Measurement-based quantum error correction

Janna Hinchliff
University of Bristol,
Quantum Engineering Centre for Doctoral Training

Introduction

Measurement-based (or one-way) quantum error correction (MBQEC) is a method with the capability to detect and correct any errors present in a measurement-based quantum computation (MBQC) setup [1]. There are a variety of methods and protocols we can use to perform QEC, although few that have been successfully implemented experimentally [2, 3]. An MBQC protocol requires a resource state and will inevitably experience errors, which may be caused by a wide variety of factors, including coherent, systematic control errors, environmental decoherence, channel loss and measurement errors [4]. This essay will discuss the theoretical methods utilised for MBQEC and how these may be implemented. Our long-term aim is to create a quantum computation setup that successfully corrects against a reasonable threshold percentage of errors. Such a thing is said to be fault tolerant.

Before leaping straight into the process of MBQC, it seems prudent to first discuss the finer details of QEC. We know that all forms of quantum computation will experience errors, leaving scientists with the crucial, yet difficult task of developing an implementable method of rectifying this. Previous research into quantum information has shown that there are two main types of error, a bit-flip error and a phase-flip error. A bit-flip error occurs when two bits in a state are swapped. For instance, in the single qubit state defined by $|\psi\rangle_{in} = \alpha|0\rangle + \beta|1\rangle$ a bit-flip error will output $|\psi\rangle_{out} = \alpha|1\rangle + \beta|0\rangle$. This is equivalent to performing an X operation on the state, meaning that this may also be referred to as an X error. A phase-flip error incurs a change in sign, giving $|\psi\rangle_{out} = \alpha|0\rangle - \beta|1\rangle$, also known as a Z error. Classically, although we still experience errors in our states, the system is easily correctable, as we can simply send many copies of the same state and take the outcome that occurs with the highest probability to be correct [5]. For a QEC procedure, we may not simply reproduce the classical configuration in a quantum setting as this relies on our ability to copy states, a well-known caveat of quantum mechanics [6]. Instead, we must consider alternative correction methods. One well-established method is the Shor code, which corrects one error on a system of nine qubits (or three logical qubits) [7]. However, this method is only useful in a quantum circuit setup, so we must instead discuss possible correction methods for a resource state.

Measurement-based quantum computation and resource states

MBQC is a quantum computing method that uses a resource state, allowing all computations to be done using only single qubit operations. So, what exactly is a resource state, and how can it be implemented? Most commonly, we consider a cluster state. This is a multi-qubit highly entangled state. Generation can be achieved efficiently in a d-dimensional lattice with a quantum Ising-type interaction through use of the CZ quantum gate

$$CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

$$\begin{aligned} \mathrm{CZ}\,|00\rangle &= |00\rangle \\ \mathrm{CZ}\,|01\rangle &= |01\rangle \\ \mathrm{CZ}\,|10\rangle &= |10\rangle \\ \mathrm{CZ}\,|11\rangle &= -|11\rangle \end{aligned}$$

which is used to create entanglement between two neighbouring qubits, which will be prepared in the state

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}.$$

We can think of a cluster state as a stabiliser state of qubits in a lattice configuration. To fully understand this, we must define stabiliser formalism. We may write the ground state and excited state of a particle as $|0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and apply the Pauli X and Z operators to these states where

$$\begin{aligned} X &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = |0\rangle \langle 1| + |1\rangle \langle 0| \\ Z &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = |0\rangle \langle 0| - |1\rangle \langle 1| . \end{aligned}$$

Application of the Pauli Z operator to the $|0\rangle$ and $|1\rangle$ states returns the eigenvalues $+1$ and -1 respectively. As the ground state retains its eigenvalue under this operation, we can say that Z is a stabiliser state for $|0\rangle$. Similarly the X operator acts as a stabiliser for the $|+\rangle$ state. The calculation for this is simple (assuming orthogonality):

$$\begin{aligned} X|+\rangle &= X\Big(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\Big) \\ &= \frac{X|0\rangle + X|1\rangle}{\sqrt{2}} \\ &= \frac{(|0\rangle \langle 1| + |1\rangle \langle 0|)|0\rangle + (|0\rangle \langle 1| + |1\rangle \langle 0|)|1\rangle}{\sqrt{2}} \\ &= \frac{|1\rangle + |0\rangle}{\sqrt{2}} \\ &= |+\rangle . \end{aligned}$$

