


# Chapter 1

## The Power of Small Toric Codes

### 1.1 Introduction to the Toric Code

The ability to correct and recover from errors is important for any quantum computing scheme. The discovery of the first error correcting quantum codes by Shor in 1995  vital for demonstrating that quantum computing was theoretically viable. The toric code is an error correcting code proposed by Kitaev [1, 2], arising from work using quantum mechanics to provide simple models of topological order. It is the simplest and most elegant in the class of such codes, collectively known as surface codes [3, 4].

One of the key advantages of surface codes is their local nature - the codes can be implemented on a two-dimensional lattice using only nearest neighbour interactions. In this realistic experimental scenario, surface codes can tolerate per-operation error rates of up to 1% [5, 6], compared with rates in of the order of  $10^{-4}$  for the early error correcting codes [7].

In this chapter we look towards an experimental realisation of the toric code and study the code for small values of  $n$ . We pre-compute a decoding library, an approach that quickly becomes intractible for values of  $n$  above those considered

here, but has the advantage that we obtain a provably optimum decoding success rate. We use our pre-computed decoder to assess the *encoding power* of the code: we say the code has positive encoding power if the encoded qubits exhibit a lower error rate than the same number of unencoded qubits subjected to the same noise.

## 1.2 Shor's Code Revisited

Before moving on to the toric code, we first look at a simple example of an error correcting code in order to introduce some of the notation and concepts that we use later. For this purpose we pick the first stage of Shor's code [8]. The full code uses nine qubits to protect against general errors; the code we analyse here uses only three qubits and protects only against a single error channel. We describe the code by specifying how to encode a general qubit:

$$\alpha |0\rangle + \beta |1\rangle \rightarrow \alpha |000\rangle + \beta |111\rangle. \quad (1.1)$$

The idea here is very simple: each logical qubit is encoded into three physical qubits. Shor constructed a circuit such that if there was a bit-flip error on any one of the three qubits, the state would automatically be corrected. The system basically implements a 'majority vote' error correction scheme.

We now analyse this system in a different way, which will directly translate to our treatment of the toric code later in the chapter. In order to do this we first introduce the standard qubit operators:

$$Z = |0\rangle \langle 0| - |1\rangle \langle 1| \quad (1.2)$$

$$X = |0\rangle \langle 1| + |1\rangle \langle 0| \quad (1.3)$$

$$Y = |0\rangle \langle 1| - |1\rangle \langle 0| = ZX. \quad (1.4)$$

| State         | $Z_1Z_2$ | $Z_2Z_3$ | $Z_1Z_3$ | $Z_1$ |
|---------------|----------|----------|----------|-------|
| $ 000\rangle$ | +1       | +1       | +1       | +1    |
| $ 001\rangle$ | +1       | -1       | -1       | +1    |
| $ 010\rangle$ | -1       | -1       | +1       | +1    |
| $ 100\rangle$ | -1       | +1       | -1       | -1    |
| $ 110\rangle$ | +1       | -1       | -1       | -1    |
| $ 101\rangle$ | -1       | -1       | +1       | -1    |
| $ 011\rangle$ | -1       | +1       | -1       | +1    |
| $ 111\rangle$ | +1       | +1       | +1       | -1    |

Figure 1.1: The eigenvalues of the operators on their simultaneous eigenstates

In what follows we will need to apply these operators to states of multiple qubits. To do this we will introduce subscripts to the operators, so that, for example,  $Z_1$  is the  $Z$  operator acting on the first qubit. Operators acting on different qubits can be seen to commute with one another.

We will start by considering a state of three qubits and the set of operators  $S = \{Z_1Z_2, Z_2Z_3, Z_1Z_3\}$ . As these operators commute with one another it is possible to find states that are simultaneous eigenstates of all three. For the set  $S$  that we have chosen we can take these to be the standard basis elements as shown in Fig. 1.2. The operators we have chosen are not actually independent as


$$(Z_1Z_2)(Z_2Z_3) = Z_1Z_3. \quad (1.5)$$

The eigenvalue of the third operator can be inferred from the other two. To break the degeneracy in the eigenstates we introduce a fourth operator  $Z_1$  that commutes with the elements of  $S$ .

We say that the set  $T = \{Z_1Z_2, Z_2Z_3, Z_1\}$  is an *independent* set of commuting operators. The eigenvalues of these operators uniquely define the basis states. We say that the set  $T$  *stabilises* the state  $|000\rangle$ , as  $t|000\rangle = |000\rangle$  for all  $t \in T$ . Similarly  $\{Z_1Z_2, -Z_2Z_3, Z_1\}$  stabilises  $|001\rangle$ . We call  $T$  the *stabiliser representation* of the

state  $|000\rangle$ ; by specifying three independent, binary-outcome operators in a Hilbert space of dimension  $2^3$  we have uniquely defined the state.

We can also use the stabiliser representation to define a subspace of the Hilbert space. The set  $\{Z_1Z_2, Z_2Z_3\}$  uniquely defines a subspace spanned by the states  $\{|000\rangle, |111\rangle\}$ , which is precisely the codespace defined in Eqn. (1.1). If we were to perform measurements corresponding to the operators  $Z_1Z_2$  and  $Z_2Z_3$  on our encoded state we would expect the outcome  $+1$  in both cases, no matter how many times the measurement was repeated. We can consider a measurement of the operator  $Z_1$  as a measurement of the encoded logical qubit.

