Fault-Tolerant Quantum Computation with Topological Codes

Group 11: Kaixin Huang, Yixu Wang, Yong-Chan Yoo

1 Motivation and Goals

Quantum computation is accompanied by errors intrinsically. The two main sources of the errors are the decoherence from the perturbance of the environment and the limited precision of implementing an element of a Lie group from discretized operations. The inevitable errors make quantum error correction (QEC) and fault-tolerant quantum computation (FTQC) essential for the manifestation of practical and reliable quantum computation. [1]

Topological features of a system are characters that are invariant under continuous deformation or local perturbations. So, one can expect if quantum information is encoded in some topological feature of the quantum system, it should be protected robustly against any local error. Performing FTQC in a topological manner is resource- and time-saving. The topological codes are interesting from the aspects of physics as well. They are deeply related to many fields with profound implications, such as fractional quantum hall effect, categorization of symmetries and phases of matter, topological quantum field theory, etc.

We would like to review different aspects of topological code modes. We start by introducing the stabilizer formalism, on which important concepts of QEC codes are based. We introduce two types of topological stabilizer codes which are based on the notion of Z_2 chain complex on torus and plane respectively. For planar stabilizer code, we further discuss the methods of creating logical qubits and performing fault-tolerent quantum computation for that. Finally we discuss about the topological error correction and decoding methods.

2 Stabilizer Formalism

Stabilizer formalism focuses on the group of the Pauli matrices. For an n-qubit system, we consider the set

$$G_n = \{\pm 1, \pm i\} \times \{I, X, Y, Z\}^{\otimes n} \tag{1}$$

The set is closed under the matrix multiplication so forms a group. An interesting property of G_n is that the group elements either commute or anti-commute with each other.

We are interested in an Abelian subgroup of G_n, denoted by S, such that

$$S = \{g | g \neq I, g \in G_n \text{ and } \forall i, j, [g_i, g_j] = 0\} \tag{2}$$

One can show that if $-I \notin S$, then the eigenvalues of any element of S can only be ± 1. Since S is an Abelian group, there always exists a vector space V_s which is spanned by the common eigenvectors of S with eigenvalue 1. We call V_s the space stabilized by S, and S its stabilizer.

An efficient way of describing the stabilizer group is to specify its generators. The generators $\{g_1, g_2, ..., g_l\}$ form a subset of S such that any element g of S can be written as the product of the generators.

$$g = \prod_{i=1}^{l} g_i^{n_i}, n_i \in \{0, 1\}. \tag{3}$$

Furthermore, to have a minimal description, we require that any g_i cannot be written as the product of other generators. In the case of the stabilizer group, a group of 2^n elements needs only n generators, which is efficient for description. For an n-qubit system, if a stabilizer group is specified by k generators, then the corresponding V_s is of dimension 2^{n-k}.

The stabilizer formalism is efficient in describing the dynamics of the system as well. If a state $|\psi\rangle \in V_s$ is evolved by a unitary U, then

$$U|\psi\rangle = Ug|\psi\rangle = UgU^\dagger U|\psi\rangle \tag{4}$$

This means that the vector $U|\psi\rangle$ is stabilized by the group $USU^\dagger = \{UgU^\dagger, g \in S\}$. Correspondingly, $USU^\dagger$ is the stabilizer of linear space UV_s.

An interesting type of unitary evolution keeps the stabilizer group invariant, i.e. $USU^\dagger = S$. We denote the set of such U's as $N(s)$, the normalizer of S. Obviously $S \subset N(S)$. For 1-qubit stabilizer group, the Hadamard gate H and the phase gate $\left(\begin{smallmatrix} 1 & 0 \\ 0 & i \end{smallmatrix}\right)$ are in $N(S)$. For 2-qubit stabilizer group, the CNOT gate is a normalizer of 2-qubit stabilizer group as well. Actually any normalizer of the n-qubit stabilizer group can be constructed by the Hadamard, phase and CNOT gates.

The stabilizer formalism can describe certain types of measurement. We would like to measure an element of the Pauli group $g \in G_n$.There are two possible relations between g and the elements of the generators. The first case is that g commutes with all the generators of S. In this case, g and S share the same set of eigenvectors. This means if we measure the operator g in the space V_s, we get the probability 1 or -1 with probability 1. The second case is that g anticommutes with some of the generators of S. In this case, one can show that the probability 1 or -1 is $\frac{1}{2}$ respectively. In particular the posterior state after measurement of the operator g is given by

$$|\psi^{(\pm)}\rangle = \frac{I \pm g}{\sqrt{2}}|\psi\rangle \tag{5}$$

where the choice of the sign depends on whether the result of the measurement is ± 1.