For any particular qubit $k \in \mathcal{C}$, where $\mathcal{C}$ is the lattice, we may define a neighbourhood N_k of it's nearest-neighbour qubits. Using this, any cluster state $|\psi\rangle_{\mathcal{C}}$ can be defined by the generators

$$G_k = X_k \bigotimes a \in N_k Z_a, \forall k \in \mathcal{C}$$

where $a \in \mathcal{C}$ is a second qubit, and we see that $G_k |\psi\rangle_{\mathcal{C}} = |\psi\rangle_{\mathcal{C}}$, i.e. G_k stabilises the cluster state. If we act this generator on the cluster state we have an eigenvalue equation:

$$X_k \bigotimes_{a \in N_k} Z_a |\psi\rangle_{\mathcal{C}} = \pm |\psi\rangle_{\mathcal{C}}$$

where the eigenvalues are ± 1. These equations will be used later to show how we can identify errors in a system.

It is possible to construct both two- and three-dimensional cluster states using only the nearest-neighbour Ising interaction. Qubits will interact only once with their neighbours at the beginning of the computational process. Any quantum gate can be implemented using the cluster state method, with a CNOT gate being possible using a minimum construction of only four qubits [8]. Now that we have this cluster state method in place, we may perform universal quantum computation using only single-qubit operations.

The two resource states discussed here are the Raussendorf code and the surface code. The so-called surface code is a topological QEC code that comprises of a combination of X and Z operators and the Raussendorf code is simply a three-dimensional modification of this. Much of the following material can be found in [9]. By topological, we mean that the system depends on global characteristics rather than local ones and has highly entangled degenerate ground states [10]. It is necessary to distinguish between the Raussendorf code and the surface code for the simple reason that the Raussendorf code is robust against dynamic errors whereas the surface code is not. However, for the purposes of this error correction protocol the surface code is perfectly sufficient. Both have evolved from a code pioneered by Alexei Kitaev, known as the toric code, which distributes qubits on the surface of a toroid [11]. More recently, it has become clear that the toroidal geometry is not a necessity and that we can in fact just as easily use a planar surface [9]. Using the surface code we can then construct a two-dimensional code space that encodes a single

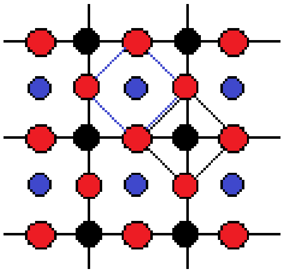

Figure 1: *Layout of a surface code. Red circles are data qubits used to store quantum information. Black circles are X stabilisers and blue circles are Z stabilisers. The section of 5 qubits indicated by the blue lines is a plaquette whereas the black lines represent a vertex. This system can encode a single logical qubit using 25 qubits, 13 of which are data qubits which can correct a single X and/or Z error.*

logical qubit. An example of this is given in Figure 1 for a 5 × 5 structure. If we define vertices and plaquettes as in Figure 1 and call vertices v and plaquettes p we can define the stabiliser operators to be

$$A_v = \prod_{i \in v} \sigma_i^x, \quad B_p = \prod_{j \in p} \sigma_j^z$$

where $i \in v$ are the edges touching each vertex and $j \in p$ are the edges surrounding each plaquette. Hence, as this acts as a stabiliser we can say that for any state $|\psi\rangle$,

$$\begin{aligned} A_v |\psi\rangle &= |\psi\rangle, \forall v, \\ B_p |\psi\rangle &= |\psi\rangle, \forall p. \end{aligned}$$

Equivalently, for a plaquette, we can say that $Z_1 Z_2 Z_3 Z_4 |\psi\rangle = |\psi\rangle$ where Z_i are the four Z stabilisers surrounding the X qubit starting from the top and going in a clockwise direction (similarly for X). We know that $[X, X] = 0, [Z, Z] = 0$ but $[X, Z]_+ = 0$. Now let's assume that the third qubit in our plaquette has experienced an X error. We find that now when we measure the stabilisers surrounding this state we get the result $Z_1 Z_2 X_3 Z_4 |\psi\rangle = -|\psi\rangle$, allowing us to detect the error. The beauty of this system is that by alternating a row of Z operators with a row of X operators, we can tell not only where an error has occurred, but which type of error, as a bit-flip error will return a negative parity in the Z operator but not in the X operator whereas a phase error will return a negative X parity. This seems like a foolproof method, but problems do start to arise when we have several errors close to each other in the system. We can always detect and resolve a single error, but we now need to decide whether there comes a point when repairing the system is no longer possible. Consider the scenario posed in Figure 2b). How can we distinguish between these two possibilities? The answer to this, surprisingly, is that we don't need to. We find that a correct parity is sufficient to be able to perform successful operations on the system. This means that any number of errors can be left uncorrected as long as they are formed in a loop, as then all parities are positive. This means that we only need to be able to correct single errors or chains of errors and such a method is indeed possible.