Following Shor's approach  we model the introduction of errors as the  $X$  and  $Z$  operations acting on one or more of the qubits. To take a concrete example consider the error represented by  $X_1$  acting on the state:

$$X_1(\alpha|000\rangle + \beta|111\rangle) = \alpha|100\rangle + \beta|011\rangle. \quad (1.6)$$

If we evaluate the stabilisers now we find  $Z_1Z_2 \rightarrow -1$  and  $Z_2Z_3 \rightarrow +1$ . We can see from the stabiliser outcomes that we are no longer in the codespace and that an error has occurred. You will notice that the error was visible only in the outcome of the stabiliser that anti-commuted with the error operator:  $(Z_1Z_2)X_1 = -X_1(Z_1Z_2)$ .

Although we can tell that an error occurred, we do not know if it was  $X_1$  or  $X_2X_3$ . In Shor's scheme we correct the most likely. A different way of looking at this is that we can return to the codespace by applying  $X_1$ : if the error was actually  $X_1$ , the logical qubit will be intact, but if the error was  $X_2X_3$  the overall operation will be  $X_1X_2X_3$  and there will be an error on the logical qubit.

If we instead have a phase flip  $Z_1$  we will not be able to detect the error, as  $Z_1$  commutes with all the stabilisers and will go unseen. In Shor's original paper he fixes this by expanding the codespace.

We have introduced the stabiliser notation to describe a familiar code and have seen how by measuring stabiliser operators we can detect certain errors. We have an operator that completes the commuting set that defines an operation on the logical qubit and that we can detect errors provided they do not commute with all of the stabilisers. We will now use this language to introduce the toric code.

### 1.3 Definition of Toric Codespace

The toric  $2n$ -code (Fig. 1.2) uses  $2n^2$  physical qubits to encode two logical qubits. The codespace is most elegantly described using the stabiliser formalism. In particular we shall specify a set of  $2n^2 - 2$  independent stabilisers to give a codespace of dimension four.

To describe the code, it is useful to picture the  $2n^2$  physical qubits positioned on the odd diagonals of a  $2n \times 2n$  lattice (i.e. in the positions with coordinates  $(i, j)$ , where  $i + j$  is odd). On the sites  $(i, j)$  where  $i$  and  $j$  are both even we construct an  $X$ -stabiliser,  $s_k^X$ , by taking the product of  $X$  operators of qubits on the neighbouring squares. When considering what we mean by ‘neighbouring’ we identify opposite edges of the lattice, hence the ‘toric’ nature of the code. On the sites  $(i, j)$  where  $i$  and  $j$  are both odd, we define the  $Z$ -stabilisers,  $s_k^Z$ , in a similar manner.

We let  $S_X = \{s_k^X\}$ . There are  $n^2$  elements in  $S_X$ , but only  $n^2 - 1$  independent elements, as the final element is the product of all the others. We similarly define  $S_Z = \{s_k^Z\}$ . We will later use the sets  $S_X$  and  $S_Z$  to define the stabilisers of the codespace. Practically we will make measurements of these operators to check whether our state is still in the codespace.

Just as in Shor’s code, to fully define a state we will need to add additional commuting operators to our set of stabilisers. We now construct a set of operators that will be used for this purpose. We define  $X_h$  (horizontal X) to be the product

|         |         |         |         |
|---------|---------|---------|---------|
| $s_0^X$ | $q_0$   | $s_1^X$ | $q_1$   |
| $q_2$   | $s_0^Z$ | $q_3$   | $s_1^Z$ |
| $s_2^X$ | $q_4$   | $s_3^X$ | $q_5$   |
| $q_6$   | $s_2^Z$ | $q_7$   | $s_3^Z$ |

Figure 1.2: Qubit lattice for the 4-code.  $s_k^X$  and  $s_k^Z$  are  $X$ - and  $Z$ -stabilisers represented by the sites respectively.  $q_i$  are qubits. By way of example,  $s_0^X = X_0 X_2 X_4 X_6$  and  $s_0^Z = Z_1 Z_3 Z_5 Z_7$ .

of  $X_i$ s acting on qubits in an even row,  $X_v$  (vertical  $X$ ) the product of  $X_i$ s acting on qubits in an even column, and similarly,  $Z_h$  to be  $Z_i$ s on an odd row, and  $Z_v$  to be  $Z_i$ s on an odd column. Any of the pairs  $\{X_v, X_h\}$ ,  $\{X_v, Z_v\}$ ,  $\{Z_v, Z_h\}$ , and  $\{Z_h, X_h\}$  will suffice to fully define the state. The remaining pairs,  $\{X_v, Z_h\}$  and  $\{X_h, Z_v\}$ , satisfy the standard qubit commutation relation,  $[X, Z] = 2iY$  and so we can use this set of *logical operators*,  $L = \{X_v, X_h, Z_v, Z_h\}$ , to define a pair of independent *logical qubits*.

We will now turn to how errors are detected in the toric code. As with the treatment of the Shor code we will consider an error to be a set of individual  $X$  and  $Z$  errors applied to a subset of the physical qubits. An  $X$ -error on a single qubit will flip the eigenvalue of the neighbouring  $Z$ -stabilisers, but commutes with the  $X$ -stabilisers and will not be detected by them. Similarly, a  $Z$  error will be detected by the  $X$ -stabilisers and not the  $Z$ -stabilisers, and a  $Y$  will be detected by both sets. Additional errors will cause the stabiliser eigenvalues to toggle. If errors occur sequentially they will be detected by the stabilisers at either end of the chain