We can in general define a stabilizer code. In quantum error correction, an $[n, k]$ code means we store the information represented by k qubits (logical qubits) in a system of n physical qubits. An $[n, k]$ stabilizer code is defined to be the vector space V_s stabilized by an Abelian subgroup S of G_n with $n - k$ generators and $-I \notin S$. We call this code $C(S)$. $C(S)$ is of dimension 2^k. A systematic way of generating the computational basis of $C(S)$ is to find a set of k operators $\{\bar{Z}_1, ..., \bar{Z}_k\}$ in G_n such that they mutually commute and commute with the generators of S $\{g_1, ...g_{n-k}\}$ as well. The elements in this set gets the name of logical operators. The computational basis of $C(S)$, $|x_1, ..., x_k\rangle$ is defined to be

the vector stabilized by the generators

$$\langle g_1, ...g_{n-k}, (-1)^{x_1}\bar{Z}_1, ..., (-1)^{x_k}\bar{Z}_k\rangle. \tag{6}$$

The above set of $\{\bar{Z}_1, ..., \bar{Z}_k\}$ gets the name of the logical Z operators for each logical qubit. We can define the corresponding logical X operators $\{\bar{X}_1, ..., \bar{X}_k\}$ by requiring $\bar{X}_i$ anti-commutes with $\bar{Z}_i$, but commutes with all other $\bar{Z}_j$'s, $\bar{X}_j$'s and generators. We mention that this section is transcribed from Chapter 10.5 of [2].

3 $\mathbf{Z_2}$ chain complex

In this part we will go through some important concepts about the $\mathbf{Z_2}$ chain complex, which is a useful tool towards the surface code.

1. A surface: $G = (V, E, F)$, vertices: $V = \{v_k\}$, edges: $E = \{e_l\}$, faces: $F = \{f_m\}$.
   The corresponding Abelian groups:
   $C_0 : c_0 = \sum_k z_k v_k, C_1 : c_1 = \sum_l z_l e_l, C_2 : c_2 = \sum_m z_m f_m$, where $z_k, z_l, z_m = \{0, 1\}$referred to the 0-chain, 1-chain and 2-chain.

2. A homomorphism (boundary operator): $\partial_i : C_i \to C_{i-1}$ s.t. $\partial_i \circ \partial_{i-1} = 0$. $\partial_i c_i$ is an $(i-1)$-chain which is the boundary of c_i.

3. c_i is called a cycle if it's in a kernel of ∂_i, and is called a trivial i-cycle if there exists an $(i+1)$-chain s.t. $c_i = \partial c_{i+1} (c_i \in Img(\partial_{i+1}))$.
   A homology group h_i is defined as a quotient group: $h_i = ker(\partial_i)/Img(\partial_{i+1})$. An element in it is called a homology class. For any c_i and c_i' chains that belong to the same homology class, there exists an (i+1)-chain s.t. $c_i = c_i' + \partial c_{i+1}$, and they are said to be homologically equivalent.

4. A qubit is defined on each edge $e_l \in E$ of the surface G (or equivalently the dual edge $\bar{e}_l$ of the dual surface $\bar{G}$). The corresponding Pauli product: $W(c_1) = \prod_l W_l^{z_l}$, where $W_l \in \{X_l, Y_l, Z_l\}$. The commutability of two operators is determined by the inner product of the chains.

5. $M(\partial_i)$ is the matrix representation of ∂_i, $(M(\partial_i)c_i) \cdot c_{i-1} = c_i \cdot (M(\partial_i)^T c_{i-1}), M(\partial_1) = M(\bar{\partial}_2)^T$.

6. Since $\partial_1 \circ \partial_2 = 0$, $M(\bar{\partial}_2)^T M(\partial_2) = 0$, we have $\bar{\partial}_{\bar{c}_2} \cdot \partial_{c_2} = 0$, which tells us that $X(\partial_{c_2})$ and $Z(\bar{\partial}_{\bar{c}_2})$ always commute.

4 Surface Codes

Now that we have defined what is a surface, we could apply the qubits on the edges of the surface to make a basic construction for a surface code. In other word, a group of qubits are defined on a 1-chain, on which we can define the stabilizer generators and the logical operators.

4.1 Toric Code

Figure 1 shows the basic idea of the toric code: mapping the surface code on a torus. The black square represents a square lattice G, while the gray one stands for the dual lattice. The stabilizer generators are defined as,

$$A_m = Z(\partial f_m), B_k = X(\partial \bar{f}_k) \tag{7}$$

They commute and are called plaquette and star operators. The eigenvalues of the corresponding code states $|\Psi\rangle$ are $+1$, which means the states are protected.
Next, we want to find a way to define the logical operators, that could change the code state. As we have learned from the notes, for two homologically equivalent 1-chains c_1, c_1', the actions of $Z(c_1), Z(c_1')$ on the code state are identical. This also holds for the actions of X on dual 1-chains.
Also, since the logical operators $Z(c_1), X(\bar{c}_1)$ have to commute with all stabilizer generators and also be independent, we can derive that $c_1, \bar{c}_1$ must be two non-trivial cycles belonging to different homology classes, as is shown in Figure 1. Thus we can define two groups of logical Pauli operators,

$$\{L_Z^{(1)} = Z(c_1^{(1)}), L_X^{(1)} = X(\bar{c}_1^{(1)})\}, \quad \{L_Z^{(2)} = Z(c_1^{(2)}), L_X^{(2)} = X(\bar{c}_1^{(2)})\} \tag{8}$$

$$L_Z^i|\Psi_Z(s_1, s_2)\rangle = (-1)^{s_i}|\Psi_Z(s_1, s_2)\rangle, \quad L_X^i|\Psi_X(s_1, s_2)\rangle = (-1)^{s_i}|\Psi_X(s_1, s_2)\rangle. \tag{9}$$

