Understanding surface quantum error-correcting codes

November 23, 2005

Abstract

Effective quantum error correction is necessary for reliable storage of quantum information and fault-tolerant universal quantum computation. In this paper, we describe the family of quantum error-correcting codes called surface codes, which are introduced by Kitaev. Surface codes are favourable because of their nice locality properties. In these codes, qubits are arranged on a surface of nontrivial topology, and the allowed quantum operations on the code space are associated with nontrivial homological cycles of the surface.

We focus on a toric code and a planar code as examples of surface codes. Toric codes are easy to analyze because of its periodic boundary condition while the planar version is more convenient for implementation. In the codes we look at, the stabilizer generators are defined on sites or plaquettes with coordination number four. We then describe the procedures for error detection and recovery, as well as encoding and measurement. We briefly discuss the efficacy of error recovery, and review the estimate of accuracy threshold for surface codes.

1 Introduction

A general quantum state can be expressed as some superposition of the eigenfunctions of any observable. Such a superposition of readily distinguishable macroscopic states differs from a classical mixture of these states in that it contains important phase information, and exhibits quantum interference between the different portions. Such a quantum superposition is highly unstable

since the environment constantly tries to measure it. The measurement can be thought of as an entanglement procedure between the quantum system and its environment. The environment often has many degrees of freedom, hence it is unlikely that the different portions of the quantum superposition couples to the environment in the same manner. Different couplings lead to phase decoherence of the quantum state. Due to unavoidable interactions with the classical surroundings, decoherence of a quantum state is pervasive and rapid. The decoherence problem is a serious challenge for the implementation of quantum computing as well as for reliable storage of quantum information.

The ubiquitousness of quantum decoherence necessitates the study of quantum error-correcting codes. The goal of quantum error correction is to battle against the effect of decoherence, thereby allowing the computation time to exceed the decoherence time of the system.

The threshold theorem asserts that an arbitrarily long quantum computation can be executed with arbitrarily high fidelity, provided that the error rates of the computer's fundamental quantum gates are below a certain critical value, the *accuracy threshold* [7, 8, 9, 10, 11]. The numerical value of the accuracy threshold sets the standard to be met by designer of quantum hardware. The critical error probability per gate p_c has been estimated to be $p_c \gtrsim 10^{-4}$. Loosely speaking, it means that if decoherence is the only source of error, a quantum computer can carry out robust quantum computation if the decoherence time of the stored qubits is at least 10^4 longer than the execution time of one fundamental quantum gate. This estimate of the accuracy threshold is obtained by analyzing the efficacy of a concatenated code, that is, a hierarchy of codes within codes. The estimate rests on a number of assumptions, one of which is that a quantum gate can act on any pair of qubits with a fidelity that is independent of the spatial separation between them. This assumption is clearly unrealistic.

Fortunately, it has been shown in [8, 12] that the threshold theorem still holds even if quantum gates are required to be local. An estimate of the threshold value for this more realistic case has been shown in [1] based on surface codes with excellent locality properties. They obtained a critical error probability per gate and per time step of $p_c \gtrsim 1.7 \times 10^{-4}$ for reliable storage based on a series assumptions which will be outlined in section 7. While they did not obtain a critical value for quantum computation, they argued that the value is not likely to be substantially different.

Following the approach to quantum fault tolerance in [1], which is based

on the surface codes introduced in [2, 3], we study the properties of surface codes in which the quantum operations required to control errors are local. Locality is a very desirable property to have in view of the capabilities and limitations of foreseeable future quantum computers. For this reason, surface codes might suggest a promising approach to quantum computing architecture.

In section 2, we cover some background material in quantum error correction. Then we provide two examples of surface codes, toric codes and planar codes, in section 3. Before proceeding to describe error syndromes and recovery methods mainly for toric codes in section 5, we introduce in section 4 a few concepts from homology theory in the context of surface codes of which we shall make use throughout the paper. After focusing for a long time on the encode block, we shift our attention in section 6 to the encoding and measurement procedure for the planar code which encodes one single qubit. Finally, we conclude in section 8.