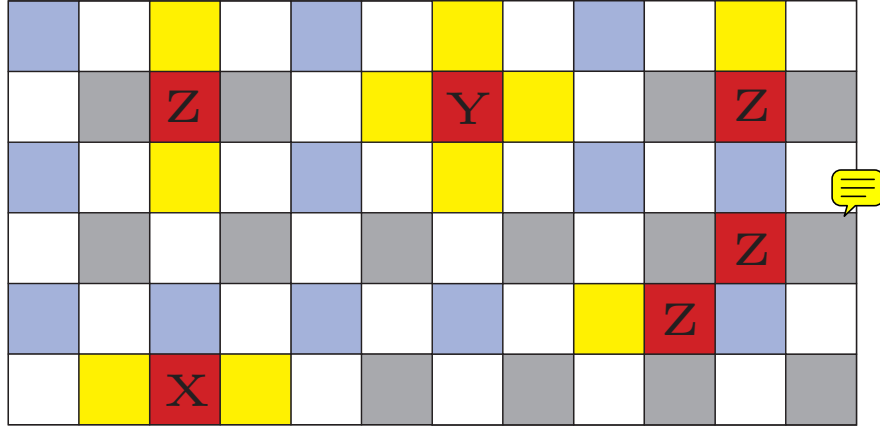


Figure 1.3: Error detection of isolated  $Z$ -,  $X$ -,  $Y$ -errors and a chain of  $Z$ -errors.

## 1.4 Code Variants


In the rest of the chapter, we will consider three variants of the toric code, that we refer to as the *classical toric code*, the *reduced quantum toric code* and the *full quantum toric code*.

### 1.4.1 Classical Toric Code

In the classical toric code we completely ignore the phase of each qubit, concerning ourselves only with the outcomes of measurements on the logical qubits in the  $Z$ -basis. To define the codespace we take our set of stabilisers  $S$  to be the set  $S_Z$  and the set of logical operators  $L = \{Z_h, Z_v\}$ . The codespace is massively degenerate but any state in it will be sufficient for our purposes.

We simplify the analysis by considering only  $X_i$  errors on the individual qubits, as these are the only ones that can affect the measurement outcomes we are interested in. We define the complete set of error operators,  $E$ , to be the set generated by the individual qubit  $X_i$  operators - that is the set of all products of such operators. As the  $2n^2$  individual qubit operators  $X_i$ , commute with one another and as  $X_i^2 = \mathbf{1}$

we can show that  $|E| = 2^{2n^2}$ .


We also introduce the set of *syndromes*,  $A$  - the possible outcomes when the code stabilisers  $S$  are measured. When each stabiliser measurement outcome is  $+1$  we say we have obtained the *zero-syndrome*, which we denote by the element  $a_0 \in A$ . If we obtain the zero-syndrome we know that we are in the codespace. In the classical toric code we have that  $|A| = 2^{n^2-1}$ , as each of the  $n^2$  elements of  $S$  can report  $\pm 1$ , but the last outcome is fixed by the other .

### 1.4.2 Full Quantum Toric Code

For the full quantum toric code, we aim to fully protect our encoded qubits against all sources of error. We take a full set of code stabilisers,  $S = S_X \cup S_Z$ , and the full set of logical operators  $L = \{Z_h, Z_v, X_h, X_v\}$ . The complete set of errors is generated by the set of all individual qubit operators  $\{X_i, Z_i\}$ , so that  $|E| = 4^{2n^2}$ .

Following similar reasoning to the classical toric code we can deduce the size of the set of syndromes,  $A$ : the outcomes of each of the  $2n^2 - 2$  independent stabilisers in  $S$  completely define a syndrome, so that  $|A| = 2^{2n^2-2}$ .

### 1.4.3 Reduced Quantum Toric Code

The reduced quantum toric code is slightly more subtle than the previous two cases. We take the full set of stabilisers  $S = S_X \cup S_Z$  and logical operators  $L = \{Z_h, Z_v, X_h, X_v\}$ , but aim only to protect against individual qubit  $Y_i$  errors. As previously, it is straightforward to show that  $|E| = 2^{2n^2}$ . It is more difficult to determine the size of  $A$  in this case, as not all syndromes are possible from the errors that we consider. It is possible to show that  $|A| = 2^{2n(n-1)}$ . .



## 1.5 The Structure of the Code

We now investigate the structure of the set of errors,  $E$ , and how they relate to the syndromes,  $A$ , and logical errors,  $L$ . We begin by defining the function

$$\text{synd} : E \rightarrow A \quad (1.7)$$

that maps an error onto its associated syndrome. We are then able to define


$$E_a = \{e \in E : \text{synd}(e) = a\}, \quad (1.8)$$

the set of all possible errors corresponding to a syndrome  $a$ . For convenience we define  $E_0 = E_{a_0}$  to be the set of errors corresponding to the zero-syndrome,  $a_0$ . If an error in  $E_0$  occurs it will go undetected by the code. We define a function to act on the set  $E_0$ ,

$$\text{lerr} : E_0 \rightarrow L, \quad (1.9)$$

taking an error to the corresponding logical error on the encoded qubits. Finally we define the set

$$E_{0,l} = \{e \in E_0 : \text{lerr}(e) = l\}. \quad (1.10)$$

It is easy to see that each  $E_{0,l}$  contains at least one element, as  $l$  itself  a member: by applying a set of errors corresponding to a logical operator, we apply this operator to the encoded qubits, inducing a logical error. We will now show that the sets  $E_{0,l}$  have exactly the same number of elements - our first important result about the structure of our toric codes.