The number of stabilizer generators of a $n \times n$ square lattice on such torus is $|F|+|V|-2 = 2n^2-2$, while the number of qubits $|E|$ is $2n^2$. Thus we have a 2^2-dimensional stabilizer subspace, which can describe the degrees of freedom in the code space. The relationship between the surface code and the Z_2 chain complex is included in Table 1 as follows:

stabilizer code	chain complex
Z-type generator	f_m
X-type generator	$v_k(f_k)$
Z- & X-type operators	Boundaries of 2-chains
logical Z & X operators	nontrivial cycles $c_1, \bar{c}_1$
Z & X errors	1-chains $c_1, \bar{c}_1$
Z & X error syndromes	$\partial c_1, \partial \bar{c}_1$

Table 1: The relationship between the surface code and the Z_2 chain complex.

4.2 Planar Surface Code

Similar to the toric code, we can construct the codes on a planar $n \times (n-1)$ square lattice with 3-qubit star operators on the top and bottom boundaries while the plaquette operators on the left and right boundaries. The X operators on dual 1-chain can determine the smooth boundaries (top and bottom), while the Z operators for the rough boundaries (left and right). Beside the boundaries, the structure of planar surface code is quite similar to the toric code, with the vertical and horizontal 1-chains representing the new $c_1, \bar{c}_1$ chains. The related numbers now become $|E| = 2n^2 - 2n + 1, |V| = n^2 - n, |F| = n^2 - n$, which means the stabilizer subspace is a subspace of dimension 2.

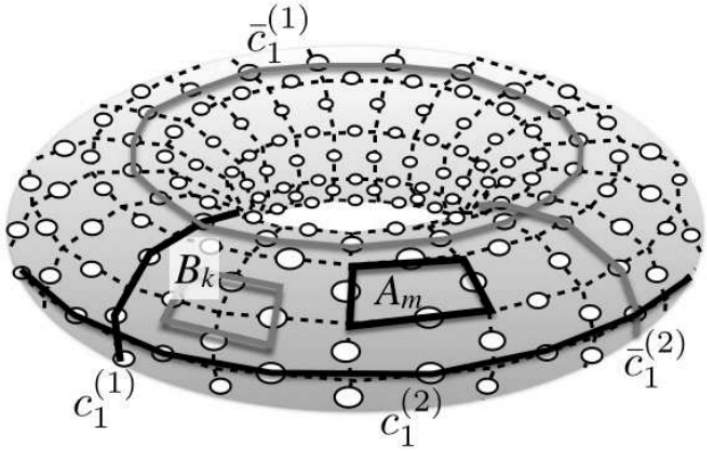

Figure 1: The surface code on a torus. Figure credit: [4]

In fact, the number of the logical qubits is closely related to the number of the rough boundaries and smooth boundaries. If we have at least two boundaries of the same type, say two smooth boundaries, a set of logical operators can be defined such that the logical Z operators is a 1-chain that is homologous to the smooth boundary and the corresponding logical X operators is a dual 1-chain that connects the two smooth boundaries. Discussion specialized on planar surface codes include such as [3].

4.3 Defect Pair Qubits and CNOT Gate by Braiding

Since the number of the logical qubits on a planer surface code depends actually on the numbers of each type of the boundaries, if we want to make the most use of the grid of the physical qubits, we should find ways to introduce more 'boundaries' on a plane. In the previous section, this is done by deleting physical qubits from the grid, thus introducing a physically rigid boundary. However, it changes the structure of the physical system, and is technically difficult to make further adjustments. So, it is more practical to introduce the so-called 'defect qubit pairs' by just making measurements on the original grid.

To simplify the discussion, we work on an infinite grid. Let the state stabilized by the all plaquette and star operators be $|v\rangle$. So that for any face f and vertex v, $A_f|v\rangle = B_v|v\rangle = |v\rangle$. This is the state without defect. We can create a defect area D by measuring the X operators for all the qubits inside D. By being inside we mean the boundaries of the region are not included. According to the outcome of the result, we can apply Z operator to those physical qubits whose outcomes are $|-\rangle$ so that all the measured states are $|+\rangle$. The posterior state with defect is

$$|D\rangle = \prod_{e \in D} \left(\frac{1 + X_e}{\sqrt{2}} \right) |v\rangle \tag{10}$$

according to Equation 5.