2 Basics of quantum error correction

To perform quantum error correction on a block of k qubits, we first encode the original block in an even larger block, say of size n. After some time, errors might occur in the enlarged code block. The redundancy will help us identify the correct way to fix an error upon its detection if the error rate is small. To recover the original message, we detect the location and nature of errors by attaching some ancilla qubits and measuring the error syndrome. Assuming that the detection is perfect, we can undo the error and thereby revert the erroneous message to the original message.

Any 2×2 matrix can be expressed as a linear combination of the 2×2 identity, I, and Pauli matrices, X, Y and Z where

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{1}$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2}$$

The tensor product of n operators, each of which chosen from $\{I, X, Y, Z\}$, forms a Pauli operator acting on n qubits. It is easy to see that there are 4^n Pauli operators on n qubits, and that they span the space of $2^n \times 2^n$

matrices. Hence, by Theorem 2 in [6], we need only consider the Pauli errors (errors resulting from applying Pauli operators on the state) on the n qubits. The Pauli operators have the nice property that they are unitary as well as Hermitian.

Theoretically, we can formulate an error-correcting code in terms of its stabilizer. The Pauli operators with the an overall phase of ± 1 or $\pm i$ form the n-qubit Pauli group $\mathcal{P}_n$. Two members of the Pauli group either commute or anticommute. The stabilizer of an error-correcting code is an Abelian subgroup S of the Pauli group $\mathcal{P}_n$ which does not contain $-I$. Then the stabilizer uniquely defines a nontrivial quantum code $T(S)$ as the simultaneous eigenspace of the stabilizer with eigenvalue one; that is

$$T(S) = \{|\psi) \in \mathcal{H}_n : M|\psi) = |\psi) \ \forall M \in S\} \tag{3}$$

where $\mathcal{H}_n$ is the Hilbert space of n-qubit states. A minimal set of generators for the stabilizer provides a set of error syndrome measurements.

The normalizer $N(S)$ of S is composed of Pauli operators which commute with S. It is a subgroup of $\mathcal{P}_n$. A Pauli error cannot be detected if and only if it is in $N(S) \backslash S$. However, we may not be able to recover the original message correctly even if we can detect the presence of an error since the same error syndrome might be caused by different errors.

If n is the number of qubits involved in a stabilizer code, and a represents the minimal number of operators required to generate the stabilizer, then

$$n - a = k \tag{4}$$

where k is the number of encoded qubits.

3 Two examples of surface codes

In this section, we examine surface codes, a family of quantum error-correcting codes introduced in [2, 3]. Surface codes are capable of encoding and correcting errors for a block of qubits. The main advantage of surface codes is that the quantum operations for error syndrome measurement is highly local, which makes them especially suitable for fault-tolerant implementation.

A surface code is a special type of "stabilizer code" [4, 5]. Here, we describe two classes of surface codes, toric codes and planar codes, as stabilizer codes; that is, we specify for each of them the check operators which form

a minimal set of stabilizer generators. Toric codes reside on a surface that is topologically equivalent to a torus whereas planar codes reside on a Euclidean plane with unidentified edges. Other surface codes involve different tessellations of a two-dimensional surface. Toric codes and planar codes are of particular usefulness. We describe them both in this paper because toric codes have nice periodic boundary condition which simplifies the error analysis whereas the planar code is advantageous for practical implementation.

3.1 Toric codes

It is convenient to picture toric codes on a square with opposite edges identified. Such a square is topologically equivalent to a torus. Qubits are in one-to-one correspondence with the links of a square lattice drawn on the square. An $L \times L$ lattice has $2L^2$ links; hence, it represents a block of $n = 2L^2$ qubits. This gives a large number of degrees of freedom onto which we will encode two qubits.

The set of check operators acting on the Hilbert space of dimension 2^n consists of two types, the site operators X_S and the plaquette operators Z_P. X_S is defined on each site (intersection of four links) as the tensor product of the four X's, each acting on a qubit that radiates from the site, with the identity operator on the remaining qubits. In contrast, an operator Z_P lives on a plaquette (interior of an elementary cell, bordered by four links); it is the tensor product of the four Z's, each of which acts on the bordering four qubits, with the identity on the remaining qubits. The check operators belong to the Pauli group $\mathcal{P}_n$. As depicted in figure 1, they act nontrivially on and only on those qubits in contact with the site or plaquette.