To show that the  $E_{0,l}$  are all the same size, we will show that they all have the

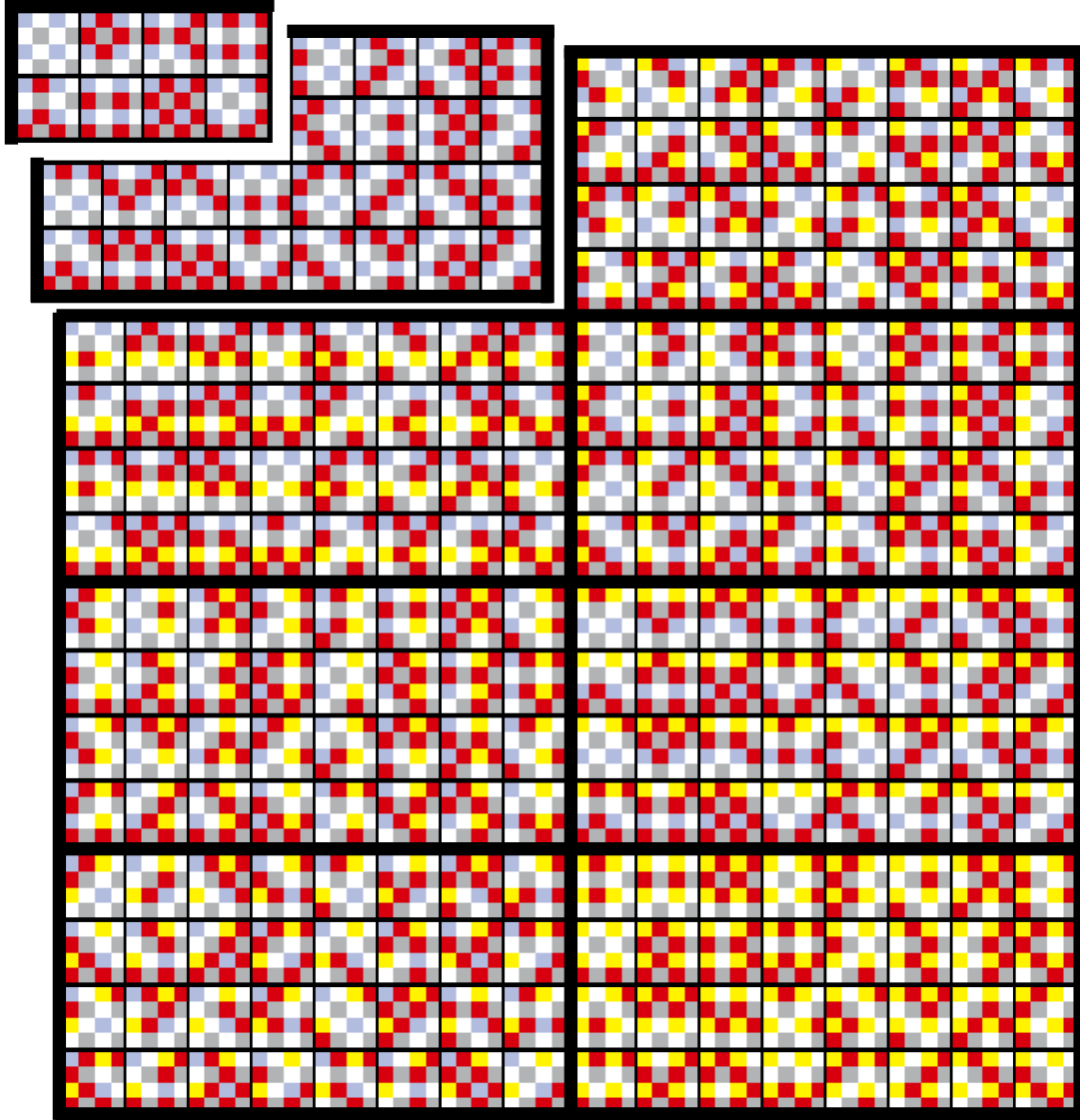


Figure 1.4: Full set of errors,  $E$ , for the classical 4-code.

same number of elements as  $E_{0,0}$  - the set of elements of  $E_0$  that introduce no logical error. To see this, take element  $e_1 \in E_{0,l}$  and  $e_2 \in E_0$ . We claim that

$$\text{⌋} \in E_{0,l} \Leftrightarrow e_1 e_2 \in E_{0,0}. \quad (1.11)$$

The proof of the claim follows from the fact that all logical operators  $l \in L$  are products of individual qubit operators, so that  $l^2 = \mathbf{1}$ : by applying the same logical

error twice we undo it. If by applying  $e_1$  after  $e_2$  we have no logical error, they must be in the same error class, and the converse is also true. We now consider the set

$$e_1 E_{0,0} = \{e_1 e : e \in E_{0,0}\} \quad (1.12)$$

and claim that, given  $e_1 \in E_{0,l}$ , we must have  $E_{0,l} = e_1 E_{0,0}$ . This follows as,

$$e_2 = e_1 e \Leftrightarrow e_1 e_2 = e, \quad (1.13)$$

where we have used that  $e_1^2 = \mathbf{1}$ . Therefore we can use  $e_1$  to establish a bijection between  $E_{0,l}$  and  $E_{0,0}$  and so the sets must have the same number of elements.

We now move on to the sets of errors,  $E_a$ , corresponding to a general syndrome,  $a \in A$ . As we have taken  $A$  to be the set of obtainable syndromes, it follows that there must be at least one element in each set  $E_a$ . It is useful to assume that we have chosen an arbitrary member from  $E_a$ . We call this a *matching* for the syndrome, and denote our specific choice of matching  $m^*(a)$ . In practice there are many ways to construct a specific matching for each syndrome  $a$ , for example see Fig. 1.5. If we observe the syndrome  $a$  and apply the matching  $m^*(a)$  we are guaranteed to be back in the codespace.

Following a similar procedure to before we can show that  $E_a = m^*(a)E_0$ , and therefore each of the sets  $E_a$  have the same number of elements. Furthermore we can split the set  $E_a$  up into a set of classes

$$E_{a,l} = m^*(a)E_{0,l}. \quad (1.14)$$

The set  $E_{a,l}$  can be interpreted as the set of errors that when corrected by  $m^*(a)$  induce the logical error  $l$  on the encoded qubits. All the sets  $E_{a,l}$  have the same number of elements as  $E_{0,0}$ .