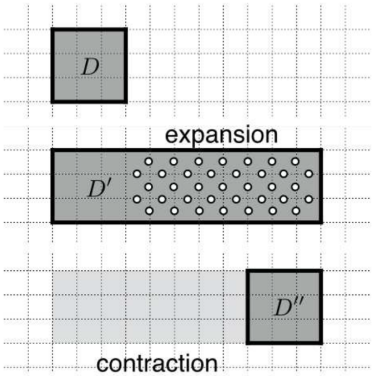

Figure 2: The figure shows how to move a defect region on the grid by expanding and contracting a single defect region. The operators are composed by local operators and measurements. Figure credit: [4]

Obviously the state $|D\rangle$ is no longer stabilized by the plaquette operators B_f whose boundaries overlap with the interior of region D, since they do not commute with the projection opeartors $\left(\frac{1+X_e}{\sqrt{2}}\right)$. However, the star operators B_v's are still the stabilizers of $|D\rangle$, and so is the operator $Z(\partial D)$. The effect of the measurement is to project the qubits inside D onto a certain state, which is equivalent to delete this part of the degrees of the freedom from the full system. Also, the A_f's and the B_v's which contain these qubits are removed from the set of stabilizer generators. The effect of the defect creating is shown in the left panel of Figure 5. To make a logical qubit, we need to create two disconnected regions D and D'. Now $Z(\partial D + \partial D')$ is a stabilizer, which can be composed of all the B_f operators whose $f \notin D \cup D'$. But $Z(\partial D)$ would act nontrivially on the state $|D, D'\rangle$ so is chosen as the logical Z operator. Since $Z(\partial D)Z(\partial D + \partial D') \sim Z(\partial D')$, $Z(\partial D')$ and $Z(\partial D)$ would act equivalently on the logical qubit. The logical X operator is chosen to be a dual 1-chain connecting the boundaries of D and D'.

The measurement is equivalent to creating a physical hole on the grid, but it is recoverable. To see how to recover the defect region, we can measure the state $|D\rangle$. Note that $|D\rangle$ in Equation 14 is a uniform superposition of all the possible configurations of applying X_e operators inside the defect region D. By measuring, it collapses to a certain configuration. We can then recover to state $|v\rangle$ just by applying X_e's on the corresponding positions. A defect region can be moved by expanding and recovering the defect region as described in Figure 4.

By measuring X_e's, we create a defect region on the primal lattice. We can create dual defect pairs on the dual lattice by measuring Z_e's as well. If we have a dual defect pair, as shown in the right panel of Figure 5, the boundary $X(\partial \bar{D})$ acts as a logical X operator while a 1-chain that connects the two dual defect regions is the logical Z operator. The techniques of moving a dual defect is similar as above discussion for the primal defects.

With the techniques of defect movement, we can implement the CNOT gate by braiding the primal and dual defects with each other. The procedure is demonstrated in Figure 6. To implement the CNOT gate, we need a primal and a dual defect pair. Denote the logical qubit they represent as 1 and 2 respectively. First, we wind one of the primal defect region around a dual defect region. As is shown on the left part of Figure 6. The logical X operator for the primal pair, which is the dual 1-chain that

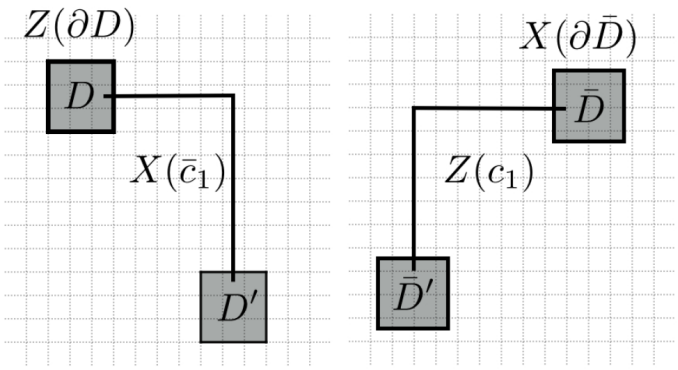

Figure 3: One can make defects on the grid by measurement rather than physical defects. A pair of the defect regions together defines a logical qubit (left). We can have a logical qubit defined on the dual defect regions as well (right). Figure credit: [4]

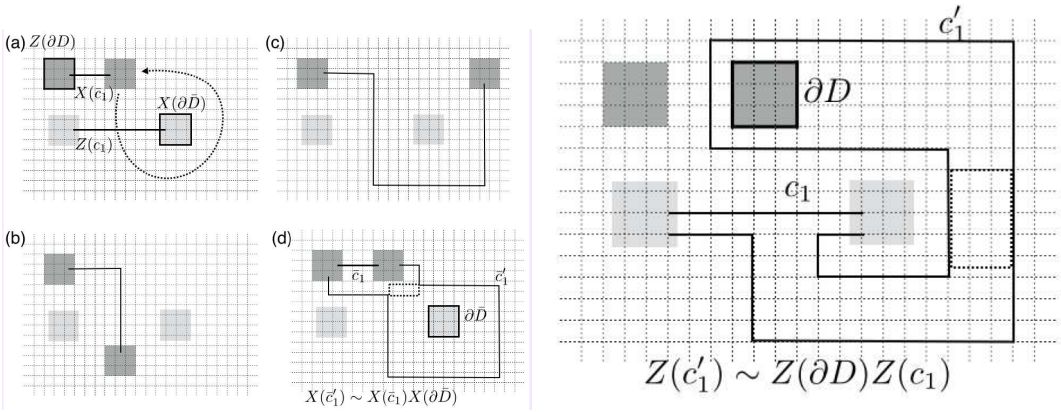

Figure 4: A logical CNOT operation by braiding a primal defect around a dual defect, and then braiding a dual one around a primal one. Figure credit: [4]