The check operators defined above are mutually commuting. Clearly, operators of the same type commute; that is, site operators commute with site operators, and plaquette operators with plaquette operators. This is because $[X, X] = [Z, Z] = [I, X] = [I, Z] = 0$. Furthermore, site operators commute with plaquette operators due to the geometrical constraint that a site operator and a plaquette operator act nontrivially either on two disjoint sets of links or on sets sharing exactly two links. In the first case, the operators commute because $[I, X] = [I, Z] = 0$. Whereas in the second case, the Pauli matrices corresponding to the site and plaquette operators acting on a qubit in the intersection anticommute, hence giving a minus sign when the order of operations is reversed; however there are two qubits in the intersection, the two minus signs cancel.

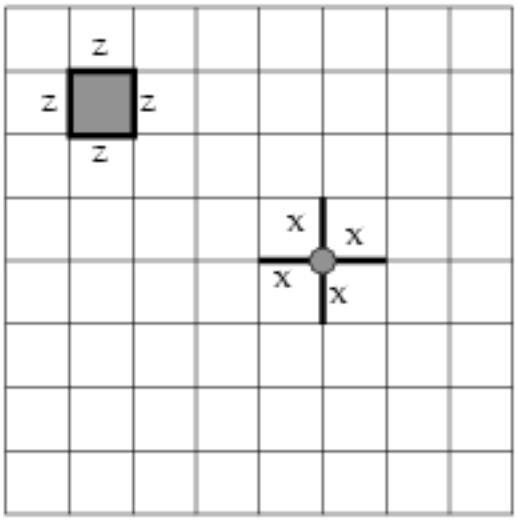

Figure 1: Figure and caption verbatim from [1]. Check operators of the toric code. Each plaquette operator is a tensor product of Z's acting on the four links contained in the plaquette. Each site operator is a tensor product of X's acting on the four links that meet at the site.

Since the check operators are mutually commuting, the group generated by the check operators is Abelian. Furthermore, this group does not contain $-I$. We let this group be the stabilizer of the toric code. Then the toric code is the simultaneous eigenspace of the check operators with eigenvalue one.

Due to the periodic boundary condition, the product of all L^2 site operators is the identity on the $2L^2$ qubits. Moreover, a site operator is the inverse of itself; therefore, it can be written as the product of all other $L^2 - 1$ site operators. The same argument also holds for the plaquette operators. For this reason, there are only $2(L^2 - 1)$ independent check operators although there are L^2 sites and L^2 plaquettes on an $L \times L$ lattice, The number of encoded qubit is $2L^2 - 2(L^2 - 1) = 2$ by equation 4.

3.2 Planar codes

To keep all the quantum operations local for the toric code, the qubits must be arranged on a topologically nontrivial surface, the torus. In practice, the toroidal topology would be difficult to implement especially when interactions between different tori are required. Fortunately, we can construct surface codes on planar sheets. These codes are called planar codes. Since the incorporation of planar sheets into quantum computing hardware can be achieved more easily, the use of planar codes offers a more feasible alternative.

For planar codes, qubits and check operators are arranged in a similar manner as for toric codes. However, the boundary condition is no longer periodic, giving rise to check operators which act on only three qubits instead of four at the boundary of the surface. As shown in figure 2, the boundary has two different types of edges. Along a "rough edge" (top or bottom edge in figure 2, each plaquette check operator acts on three qubits. Along a "smooth edge" (left or right edge), each check operator is a three-qubit site operator.

For a planar code with L sites between the two smooth edges and L plaquettes between the two rough edges, the lattice has $L^2 + (L-1)^2$ links, $L(L-1)$ plaquette operators and $L(L-1)$ site operators. All plaquette and site operators of the planar code are independent. Hence the number of encoded qubit is $L^2 + (L-1)^2 - 2L(L-1) = 1$ as given by equation 4.

