscr{D} \circ R\left(\rho^{\otimes 3}\right), \tag{4.23}$$

for any ρ. For a proof see Section 4.5.3.

Consider a recursion relation of the form

$$\langle K_{\mathbf{a},\mathbf{b}}(\rho'_D)\rangle = f_{\mathbf{a},\mathbf{b}}\left(\{\langle K_{\mathbf{i},\mathbf{j}}(\rho_D)\rangle\}\right), \tag{4.24}$$

with $f_{\mathbf{a},\mathbf{b}}$ some function depending on $\mathbf{a}, \mathbf{b}$ as in Eq. (4.13). Now,

$$\begin{aligned}
\langle K_{\mathbf{a},\mathbf{b}}(\rho'_D)\rangle &= \mathrm{Tr}\left[K_{\mathbf{a},\mathbf{b}} R\left((\mathscr{D}\rho)^{\otimes 3}\right)\right] \\
&= \mathrm{Tr}\left[K_{\mathbf{a},\mathbf{b}} \mathscr{D} \circ R\left(\rho^{\otimes 3}\right)\right] && \text{[by Eq. (4.23)]} \\
&= \mathrm{Tr}\left[\mathscr{D}^{\dagger}(K_{\mathbf{a},\mathbf{b}})\rho'\right] && \text{(trace cyclicity)} \\
&= \langle K_{\mathbf{a},\mathbf{b}}(\rho')\rangle. && (\mathscr{D}^{\dagger} \equiv \mathscr{D})
\end{aligned}$$

Similarly, $\langle K_{\mathbf{i},\mathbf{j}}(\rho_D)\rangle = \langle K_{\mathbf{i},\mathbf{j}}(\rho)\rangle$, such that

$$\langle K_{\mathbf{a},\mathbf{b}}(\rho')\rangle = f_{\mathbf{a},\mathbf{b}}\left(\{\langle K_{\mathbf{i},\mathbf{j}}(\rho)\rangle\}\right). \tag{4.25}$$

Thus, a recursion relation of the form of Eq. (4.24) such as Eq. (4.19) holds for all states ρ and not just for diagonal states $\rho_D = \mathscr{D}\rho$.

4.3 Improved Protocols

4.3.1 Error Model

In the following, we consider a scenario where graph states are created locally from product states, then distributed to several parties and subsequently purified. Errors occur in each of these steps–specifically, the following

- There is a two-qubit error T, Eq. (4.15), associated with each controlled-PHASE gate in the creation of the graph state, with probability p_2.
- A local depolarizing error with probability p_1 occurs on each graph state qubit during transmission.
- Every CNOT gate used in purification carries a two-qubit error, Eq. (4.15), with error probability p_2. Every measurement is modeled by a one-qubit depolarizing channel with error probability p_2 followed by a perfect measurement.

4.3.2 Bandaid Protocol

In order to raise the threshold of the three-copy protocol, we will try to combine the strategies of error correction and post-selection (which has a higher threshold). One way to do this is to use small highly purified GHZ states–i.e., bandaids, to purify the graph one vertex at a time. The usual MCNOT is performed between the bandaid and the large graph state as shown in Figure 4.3(b). This copies information about the central vertex into the bandaid which is then measured to give a syndrome. Since the bandaid is highly purified (for example, by post-selection), it does not pollute the large state much. It is important to note that the error correction is still local, and we expect the recursion relations to be decoupled as in the case of the three-copy protocol.

The bandaid protocol also has two subprotocols. The first one $P1$ is the following.

i. Partition the graph into two colors A and B ($V = V_A \cup V_B$ and $V_A \cap V_B = \varnothing$).

ii. The bandaids are placed over the large state such that each central qubit of the bandaid is over a vertex of qubit A for all qubits of color A. Perform the MCNOT as shown in Figure 4.3(b).

iii. Measure the central qubit of each bandaid in the X basis and the other qubits in the Z basis. For each bandaid multiply the measured eigenvalues. If the product is (-1) +1 then the syndrome bit σ_j is (1) 0.

iv. Apply the correction $\prod_{j\in A} Z_j^{\sigma_j}$ to the large state.

$P2$ is the same as $P1$, with the roles of colors A and B reversed.

Consider subprotocol $P1$. For qubits of color B the argument is very similar to the three-copy protocol, except that each qubit is affected by two gates from each of d bandaids. Thus,

$$\langle K_j\rangle' = (1-p_2)^{2d}\,\langle K_j\rangle\,\langle K_j\rangle_b^d\,, \tag{4.26}$$

where $\langle K_j\rangle_b$ is the constant initial purity of the bandaid.

For qubits of color A, first suppose that the CNOT gates are ideal. Then, a simple transfer of purity occurs:

$$\langle K_j\rangle' = \langle K_j\rangle_b\,. \tag{4.27}$$

If the gates are noisy, Eq. (4.27) is multiplied by a noise factor of the form $(1-p_2)^{f(d)}$ as in the case of the three-copy protocol. There is a subtlety involving the temporal ordering of the bandaids. The bandaids do not all commute with each other. There are $1+d(d-1)$ bandaids that affect qubit j. One of them is the bandaid that is used to purify the qubit. On average $k = \frac{d(d-1)}{2}$ of the rest will be applied before the purifying one. Any effect from the k prior bandaids will be erased by the purifying bandaid [see Eq. (4.27)]. The purifying bandaid has $d+1$ noisy CNOTs affecting $\langle K_j\rangle$; since the noisy MCNOT is modeled as an ideal MCNOT followed by noise, no information about the noise is propagated to the bandaid. Thus, the noise will commute with the error correction procedure. Since a measurement error that flips the central qubit of the bandaid will cause us to apply the wrong error correction operator, it can also be reduced to an effective error as given by Eq. (4.16). Thus, $f(d) = 2(d+1) + k$ and we have

$$\langle K_j \rangle' = (1-p_2)^{\frac{d(d+3)+4}{2}} \langle K_j \rangle_b . \tag{4.28}$$

Combining subprotocols $P1$ and $P2$, we get the recursion relations for the bandaid protocol with noisy gates as well as noisy measurements

$$\langle K_j \rangle' = \begin{cases} (1-p_2)^{\frac{d(d+7)+4}{2}} \langle K_j \rangle_b^{d+1} & \text{for } j \in A, \\ (1-p_2)^{\frac{d(d+3)+4}{2}} \langle K_j \rangle_b & \text{for } j \in B. \end{cases} \tag{4.29}$$

The behavior of qubits of color A is worse, and we will use their purity as the final purity of the large state.

As per our error model in Section 4.3.1, the noisy CPHASE, CNOT and measurement gates are parametrized by p_2. The noisy transmission channel is parametrized by p_1. For the final result, we need to know the quality of the bandaids. We assume that these are also created locally, then transmitted and purified. The bandaids, however, are of fixed size and may thus be purified by the post-selection protocol [ADB05] with the higher threshold. The output quality of the purified bandaids is, to leading order in p_2,

$$\langle K_j \rangle_b = 1 - (d+1)p_2, \tag{4.30}$$

such that

$$\langle K_j \rangle = 1 - \frac{1 - d(3d+11)+6}{2} p_2, \tag{4.31}$$

for small p_2 [from Eqs. (4.29) and (4.30)]. As Eq. (4.31) shows, with increasing graph degree the effect of errors in the purification process is strongly enhanced. One may therefore ask the question whether it is useful to purify at all or whether the transmitted state should be used right away. To decide this we compute $\langle K_j \rangle$ after graph-state creation and transmission,

$$\langle K_j \rangle = (1-p_2)^{\frac{d(d+1)}{2}} (1-p_1)^{d+1}. \tag{4.32}$$

See Section 4.5.4 for a derivation. We compare this expression with Eq. (4.29) and find that there is indeed a parameter region where it makes sense to purify. This region is displayed for graphs of degree $d = 4$ in Figure 4.4. It is bounded from above and right by the curve which indicates the breakdown of the bandaid purification according to the post-selection protocol [ADB05]. If we use post-selection to obtain bandaids of high purity, then the threshold of the bandaid protocol for degree d graph states equals the threshold for purification of a $d+1$-qubit GHZ state with the post-selection protocol [8]. However, the output purity of the bandaid protocol is smaller. Only above the ascending curve is it advantageous to purify.