connects the two primal pairs, would now be homologous to a dual 1-chain, plus an extra part that winds around the dual defect region. It's written as $X(1) \to X(1)X(2)$, and $Z(1)$ is unchanged. Now we wind a dual defect region around a primal defect region. The 1-chain that connects the two dual defect regions would change similarly. This reflects the change in the logical operators: $Z(2) \to Z(1)Z(2)$ and $X(2)$ is left invariant. By recalling the effect of the CNOT gate on the control and target qubit:

$$\begin{aligned} &\text{CNOT } Z_c I_t \text{ CNOT}^\dagger = Z_c I_t, \text{ CNOT } I_c Z_t \text{ CNOT}^\dagger = Z_c Z_t, \\ &\text{CNOT } X_c I_t \text{ CNOT}^\dagger = X_c X_t, \text{CNOT } I_c X_t \text{ CNOT}^\dagger = I_c X_t, \end{aligned} \tag{11}$$

we see that the above procedure indeed implements the CNOT gates on the logical qubits where the primal and dual defect qubit acts as the control and target qubit respectively. This braiding procedure gives an example of fault-tolerant quantum computation on the planer surface code.

5 Topological Error Correction

In order to describe how topological error correction can be made, consider the toric code defined by the $n \times n$ square lattice on the torus again. There are 2^{2n^2-2} orthogonal eigenspaces of the stabilizer generators. These eigenspaces are called the syndrome subspaces, which are important to perform the error correction. Consider two kinds of errors, the $X(\bar{c}_1^e)$ and $Z(c_1^e)$ errors which are defined by 1-chains on the lattice. Using the anticommutivity of $X(\bar{c}_1^e)$ and $A_m = Z(\partial \bar{c}_1^e)$, the error syndromes can be defined as $\partial \bar{c}_1^e \equiv c_2^s$. Similarly, the anticommutivity of $Z(c_1^e)$ and $B_k = X(\partial c_1^e)$ gives $\partial \bar{c}_1^e \equiv c_2^s$. Since the stabilizer generators A_m and B_k are on the face f_m and vertex v_k, respectively, the error syndromes found above can be written as $c_2^s = \sum_m z_m^s f_m$ and $c_0^s = \sum_k z_k^s v_k$. Then, it follow that the eigenvalues of the stabilizer generators A_m and B_k associated with c_2^s and c_0^s are $(-1)^{z_m^s}$ and $(-1)^{v_k^s}$ respectively.

Here is the important point. The error correction for these error 1-chains is construction of the corresponding 'recovery' 1-chains $\bar{c}_1^r$ and c_1^r, which satisfy $\partial(\bar{c}_1^r + \bar{c}_1^e) = 0$ and $\partial(c_1^r + c_1^e) = 0$. Suppose each Z error is made with an independent and identical probability p. (X error analysis can be done in the similar way.)The posterior probability of an error $Z(c_1)$ with $c_1 = \sum_l z_l e_l$ conditioned on the error syndrome c_0^s is given by

$$P(c_1|c_0^s) = \mathcal{N} \prod_l \left(\frac{p}{1-p} \right)^{z_l} \Bigg|_{c_0^s} \tag{12}$$

A good recovery chain can make the posterior probability maximum, that is, c_1^r maximizes $\sum_l z_l e_l$ under the condition of the error syndrome. Equivalently, the overlap between the error and the recovery chains must be minimized. In this sense, it is called minimum distance decoding. Also, the triviality of the cycle $c_1^r + c_1^e$ is important. As stated before, $\partial(c_1^r + c_1^e) = 0$ means that the cycle $c_1^r + c_1^e$ is trivial, then the error would be corrected. If it is non-trivial, then there is a logical operation $Z(c_1^r + c_1^e)$ which can cause a logical error. This is shown in Figure 2.

Note that the homologically equivalent 1-chains result in the same actions on the code state. This can cause degeneracy problem of the surface code. That is, if each error syndrome is not uniquely determined, the combinatorics from those syndromes should also be considered. Let $c_1^r + c_1^{r'} \in h_i$ where h_i be the homology class. Then $c_1^{r'}$ gives the same result, so the sum of the posterior probabilities represented below should be maximized:

$$p_i = \sum_{c_1^{r'}} P(c_1^{r'}|c_0^s) \qquad \text{with the condition } c_1^r + c_1^{r'} \in h_i \tag{13}$$

The posterior probability can be calculated with the stabilizer and logical operators. Note that the recovery operator $Z(c_1^r)$ can be determined if an error syndrome c_0^s is given. Then, an error operator can be decomposed as $Z(c_1^e) = Z(c_1^r) G L_i$. Here, G and L_i are a stabilizer and logical operator, respectively. It follows that the posterior probability of the logical operator L_i is

$$p_i = P(L_i|c_0^s) = \frac{1}{\mathcal{N}} \sum_{G \in \mathcal{G}} P[Z(c_1^e) = Z(c_1^r) G L_i] \qquad \text{where } \mathcal{G} \text{ is the stabilizer operator group} \tag{14}$$

Let L_{i*} is the most probable logical operator that makes p_i maximum, then the error correction can be

