CMSC 657 Final Report

Eric Kubischta & Ian Teixeira
erickub@math.umd.edu, ianteixeira1@gmail.com

Due: 11:59 PM, December 13, 2018

Part I

Topic. To recall, our topic is on **toric codes**, a type of topological quantum error correcting code. We chose this topic due to its relationship with *topological* quantum computing, which both of us find interesting especially due to our mathematical backgrounds.

Summary of relevant literature.

- *Nielsen and Chuang's Quantum Computation and Quantum Information* offers a very good introduction to Error Correcting codes and Stabilizer theory.
- *Stabilizer Codes and Quantum Error Correction* is the thesis of the well known Daniel Gottesman. In this work he describes in very rich detail the formalism of stabilizer codes, which we will use as our basis for understanding toric codes. In particular, Chapter 3 is a well written introduction to stabilizer codes.
- *Topological Quantum Computation* by Zhenghan Wang, a mathematician working on Topological Quantum computing at Microsoft Station Q, has a fully mathematically rigorous definition of a Toric code given in section 8.1.1. He begins with a more general definition of quantum doubles, but immediately gives the definition of toric codes as an example.
- *Introduction to Topological Quantum Computation* by Jiannis K. Pachos is a detailed resource on toric codes, specifically section 5.2.1. It is very clear, detailed, and mathematical, addressing in depth the relation between the topology and quantum error correction.
- *A pedagogical overview about the 2D and 3D Toric Codes and the origin of their topological orders* by M. F. Araujo de Resende gives the clearest exposition on toric codes of all the resources listed.

Part II

To understand the Toric code, we find it instructive to first understand the idea behind error correcting codes and their generalization to stabilizer codes, since the Toric code is an example of each of these objects.

Error Correcting Codes. Suppose we have in our possession some quantum information, that is, we have some quantum state $|\psi\rangle$. We would like to *preserve* our information through time or space. What we mean by this is that if we transport $|\psi\rangle$ across space or if we leave $|\psi\rangle$ fixed in place but wait a long time, we would like $|\psi\rangle$ to be in the exact same state as when we started. The reason $|\psi\rangle$ may not be in the exact same state is due to an effect called "noise". Noise is basically any physical perturbation affecting our state over the course of time or space and can be classical or quantum (such as decoherence). To model it, we use a quantum channel $\mathcal{T}$ applied to $|\psi\rangle$. In basic cases, we can think of $\mathcal{T}$ as a simple tensor product of unitary operations applied independently to input qubits.

To protect our state, that is, to make sure that $\mathcal{T}|\psi\rangle = |\psi\rangle$, we employ the use of an error correcting code. The idea behind such a code is to add enough redundancy to our state so that when it passes through the channel, the output state is unequivocally *close* to the original state. In more detail, we *encode* our original state $|\psi\rangle$ into some pre-channel state $|\psi'\rangle$ using some unitary encoding operating $\mathcal{E}$ that makes $|\psi'\rangle$ significantly more redundant. Then, $|\psi'\rangle$ passes through the channel $\mathcal{T}$ to make an output state $\mathcal{T}|\psi'\rangle = |\psi''\rangle$. Lastly, we *decode* $|\psi''\rangle$ into an estimate $|\psi'''\rangle$ of $|\psi\rangle$ by suitable measurements and unitary operations. Assuming we added enough redundancy in the encoding step, $|\psi'''\rangle$ will equal $|\psi\rangle$ and we will have protected our information.

Stabilizer codes. The main idea of the stabilizer formalism can be illustrated with an example. Consider the Bell basis state $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Then $|\psi\rangle$ is a $+1$ eigenvector of both $X^{\otimes 2}$ and $Z^{\otimes 2}$. In other words, $|\psi\rangle$ remains unchanged after either of these two unitary operators is applied to $|\psi\rangle$. Moreover, up to a global phase, $|\psi\rangle$ is the *unique simultaneous* $+1$ eigenvector of $X^{\otimes 2}$ and $Z^{\otimes 2}$. Thus, there is a one-to-one correspondence between $|\psi\rangle$ and these two operators. In general, we can always find a correspondence between state vectors and the operators they are simultaneous $+1$ eigenvectors for. In this sense, we can work with either the state vectors or the corresponding operators, whichever is easy. What is perhaps surprising is that working with the operators can actually be easier than working with the state vector and it is all because we employ group theory.