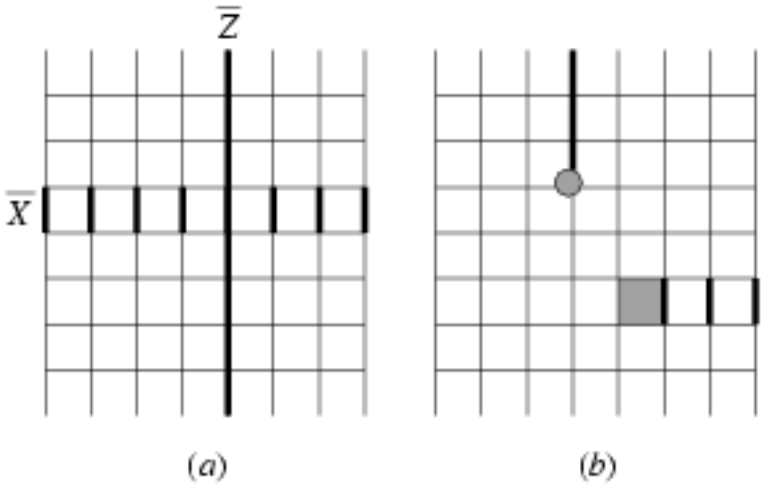

Figure 2: Figure verbatim from [1]. A planar quantum code. (a) At the top and bottom are the "rough edges" where the three-qubit plaquette operators are located, and at the left and right are the "smooth edges" where the three-qubit site operators reside. The logical operator $\bar{Z}$ for the planar code is a tensor product of Z's acting on a chain running from one rough edge to the other; whereas, the logical operator $\bar{X}$ is a tensor product of X's acting along a chain of the dual lattice running from one smooth edge to the other. For the lattice shown, the code's distance is $L = 8$. (b) Not used in this paper.

4 Homology of error chains

Here, we introduce some vocabulary for describing errors in an encoded block. A (Z_2-valued) 0-chain is a mapping which assigns values from $Z_2 = \{0, 1\}$ to the sites of a lattice. Similarly, a 1-chain (or equivalently 'chain') assigns elements of Z_2 to each link, and a 2-chain assigns elements of Z_2 to the lattice plaquettes. The term 1-chain can also be used to mean the set of all links that are assigned the value 1 by such mapping. The sum $C_1 + C_2$ of two chains C_1 and C_2 is the disjoint union of the chains, or alternatively, the modulo 2 addition of the two maps. For $i = 0, 1 and 2$, the set of all i-chains defined on a lattice forms a vector space. The linear boundary operator ∂ is defined in the obvious way in which the boundary of a plaquette is the sum of the four links comprising the plaquette, and the boundary of a link is the sum of the two endpoints (or sites). We observe that the operator ∂ takes 2-chains to 1-chains, and 1-chains to 0-chains.

In describing the properties of a surface code, the following notions are of particular importance. A cycle is a 1-chain whose boundary is trivial. A 1-cycle is said to be homologically trivial if it can be expressed as the boundary of a 2-chain; otherwise, it is homologically nontrivial. See figure 3 for a pictorial comparison between a homologically trivial chain (a) and a nontrivial chain (b).

5 Syndrome measurement and code recovery

We are now ready to describe the procedure of syndrome measurement and code recovery using the language of homology.

For a toric code, the defects obtained by measuring all the site operators is the boundary of some 1-chain. We say that there is a defect at a site s if the site operator X_s is measured to be -1. Assuming that there are no syndrome measurement errors, we can correct these errors by applying Z to the links along a chain of minimal length whose boundary consists of the sites with defects.

The selection of a 1-chain to which we apply Z is ambiguous as shown in figure 3. If the guessed error chain and the actual error chain differ only by a homologically trivial cycle, the recovery by applying Z to the links in the guessed error chain would correct all the phase errors since Z applied to a homologically trivial cycle acts trivially in the coding space. However, if