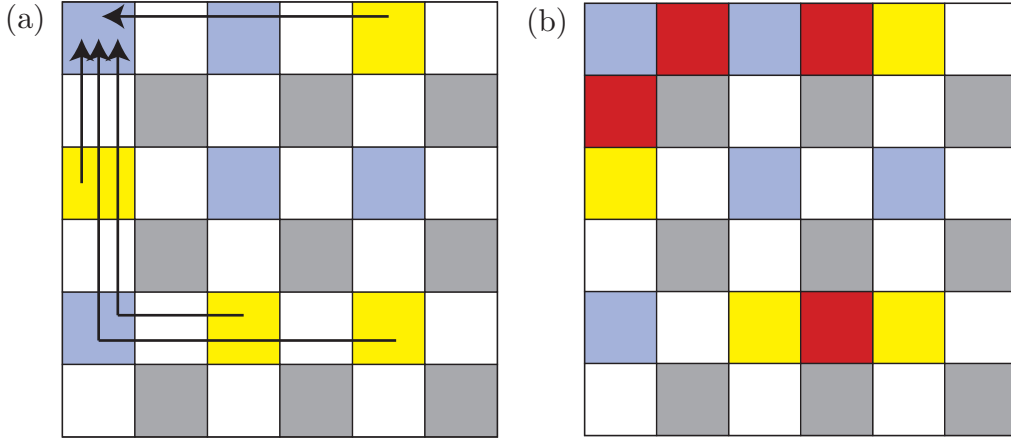


Figure 1.5: Constructing a specific matching for a classic code: the detected errors are first moved to the left, then to the top, of the lattice (a); we apply a  $Z$  operator to any qubit that has been passed over an odd number of times (b).

| Code            | $ E $      | $ A $         | $ L $ | $E_{0,0}$    |
|-----------------|------------|---------------|-------|--------------|
| Classical       | $2^{2n^2}$ | $2^{n^2-2}$   | 4     | $2^{n^2-2}$  |
| Reduced Quantum | $2^{2n^2}$ | $2^{2n(n-1)}$ | 4     | $2^{2n-2}$   |
| Full Quantum    | $4^{2n^2}$ | $2^{2n^2-2}$  | 16    | $2^{2n^2-2}$ |

Figure 1.6: Sizes of the sets involved with our three toric code variants.

In the preceding paragraphs we have split up the total set of errors  $E$  into identically sized subsets  $E_{a,l}$  indexed by  $a \in A$  and  $l \in L$ . It follows immediately that

$$|E| = |A| \cdot |L| \cdot |E_{0,0}|. \quad (1.15)$$

You can see that this relationship holds for our three code variations in Fig. 1.5.

We have shown that given a syndrome always corresponds to multiple logical error classes, of which we can correct only one. We still have to answer the question

of which one to correct. In the Shor code case we picked the matching to minimise the probability of introducing a logical error. Ideally that is what we would do here too. This of course depends on the error model, which we have not discussed yet. It is also not straightforward to do. The number of matchings in each class grows exponentially with the size of the grid. No efficient way of calculating the probabilities is known, even for the simplest error model.

The job of picking a matching given a syndrome falls to a decoder. These are often heuristic procedures, designed to balance calculation time with identifying the most likely class as often as possible. Determining a good decoding strategy is currently a very active area of research and there are a number of different approaches [9, 6, 10]. It is not only important that a decoder maximises the probability of returning to the codespace without inducing an error on the logical qubits, but also that the decoder can run in a reasonable amount of time.

## 1.6 Error Model

We consider depolarising noise distributed uniformly over the lattice with a given, constant probability:

$$D(\rho) = p_x X \rho X + p_z Z \rho Z + p_y Y \rho Y + (1 - p_x - p_y - p_z) \rho. \quad (1.16)$$

The values of  $p_x$ ,  $p_y$  and  $p_z$  vary over our different code variants accordingly

1. Classical case:  $p_x = p$ ,  $p_y = p_z = 0$
2. Reduced quantum case:  $p_y = p$ ,  $p_x = p_z = 0$
3. Full quantum case:  $p_x = p_y = p_z = p/3$ .

We also consider the effect of stabiliser sites mis-reporting the stabiliser measurement outcomes: that with some probability, the stabiliser reports that it fired (/did not fire) when it did not (/did). The normal approach here is to consider multiple faulty stabiliser evaluations as extending the code into a third dimension. By observing the stabiliser outcomes over multiple rounds it is possible to increase the chance of correctly identifying any mis-reported outcomes.

We just consider a single round of stabiliser evaluation, and so this extra information is not available to us. Instead we try to identify and correct the right syndrome based on the syndrome we see. Even in multi-round systems the final step will look like this and so our results are an upper bound on the possible decoding probabilities for small systems.

## 1.7 A Precomputed Decoder

We begin by looking at the probability of getting a particular logical error,  $l$ , given that syndrome,  $a$ , occurs and we attempt to correct it using our chosen matching method by applying  $m^*(a)$ :

$$p(l|a) = \sum_{e \in E_{a,l}} \frac{p(e)}{p(a)}. \quad (1.17)$$

For each stabiliser outcome  $a$  we identify the most likely logical error  $l_a$ , such that  $p(l_a|a)$  obtains a maximum. In cases where the most likely logical error is not unique we pick arbitrarily between the candidates. We may then correct our earlier attempt by applying the operator  $l_a$ , so that the total correction operation on the state will be  $l_a m^*(a)$ . The overall successful decoding probability from following this procedure

is

$$P_d = \sum_{a \in A} p(l_a|a)p(a) \quad (1.18)$$

$$= \sum_{a \in A} \max_{l \in L} \left\{ \sum_{e \in E_a} \frac{p(e)}{p(a)} \right\} p(a) \quad (1.19)$$

$$= \sum_{a \in A} \max_{l \in L} \left\{ \sum_{e \in E_a} p(e) \right\}. \quad (1.20)$$