Let X, Y and Z be the familiar Pauli operators (matrices) and I be the identity operator (matrix). Let $\Pi = \{I, X, Y, Z\}$ and define the Pauli group as

$$G_n = \{e^{i\frac{\pi}{2}m} M_1 \otimes \cdots \otimes M_n \,|\, M_j \in \Pi,\, m = 0, 1, 2, 3\}.$$

Basically the n-th Pauli group G_n is the n-fold tensor product of Pauli operators with the addition of the factors ± 1 and $\pm i$ so that it is indeed a group with respect to matrix multiplication.

Given n, consider a composite system of n qubits represented by a state vector $|\psi\rangle$, and let S be a subgroup of G_n. Define

$$V_S = \{|\psi\rangle \,:\, g\,|\psi\rangle = |\psi\rangle\,,\, \forall g \in S\},$$

that is, V_S is the space of simultaneous $+1$ eigenvectors of S or more intuitively, if $|\psi\rangle$ is in V_S, then if we apply *any* operator from S to $|\psi\rangle$, the effect is as if nothing was applied to $|\psi\rangle$.

It is clear that the zero vector is in V_S and that V_S is closed under linear combinations, thus, V_S is a vector subspace. We call V_S the *vector space stabilized by* S and we call S the *stabilizer* of V_S. To ensure that V_S is not a trivial vector space, it turns out that it is necessary and sufficient that S is abelian (everything in S commutes with everything else in S) and that $-I \notin S$.

To talk about S more succinctly, we can express S in terms of its *generators*. If $g_1, \ldots, g_m \in S$ and any element $g \in S$ can be written as a finite product of elements from $g_1, \ldots, g_m$, then we call $g_1, \ldots, g_m$ the *generators* of S; in this case we will write $S = \langle g_1, \ldots, g_m \rangle$. We say that generators $g_1, \ldots, g_m$ are *independent* if removing any of them makes the group they generator smaller.

With the previous in mind we have the following important theorem. Let n be the number of qubits in our composite system. If $g_1, \ldots, g_{n-k}$ are independent and commuting elements of G_n such that $S = \langle g_1, \ldots, g_{n-k} \rangle$ and $-I \notin S$, then V_S has dimension 2^k as a vector space.

Now we finally come to stabilizer codes. Fix n and consider a subgroup of G_n given as $S = \langle g_1, \ldots, g_{n-k} \rangle$ such that the generators are commuting and independent and $-I \notin S$. Then an $[n, k]$ *stabilizer code* is defined to be the vector space V_S (that is, the vector space stabilized by S). We denote such a code by $C(S)$ since the code is completely determined by the choice of S. Thus by the previous paragraph, the code $C(S)$ is just a specific 2^k vector subspace of the 2^n qubit vector space; it is noteworthy that this is a direct analog to linear codes from classical error correction coding theory. Intuitively, a $[n, k]$ stabilizer code encodes k information qubits to n physical qubits.

Toric Code Motivation. Recall that elements of the Pauli group G_n take the form

$$e^{i\frac{\pi}{2}m} M_1 \otimes \cdots \otimes M_n$$

for $m = 1, 2, 3, 4$ and all the $M_i \in \{I, X, Y, Z\}$. We say an element of G_n has length k if k of the M_i are non-identity Pauli matrices and the rest of the M_i are the identity. For example, the element

$$+iX \otimes Z \otimes I \otimes I \otimes I \otimes I \otimes I \otimes I$$

of G_8 has length 4. We call a subgroup $S \subset G_n$ a k-local stabilizer code if S is generated by operators of length k. This is worth defining because the Toric code turns out to be a 4-local code. That is, the stabilizer group for the Toric code is generated by length 4 elements of the Pauli Group G_n. To motivate the Toric code we can first consider three other natural 4-local codes on n qubits.