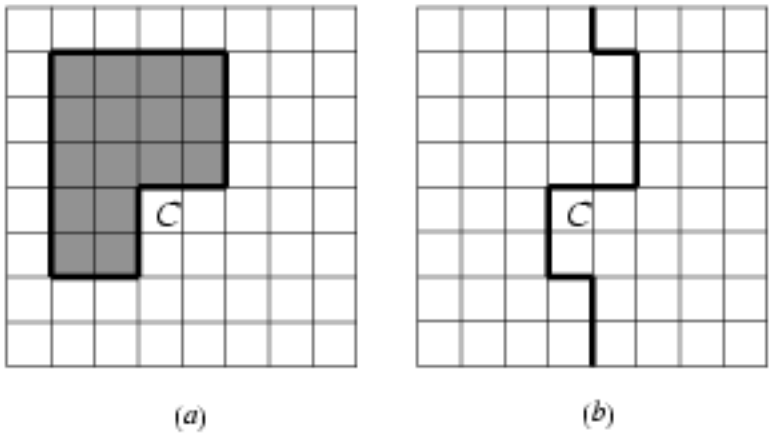

Figure 3: Figure and caption verbatim from [1]. Cycles on the lattice. (a) A homologically trivial cycle bounds a region that can be tiled by plaquettes. The corresponding tensor product of Z's lies in the stabilizer of the toric code. (b) A homologically nontrivial cycle is not a boundary. The corresponding tensor product of Z's commutes with the stabilizer but is not contained in it. It is a logical operation that acts nontrivially in the code subspace.

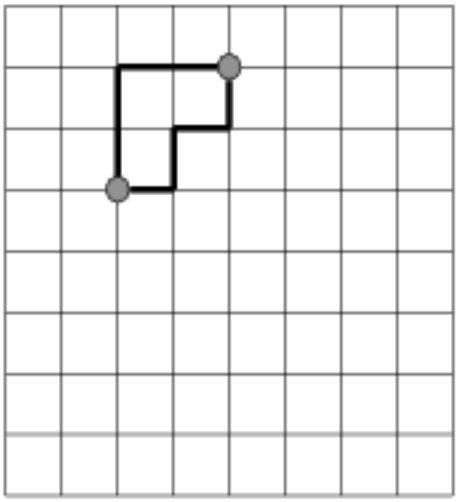

Figure 4: Figure and caption verbatim from [1]. The highly ambiguous syndrome of the toric code. The two site defects shown could arise from errors on either one of the two chains shown. In general, error chains with the same boundary generate the same syndrome, and error chains that are homologically equivalent act on the code space in the same way.

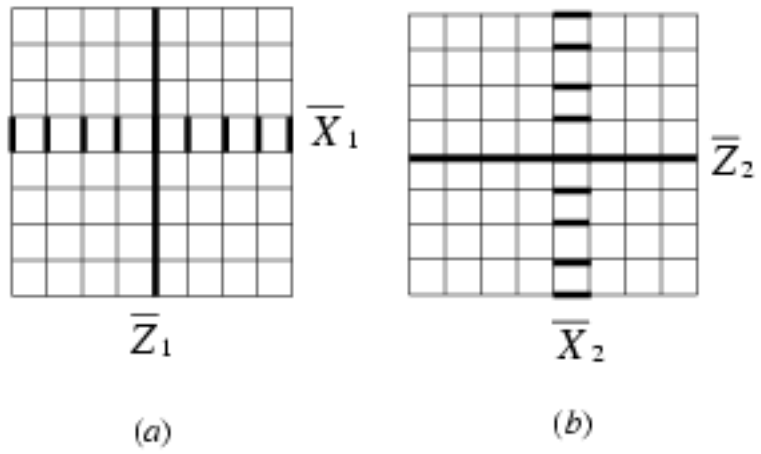

Figure 5: Figure and caption verbatim from [1]. Basis for the operators that act on the two encoded qubits of the toric code. The logical operators $\bar{Z}_1$ and $\bar{Z}_2$ are tensor products of Z's associated with the fundamental nontrivial cycles of the torus constructed from links of the lattice. The complementary operators $\bar{X}_1$ and $\bar{X}_2$ are tensor products of X's associated with nontrivial cycles constructed from links of the dual lattice.

the guessed error chain differs from the actual error chain by a homologically nontrivial cycle, then applying Z along the guessed error chain would be equivalent to applying $\bar{Z}_i$ to the encoded block where $\bar{Z}_1$ and $\bar{Z}_2$ are tensor products of Z's acting along the two fundamental nontrivial cycles respective (see figure 2). In the case of discrepancy by a nontrivial cycle, the error recovery based on the guessed error chain would fail. For this reason, the minimal weight of an operator which acts nontrivially in the coding space and can escape the error detection scheme is L for the surface codes described in section 3. Hence, those codes have distance L.