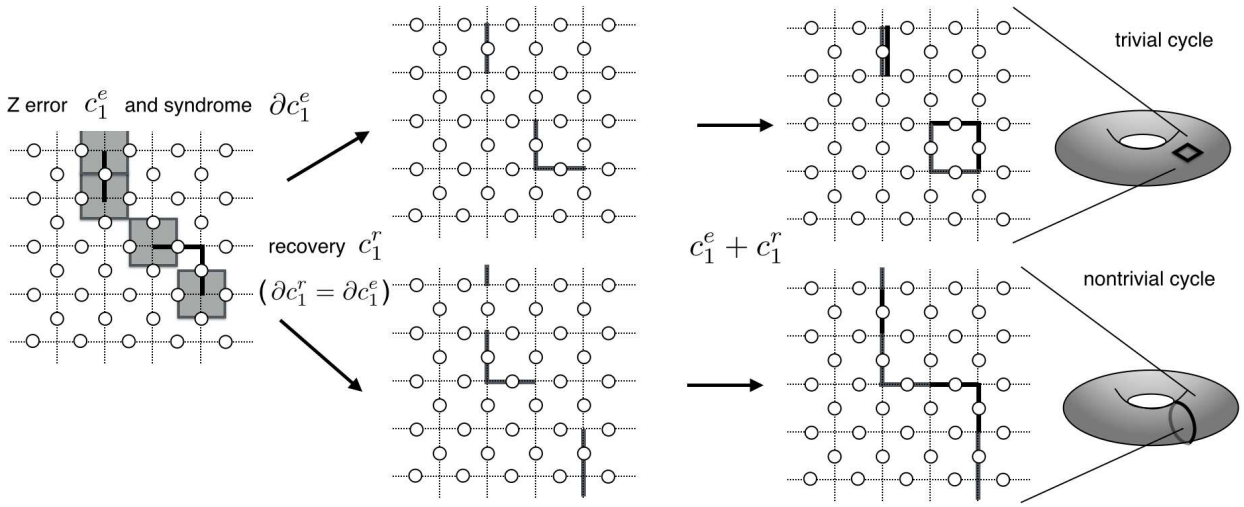

Figure 5: Schematic diagram of the topological error correction. Figure credit: [4]

done with an application of $L_{i*}Z(c_1^r)$. In this case, the logical error probability is $1 - p_{i*}$.

6 Decoding Methods

Considering physical realization of the fault-tolerant computation, there must be decoherence of quantum information systems. Therefore, efficient decoding algorithm is one of the most critical part for the realization of the fault-tolerant quantum computation. According to the several papers, it is known that the minimum distance decoding is NP-hard problem in general. However, with the condition that 1-chain that connects pairs of two vertices with a minimum Manhattan length, there is a classical algorithm, so-called Edmond's minimum-weight perfect matching (MWPM) algorithm, with the complexity of $\mathcal{O}(n^6)$ can implement this minimum distance decoding. There is an important point for the minimum distance decoding. That is, if the error probability p is sufficiently smaller than some threshold, then the logical error probability decreases as the lattice size increases. If specific weights imposed on each pairs of vertices having -1 eigenvalues, then the number of edges is greatly reduced from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^2)$ and the searching space of the pairs of vertices is also reduced.

Another type of decoding method can be implemented with a renormalization technique by introducing a hierarchal structure on the surface code. In this method, there are l-levels of those hierarchal structure and each level can be defined by recurrence. The level-1 logical qubits, which is constructed by using 12 qubits on the surface code, consist of 2 pairs of logical operators $\mathcal{L}^{(1)}$, 6 stabilizer generators $\mathcal{G}^{(1)}$, 6 pure errors $\bar{\mathcal{G}}^{(1)}$ (each of which anticommutes with $\mathcal{G}^{(1)}$), and 4 edge operators $\mathcal{E}^{(1)}$ (generate the Pauli group of the 12 qubits). They are described in Figure 3. By the similar argument of the above error 1-chain, $\bar{\mathcal{G}}^{(1)}$ is uniquely matched with the error syndrome $\mathcal{S}^{(1)}$. Then the posterior probability of