When considering mis-reported stabiliser outcomes we must relax the restriction that there will be an even number of firing stabiliser sites. We let  $A'$  be the set of all possible stabiliser outcomes  $A' = \{0, 1\}^{\otimes n^2}$ . If we are to successfully decode a syndrome  $a'$ , we must first identify the correct syndrome  $a$ , on which to perform the arbitrary matching, and then identify the most likely logical error  $l_a$  to correct. If we identify the wrong  $a$ , the final state will not even be in the code space. Given  $a'$  we must pick  $a$  and  $l$  to provide the maximum probability of a successful decoding in the case of mis-reporting stabiliser outcomes:

$$P'_d = \sum_{a' \in A'} p(\text{success}|a')p(a') \quad (1.21)$$

$$= \sum_{a' \in A'} \max_{a \in A} \{p(l_a|a)p(a|a')\} p(a') \quad (1.22)$$

$$= \sum_{a' \in A'} \max_{a \in A} \left\{ \max_{l \in L} \left\{ \sum_{e \in E_a} \frac{p(e)}{p(a)} \right\} p(a \cap a') \right\} \quad (1.23)$$

$$= \sum_{a' \in A'} \max_{a \in A} \left\{ \max_{l \in L} \left\{ \sum_{e \in E_a} p(e) \right\} p(a'|a) \right\}. \quad (1.24)$$

While equations (1.20) and (1.24) are conceptually simple, calculations are beset by difficulties due to the rapid growth in the size of the sets  $E_{a,l}$ ,  $A$ , and  $A'$ : in the single noise channel case for the  $2n$ -code  $|E_{a,l}| = |A| = 2^{n^2-1}$ ; for the full noise channel  $2n$ -code  $|E_{a,l}| = |A| = 2^{2n^2-2}$ . For example, the total number of error configurations,

$|E|$ , for the 6-code is  $2^{36}$ , which would require  $\sim 200\text{GB}$  of storage space.

Thankfully, we do not need the complete set of error configurations to compute the value of  $p_{\text{success}}$  in Eq. (1.20). For each configuration  $e \in E$  the probability is

$$p(e) = (1 - p)^{2n^2 - n(e)} p^{n(e)} \quad (1.25)$$

where  $n(e)$  counts the number of errors in error configuration  $e \in E$ . When summing over the errors consistent with a given logical error and syndrome we get

$$\sum_{e \in E_{a,l}} p(e) = \sum_{e \in E_{a,l}} (1 - p)^{2n^2 - n(e)} p^{n(e)} \quad (1.26)$$

$$= (1 - p)^{n^2} \sum_{i=0}^{2n^2} d_{a,l}^{(i)} \left( \frac{p}{1 - p} \right)^i \quad (1.27)$$

$$=: (1 - p)^{n^2} \chi_{a,l} \left( \frac{p}{1 - p} \right) \quad (1.28)$$

where  $d_{a,l}^{(i)} = |\{e \in E_{a,l} : n(e) = i\}|$  and we have used the final line to define the characteristic function  $\chi_{a,l}$  of the class  $E_{a,l}$ . By computing and storing the coefficients  $d_{a,l}$  we are able to calculate the success probabilities for a range of values of  $p$ .

The full source code used for our calculations, along with the computed tables of characteristic function coefficients, is available online [11].

## 1.8 Simulation Results

### 1.8.1 Classical Toric Code

Using the method described, we calculated the success probabilities  $P_d(n, p)$ , and compared them with  $p_{\text{bare}} = (1 - p)^2$ , the probability that two non-encoded qubits would remain error-free (Fig. 1.7).



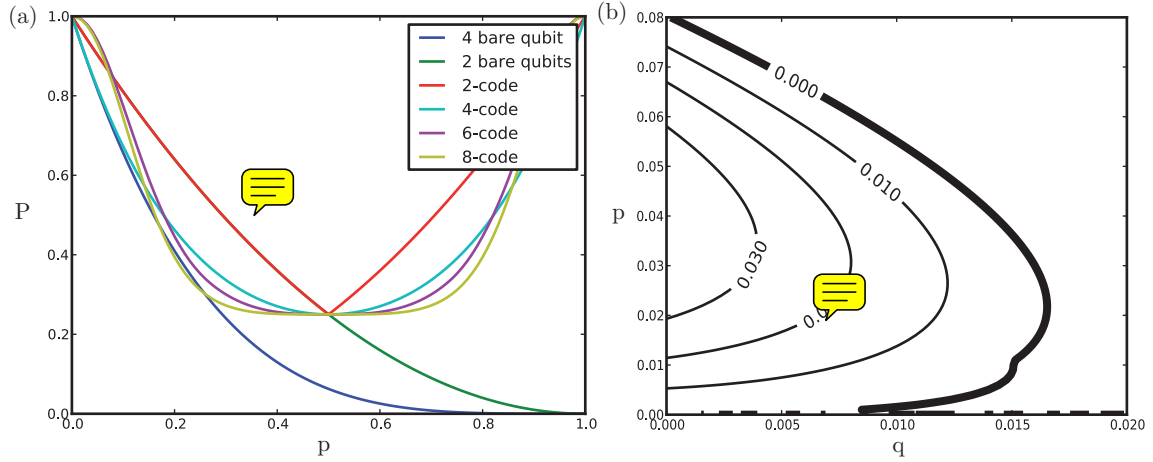


Figure 1.7: Success probabilities for classical toric code: (a) shows the success probability,  $P$ , for ideal version of the code vs. the error rate,  $p$ , with two and four unprotected qubits for comparison; (b) shows the power of the 6-code in the case when stabiliser outcomes are misreported with probability  $q$ .