One 4-local code is the "A" code that has the stabilizer group S generated by all the

$$e^{i\frac{\pi}{2}m} M_1 \otimes \cdots \otimes M_n$$

such that exactly 4 of the M_i are the Pauli X matrix, and the rest of the M_i are the identity.

Another obvious 4 local code is the "B" code, which has a stabilizer group S generated by all the

$$e^{i\frac{\pi}{2}m} M_1 \otimes \cdots \otimes M_n$$

such that exactly 4 of the M_i are the Pauli Z matrix, and the rest of the M_i are the identity.

We could try to form a third 4 local stabilizer code ("AB" code) with stabilizer group S generated by all finite products of the elements of the A code with elements of the B code. However such an S is not commutative, since $XZ = -ZX$ so the codespace would be empty! It turns out, fortunately, that this code can be salvaged, however. The toric code sits inside the "AB" code as an abelian subgroup which has a natural topological interpretation.

Toric Code Definition. We create the toric code as follows. We embed a square lattice as an n edge graph Γ on a torus $\mathbb{T}^2$ as in Figure 1. We then assign a qubit to each of the n edges of the graph so that we have a $2n$ dimensional Hilbert space $L = \otimes_{\text{edges}} \mathbb{C}^2$.

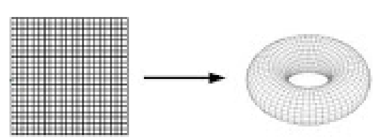


FIGURE 1

Consider Figure 2. For each vertex v of the graph Γ (i.e. a point on the original square lattice) define an operator A_v on $L = \otimes_{\text{edges}} \mathbb{C}^2$ as a tensor product of Pauli X matrices and identities such that A_v acts as X on the each of the four qubits corresponding to the four edges adjacent to v and otherwise A_v acts as the identity. Similarly, define a face operator B_f for each face f as an operator acting on L as a tensor product of Pauli Z and identities such that B_f acts as Z on each of the 4 qubits corresponding to the edges enclosing the face f, and otherwise acts as the identity.

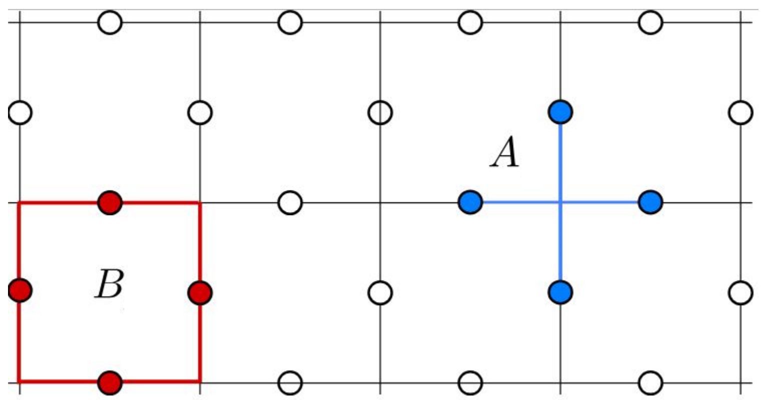


Figure 2

Now let S be generated by every vertex operator, A_v, and every face operator, B_f, that is,

$$S = \langle A_{v_1}, A_{v_2}, \cdots, B_{f_1}, B_{f_2}, \cdots \rangle.$$

Then S is definitely a subgroup of G_n. Furthermore, note that for any v, v' we have $A_v A_{v'} = A_{v'} A_v$ and for any f, f' we have $B_f B_{f'} = B_{f'} B_f$ since X and Z operators each commute with themselves respectively. But also note that if v is not a vertex adjacent to the face f then B_f and A_v act on completely different qubits, so they naturally commute. Otherwise, if v is a vertex adjacent to the face f, the operators overlap on exactly two edges. But since X and Z anti-commute, we have $(-1)^2 = 1$ so we still have that A_v and B_f commute. Hence, S is a subgroup generated by *commuting* generators.