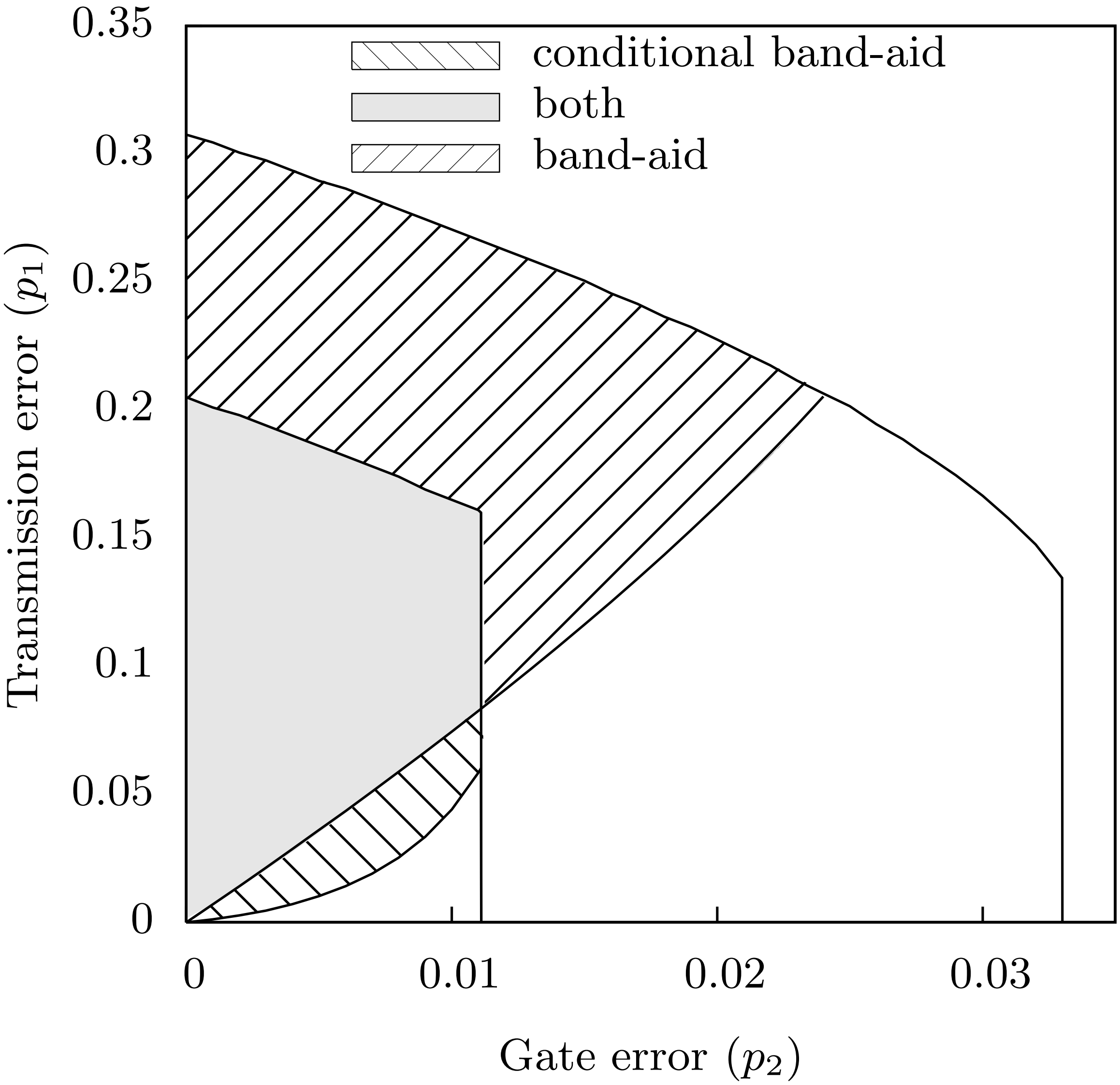


Figure 4.4 – Trade-off curves for the bandaid and conditional bandaid protocols ($d = 4$). The decreasing curves represent the breakdown of the post-selection protocol, when there is too much error. The increasing curves demarcate the region where the final purity of the purified states is higher than the purity of the unpurified states. It makes sense to purify in the shaded regions.

4.3.3 Conditional Bandaid Protocol

In order to correct the d^2 dependence of the fixed point in the bandaid protocol, we will combine it with the three-copy protocol. The hybrid protocol, called the conditional bandaid protocol, sacrifices in threshold to improve the fixed point. The fixed-point behavior, at least to linear order in gate noise, is almost as good as that of the post-selection protocol.

This protocol proceeds in the same fashion as the three-copy protocol, except that two copies are used per round, and wherever a measurement of K_j yields eigenvalue -1 (i.e., an error), a post-selected bandaid is applied to purify qubit j (see Figure 4.3). For small gate noise, we expect to have to apply only a few bandaids per round, nonetheless, the threshold is set by the qubits to which we have to apply bandaids. Locations where a measurement of K_j yields 1 are error free to lowest order. Once again, we have two subprotocols, P_1 and P_2 each purifying a different color.

The analysis is similar to that used in arriving at Eq. (4.19) for the three-copy protocol. However, the situation is complicated by the fact that the bandaids are applied conditioned on the results of measuring $\rho^{(1)}$. As a result, the recursion relations for the one point correlators are no longer completely decoupled. We can, however, find a simple lower bound on them.

Define $\langle K_b \rangle$ to be the minimum purity of the post-selected bandaid. It is a constant. For simplicity we assume that all qubits in the bandaid have this purity. As before, we assume that the graph of the large state is translationally invariant, i.e., all vertices have the same degree. The definition $\beta \equiv (1-p_2)^2 \langle K_b \rangle$ will be useful. Consider qubits of color A in subprotocol P_1, then, by a derivation similar to Eq. (4.18),

$$\langle K_j \rangle' = \frac{\alpha}{2} \left(2\alpha \langle K_j \rangle + \langle K_b \rangle - \alpha \langle K_b \rangle \langle K_j \rangle^2 \right), \tag{4.33}$$

where $\alpha = (1-p_2)^{d+1}$ as before.