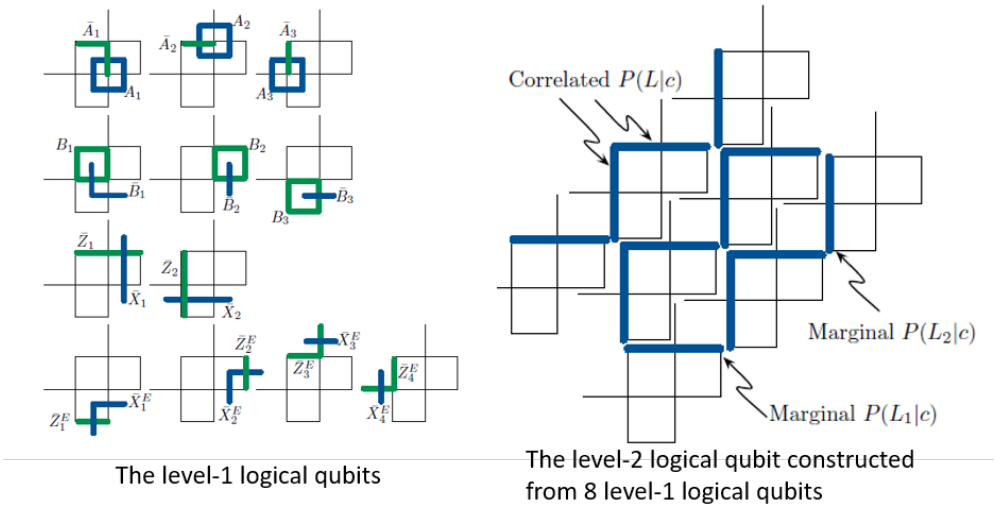

Figure 6: The Pauli operator basis in the renormalized algorithm. The level-1 logical qubits (left): The first and the second rows: the stabilizer generators and the pure errors; The third row: the logical operators; The last row: the edge operators. The level-2 logical qubits (right). Figure credit: [5]

the level-1 logical operator is given by

$$P(L^{(1)}|S^{(1)}) = \sum_{G^{(1)}\in\mathcal{G}^{(1)}, E\in\mathcal{E}^{(1)}} P(A = L^{(1)}G^{(1)}\bar{G}^{(1)}E^{(1)}|S^{(1)}), \tag{15}$$

where A is the decomposed Pauli operator in terms of the level-1 logical qubits. Then, a level-k unit cell can be constructed similarly with 8 level-$(k-1)$ unit cells and the 12 pairs of the level-$(k-1)$ operators on it. It follows that associated level-k logical qubits $\mathcal{L}^{(k)}$, $\mathcal{G}^{(k)}$, $\bar{\mathcal{G}}^{(k)}$, $\mathcal{E}^{(k)}$ and level-k error syndromes $\mathcal{S}^{(k)}$. At the end of construction, the posterior probability $P(L^{(l)}|S^{(l)})$ can be obtained under all imposed conditions imposed on the sub levels, and thus the most probable logical operator L^*. Therefore, it is called the renormalization decoding.

7 Faulty Syndrome Measurement

The fault-tolerant quantum computation must be able to deal with any source of noise, so the quantum error correction method must also be expanded by taking all these accounts. Consider the all possible sources of error including faulty syndrome measurements, which can happen in measuring the code state. Similar to the errors in the code space, the faulty syndrome measurement can result in an incorrect eigenvalue for the syndrome.

Since any Kraus operator of noise can be written as a superposition of Pauli operators, the error can be assumed as Pauli errors for modeling the faulty syndrome measurement. In addition to that, the Pauli Z and X operators are used to describe the errors for consistency. Just like the previous case, consider that there are Z errors in the topological code for simplicity. However, in this case, time

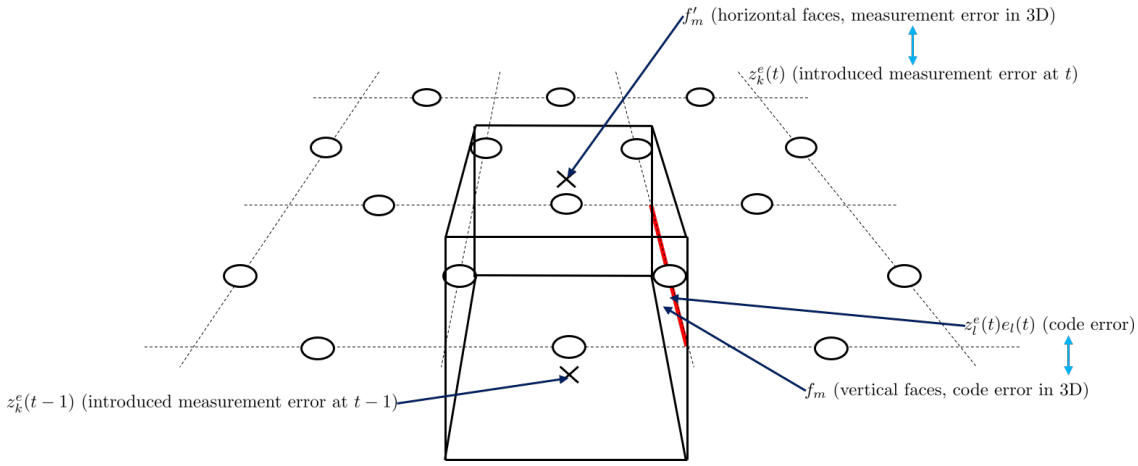

Figure 7: A schematic relation between the chain complex in 2D and 3D with time evolution.

dependence of each element of the chain complex is introduced. So, the error 1-chain is now defined as $c_1^e(t) = \sum_l z_l^e(t)e_l$. Similarly, the Z operators at time t can be defined by $Z[c_1(t)] = Z[\sum_l z_l(t)]$. Then, the time evolution of the Z operator is given as following equation of motion:

$$Z[c_1(t+1)] = Z[c_1(t) + c_1^e(t+1)]. \tag{16}$$

The measurement outcome at time step t is obtained by applying the star operator $B_k(t)$ (the plaquette operator $A_m(t)$ for the X errors):

$$m_k(t) = \left(\bigoplus_{e_l \in \delta v_k} z_l(t) \right) \oplus z_k^e(t), \tag{17}$$

where $z_k^e(t) \in \{0, 1\}$ is the error on the measurement error from the syndrome measurement. A relation with only in term of the contribution of measurement error from the faulty syndrome measurement can be calculated by the parity of the measurement outcomes at time steps t and $t-1$:

$$s_k(t) = m_k(t) \oplus m_k(t-1) = \left(\bigoplus_{e_l \in \delta v_k} z_l^e(t) \right) \oplus z_k^e(t) \oplus z_k^e(t-1). \tag{18}$$

This indicates that the errors arising from the faulty syndrome measurement can be detected by the parity $s_k(t)$. So, it is natural to define $\{s_k(t)\}$ as an error syndrome of the space-time errors.