The toric code places qubits at every edge on our lattice, hence $n = |\text{edges}|$. But for the torus, it is easy to check using the Euler characteristic that $|\text{faces}| = |\text{vertices}| = n/2$ so that there is actually $n/2$ A_v operators and $n/2$ B_f operators. Hence, S is a subgroup generated by $n/2 + n/2 = n$ commuting generators. However, they are not independent generators since we clearly have the relations

$$\prod_{v \in \{\text{vertices}\}} A_v = \prod_{f \in \{\text{faces}\}} B_f = I.$$

But two relations means we lose two degrees of freedom from the previous n commuting generators and we are left with some $n-2$ commuting independent generators.

Furthermore, it is clear that $-I \notin S$. Thus by the above result about stabilizer codes, the Toric code must have a code space V_S of size

$$2^{n-(n-k)} = 2^{n-(n-2)} = 2^2 = 4$$

and so we can conclude that the toric code is a $[n, 2]$ stabilizer code. Hence, we can encode 2 information qubits using the Toric code using n physical qubits placed on each of the n edges of the lattice embedded on the torus.

Essential Equations of the Toric Code. The Hamiltonian (the operator roughly corresponding to the total energy in the system) for the toric code is given by

$$H = -\sum_{v\in\{\text{vertices}\}} A_v - \sum_{f\in\{\text{faces}\}} B_f.$$

The *ground state* of the Hamiltonian is the eigenstate of H corresponding to the smallest eigenvalue. Since there are -1 factors in front of each sum in the Hamiltonian and the eigenvalues of each A_v and B_f are ± 1, the ground state is simply the state that is the simultaneous $+1$ eigenstate of *each* A_v and B_f. In fact, the eigenvalue of the ground state, that is, the ground energy, is given by

$$\begin{aligned} E_0 &= -(\#\text{ of vertex operators}) - (\#\text{ of face operators}) \\ &= -(\#\text{ of edges}) \\ &= -(\#\text{ of qubits}) \\ &= -n. \end{aligned}$$

A ground state of the Hamiltonian is given by

$$|\zeta\rangle = \prod_v \frac{1}{\sqrt{2}}(I + A_v)|00\cdots 0\rangle.$$

We note that the ground state $|\zeta\rangle$ lies in the code space since again it is a simultaneous $+1$ eigenvector for all the $A(v)$ and all the $B(p)$. For this reason, it is only natural to let $|\zeta\rangle$ be one of the four basis states of our Toric code vector space as the "resting" state of our n physical qubits is $|\zeta\rangle$. What is surprising now is that this actually isn't the *only* ground state!

Suppose the qubits on the lattice embedded on the torus are in the ground state $|\zeta\rangle$. Then suppose we apply a Z operator to the qubit on edge α. This raises the energy of the system to $E_0 + 4$ and thus is referred to as an excitation. Measuring the two A_v operators on the two vertices adjacent to α yields a -1 measurement each. When this happens we say an e-anyon has been created at each of these two vertices. Note that the e-anyons' must come in pairs. Now suppose we apply a Z operator to consecutive qubits on the lattice in the form of a path such that the path is non-closed. This again raises the energy of the system to $E_0 + 4$ and creates an e-anyon pair at the boundary of such a path. However, if we consider a closed path, it turns out that the energy of the system is not raised and no e-anyon pairs are created. For this reason, we would suspect that such a scenario corresponds to a new ground state, however, careful analysis reveals that this is not able to change the ground state since Z operators are not able to flip $|0\rangle$'s and $|1\rangle$'s.

Now what if we do the exact same analysis except with the X operator in place of the Z operator on the lattice? Since some X loops fail to commute with some B_f, this will take $|\zeta\rangle$ out of the code space entirely. However, we can define a dual-lattice Γ^*. We define the dual lattice as a new lattice that has lattice vertices as faces and lattice faces as vertices. Here, if we apply the X operator on a path as before we really will get a new ground state! In fact, we get a new ground state whenever we the paths are distinct.

Path and Loop Operators: Storing Information In The Toric Code. What is perhaps surprising is that each of the basis vectors of V_S corresponds to a certain distinct *loop* of the torus. In this way, the loops are what are storing the logical information. Since we have designed the toric code in such a fashion, the basis vectors are actually the degenerate ground states of the quantum system so these basis states are incredibly stable. Hence, each loop corresponds to a ground state.