So far, we have been exact. Now consider subprotocol P_2. Again focus on qubits of color A. Break P_2 down into two steps. In step one, we apply the MCNOT to $\rho^{(0)}$ and $\rho^{(1)}$. It can be readily verified that $\langle K_j \rangle \mapsto \alpha \langle K_j \rangle^2$. In step two, bandaids conditioned on the measurement outcome are applied to qubits of color B. Let $\mathbf{y} \in \{0,1\}^d$ be the measurement results for the neighbors of qubit j. A measurement result of one means a bandaid must be applied at that location. If a bandaid is applied to a neighbor of j, $\langle K_j \rangle$ is affected by the errors on the bandaid, characterized by $\langle K_b \rangle$ and by two noisy CNOTs. Thus $\langle K_j \rangle \mapsto \beta^{|\mathbf{y}|} \langle K_j \rangle$. Summing over measurement outcomes and including step one, we get

$$\langle K_j \rangle' = \alpha \left(\sum_{k=0}^{d} \left(\sum_{|\mathbf{y}|=k} q_{\mathbf{y}} \beta^k \right) \right) \langle K_j \rangle^2, \tag{4.34}$$

where $q_{\mathbf{y}}$ is the probability of measurement outcome $\mathbf{y}$. Unfortunately, $q_{\mathbf{y}}$ is a function of the general stabilizer expectation values $\langle K_{\mathbf{a},\mathbf{b}} \rangle$, so we will resort to finding a lower bound. Since $q_0 = 1 - \sum_{\mathbf{y}\neq 0} q_{\mathbf{y}}$, we can rewrite the above equation as

$$\begin{aligned}\langle K_j\rangle' &= \alpha_a\left((1-\sum_{\mathbf{y}\neq 0} q_{\mathbf{y}}) + \sum_{k>0}^{d}\sum_{|\mathbf{y}|=k} q_{\mathbf{y}}\beta^k\right)\langle K_j\rangle^2\\ &\geq \alpha\left(1-(1-\beta^d)\sum_{\mathbf{y}\neq 0} q_{\mathbf{y}}\right)\langle K_j\rangle^2,\end{aligned}$$

using $\beta \leq 1$ to arrive at the inequality.

Now, q_0 is just the probability that no error is detected on any of the neighbors of j. Let p_j be the probability of detecting an error on site j. Then, by definition, $\langle K_i\rangle = 1 - 2\sum_{y|y_i=1} q_y$. This implies that $\sum_{\mathbf{y}\neq 0} q_{\mathbf{y}} \leq \sum_{i\in\mathcal{N}(j)} \frac{1-\langle K_i\rangle}{2}$. Putting this into the above inequality,

$$\langle K_j\rangle' \geq \alpha\left(1-\frac{d}{2}(1-\beta^d)(1-\langle K_i\rangle)\right)\langle K_j\rangle^2, \tag{4.35}$$

where $\langle K_i\rangle$ is the purity of qubits of color B from the previous round.

Solving for the fixed point, we get, to leading order in gate noise p_2,

$$\langle K_j\rangle = 1 - 2(d+1)p_2. \tag{4.36}$$

Comparing this to Eq. (4.30), we see that the fixed-point scaling with degree is almost as good as in the post-selection protocol. We now apply the conditional bandaid protocol to the same situation–of a graph state being shared among widely separated parties, as for the bandaid protocol. The results for a degree four state are plotted in Figure 4.4. We see that the threshold (upper) curve is worse, whereas the fixed-point (lower) curve is better for this protocol, as compared to the bandaid protocol. The total purifiable area is smaller, indicating that it breaks down faster. In some sense, we have traded threshold for fixed point. These conclusions hold for arbitrary degree, and the curves are independent of the size of the state, making this protocol eminently suitable for the purification of large bi-colorable graph states.

4.4 Conclusion

We have described novel purification protocols for bi-colorable graph states and discussed their performance. The criteria for our protocols are that they do not breakdown in the presence of small amounts of noise in the purification process, that they have a high purification threshold and good output quality, scale efficiently, and be analytically tractable.

Our final protocol can, for relevant graph states of degree 4, tolerate 1% gate or 20% local transmission error. These are about 1/3 and 2/3 of the respective values for the post-selection protocol [ADB05; DAB03]. However, in contrast to this reference protocol, our protocol scales efficiently with the graph size.

All our protocols can be treated analytically. In particular, for the three-copy protocol we derive closed, exact one-dimensional recursion relations in the appropriate observables, irrespective of the size of the state.

We would like to comment on the influence of the graph degree for the purification threshold. First note that for the three-copy protocol of Section 4.2, in the case of perfect purification gates, the recursion relations (Eq. (4.10)) are completely independent of the graph structure, and so are the thresholds (Eq. (4.11)). This behavior changes if noise is included in the purification. The critical noise level per purification gate—at which the protocol breaks down—scales inversely proportional with the graph degree. The unfavorable dependence on the graph degree is present in all three protocols we discuss. Thus, the lesson we learn for the case of noisy purification is to beware of large graph degrees. Large graph degrees occur, for example, in graph states corresponding to codewords of concatenated CSS codes.

We would also like to comment on the structure of the nontrivial fixed point in our protocols. In the case of erroneous purification gates, the nontrivial fixed point is not completely specified by the lowest order expectation values $\langle K_j \rangle$ and it remains to be discussed which error correlations are removed by the purification protocol. As a first result in this direction, for the three-copy protocol discussed in Section 4.2 we have shown (in Section 4.5.1.3) that correlations of stabilizer expectation values located on non-overlapping supports are not introduced by the purification procedure if they are absent initially. This implies that such correlations are absent in all purified states which end up at the same fixed point as the perfect state. We show in Section 4.5.2 that the fixed point for two-generator correlations with distinct support is unique, which is enough to establish the result that all states at the fixed point obey the relation $\langle K_i K_j \rangle = \langle K_i \rangle \langle K_j \rangle$ for such correlations.

A question of further interest is whether the nontrivial fixed point of the protocol is unique at all levels of correlations. This would imply $\langle K_{\mathbf{i}+\mathbf{j}} \rangle = \langle K_{\mathbf{i}} \rangle \langle K_{\mathbf{j}} \rangle$ for all correlations with distinct supports.

Another question of further interest is whether the described or related protocols may be used to boost the threshold value for fault-tolerant quantum computation [ND05; DHN06; AL06; VBR05; RHG05] based on graph states.

4.5 Appendix

4.5.1 Generalized Recursion Relations

We now derive the generalized recursion relations [Eq. (4.13)] for the three-copy protocol. While the method used for this derivation is less intuitive, it yields recursion relations for arbitrary stabilizer elements and can handle noisy gates easily.

4.5.1.1 Noiseless Gates

In order to derive Eq. (4.13) we work in the *stabilizer basis*. Because $\rho^{(0)}$ is diagonal and the set $\{\langle K_{\mathbf{a},\mathbf{b}} \rangle\}$ where $\mathbf{a} \in \{0,1\}^{|V_A|}, \mathbf{b} \in \{0,1\}^{|V_A|}$ forms a complete set of observables, we can write an expansion $\rho^{(0)}=\frac{1}{2^{|V_A|+|V_B|}}\sum_{\mathbf{a},\mathbf{b}}\langle K_{\mathbf{a},\mathbf{b}}\rangle K_{\mathbf{a},\mathbf{b}}$.