Next, we must measure the eigenvalues of all lattices, a task which can be completed in parallel and we can then generate a graph showing the positions of the parity flips. This allows us to find error chains, although we are only able to tell from the parity where the starting and ending points of such chains are. However, this is still correctable, although probabilistic. If we define a physical error probability, $p < 1$, we see that the probability of finding one error is p, the probability of finding two errors is $\mathcal{O}(p^2)$ and so on, such that the probability of finding multiple errors becomes increasing unlikely. If we then take each cell that returns a negative parity we can construct a weighted graph, where the weight between two cells corresponds to the numbers of errors needed to travel from one to the other. We want to find the solution that gives the overall total minimum weight which can be done computationally using a classical algorithm of polynomial resource known as Edmond's minimum weight perfect-matching algorithm [12]. We then perform iterations of the full process in order to get a high fidelity representation of the scheme. We may consider each iteration as a time step, transforming the system into a three-dimensional graph with time as the third axis. Another element to consider is that we may in fact experience errors on the measurement qubits. These propagate temporally, differentiating them from bit and phase flip errors, which propagate spatially. As Edmond's algorithm can operate in three-dimensions, we simply run matching on all sheets as a three-dimensional structure.

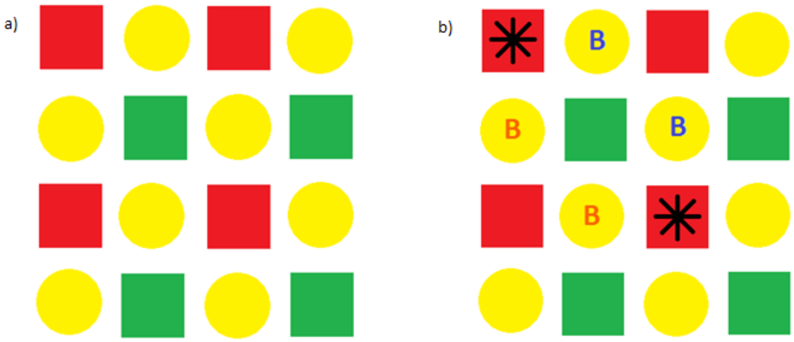

Figure 2: *a) shows the setup of a surface code, where the red squares represent Z operators, green squares are X operators and the yellow circles are data qubits. In b), the black stars on the Z operators show that these have both returned a negative parity, meaning that there are two bit-flip errors. The two pairs of coloured Bs show the two possible places the errors could have occured, but it is impossible to tell which of the pairs are erroneous and which are not.*

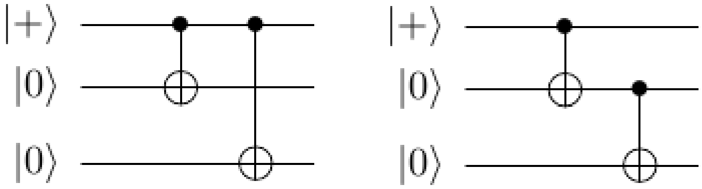

Figure 3: *Two alternative circuits to carry out the same operation - namely taking* $|+\rangle|00\rangle \longrightarrow \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$

To encode multiple logical qubits, we need to consider a lattice with closed boundaries, i.e. where the qubits lie on the boundaries unlike the system shown in Figure 1. This is where the toric code mentioned above falls down, as this only has the capability to encode two logical qubits as a maximum, however a closed boundary periodic surface code lattice can encode an arbitrary number of qubits. In this setup, if we have N data qubits we need N stabilisers, but we no longer have any degrees of freedom. The solution is to add a 'defect'. To do this, we stop measuring the parity of one of the stabilisers in the lattice at random, leaving a hole which gives the necessary degree of freedom and error correction is possible as described above [9].