The Toric code can store 4 different logical quantum states:

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle.$$

Starting from a Torus with a spin lattice in the state

$$|00\ldots 0\rangle$$

we can apply the operator

$$\prod_{v \in \text{vertices}} \frac{1}{\sqrt{2}}(I + A_v)$$

to store the state $|00\rangle$. Let γ be a set of qubits that a closed loop around the Torus on the dual lattice. That is the dual edges that intersect the edges on which the qubits are placed actually are connected and form a closed loop around the dual lattice. Then let O^x_γ be the element of the Pauli group that acts as the X operator on every qubit in γ and acts as the identity on every other qubit. We call such an operator a path operator. It turns out that for any paths γ, γ' that wrap around the Torus vertically we have

$$O^x_\gamma|\zeta\rangle = O^x_{\gamma'}|\zeta\rangle$$

Similarly, if β and β' are any two paths that wrap around the torus horizontally we have

$$O^x_\beta|\zeta\rangle = O^x_{\beta'}|\zeta\rangle$$

In the Toric code we store logical qubits by

$$\begin{aligned} |00\rangle &\equiv |\zeta\rangle \\ |01\rangle &\equiv O^x_\gamma|\zeta\rangle \\ |10\rangle &\equiv O^x_\beta|\zeta\rangle \\ |11\rangle &\equiv O^x_\beta O^x_\gamma|\zeta\rangle \end{aligned}$$

where on the left hand side the kets represent logical quantum states and on the right hand side the states are of the noisy, physical, quantum spin lattice. To summarize, the four logical states are precisely the four elements of V_S.

Broken Loops and Anyons: Detecting and Correcting Errors in the Toric Code. Suppose we are using the Toric code as a quantum memory for example to store the state

$$|01\rangle \equiv O^x_\gamma|\zeta\rangle.$$

What happens if some qubit on the vertical loop γ experiences a bit flip error? Can we detect such an error? Yes we can! Imagine a bit flip error as deleting a edge in the loop γ. Now what remains is actually a path with two endpoints and the Toric code system is still in an eigenstate. We can measure every A_v and B_f in S and read out their eigenvalues. For any face f that is adjacent to the flipped qubit we will read out the eigenvalue

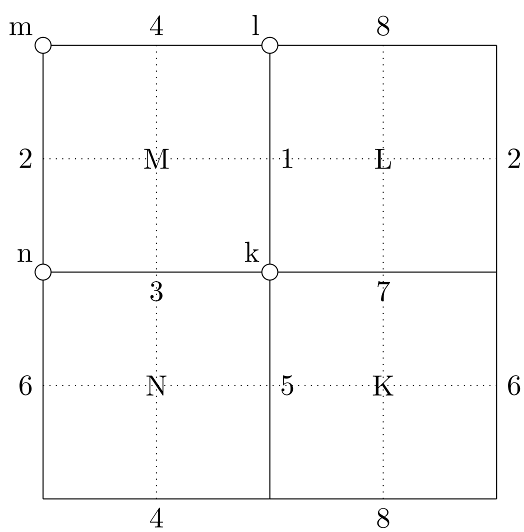
$$(1)(1)(1)(-1) = -1 \neq 1.$$

All four states in the code space have eigenvalue 1 for all B_f, so we know something is wrong! Similarly, for any face adjacent to k edges that have experienced a bit flip error we will read out an eigenvalue of

$$(-1)^k$$

so whenever k is even we detect nothing, but when k is odd we see an error. And we call this error an "anyon" as previously mentioned. One bit flip produces two anyons, and two errors on consecutive (dual) edges produce a path (on the dual lattice), the endpoints of which show up as eigenvalues of -1.

Toric Code 2 × 2 Example.