Consider subprotocol P_1, which purifies the A subgraph. The initial state is $\rho^{(0)}\otimes\rho^{(1)}\otimes\rho^{(2)}$, which can be rewritten as a sum over $\mathbf{a},\mathbf{b}$ of terms of the form

$$\langle K_{\mathbf{a}^{(0)},\mathbf{b}^{(0)}}\rangle\langle K_{\mathbf{a}^{(1)},\mathbf{b}^{(1)}}\rangle\langle K_{\mathbf{a}^{(2)},\mathbf{b}^{(2)}}\rangle\times K_{\mathbf{a}^{(0)},\mathbf{b}^{(0)}}K_{\mathbf{a}^{(1)},\mathbf{b}^{(1)}}K_{\mathbf{a}^{(2)},\mathbf{b}^{(2)}}. \tag{4.37}$$

The protocol is linear, so we track the evolution of each term separately. Performing step (ii), this term becomes

$$\langle K_{\mathbf{a}^{(0)},\mathbf{b}^{(0)}}\rangle\langle K_{\mathbf{a}^{(1)},\mathbf{b}^{(1)}}\rangle\langle K_{\mathbf{a}^{(2)},\mathbf{b}^{(2)}}\rangle\times K_{\mathbf{a}^{(0)}+\mathbf{a}^{(1)}+\mathbf{a}^{(2)},\mathbf{b}^{(0)}}K_{\mathbf{a}^{(1)},\mathbf{b}^{(0)}+\mathbf{b}^{(1)}}K_{\mathbf{a}^{(2)},\mathbf{b}^{(0)}+\mathbf{b}^{(2)}}. \tag{4.38}$$

Now consider step (iii). Suppose we get measurement outcomes $\lambda^{(1)},\lambda^{(2)}$ for the stabilizers in subgraph A on copies $\rho^{(1)},\rho^{(2)}$. Then the resultant state is given by applying the projector

$$\frac{1}{2^{2|V_A|}}\prod_{j=1}^{|V_A|}[I]\otimes\left([I]+(-1)^{\lambda_j^{(1)}}K_j^{(1)}\right)\otimes\left([I]+(-1)^{\lambda_j^{(2)}}K_j^{(2)}\right). \tag{4.39}$$

All the single-site operators involved commute, so this term is a product of stabilizers in $\mathbf{b}$ and terms of the form

$$\left([I]+(-1)^{\lambda_j^{(k)}}K_j^{(k)}\right)\left(K_j^{(k)}\right)^{a_j^{(k)}}.$$

Here $k=1,2$. Discarding $\rho^{(1)},\rho^{(2)}$, we perform a partial trace over these systems (recalling that $K_{\mathbf{a},\mathbf{b}}$ are all traceless except $K_{\mathbf{0},\mathbf{0}}=I$). In the above term, only the coefficient of $[I]$ contributes, which is $(-1)^{\lambda_j^{(k)}a_j^{(k)}}$. Including the stabilizer operator, we are left with

$$\delta_{\mathbf{b}^{(0)},\mathbf{b}^{(1)}}\delta_{\mathbf{b}^{(1)},\mathbf{b}^{(2)}}\frac{(-1)^{\lambda^{(1)}\cdot\mathbf{a}^{(1)}+\lambda^{(2)}\cdot\mathbf{a}^{(2)}}}{2^{2|V_A|}}\times K_{\mathbf{a}^{(0)}+\mathbf{a}^{(1)}+\mathbf{a}^{(2)},\mathbf{b}^{(1)}}, \tag{4.40}$$

where $\delta_{\mathbf{p},\mathbf{q}}$ is the Kronecker delta on each component of $\mathbf{p},\mathbf{q}$. Note that we must have $\mathbf{b}^{(0)}=\mathbf{b}^{(1)}=\mathbf{b}^{(2)}$ or the term is zero.

Now examine the action of the Pauli $[Z]$ operator in this basis. $[Z]K_{\mathbf{a},\mathbf{b}}=ZK_{\mathbf{a},\mathbf{b}}Z=-1^{k}K_{\mathbf{a},\mathbf{b}}$, where $k=0$ iff Z and $K_{\mathbf{a},\mathbf{b}}$ commute. Effectively, Z is a diagonal matrix with entries ±1. Identical reasoning applies to X and Y. This will make it very easy to add gate noise into the analysis. It also allows us to say that the net effect of the error-correction step iv is to multiply Eq. (4.40) by a factor of $(-1)^{(\lambda^{(1)}\times\lambda^{(2)})\cdot(\mathbf{a}^{(0)}+\mathbf{a}^{(1)}+\mathbf{a}^{(2)})}$, where $(\mathbf{p}\times\mathbf{q})_j\equiv p_j\cdot q_j$. To simplify the notation, change the basis to $\mathbf{a}\equiv\mathbf{a}^{(0)}+\mathbf{a}^{(1)}+\mathbf{a}^{(2)}$, $\mathbf{b}\equiv\mathbf{b}^{(0)}$. Then the term becomes

$$\delta_{\mathbf{b},\mathbf{b}^{(1)}}\delta_{\mathbf{b},\mathbf{b}^{(2)}}\frac{(-1)^{\lambda^{(1)}\cdot\mathbf{a}^{(1)}+\lambda^{(2)}\cdot\mathbf{a}^{(2)}+(\lambda^{(1)}\times\lambda^{(2)})\cdot\mathbf{a}}}{2^{2|V_A|}}K_{\mathbf{a},\mathbf{b}}.$$

In this notation and ignoring the delta functions, the original coefficient in Eq. (4.38) is $\langle K_{\mathbf{a}+\mathbf{a}^{(1)}+\mathbf{a}^{(2)},\mathbf{b}}\rangle \langle K_{\mathbf{a}^{(1)},\mathbf{b}}\rangle \langle K_{\mathbf{a}^{(2)},\mathbf{b}}\rangle$. We will now get conditions under which this term contributes to the coefficient of $K_{\mathbf{a},\mathbf{b}}$.

Summing over measurement outcomes, the coefficient of $K_{\mathbf{a},\mathbf{b}}$ is

$$\langle K_{\mathbf{a}+\mathbf{a}^{(1)}+\mathbf{a}^{(2)},\mathbf{b}}\rangle \langle K_{\mathbf{a}^{(1)},\mathbf{b}}\rangle \langle K_{\mathbf{a}^{(2)},\mathbf{b}}\rangle \times \frac{1}{2^{2|V_A|}} \sum_{\lambda^{(1)},\lambda^{(2)}} (-1)^{\lambda^{(1)}\cdot\mathbf{a}^{(1)}+\lambda^{(2)}\cdot\mathbf{a}^{(2)}+(\lambda^{(1)}\times\lambda^{(2)})\cdot\mathbf{a}}.$$

The sum can be re-expressed as

$$\prod_{j=1}^{|V_A|} \sum_{\lambda_j^{(1)},\lambda_j^{(2)}=0}^{1} (-1)^{\lambda_j^{(1)}a_j^{(1)}+\lambda_j^{(2)}a_j^{(2)}+\lambda_j^{(1)}\lambda_j^{(2)}a_j}.$$

If $a_j=0$, then the jth factor is zero unless $a_j^{(1)}=a_j^{(2)}=0$, in which case it is 4. Hence, for the term $\langle K_{\mathbf{a}+\mathbf{a}^{(1)}+\mathbf{a}^{(2)},\mathbf{b}}\rangle \langle K_{\mathbf{a}^{(1)},\mathbf{b}}\rangle \langle K_{\mathbf{a}^{(2)},\mathbf{b}}\rangle$ to survive the procedure, we must have $\mathbf{a}^{(1)},\mathbf{a}^{(2)} \ll \mathbf{a}$. If this holds, then an overall factor of $4^{|V_A|-|\mathbf{a}|}$ comes out. If $a_j=1$, then a straightforward calculation shows that the jth factor contributes a factor of $2(-1)^{a_j^{(1)}a_j^{(2)}}$. The overall numerical factor is thus $\frac{1}{2^{|\mathbf{a}|}}$. To get the new value of $\langle K_{\mathbf{a},\mathbf{b}}\rangle$, we simply sum over $\mathbf{a}^{(1)},\mathbf{a}^{(2)}$ since these and only these will contribute to the support of $K_{\mathbf{a},\mathbf{b}}$ under P1. This gives Eq. (4.13).