For a closed orientable Riemann surface of genus g, of which a torus is an example, $2g$ qubits can be encoded because for each handle introduced to a surface, 2 new logical $\bar{Z}$'s are introduced where $\bar{Z}$ corresponds to a homologically nontrivial cycle of the surface. Bit flip errors can be detected

and corrected using the same procedure applied to the dual lattice.

In the planar code described in section 3, the error analysis is more complicated since the check operators at the boundary of the surface need be treated differently from those in the interior. Some discussion about the error chains in the planar code can be found in [1]. Planar codes can also be used to encode more than one qubit. A surface with e distinct rough edges separated by e distinct smooth edges can encode $e-1$ qubits which are associated with the relative 1-cycles that connect one rough edge with any of the others.

In reality, there will be some errors in the syndrome measurement. This substantially complicates the syndrome measurement and error recovery procedure. [1] devised a way to recover the code from errors in this case. The error syndrome measurements have to be repeated in order to determine a reliable recovery strategy.

6 Implementation

After discussing the theory of error detection and recovery, we turn to look at the practical aspect of how to implement a surface code. In order to make use of a stabilizer code, we need to first encode a block of qubits. At the end of a period of data storage or a quantum computation, we need to read out the information by measuring the evolved encoded block. In this section, we use the planar code with two rough edges and two smooth edges, which encodes one qubit, as an example.

6.1 Understanding the coding space and encoded operations

To carry out quantum error correction, the first procedure is to encode a quantum state. The encoding process for k qubits is one which maps a state in the Hilbert $\mathcal{H}_k$ into a subspace of the same dimension in a larger Hilbert space $\mathcal{H}_n$ where n is the number of qubits in the error-correcting code. The map must be injective. To understand the process of encoding, we first consider the coding space and the encoded operations for our stabilizer code.

For the planar code which encodes one qubit, the coding space is a 2-dimensional complex subspace corresponding to the $+1$ eigenvalue for all check operators. $\bar{Z}$ and $\bar{X}$ act on a nontrivial homology cycle of the code;

they commutes with the stabilizer and acts nontrivially on the coding space. Therefore, $\bar{Z}$ and $\bar{X}$ are two encoded operations which can be performed on the encoded block. We have the freedom to choose a pair of orthonormal eigenstates as our basis of the coding space. A convenient choice of a basis would be the eigenfunctions of $\bar{Z}$ or the eigenfunctions of $\bar{X}$.

6.2 Encoding of known states

Before we store a qubit or start a quantum computation, we need to prepare encoded qubits in eigenstates of the encoded operations. Suppose we are given a state

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{5}$$

to start with. We encode it by mapping it to

$$\alpha|\bar{0}\rangle + \beta|\bar{1}\rangle \tag{6}$$

where $|\bar{0}\rangle$ and $|\bar{1}\rangle$ are 1 and -1 eigenstates of $\bar{Z}$ respectively.

If there were no errors in syndrome measurements, $|\bar{0}\rangle$, a $\bar{Z} = 1$ eigenstate in the coding space, can be prepared in the following way. We start with the state $|0\rangle^{\otimes n}$ where n is the number of qubits in the code. Since $|0\rangle$ is a $+1$ eigenstate of the operator Z, the initial state is a simultaneous eigenstate with eigenvalue 1 of all plaquette operators Z_p and of the operator $\bar{Z}$. However, the initial state does not belong to the $+1$ eigenspace of the site operators. We wish to modify the state so that it becomes a $+1$ eigenstate of X_s as well as for the plaquette operators and $\bar{Z}$. Therefore, we measure all the site operators, and apply Z to each link of an arbitrary 1-chain whose boundary consists of the positions of all the site defects. By applying these operations, we impose $X_s = 1$ at each site. Since the site operators commute with the plaquette operators and $\bar{Z}$, measuring the site operators does not change their values. Applying any tensor product of Z's on the initial state, which is a tensor product of n eigenstate of Z, takes it to a state which is colinear with the initial state and therefore corresponding to the same eigenvalue. The resulting state is a simultaneous eigenstate of the logical operator $\bar{Z}$, the plaquette operators and the site operators. When choosing an arbitrary 1-chain, we have the freedom to apply $\bar{Z}$ to the code block by applying Z to a homologically nontrivial path. However, the state before applying $\bar{Z}$ is an eigenstate of $\bar{Z}$, therefore it remains to be an eigenstate corresponding to the same eigenvalue after the operation.