The 2-code performs exactly the same as the bare qubits. This is not surprising, as the 2-code has 2 physical code qubits and no error detect capacity - there is only one possible syndrome. The 4-code performs worse than the bare qubits. Although the 4-code has error-detect capacity, it has no error correct capacity. This can be seen by considering the case when a single error occurs - due to the translational symmetry of the lattice, the syndrome can give us no information about whether our correction introduced a logical error or not. For the 4-code to be successful there must be no errors on the 4 qubits that  $X_h$  and  $X_v$  measure, an event with probability approximately equal to  $(1 - p)^4$  for small  $p$ .

The 6-code is the smallest code that exhibits encoding power for some values of  $p$ . Provided that  $p < 0.1$ , the 6-code outperforms the bare qubits. The 8-code appears to offer roughly the same performance as the 6-code. We weren't able to go beyond  $2n = 8$  with the computing resources available, but we expect that as  $n$  increases the curves would tend towards a step function at the one-channel threshold. We looked at the mis-reported stabiliser outcome case for the 6-code. There is a small

region with positive encoding power, that requires stabiliser fidelity in the region of 1%.

### 1.8.2 Reduced Quantum Code

For the reduced quantum code, we find that the 4-code is the first code to offer encoding power in this scenario (Fig. 1.8). As before the 2-code has no error detect capacity. Unlike before, the 4-code is able to detect and correct errors, as it can use both  $X$ - and  $Z$ -stabiliser information to break the symmetry and pinpoint the error.

We investigated the case of mis-reported stabiliser measurement outcomes for the 4-code. We obtain positive encoding power for mis-reporting rates of up to 10%. Overall the reduced quantum code shows remarkable encoding power, even for small codes. In using the full stabiliser information to detect a reduced set of errors, we have managed to hugely reduce the size of each error class (Fig. 1.5).

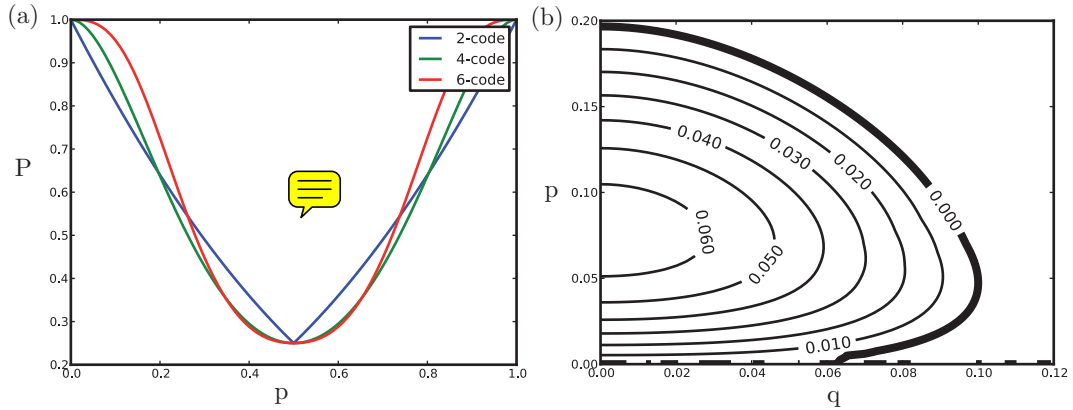


Figure 1.8: Success probabilities for reduced quantum toric code: (a) shows the success probability,  $P$ , for ideal version of the code vs. the error rate,  $p$ , with two and four unprotected qubits for comparison; (b) shows the power of the 4-code in the case when stabiliser outcomes are misreported with probability  $q$ .

The circumstances that could justify the assumptions of the reduced quantum code are not unrealistic: we envisage a system with a single dominant error channel.

In picking this to be the  $Y$  channel, we exploit the lack of symmetry in our code effectively ~~taylor~~<sup>tailor</sup>ing our code to be effective against the dominant error channel.

### 1.8.3 Full Quantum Code

Finally we look at the full code, protecting against full depolarizing noise, taking  $p_x = p_y = p_z$  in equation (1.16).

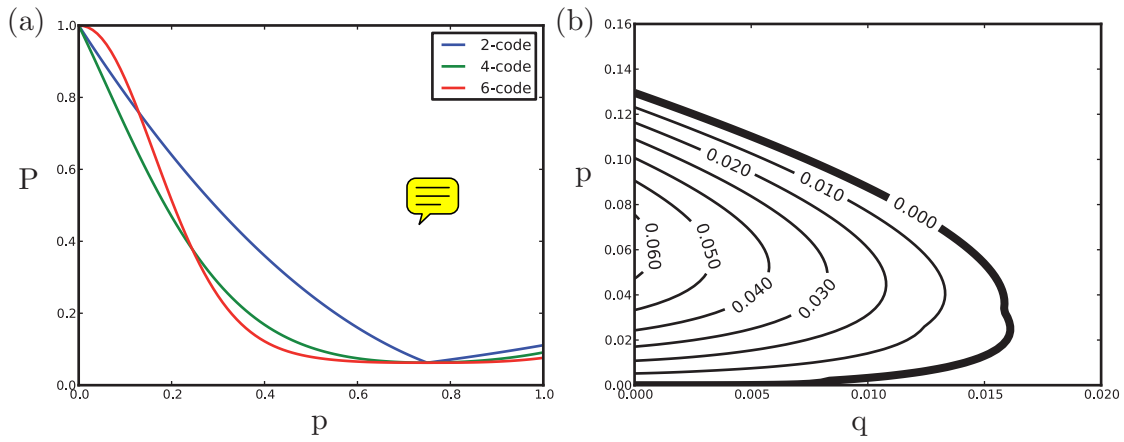


Figure 1.9: Success probabilities for full quantum code: (a) shows the success probability,  $P$ , for ideal version of the code vs. the error rate,  $p$ , with two and four unprotected qubits for comparison. Note that the graph has a minimum at  $p = 0.75$  when the probability of each error is equal to the probability of no error. (b) shows the power of the 6-code in the case when stabiliser outcomes are misreported with probability  $q$ .