4.5.1.2 Noisy Gates

Adding noise to the gates requires very little additional work. We can rewrite the depolarizing channel on qubit j of copy k as

$$\begin{aligned} D_j^{(k)}[\rho] &= \frac{1}{2}\left([I]+[Z]_j^{(k)}\right)\frac{1}{2}\left([I]+[X]_j^{(k)}\right)[\rho] \\ &\equiv P_Z^{(j,k)}P_X^{(j,k)}. \end{aligned}$$

It was shown above that $[X],[Z]$ have ±1 on the diagonal. Thus writing the noise channel in this form illustrates how the noise components act as projectors $P_{Z_j}^{(k)}$, $P_{X_j}^{(k)}$. If a specific ket is affected by noise on site j of copy k, it will be an eigenvector of $D_j^{(k)}$ with zero eigenvalue.

The noise from a CNOT at site j between copies i and k is

$$E_j^{(i),(k)} \equiv (1-p_2)+p_2(P_{Z_j}^{(i)}P_{X_j}^{(i)})(P_{Z_j}^{(k)}P_{X_j}^{(k)}). \tag{4.41}$$

If a ket $K_{\mathbf{a},\mathbf{b}}$ is affected by any of these noise terms (that is, if the noise anticommutes with $K_{\mathbf{a},\mathbf{b}}$), it will be projected to zero and thus acquire a $(1-p_2)$ multiplier overall.

The noise from the first MCNOT is $E_{01}\equiv\prod_j E_j^{(0),(1)}$ and from the second MCNOT is $E_{02}\equiv\prod_j E_j^{(0),(2)}$. Clearly the overall multiplier is independent of the measurement outcomes, so the analysis for Eq. (4.39) still holds. The recursion relations are then similar in structure to Eq. (4.13), except that coefficients dependent on $(1-p_2)$ are inserted before each term.

We illustrate this by calculating the recursion relations for $\langle K_j \rangle$. If $j \in B$, there is no sum in Eq. (4.13), and $\langle K_j \rangle \to E_j \langle K_j \rangle^3$. The only noise terms that anticommute with K_j [and hence give factors of $(1-p_2)$] are those in $j \cup N_j$. There are $2(d+1)$ of these (since there are two sets of noisy gates), so $\langle K_j \rangle \to (1-p_2)^{2(d+1)} \langle K_j \rangle^3$, which is Eq. (4.17).

Now suppose $j \in A$. Let $\mathbf{j} = (0,\dots,0,j,0,\dots,0)$. Our sum is over $\mathbf{a}^{(1)}, \mathbf{a}^{(2)} \in \{\mathbf{0},\mathbf{j}\}$, and $\mathbf{b} = \mathbf{0}$. Since we are interested only in $\langle K_j \rangle$, our effective noise model is $[X]_k \mapsto [Z]_j \forall k \in N_j$ and $[X]_j \mapsto [I]$. All other noise terms do not affect the state. Then

$$E_{01} \mapsto \left[(1-p_2) + p_2 P_Z^{(j,0)} P_Z^{(j,1)}\right]^{d+1}. \tag{4.42}$$

A similar replacement holds for E_{02}. E_{01} acts on terms $K_{\mathbf{j}+\mathbf{a}^{(1)}+\mathbf{a}^{(2)},\mathbf{0}} K_{\mathbf{a}^{(1)},\mathbf{0}}$ and gives a factor of 1 iff $\mathbf{j}+\mathbf{a}^{(1)}+\mathbf{a}^{(2)} = \mathbf{0}, \mathbf{a}^{(1)} = \mathbf{0} \Rightarrow \mathbf{j} = \mathbf{a}^{(2)}, \mathbf{a}^{(1)} = \mathbf{0}$, and a factor of $(1-p_2)^{d+1}$ otherwise.

Performing the MCNOT between $\rho^{(0)}$ and $\rho^{(2)}$, the noise channel E_{02} acts on the kets $K_{\mathbf{j}+\mathbf{a}^{(1)},\mathbf{0}} K_{\mathbf{a}^{(2)}}$, which gives a factor of 1 iff $\mathbf{j}+\mathbf{a}^{(1)} = \mathbf{0}, \mathbf{a}^{(2)} = \mathbf{0}$ and $(1-p)^{d+1}$ otherwise. Putting in each of the four cases $a_j^{(1)}, a_j^{(2)} \in \{0,1\}$ gives us Eq. (4.18).

4.5.1.3 Behavior of correlations

If we take two qubits j,k such that $\mathcal{N}(j) \cap \mathcal{N}(k) = \emptyset$, then the noise terms on sites in $\mathcal{N}(k) \cup k$ do not affect terms involving j and vice versa. Hence the sum over terms in the recursion relation for $\langle K_{jk} \rangle$ will factor into $\langle K_j \rangle \langle K_k \rangle$. If initially $\langle K_j K_k \rangle = \langle K_j \rangle \langle K_k \rangle$, then the three-copy protocol will not generate any new correlations between these regions.

4.5.2 Uniqueness of the Fixed Point

Here we show that the three-copy protocol has a unique fixed point for stabilizer elements $\langle K_{\mathbf{a},\mathbf{b}} \rangle$ with weight $w = |\mathbf{a}| + |\mathbf{b}| \le 2$. The recursion relations for stabilizer elements of weight $w > 1$ [see Eq. (4.13)] depend only on stabilizer elements whose weight is at most w. Thus, we can use an inductive argument. If all the stabilizer elements of weight less than w have reached a fixed point, they become constants and then the recursion relation for elements of weight w will have the same form as those for weight one (i.e., they will depend only on stabilizer elements of weight w). First consider the case when $|\mathbf{a}|, |\mathbf{b}| \le 1$. For this case, the three-copy recursion relations Eq. (4.13) have the form

$$f(z) = az + bz^3,$$
$$g(z) = cz + dz^3,$$

with $a, c > 0$ and $bd < 0$. The presence of noise does not change the form of the recursion relations, it only multiplies each term by a number between 0 and 1 (see Appendix 4.5.1.2). Let $y = z^2$ and $x = dy + c$. Define

$$p(x) := f(g(z))/z - 1 = bx^4 - bcx^3 + adx - d.$$

The signature of $p(x)$ is

$$\begin{aligned} p(x)&: -++-, \\ p(-x)&: ----. \end{aligned}$$

Then by Descartes' rule of signs [SL54], $p(x)$ has at least two complex roots. Thus the recursion relation $f(g(z)) = z$ has at most two positive fixed points. The recursion relation $g(f(z)) = z$ can be analyzed identically. It was already argued in Section 4.2.2 that this means that there is a unique attractive fixed point.