In the real world, however, syndrome measurement may be afflicted by errors. Therefore, measuring the site operators and applying Z on the syndrome chain might not ensure $X_S = 1$ at all sites. In this case, we need to repeatedly measure both the X and Z syndromes of order L times, where L is the linear size of the lattice, and use the global recovery method described in [1] to control errors. After the error correction procedure, we end up with the desired state $|\bar{0}\rangle$.

Using a similar procedure, we can prepare a $\bar{Z} = -1$ eigenstate, $|\bar{1}\rangle$ in the coding space, except that we now start from $|1\rangle^{\otimes n}$ instead of $|0\rangle^{\otimes n}$.

Alternatively, we may choose to encode a qubit in the eigenstates of $\bar{X}$ instead. Eigenstates corresponding to $\bar{X} = 1$ and $\bar{X} = -1$ in the coding space can be prepared by the dual procedure and starting with the state $[(|0\rangle + |1\rangle)/\sqrt{2}]^{\otimes n}$ and $[(|0\rangle - |1\rangle)/\sqrt{2}]^{\otimes n}$ respectively.

6.3 Encoding of unknown states

The ability to encode an unknown quantum state is not necessary for quantum computation since we can initialize the computer by encoding known states and then execute a known quantum circuit.

However, if we are presented with an unknown state for storage, we need to be able to encode the state before placing it in quantum memory. [1] claimed that quantum error-correcting codes are capable of protecting unknown coherent quantum states.

6.4 Measurement of an encoded state

At the conclusion of a period of data storage or a quantum computation, We need to measure some qubits in order to read out the stored information or the result of a quantum computation. As an example, we shall describe how to measure the logical operator $\bar{Z}$ for the planar code which encodes one qubit, that is, to measure the encoded block in the basis $\{|\bar{0}\rangle, |\bar{1}\rangle\}$.

The naive approach would be to measure every qubit in the basis $\{|0\rangle, |1\rangle\}$, by which operation we also destroy the encoded block. Assuming that there are no errors in the code block at the time of measurement, and there are no measurement errors, then we can choose any homologically nontrivial path on the lattice and multiply together all the outcomes for the links along that path. If the result is 1, then the encoded block is in the state $|\bar{0}\rangle$; if the result is -1, then the encoded state is $|\bar{1}\rangle$.

However, the real situation is not as ideal as that we assumed. There will be errors in the code block and the measurements of individual qubits are not perfect. The procedure for measurement outlined above is very sensitive to such errors since a single bit flip along the path could would alter the parity of the measurement outcome. Therefore, we need a fault-tolerant procedure to measure the encoded qubit. A fault-tolerant procedure is described in [1] where we first correct the bit flip errors by measuring all the plaquette operators and flipping bits along the identified error chain (both qubit error and measurement error). This task is manageable if we assume that the measurement errors are rare and uncorrelated, in which case they can be identified by looking at the world line (path in the space-time lattice) of the defect. After the error correction, we can perform the above procedure to determine the parity of Z measurements along a homologically nontrivial cycle.

7 Error modeling and estimate of accuracy threshold

planar has value close to that for toric

In [1], Dennis et al. related error recovery to statistical-mechanical models with local interactions. In the unrealistic case where syndrome measurements are perfect, they described error recovery by the two-dimensional Ising model with quenched disorder, whose phase diagram has been studied using Monte Carlo simulations. In the more realistic case where we take into account of the imperfections in syndrome measurements, error recovery can be modeled using a three-dimensional Z_2 gauge theory with quenched disorder. The third dimension which can be interpreted as time emerges because we must repeat the measurement many times before the correct way to recover from the errors can be concluded upon.

[1] also estimated the accuracy threshold to be $p_c \gtrsim 1.7 \times 10^{-4}$ for quantum storage using toric codes by identifying the threshold with an order-disorder phase transition in the statistical-mechanical models. Although the estimate is made based on toric codes, the value is believed to be similar for a planar code.