Fault-tolerance

A process is defined to be fault-tolerant if the total amount of errors in the system is less than the number the system can handle. From the discussion above, we can define the error rate for one logical qubit as

$$p_L \quad = \quad p + \mathcal{O}(p^2) + \mathcal{O}(p^3) + \cdots + \mathcal{O}(p^N)$$

for an N qubit system. We know that one logical qubit, $|\psi_L\rangle$ can correct a single error, so we may deterministically remove the first error, giving

$$p'_L \quad = \quad \mathcal{O}(p^2) + \mathcal{O}(p^3) + \cdots + \mathcal{O}(p^N).$$

This process can be repeated until all errors are removed, however, we also have to consider the problem of errors cascading through the system and propagating more and more errors. For example, if we consider a simple system of CNOT gates, we can show that the two operations in Figure 3 are equivalent. However, if an error occurs on the control qubit before a CNOT operation, this transfers to an error on both qubits after the operation has been performed, but if it occurs on the target qubit, there is still only a single error after the operation has been performed, meaning that the two circuits in Figure 3 do in fact differ in terms of output after errors. Because of this, we should use the first circuit, as this will only cause an error cascade if the error occurs in qubit 1. Once we are convinced that we have fault-tolerant QEC and gates, we can define a logic unit: a logical state acted upon by a QEC process followed by a set of quantum states followed by a further QEC process. We say that $p_{L_1} \approx \mathcal{O}(p^2) = c(p)^2$ where c is the physical failure rate per qubit. As the logic units are self-similar circuits, we can apply a concatenation process, so that for the kth order term we find

$$p_{L_k} \approx \frac{(cp)^{2^k}}{c}$$

and from this we define the relation

$$p < \frac{1}{c}.$$

This condition is known as the threshold theorem [13] and once satisfied, we find that for a linear increase in qubits, we get an exponential suppression in error. c is specific to the type of code, and we can achieve a particularly good rate for a surface code in comparison to many other architectures.

Paper analysis

Fault-tolerant quantum computation with high threshold in two dimensions [14]

This paper presents '...a scheme for fault-tolerant universal quantum computation on a two-dimensional lattice of qubits requiring only a nearest-neighbour translation-invariant Ising interaction and single-qubit preparation and measurement.' [14] We are shown how to construct fault-tolerant universal gates in three spatial dimensions, which are then mapped to two-dimensions and the threshold value is calculated. The scheme used is well suited to large qubit models such as ion traps [15] and optical lattices [16]. The toric code features in the protocol along with the additional phenomenon of magic state distillation [17]. Magic states are quantum states containing a level of error described as acceptably low. To create these states, we begin with a set of so-called 'noisy' quantum states and perform distillation on them, leaving ourselves with fewer states but each with a higher fidelity than those we had originally. The distillation process requires a very large amount of qubits, which is a serious consideration when thinking of building a quantum computer. However, the process requires only a combination of local operations and classical communication and the use of magic states means that the error threshold is improved significantly. Recently, it has been shown that the amount of resources can be reduced significantly if the distillation is merged with the implementation of Toffoli gates [18]. This shows that there is still scope for such a method to be considered, despite its current scalability issues, with a potential research area being the combining of the distillation process with different gate sets, possibly reducing the amount of qubits needed even further. The scheme uses the surface code constructed in slices as opposed to the Raussendorf code as this is more easily visualised.

The protocol requires the lattice, $\mathcal{L}$ to be subdivided into three regions, labelled V, D and S. V encompasses the main body of the cluster and provides topological error correction. The qubits in this region are measured in the X basis whilst qubits in the D and S regions are measured in the Z and Y bases respectively. D represents the defects defined above and elements of S are well-separated qubit locations interspersed with these defects. We can call stabilisers 2-chains and errors 1-chains, then $\mathcal{L}$ can be described as a chain complex with a dual whose coefficients are in $\mathbb{Z}_2$. The lattice is made up of cubes, faces, edges and sites. The error correction procedure used in V uses the minimum weight chain matching algorithm [12] with a good fault-tolerance threshold of 2.9×10^{-2}. An alternative to this would be the three-dimensional random plaquette $\mathbb{Z}_2$-gauge model, which is able to deliver a slightly higher threshold but seems not to have been implemented due to its lower computational efficiency. We can then treat the system as a plane with pairs of either electric or magnetic holes, where an electric hole is a stabilised site and a magnetic hole is a stabilised face and each hole intersects a defect and constant-time slice. Any individual qubit needs a pair of holes supporting it with a magnetic hole pair corresponding to an X spin operator and an electric whole pair a Z spin operator. Implementation of a CNOT gate is then possible, as shown in Figure 4. To satisfy the conditions for fault-tolerant quantum computation, we also require fault-tolerant preparation and measurement of X and Z eigenstates, which, along with the CNOT gate form the topologically protected gate set. These are easily created, but must then be purified via magic distillation and used along with a Bell pair to generate the final two gates needed to complete the universal fault-tolerant gate set, namely $e^{\frac{i\pi}{4X}}$ and $e^{\frac{i\pi}{8Z}}$.