Now consider the case $|\mathbf{a}| = 2$ and $|\mathbf{b}| = 0$. The recursion relations now have the form

$$\begin{aligned} f(z) &= az^3 + bz + c \\ g(z) &= dz^3. \end{aligned}$$

It is easily checked that a, c, and d are positive. The sign of b is harder to fix, but we note that for there to be a fixed point at all, b must be negative. The case $f(g(z)) = z$ is easily analyzed, as above, to show that there are at most two positive roots. Let $p(z) = g(f(z))$. To conclude the proof we need two technical results. (i) If the smallest support expectation value $\langle K_a \rangle$ has reached its fixed point value $\langle K_a \rangle_{\mathrm{fp}}$ then the physically allowed values for $\langle K_{a+a'} \rangle$ form the interval $I = [2\langle K_a \rangle_{\mathrm{fp}} - 1, 1]$. (ii) $f(z) \geq 0$ for all $z \in I$. Proof of (i) (a) z allowed $\Rightarrow z \in I$: $P = \frac{1-K_a}{2}\frac{1-K_{a'}}{2}$, with $a \neq a'$, is a projector, hence $\langle P \rangle \geq 0$. Thus $z = \langle K_{a+a'} \rangle \geq \langle K_a \rangle + \langle K_{a'} \rangle - 1$ (*). Evaluate (*) at fixed point $\langle K_a \rangle_{\mathrm{fp}}$. $z \leq 1$ is obvious. (b) $z \in I \Rightarrow z$ allowed: For an initial state of the protocol, interpolate between $\rho_1 = \langle K_a \rangle_{\mathrm{fp}}\, \rho_{++} + (1 - \langle K_a \rangle_{\mathrm{fp}})/2\,(\rho_{+-} + \rho_{-+})$ and $\rho_2 = \langle K_a \rangle_{\mathrm{fp}}\, \rho_{++} + (1 - \langle K_a \rangle_{\mathrm{fp}})\rho_{--}$. (The signs "±" refer to the eigenvalues of K_a and $K_{a'}$, respectively.) Proof of (ii). Be $\langle K_a \rangle_{\mathrm{fp}}, \langle K_b \rangle_{\mathrm{fp}} > 0$ and $z \in I$. Assume as an hypothesis $f(z) < 0$. Apply (*) to the state after application of P1, at the fixed point $\langle K_a \rangle_{\mathrm{fp}}$, $\forall a \in A$. Hence $0 \geq f(z) \geq 2\langle K_b \rangle_{\mathrm{fp}} - 1$. (Under P1 the fixed point value $\langle K_a \rangle_{\mathrm{fp}}$ for $a \in A$ is mapped to $\langle K_b \rangle_{\mathrm{fp}}$ for $b \in B$, assuming all vertices have the same degree.) Thus, $\langle K_b \rangle_{\mathrm{fp}} \leq 1/2$. But then $\langle K_b \rangle_{\mathrm{fp}} = 0$, which is a contradiction. Hence $f(z) \geq 0$.

Now, $p''(z) = g''(f(z))f'(z)^2 + g'(f(z))f''(z)$ such that, with (ii), $p'' \geq 0$ for all $z \in I$. Thus, $p(z)$ is convex on I. With (i), I is a single interval such that $p(z)$ and z intersect at most twice in I. At most one of these fixed points is attractive.

4.5.3 The Depolarizing Operator

In order to prove that the depolarizing operator $\mathscr{D}$ defined in Eq. (4.22) commutes with the evolution operator $R = \mathrm{Tr}_{(1,2)}\, M \circ \mathscr{E} \circ U$, we note that the protocol step P1 consists of a unitary part U, an error channel $\mathscr{E}$ comprising probabilistic Pauli errors, and a measurement $\mathrm{Tr}_{(1,2)}\, M$, where M is a projector. U consists of a set of transversal CNOT-gates and acts on the stabilizer as

$$\begin{aligned} K^{(0)}_{\mathbf{a},\mathbf{b}} &\longrightarrow K^{(0)}_{\mathbf{a},\mathbf{b}} K^{(1)}_{\mathbf{a},\mathbf{0}} K^{(2)}_{\mathbf{a},\mathbf{0}}, \\ K^{(1)}_{\mathbf{a},\mathbf{b}} &\longrightarrow K^{(1)}_{\mathbf{a},\mathbf{b}} K^{(0)}_{\mathbf{0},\mathbf{b}}, \\ K^{(2)}_{\mathbf{a},\mathbf{b}} &\longrightarrow K^{(2)}_{\mathbf{a},\mathbf{b}} K^{(0)}_{\mathbf{0},\mathbf{b}}. \end{aligned} \tag{4.43}$$

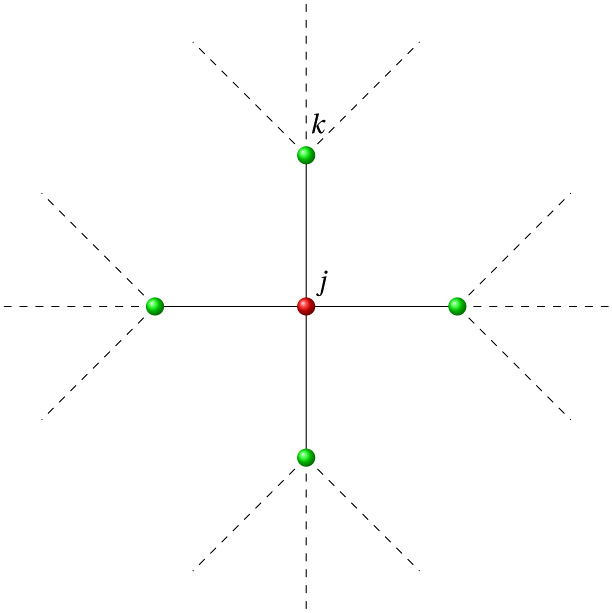


Figure 4.5 – Creation of a degree ($d = 4$) bi-colorable graph state. The figure will have the same local structure for other degrees and topologies as long as its edges are d colorable and its vertices are bi-colorable.

Now note that $\frac{[I]+[K^{(1)}_{\mathbf{0,b}}K^{(0)}_{\mathbf{0,b}}]}{2}\frac{[I]+[K^{(0)}_{\mathbf{0,b}}]}{2}\frac{[I]+[K^{(2)}_{\mathbf{0,b}}K^{(0)}_{\mathbf{0,b}}]}{2} = \frac{[I]+[K^{(0)}_{\mathbf{0,b}}]}{2}\frac{[I]+[K^{(1)}_{\mathbf{0,b}}]}{2}\frac{[I]+[K^{(2)}_{\mathbf{0,b}}]}{2}$ etc., such that

$$U \circ \mathscr{D}^{(0)}\mathscr{D}^{(1)}\mathscr{D}^{(2)} = \mathscr{D}^{(0)}\mathscr{D}^{(1)}\mathscr{D}^{(2)} \circ U. \tag{4.44}$$

The operations $\mathscr{D}^{(0)}\mathscr{D}^{(1)}\mathscr{D}^{(2)}$ and $\mathscr{E}$ commute because both are linear combinations of Pauli superoperators,

$$\mathscr{E} \circ \mathscr{D}^{(0)}\mathscr{D}^{(1)}\mathscr{D}^{(2)} = \mathscr{D}^{(0)}\mathscr{D}^{(1)}\mathscr{D}^{(2)} \circ \mathscr{E}. \tag{4.45}$$

The measurements comprising $\mathrm{Tr}_{(1,2)} M$ are of stabilizer operators $K^{(1)}_{\mathbf{0,b}}$, $K^{(2)}_{\mathbf{0,b}}$ on the states $\rho^{(1)}$, $\rho^{(2)}$, respectively. They are performed via one-qubit measurements and classical post-processing. $K^{(1)}_{\mathbf{0,b}}$, $K^{(2)}_{\mathbf{0,b}}$ commute with the Kraus operators in Eq. (4.22), such that

$$\begin{aligned} \underset{(1,2)}{\mathrm{Tr}}\, M \circ \mathscr{D}^{(0)}\mathscr{D}^{(1)}\mathscr{D}^{(2)} &= \underset{(1,2)}{\mathrm{Tr}}\, \mathscr{D}^{(0)}\mathscr{D}^{(1)}\mathscr{D}^{(2)} \circ M \\ &= \mathscr{D}^{(0)} \circ \underset{(1,2)}{\mathrm{Tr}}\, M. \end{aligned} \tag{4.46}$$