The space-time error syndrome $\{s_k(t)\}$ can be defined efficiently if a 3-dimensional chain complex is introduced as shown in Figure 7. The 3-dimensional chain complex $\{c_0, c_1, c_2, c_3\}$ can be embedded in a cubic lattice. A primal cube is defined as the 3-chain c_3 and 2-chain, 1-chain and 0-chain are defined as face, edge and vertex of the cubic lattice, respectively. Also, the dual 3-chain complex $\{\bar{c}_0, \bar{c}_1, \bar{c}_2, \bar{c}_3\}$ can be similarly defined. Then the error in the 2-dimensional chain complex at time t corresponds to the vertical face of 3-dimensional chain complex. The measurement errors at time t and $t+1$ correspond to the horizontal faces of 3-dimensional chain complex. Then, the space-time error is now described by a

2-chain (equivalently dual 1-chain) of the 3-dimensional chain complex:

$$c_2^e = \sum_m z_m^e f_m \qquad \text{(error 2-chain)} \qquad \longleftrightarrow \qquad \bar{c}_1^e = \sum_m z_m^e \bar{e}_m \qquad \text{(dual error 1-chain)} \tag{19}$$

The associated syndrome $s_k(t)$ is defined by each 3-chain (equivalently dual 0-chain) so it can be denoted by s_r. The MWPM algorithm can be applied to this 3-dimensional structure to perform topological quantum error correction including the faulty syndrome measurement.

8 Noise Threshold and Noise Models

The noise threshold is a limit that can be tolerated by error correction code with no logical error. The noise threshold depends on specific noise models. Here, two basic noise models are introduced.

First, consider the simplest noise model, which is called the phenomenological noise model. This is a naturally expanded version of the previous noise model discussed in the 2-dimensional case. In this model, each error is located at dual 1-chain with an identical probability p. Moreover, the errors are independent of each other at each time step t. Furthermore, the probability of flipping the measure syndrome is also given by the same value p, since the 3-dimensional case also includes the faulty syndrome measurement. By the MWPM algorithm, the noise threshold of the phenomenological noise model is about 2.93%.

The circuit-based noise model is a more realistic model which also take into account of errors from the gate operations during the syndrome measurement. In this model, there are four kinds of error probabilities from different sources. First, the single-qubit depolarizing noise that influences single-qubit gates with probability p_1:

$$(1-p_1)\rho + \sum_{A\in\{X,Y,Z\}} \frac{p_1}{3} A\rho A. \tag{20}$$

Second, the two-qubit depolarizing noise that influences two-qubit gates with probability p_2:

$$(1-p_2)\rho + \sum_{A,B\in\{I,X,Y,Z\}/I\otimes I} \frac{p_2}{15} (A\otimes B)\rho(A\otimes B). \tag{21}$$

Third, the error in preparation of the Pauli-basis state with probability p_p:

$$(1-p_p)\frac{I+A}{2} + p_p\frac{I-A}{2}, \qquad A\in\{X,Y,Z\}. \tag{22}$$

Lastly, the error in measurement in the Pauli-basis with probability p_m:

$$\{(1-p_m)\frac{I+A}{2} + p_m\frac{I-A}{2}, \quad (1-p_m)\frac{I-A}{2} + p_m\frac{I+A}{2}\}, \qquad A\in\{X,Y,Z\}. \tag{23}$$

The noise threshold of the circuit-based model calculated by the MWPM algorithm is $p_2 = 0.75\%$ and the probabilities are related with $p_1 = p_2 = (3/2)p_p = (3/2)p_m$. It is known that more accurate threshold values can be obtained if noise propagation and correlation are considered in noise models. The related

numerical simulations in [6] shows this statement.

References

[1] John Preskill. Fault-tolerant quantum computation. *arXiv e-prints*, pages quant–ph/9712048, December 1997.

[2] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[3] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86:032324, September 2012.

[4] Keisuke Fujii. Quantum Computation with Topological Codes: from qubit to topological fault-tolerance. *arXiv e-prints*, page arXiv:1504.01444, April 2015.

[5] Guillaume Duclos-Cianci and David Poulin. A renormalization group decoding algorithm for topological quantum codes. *arXiv e-prints*, page arXiv:1006.1362, June 2010.

[6] R. Raussendorf, J. Harrington, and K. Goyal. Topological fault-tolerance in cluster state quantum computation. *New Journal of Physics*, 9:199, June 2007.