For this figure, the ground state is (up to a suitable normalization):

$$|\zeta\rangle = (I + A_k)(I + A_\ell)(I + A_m)(I + A_n)|00000000\rangle.$$

A naive binomial expansion of this operator product is

$$I + A_k + A_\ell + A_m + A_n + A_kA_\ell + A_kA_m + A_kA_n + A_mA_n + A_\ell A_n + A_\ell A_m$$
$$+ A_\ell A_m A_n + A_k A_m A_n + A_k A_\ell A_n + A_k A_\ell A_m + A_k A_\ell A_m A_n.$$

However the product over all vertices of the A operators will hit each qubit with X exactly twice, thus since $X^2 = I$, we have that the product of the A's over all the vertices is just the identity. In our 2×2 example this relation is written explicitly as

$$A_k A_\ell A_m A_n = I.$$

Since $A^2 = I$, this implies many other relations; namely the four of the form,

$$A_\ell A_m A_n = A_k A_k A_\ell A_m A_n = A_k,$$

and the three relations of the form

$$A_m A_n = A_k A_\ell A_k A_\ell A_m A_n = A_k A_\ell.$$

Thus we can simplify our initial expansion of the operator product to

$$2(1 + A_k + A_\ell + A_m + A_n + A_k A_\ell + A_k A_m + A_k A_n).$$

The expression is in fact fixed by any multiplication A_i (the constant and singe A_i term in the sum will be switched and then the three remaining single terms will switch with the last three terms). So $|\zeta\rangle$ is clearly fixed by any A_i. The fact that it is fixed by any B is exactly the the logic from our original midterm report; the B commutes with all the I and A in the original product and so when it hits the all 0s ket it just dos nothing because Z does nothing to 0 state. So this particular ground state in our 2×2 example is proportional to

$$\begin{aligned} |\zeta\rangle = &|00000000\rangle + |01010101\rangle + |10011001\rangle + |01100110\rangle \\ &+ |10101010\rangle + |11001100\rangle + |0011001100\rangle + |11111111\rangle. \end{aligned}$$

However, it is crucial to note that there are other ground states as previously mentioned. In fact there are precisely 3 other ground states. Recall that a ground state is a state that is fixed by every vertex and every face operator. Furthermore recall that we denote the group generated by these operators as S. Define the normalizer of S in G_n to be

$$N_{G_n}(S) = \{n \in G_n : n^{-1} S n = S\}$$

then for any $n \in N_{G_n}(S)$ consider the state

$$n|\zeta\rangle$$

then for any $s \in S$ we have

$$\begin{aligned} sn|\zeta\rangle &= nn^{-1}sn|\zeta\rangle \\ &= ns'|\zeta\rangle \\ &= n|\zeta\rangle \end{aligned}$$

where s' is just some other element of S by definition of n as an element of the normalizer of S. So in fact

$$n|\zeta\rangle$$

is a simultaneous $+1$ eigenvector for all of S. That is we have found another element of the code space! Thus is very exciting. Let's apply this insight to our 2×2 example. What are some plausible elements of the Pauli group that could normalize S. It is in fact the case that the normalizer of a stabilizer code is actually equal to its centralizer. So we are really looking for elements that commute with S. If we take an operator composed solely of X and I that is a good start, because it will immediately commute with all the A_v. However, to ensure that an X type operator commutes with every B_f we must somehow guarantee that our operator acts on an even number of qubits around every face. So certainly an element of the pauli group with a single X, like

$$XIIIIIII$$

will not work. In fact if we pick any path of edges on the lattice and just act X on the qubits sitting on those edges then we encounter this same problem. For example

$$X_2X_6, X_1X_5, X_4X_2X_3X_5X_4$$

all have this issue. It is essentially a problem of tangency. If there exists a face such that the path of edges touches the boundary of the face exactly once then we get out a factor of -1 and the path operator will not commute with that face operator (touching a face three times will also bring out a factor of -1 but that can just be seen as a straight path that touches once and then is deformed by the action of a face operator). Since the Z type operators B_f live on vertices of the dual lattice, and a path on the dual lattice always enters and exits the vertex, thus touching eexactly twice to get $(-1)^2 = 1$, we should instead consider paths on the dual lattice when looking to find elements of the normalizer. More than that, we must have loops! Why loops? Because only loops on the dual lattice (closed paths) are guarenteed to touch an even number of adjacent edges to each vertex on the dual lattice. Some examples of loops on the dual lattice, for our 2×2 example, are

$$X_3X_4, X_7X_8, X_4X_1X_7X_5$$

are all X type operators that just act on the qubits sitting on the edges corresponding to a path on the dual lattice. They are in fact all vertical paths. And moreover

$$X_3X_4|\zeta\rangle = X_7X_8|\zeta\rangle = X_4X_1X_7X_5|\zeta\rangle$$

This a particular case of the general fact that for any X type vertical path operator on the dual lattice we get the same element of code space. Namely

$$\begin{aligned}|\zeta_1\rangle &:= |00110000\rangle + |01100101\rangle + |10101001\rangle + |01010110\rangle \\ &+ |10011010\rangle + |11111100\rangle + |0000001100\rangle + |11001111\rangle.\end{aligned}$$