Eqs. (4.44), (4.45), and (4.46) yield Eq. (4.23)

4.5.4 Creation of a Bi-colorable Graph State

Here we discuss the noise structure of a bi-colorable graph that is created using noisy CPHASE gates. The noisy gates are modeled as the ideal gate followed by two-qubit depolarizing noise as defined in Eq. (4.15). The graph state is created by performing

CPHASE gates between qubits in the $|+\rangle$ state. The noise structure of the final state depends on the temporal ordering of these gates. If we assume that the underlying graph has constant degree d and that its edges are d colorable, then the N-qubit graph state can be created in d time steps with Nd CPHASE gates. At each time step all the gates corresponding to edges of a particular color are performed. Thus, at every time step $t \in \{1, \ldots, d\}$, each qubit is affected by an error channel of the form of Eq. (4.15).

We are interested in the value of $\langle K_j \rangle$, so we focus on the neighborhood of qubits j in the larger graph. Since the graph is bi-colorable, it contains no three cycles and one can draw a diagram of the form of Figure 4.5. The gates are represented by both solid as well as dashed lines. The noise channels corresponding to the solid lines each contribute an effective error $\mathrm{T}_{\mathrm{eff}}$ as defined in Eq. (4.16) to qubit j. Now consider the qubit k which is a neighbor of the central qubit j. Each dashed line also contributes an effective error $\mathrm{T}_{\mathrm{eff}}$ to qubit j, but only if the CPHASE gate corresponding to the solid line between k and j was performed in a previous timestep. This is because Z_k errors commute with K_j and X_k errors would be propagated by the CPHASE to $X_k Z_j$ errors, which also commute with K_j. Thus there are a total of $\frac{d(d-1)}{2} + d = \frac{d(d+1)}{2}$ noise channels affecting the qubit j. This gives

$$\langle K_j \rangle = (1 - p_2)^{\frac{d(d+1)}{2}}. \tag{4.47}$$

Bibliography

[AB02] Hans Aschauer and Hans J. Briegel. “Private Entanglement over Arbitrary Distances, Even Using Noisy Apparatus.” In: *Phys. Rev. Lett.* 88 (2002), p. 047902.

[ADB05] H. Aschauer, W. Dür, and H.-J. Briegel. “Multiparticle entanglement purification for two-colorable graph states.” In: *Phys. Rev. A* 71.1 (Jan. 2005), p. 012319.

[AGP06] P. Aliferis, D. Gottesman, and J. Preskill. “Quantum accuracy threshold for concatenated distance-3 code.” In: *Quant. Inf. Comput.* 6 (2006), p. 097.

[AL06] Panos Aliferis and Debbie W. Leung. “Simple proof of fault tolerance in the graph-state model.” In: *Physical Review A (Atomic, Molecular, and Optical Physics)* 73.3, 032308 (2006), p. 032308. DOI: `10.1103/PhysRevA.73.032308`.

[All02] Allen Hatcher. *Algebraic Topology.* Cambridge University Press, 2002. URL: `http://www.math.cornell.edu/~hatcher/AT/ATpage.html`.

[AP09] Panos Aliferis and John Preskill. “The Fibonacci scheme for fault-tolerant quantum computation.” In: *Phys. Rev. A.* 79 (2009). DOI: `10.1103/PhysRevA.79.012332`. arXiv: `quant-ph/0809.5063`.

[Bar+95] Adriano Barenco et al. “Elementary gates for quantum computation.” In: *Phys. Rev. A* 52.5 (1995), pp. 3457–3467. DOI: `10.1103/PhysRevA.52.3457`.

[Bel64] John S. Bell. “On the Einstein Podolsky Rosen Paradox.” In: *Physics 1* 195 (1964).

[Ben+93] Charles H. Bennett et al. “Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels.” In: *Phys. Rev. Lett.* 70.13 (1993), pp. 1895–1899. DOI: `10.1103/PhysRevLett.70.1895`.

[Ben+96a] C. H. Bennett et al. “Mixed-state entanglement and quantum error correction.” In: *Phys. Rev. A* 54.5 (1996), p. 3824.

[Ben+96b] Charles H. Bennett et al. “Purification of Noisy Entanglement and Faithful Teleportation via Noisy Channels.” In: *Phys. Rev. Lett.* 76 (1996), p. 722.

[BK05a] Sergey Bravyi and Alexei Kitaev. "Universal quantum computation with ideal Clifford gates and noisy ancillas." In: *Phys. Rev. A* 71.2 (2005), p. 022316. DOI: 10.1103/PhysRevA.71.022316. arXiv: quant-ph/0403025.

[BK05b] Sergey Bravyi and Alexei Kitaev. "Universal quantum computation with ideal Clifford gates and noisy ancillas." In: *Physical Review A (Atomic, Molecular, and Optical Physics)* 71.2, 022316 (2005), p. 022316. DOI: 10.1103/PhysRevA.71.022316. URL: http://link.aps.org/abstract/PRA/v71/e022316.

[BLN95] R. H. Byrd, P. Lu, and J. Nocedal. "A Limited Memory Algorithm for Bound Constrained Optimization." In: *SIAM Journal on Scientific and Statistical Computing* 16.5 (1995), p. 1190.

[Bra05] Sergey Bravyi. "Universal Quantum Computation with the nu=5/2 Fractional Quantum Hall State." In: (2005). arXiv: quant-ph/0511178.

[BW92] Charles H. Bennett and Stephen J. Wiesner. "Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states." In: *Phys. Rev. Lett.* 69.20 (1992), pp. 2881–2884. DOI: 10.1103/PhysRevLett.69.2881.

[CL04] Kai Chen and Hoi-Kwong Lo. "Multi-partite quantum cryptographic protocols with noisy GHZ states." In: (2004). arXiv: quant-ph/0404133.

[DAB03] W. Dür, H. Aschauer, and H.-J. Briegel. "Multiparticle Entanglement Purification for Graph States." In: *Physical Review Letters* 91 (2003), p. 107903.

[Dan97] Daniel Gottesman. "Stabilizer codes and quantum error correction." PhD thesis. Caltech, 1997. arXiv: quant-ph/9705052.

[Den+02a] Eric Dennis et al. "Topological quantum memory." In: *J. Math. Phys.* 43 (2002), pp. 4452–4505. arXiv: quant-ph/0110143.

[Den+02b] Eric Dennis et al. "Topological quantum memory." In: *J. Math. Phys.* 43 (2002), pp. 4452–4505. arXiv: quant-ph/0110143.

[DH06] Dan E. Browne and Hans J. Briegel. "One-way Quantum Computation - a tutorial introduction." In: (2006). arXiv: quant-ph/0603226v2.

[DHN06] Christopher M. Dawson, Henry L. Haselgrove, and Michael A. Nielsen. "Noise Thresholds for Optical Quantum Computers." In: *Phys. Rev. Lett.* 96 (2006), p. 020501. arXiv: quant-ph/0509060.

[Dür+05] W. Dür et al. "Standard forms of noisy quantum operations via depolarization." In: *Phys. Rev. A* 72 (2005), p. 052326.

[Dür+99] W. Dür et al. "Quantum repeaters based on entanglement purification." In: *Phys. Rev. A* 59 (1999), p. 169.