The 6-code is the first code to offer encoding power (Fig. 1.9). We looked at mis-reported stabiliser outcomes for the 6-code. Due to computational constraints we were only able to find a lower bound for  $P'_d(p, q)$ . This was obtained by modifying equation (1.24) to maximise only over  $a$  close to  $a'$ :

$$p(\text{success}) = \sum_{a' \in A'} \max_{a \in N(a', x)} \left\{ \max_{l \in L} \left\{ \sum_{e \in E_{a, l}} p(e) \right\} p(a'|a) \right\} \quad (1.29)$$

where  $N(a', x) = \{a \in A : d(a', a) \leq x\}$  with  $d(a', a)$  the number of stabilisers where

$a'$  differs from  $a$ . We took  $x = 2$  to produce a rough region, and then used  $x = 4$  to refine the boundary.

## 1.9 An Experimental Suggestion

Our aim in this section is to provide a criterion which an experimentalist can use to verify whether a given candidate toric code set-up is providing protection.


Our criterion is based on the observation that the 2-code is essentially equivalent to two unprotected qubits: the syndrome outcome is always  $+1$  so we are unable to even detect errors and the logical operations reduce to single qubit operations. By comparing the performance of the candidate code system against that of the 2-code we are able to say whether the candidate system is exhibiting protective power.


We do not specify the decoder that the experimentalist must use for the larger code (decoding is trivial in the 2-code), but our precomputed decoder will give optimal results. In our analysis we assumed that the stabiliser mis-reporting and physical qubit errors are independent events. A round of stabilisers is permitted to introduce physical qubit errors, but these must be sprinkled evenly over the lattice and not be correlated with mis-reporting sites. If strong correlations were to exist a modified decoder would give a better chance of satisfying the criterion.

We assume that an experimentalist has the ability to perform all  $X$ - and  $Z$ -stabiliser measurement operations and the logical  $X$  measurements. In a large code the logical operations are potentially tricky, given their non-local nature. Here, due to the size of the codes considered, the logical operations actually involve fewer qubits than the stabilisers, and so are likely to be less technically demanding.


Our experimental proposal is as follows:

1. Measure  $X_v$ ,  $X_h$ , and the stabilisers to find initial syndrome  $a'_i$  and logical qubit states  $(x_i^v, x_i^h)$

2. Wait. Manually introduce noise if required.
3. Measure  $X_v$ ,  $X_h$ , and the stabilisers to find final syndrome  $a'_f$  and logical qubit states  $(x_f^v, x_f^h)$
4. Decode the calculated syndrome,  $a' = a'_i \text{ XOR } a'_f$ , to find matching  $m$  
5. Modify  $(x_f^v, x_f^h)$  to reflect what the outcomes would have been if we had applied  $m$  before measurement to find  $(x_m^v, x_m^h)$
6. If  $(x_i^v, x_i^h) = (x_m^v, x_m^h)$  count the round as a success; if not, count the round as a failure.
7. Repeat steps 1 to 6 many times to calculate an experimental successful decoding probability  $P_d^{\text{expt}}$

We then repeat this procedure for the 2-code and compare the results: if the larger system outperforms the 2-code protection is provided. 

## 1.10 Conclusion

We have provided a protocol and pre-computed decoder for demonstrating toric encoding size at minimal scale. The  $2n$ -code requires a minimum of  $2n^2$  physical code qubits, plus an auxiliary qubit with which we must be able to perform CNOT operations with any of the code qubits. For the 4-code this is a requirement of 9 qubits and for the 6-code the requirement is 19 qubits. The error rates provided are not unrealistic for current experimental systems. 



# Bibliography

- [1] Kitaev, A. (2003) Fault-tolerant quantum computation by anyons. *Annals of Physics*, **303**, 2–30.
- [2] Kitaev, A. Y. (1997) Quantum computations: algorithms and error correction. *Russian Math. Surveys*, **52**, 1191–1249.
- [3] Bravyi, S. B. and Kitaev, A. Y. (1998) Quantum codes on a lattice with boundary. *arXiv:quant-ph/9811052*.
- [4] Freedman, M. H. and Meyer, D. A. (2001) Projective plane and planar quantum codes. *Foundations of Computational Mathematics*, **1**, 325–332.
- [5] Wang, D. S., Fowler, A. G., and Hollenberg, L. C. L. (2011) Surface code quantum computing with error rates over  $10^{-20}$ .
- [6] Fowler, A. G., Whiteside, A. C., and Hollenberg, L. C. L. (2012) Towards practical classical processing for the surface code. *Phys. Rev. Lett.*, **108**, 180501.
- [7] Svore, K. M., Divincenzo, D. P., and Terhal, B. M. (2007) Noise threshold for a fault-tolerant two-dimensional lattice architecture. *Quantum Info. Comput.*, **7**, 297–318.
- [8] Shor, P. W. (1995) Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, **52**, 2493–2496.
- [9] Duclos-Cianci, G. and Poulin, D. (2010) Fast decoders for topological quantum codes. *Phys. Rev. Lett.*, **104**, 050504.
- [10] Wootton, J. R. and Loss, D. (2012) High threshold error correction for the surface code. *Phys. Rev. Lett.*, **109**, 160503.
- [11] [http://github.com/TomClose/mcmc\\_code](http://github.com/TomClose/mcmc_code).