All that now remains is to calculate the error threshold. The paper assumes that each individual source of error has the same magnitude and that there is no time between preparation and measurement of a qubit, and therefore no storage error. Using numerical simulations the threshold for each physical source was calculated to be

$$p_c \quad = \quad 7.5 \times 10^{-3}$$

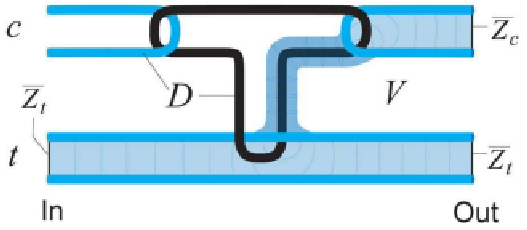

Figure 4: *The CNOT gate $\Lambda(X)_{c,t}$ (c:control, t: target) formed by topologically entangled lattice defects. Each pair of defects carries an encoded qubit. Defects exist as primal (blue) and dual (black), and are created by local measurement. The primal correlation surface (light blue) shown here converts an incoming Pauli operator Z_t into an outgoing $Z_t \otimes Z_c$ as required for a CNOT gate [14].*

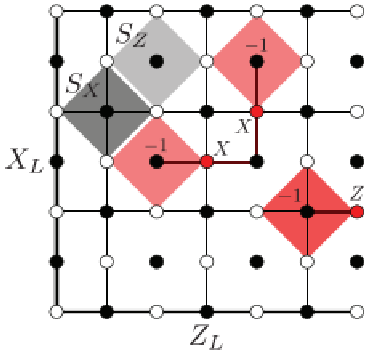

Figure 5: *Structure of the surface code for $d = 4$ where open circles signify data qubits and closed circles signify ancillary qubits. Stabilizer generators and logical operators are indicated. Chains of X and Z errors affecting data qubits will anticommute with the stabilizer generators at the end points, which will have eigenvalues equal to -1 as indicated. End points may be obscured if chains terminate on boundaries. In general, the number of data qubits is $d^2 + (d-1)^2$ [19].*

with a distillation threshold of

$$p_c = 2.8 \times 10^{-2}.$$

Fault-tolerant thresholds for error correction with the surface code [19]

This paper uses numerical simulations to investigate the error threshold of fault-tolerant quantum computation using the surface code. A surface code is constructed on a square $d \times d$ lattice, shown in Figure 5 with $d = 4$ where S_X and S_Z refer to the X and Z stabilisers respectively. As the paper's main objective is to study error correction, the system is restricted to a surface code that confines a single logical qubit. It is possible to change the topology of the surface to introduce multiple logical qubits and operators but as far as error correction is concerned, this merely adds complications to the system with no obvious benefits. To find the error syndrome, the operations shown in Figure 6 are performed. The circuits may be implemented in parallel, but need to be repeated d times in order to compensate for possible measurement errors. As a data qubit error is seen as spacelike whilst an ancillary qubit error is timelike, the error syndrome is the space-time volume of changes to the expected result. We then again apply minimum-weight perfect matching [12].

The next step requires numerical methods to determine the error threshold, firstly Monte Carlo simulations [20]. It is possible to decode the surface code using a random-plaquette gauge model [21] in three dimensions and we have the relation

$$p_l = (p - p_{th})d^{\frac{1}{\nu_0}}$$

for the logical error rate p_l where p is some physical error rate, p_{th} is the threshold error rate and ν_0 is the scaling exponent for the universality class of the model. This is then fitted to a quadratic universal scaling function and