[Edm65] J. Edmonds. "Paths, trees, and flowers." In: *Can. J. Math.* 17 (1965), p. 449.

[Eke91] A. K. Ekert. "Quantum cryptography based on Bell's theorem." In: *Phys. Rev. Lett.* 67 (1991), p. 661.

[GC99] D. Gottesman and I. L. Chuang. “Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations.” In: *Nature (London)* 402 (1999), p. 390.

[Gro01] Lov K. Grover. “A fast quantum mechanical algorithm for database search.” In: *American Journal of Physics* 69 (2001).

[HDM05] Erik Hostens, Jeroen Dehaene, and Bart De Moor. “Hashing protocol for distilling multipartite CSS states.” In: (2005). arXiv: quant-ph/0510096.

[KF05] Rochus Klesse and Sandra Frank. “Quantum Error Correction in Spatially Correlated Quantum Noise.” In: *Phys. Rev. Lett.* 95 (2005), p. 230503.

[Kru+05] Caroline Kruszynska et al. “Quantum communication cost of preparing multipartite entanglement.” In: (2005). arXiv: quant-ph/0512218.

[Lau83] R. B. Laughlin. “Anomalous Quantum Hall Effect: An Incompressible Quantum Fluid with Fractionally Charged Excitations.” In: *Phys. Rev. Lett.* 50.18 (May 1983), pp. 1395–1398. DOI: 10.1103/PhysRevLett.50.1395.

[LC99] Hoi-Kwong Lo and H. F. Chau. “Unconditional Security of Quantum Key Distribution over Arbitrarily Long Distances.” In: *Science* 283.5410 (Mar. 1999), p. 2050.

[LW05] Michael A. Levin and Xiao-Gang Wen. “String-net condensation: A physical mechanism for topological phases.” In: *Phys. Rev. B* 71.4 (2005), p. 045110. DOI: 10.1103/PhysRevB.71.045110. arXiv: cond-mat/0404617.

[Man+03a] Olaf Mandel et al. “Coherent Transport of Neutral Atoms in Spin-Dependent Optical Lattice Potentials.” In: *Phys. Rev. Lett.* 91.1 (2003), p. 010407. DOI: 10.1103/PhysRevLett.91.010407.

[Man+03b] Olaf Mandel et al. “Controlled collisions for multi-particle entanglement of optically trapped atoms.” In: *Nature* 425 (2003), p. 937. URL: http://dx.doi.org/10.1038/nature02008.

[Mar+02a] Markus Greiner et al. “Collapse and revival of the matter wave field of a Bose-Einstein condensate.” In: *Nature* 419 (2002), p. 51.

[Mar+02b] Markus Greiner et al. “Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms.” In: *Nature* 415 (2002), p. 39. URL: http://dx.doi.org/10.1038/415039a.

[MB05] Akimasa Miyake and Hans J. Briegel. “Distillation of Multipartite Entanglement by Complementary Stabilizer Measurements.” In: *Phys. Rev. Lett.* 95 (2005), p. 220501.

[MI00] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information.* Section 10.5.3. Cambridge University Press, 2000.

[Nay+08] Chetan Nayak et al. “Non-Abelian anyons and topological quantum computation.” In: *Reviews of Modern Physics* 80.3 (2008). DOI: http://dx.doi.org/10.1103/RevModPhys.80.1083. URL: http://dx.doi.org/10.1103/RevModPhys.80.1083.

[ND05] Michael A. Nielsen and Christopher M. Dawson. “Fault-tolerant quantum computation with cluster states.” In: *Phys. Rev. A* 71 (2005), p. 042323. arXiv: quant-ph/0405134.

[Ohn+04] Takuya Ohno et al. “Phase structure of the random-plaquette Z2 gauge model: accuracy threshold for a toric quantum memory.” In: *Nuclear Physics B* 69.3 (2004), pp. 462–480. URL: http://www.sciencedirect.com/science/article/B6TVC-4D0WJKM-2/2/b6f166af8cdc2a6a985902b195482bdd.

[Pom96] Carl Pomerance. “A Tale of Two Sieves.” In: *Notices of the AMS* 43 (1996).

[RB01] R. Raussendorf and H. J. Briegel. “Quantum computing via measurements only.” In: *Phys. Rev. Lett.* 86 (2001), p. 5188. arXiv: quant-ph/0010033.

[RBB03] Robert Raussendorf, Daniel E. Browne, and Hans J. Briegel. “Measurement-based quantum computation on cluster states.” In: *Phys. Rev. A* 68.2 (2003), p. 022312. DOI: 10.1103/PhysRevA.68.022312. arXiv: quant-ph/0301052.

[RHG05] Robert Raussendorf, Jim Harrington, and Kovid Goyal. “A fault-tolerant one-way quantum computer.” In: (2005). arXiv: quant-ph/0510135.

[RSA78] R. Rivest, A. Shamir, and L. Adleman. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.” In: *Communications of the ACM* 21 (1978).

[SA98] S. B. Bravyi and A. Yu. Kitaev. “Quantum codes on a lattice with boundary.” In: (1998). arXiv: quant-ph/9811052v1.

[Sho97] Peter W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer.” In: *SIAM J. Sci. Statist. Comput.* 26 (1997).

[SL54] D.E. Smith and M.L. Latham. *The Geometry of Rene Descartes with a facsimile of the first edition.* New York: Dover Publications, 1954.

[Ste96] Andrew Steane. “Multiple Particle Interference and Quantum Error Correction.” In: *Proceedings of the Royal Society of London A* 452 (1996), p. 2551. arXiv: quant-ph/9601029.

[Stu+02] D Stucki et al. “Quantum key distribution over 67 km with a plug&play system.” In: *New Journal of Physics* 4 (2002), p. 41. URL: http://stacks.iop.org/1367-2630/4/41.

[TB05a] Barbara M. Terhal and Guido Burkard. “Fault-tolerant quantum computation for local non-Markovian noise.” In: *Phys. Rev. A* 71.1 (Jan. 2005), p. 012336. DOI: 10.1103/PhysRevA.71.012336. arXiv: quanth-ph/0402104.

[TB05b] Barbara M. Terhal and Guido Burkard. “Fault-tolerant quantum computation for local non-Markovian noise.” In: *Phys. Rev. A* 71 (2005), p. 012336.

[TN04] Koujin Takeda and Hidetoshi Nishimori. “Self-dual random-plaquette gauge model and the quantum toric code.” In: *Nuclear Physics B* 686.3 (2004), pp. 377–396. URL: `http://www.sciencedirect.com/science/article/B6TVC-4C0RRCC-1/2/8df36e2d8261ae9518b000c233051991`.

[TSG82] D. C. Tsui, H. L. Stormer, and A. C. Gossard. “Two-Dimensional Magnetotransport in the Extreme Quantum Limit.” In: *Phys. Rev. Lett.* 48.22 (May 1982), pp. 1559–1562. DOI: `10.1103/PhysRevLett.48.1559`.

[VBR05] Michael Varnava, Daniel E. Browne, and Terry Rudolph. “Loss tolerant one-way quantum computation - a horticultural approach.” In: (2005). arXiv: `quant-ph/0507036`.

[WHP03] Chenyang Wang, Jim Harrington, and John Preskill. “Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory.” In: *Annals of Physics* 303.1 (2003), pp. 31–58. URL: `http://www.sciencedirect.com/science/article/B6WB1-47MS76C-1/2/533c051225c1d18d6036cb2bcb24f6a9`.