By a similar logic, the path operator for a closed horizontal loop of Xs also commutes with all the A_v and B_p. This operator can be acted on $|\zeta\rangle$ to obtain a new element of the code space

$$\begin{aligned}|\zeta_2\rangle &= |11000000\rangle + |10010101\rangle + |01011001\rangle + |10100110\rangle \\ &+ |01101010\rangle + |00001100\rangle + |1111001100\rangle + |00111111\rangle.\end{aligned}$$

The fourth and final element of the code space in our example can be obtained by acting with both the the X path operator for both a horizontal loop and a vertical loop, yielding

$$\begin{aligned}|\zeta_3\rangle &:= |11110000\rangle + |10100101\rangle + |01101001\rangle + |10010110\rangle \\ &+ |01011010\rangle + |00111100\rangle + |1100001100\rangle + |00001111\rangle.\end{aligned}$$

The bottom line is that open paths create anyons. That is, acting with an open path operator will take $|\zeta\rangle$ out of the code space, so that is useless. So we want to act with closed path (loop) operators. It turns out:

(1) The path operator for any Z loop on the lattice just maps $|\zeta\rangle$ to itself– it does not create a new element of codespace.
(2) The path operator for any Z loop on the dual lattice will fail to commute with some A_v– thus the operator take $|\zeta\rangle$ out of the codespace.
(3) The path operator for any X loop on the lattice will fail to commute with some B_f– thus the operator takes $|\zeta\rangle$ out of the codespace.
(4) The path operator for any X loop on the dual lattice is *just right* and will give us the three other states in the code space that we are looking for.

Looking Forward. The Toric code is usually called a "toy model" and it is not particularly hard to see why. We usually define the *rate* of an $[n, k]$ stabilizer code by k/n, i.e. logical qubits per physical qubits. For the toric code on n physical qubits, the rate is given by $2/n$. But what we haven't said is that the percentage of errors the toric code can correct is only close to 1 when n is large. In particular, the percent of errors the toric code can correct goes to 1 in the limit as $n \to \infty$. But in this case, the rate goes to the limit of $2/n$ as $n \to \infty$ which is 0. Thus, the toric code is somewhat impractical. However, the toric code sheds insight into the fact that we can exploit the topological nature of surfaces to build error correcting codes where the number of distinct loops corresponds to the number of logical qubits that can be encoded onto the surface.

With this idea in mind it should not be too hard to believe that if we added another "handle" to the torus to make a double-torus, we could store 3 logical qubits and in fact for a general genus g surface (that is, a generalized torus with g holes), we could store $g + 1$ logical qubits. Clearly, this shows that the studying the toric code is not in vein as other surfaces with large enough genus could actually have a reasonable coding rate!

References

[1] M. F. Araujo de Resende. A pedagogical overview about the 2D and 3D Toric Codes and the origin of their topological orders. *arXiv e-prints*, page arXiv:1712.01258, December 2017.

[2] Daniel Gottesman. *Stabilizer codes and quantum error correction*. PhD thesis, California Institute of Technology, January 1997.

[3] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, Boston, MA, USA, 2002.

[4] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[5] Jiannis K. Pachos. *Introduction to Topological Quantum Computation*. 2012.

[6] Barbara Terhal. Topological order. `https://topocondmat.org/w12_manybody/topoorder.html`, Aug 2018.

[7] Z. Wang. *Topological Quantum Computation*. Conference Board of the Mathematical Sciences. CBMS regional conference series in mathematics. Conference Board of the Mathematical Sciences, 2010.

**Note that we would like this article not to be published.*