The estimate applies to a device with strictly local quantum gates which are controlled by a perfectly reliable classical computer with clock speed much

faster than the clock speed of its quantum counterpart. Nonlocal classical processing is still permitted in this approach. Specifically, nonlocal classical processing is required for the inference of quantum gate operations needed for error recovery based on the scattered error syndrome measurements. Furthermore, it is assumed that a classical computation, with a computation time bounded above by a polynomial in the number of stored qubits, can be executed in a single time step.

Under the same set of assumptions, a nonzero value of the accuracy threshold also exists for quantum computation. Furthermore, the value is speculated to be similar to that obtained for quantum storage [1].

8 Conclusion

Today, we attempt to develop theories of quantum computation and of quantum error correction which hopefully will be applicable on the future quantum computers. This should be done with the probable specifications for the future hardware in mind.

Due to physical limitations, it is difficult to control interactions between qubits which are separated by a large distance. Therefore, the quantum gates which can be executed with high fidelity in the foreseeable future are likely to be local gates. We would also imagine that future quantum computers will include some kind of classical processors, and that classical operations can be performed much more accurately and much more rapidly than the quantum operations.

Motivated by the future prospects of quantum computers, we have adopted the computational model in which all quantum gates are local, and in which classical computations of polynomial size can be carried out instantaneously with perfect accuracy. Furthermore, it is assumed that the measurement of a qubit can be performed within the execution time of a quantum gate.

The locality requirement of quantum computations has motivated the study of topological quantum error-correcting codes where all quantum operations needed to extract the error syndrome are local. In this paper, the efficacy of quantum error correction using surface codes was investigated in the computational model described above. [1] showed that an accuracy threshold for quantum storage can be established for the two-dimensional surface codes introduced in [2, 3], and estimated its numerical value. A nonzero accuracy threshold for universal quantum computation also exists.

Our ability to control errors in a quantum system, which is naturally proned to errors, determines the viability of quantum information processing and quantum computation. Fortunately, topological codes provide a promising framework for correcting quantum errors through local quantum manipulations. Due to its nice locality properties, these codes are expected to play a major role in the future evolution of quantum technologies.

References

[1] E. Dennis, A. Kitaev, A. Landahl, J. Preskill, "Topological quantum memory," quant-ph/0110143 (2001).

[2] A. Yu. Kitaev, "Quantum error correction with imperfect gates ," in *Proceedings of the Third International Conference on Quantum Communi cation and Measurement*, ed. O. Hirota, A. S. Holevo, and C. M. Caves (New York, Plenum, 1997).

[3] A. Yu. Kitaev, "Fault-tolerant quantum computation by anyons," quant-ph/9707021 (1997).

[4] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, "Quantum error correction and orthogonal geometry," Phys. Rev. Lett. 78, 405 (1997), quant-ph/9605005.

[5] D. Gottesman, "A class of quantum error-correcting codes saturating the quantum Hamming bound," Phys. Rev. A 54, 1862 (1996), quant-ph/9604038.

[6] D. Gottesman, "An introduction to quantum error correction," quant-ph/0004072 (2000).

[7] E Knill, R. Laflamme, W. H. Zurek, "Resilient quantum computation: error models and thresholds," Proc. Roy. Soc. London, Ser. A 454, 365 (1998), quant-ph/9702058.

[8] D. Aharonov and M. Ben-Or, "Fault-tolerant quantum computation with constant error," *Proc. 29th Ann. ACM Symp. on Theory of Computing*, p. 176 (ACM, New York, 1998), quant-ph/9611025; D. Aharonov and M. Ben-Or, "Fault-tolerant quantum computation with constant error rate," quant-ph/9906129 (1999).

[9] A. Yu. Kitaev, "Quantum computations: algorithms and error correction," Russian Math. Surveys **52**, 1191-1249 (1997).

[10] J. Preskill, "Reliable quantum computers," Proc. Roy. Soc. Lond. A **454**, 385-410 (1998), quant-ph/9705031.

[11] D. Gottesman, "Stabilizer codes and quantum error correction," Caltech Ph.D. thesis (1997), quant-ph/9705052.

[12] D. Gottesman, "Fault tolerant quantum computation with local gates," quant-ph/9903099 (1999).