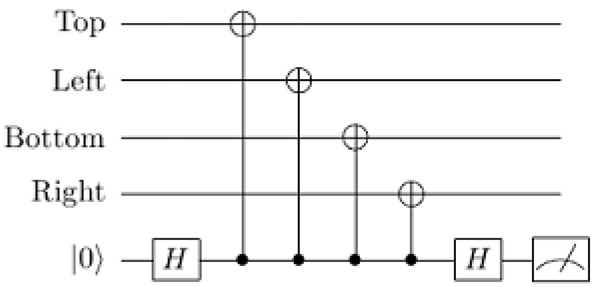

Figure 6: *The labels top - right refer to the order in which the ancillary qubits are measured around a data qubit using S_X. The circuit for S_Z takes a similar format.*

values of p_{th} and ν_0 are found using a best fit function. It turns out after calculation that the Z-error threshold is lower than the X-error threshold, so this is given first as a base value and all other results give the X-error threshold. To find the code capacity, an idealised situation is first measured, where it is assumed that the error syndrome need only be measured once, thus eliminating any timelike parts of the decoding algorithm, and giving a threshold of $p_{th} = 0.1030 \pm 0.0001$. Varying degrees of noise are then added into the system, where we see thresholds ranging from 0.00502 ± 0.00001 for a standard circuit-based noise model to 0.0290 ± 0.0001 for a phenomenological noise model. Numerical simulation plots can be found in [19]. For the most part, the thresholds found in this paper appear to be significantly better than those in other resources, with the exception of the first result, where only one round of measurements was necessary. This means that we may have to take an average, finding the logical error rate per round. However, we then find that the larger code is less reliable which would imply that the threshold is a physical error rate such that $p_{th} < p_n$ and as $n \longrightarrow \infty$, $p_n \longrightarrow p_{th}$, meaning that this may cause us to calculate an overestimate of the value of the threshold. This leaves the paper inconclusive on whether the error rates achieved are in fact better than previously derived, or whether this is a new type of error threshold based on other parameters, but shows that there is reason for us to believe that lower error rates could be a possibility and that this is a significant area for exploration.

Conclusion

There are many papers claiming error thresholds of varying degrees. Early in this research we find [22] which is a theoretical paper from 2006 which uses a three-dimensional cluster state method giving an error threshold of 0.014 for local depolarising error and 0.0011 for each source in an error model. In 2007, [23] achieved thresholds of 0.0075 for each source by reducing the system to a two-dimensional cluster state. [24], written in 2010 agrees with this value and also calculates an overall error threshold, $p_{loss} = 0.252 \pm 0.005$ which takes into account the probability of qubits being lost in the system in addition to errors on said qubits. Values of the order of 1.5×10^{-2} are found in [25], which also takes into account the phenomenon of biased noise. Experimentally, we do not yet have the resources to create such advanced systems, but [26] achieves a reasonably good threshold on the order of 0.1 per particle on a four-qubit photonic cluster state. As we have seen, [19] achieves thresholds of between 0.00502 and 0.01140 using the surface code for two spatial dimensions via numerical simulations as of 2014. This paper is moving towards a claim that the error threshold is actually better than we have proved thus far but provides insubstantial evidence to verify this claim. It seems that the next logical step should be to perform the numerical simulations used in this paper to work in earlier papers and see how this compares.

It is necessary to note that this essay focuses only on one topology used for resource states. This is purely because it is the simplest and the most easily implementable. There are many more, for example, the toric code mentioned above, the color code [27], and others such as the triangular code and tetrahedral code [28]. It may seem that these are advantageous, as theoretically they may give better thresholds than the surface code, but it is difficult to see a time when we will have the resources to be able to build such complex systems. The surface code provides us with a sensible, scalable method with a significantly high tolerance for errors that is hoped to be implementable within a reasonable time scale, thus providing a promising outlook in the field of QEC.

Bibliography

[1] Robert Raussendorf and Hans J. Briegel. A one-way quantum computer. *Phys. Rev. Lett.*, 86, 5188, 2001.

[2] J.Chiaverini *et al.* Realization of quantum error correction. *Letters to Nature*, 602:602 – 605, 2004.

[3] M. D. Reed *et al.* Realization of three-qubit quantum error correction with superconducting circuits. *Letters to Nature*, 482:382–385, 2011.

[4] Simon J. Devitt *et al.* Quantum error correction for beginners. *Rep. Prog. Phys.*, 76, 076001, 2013.

[5] Vikas Grupa and Chanderkant Verma. Error detection and correction: an introduction. *International journal of advanced research in computer science and software engineering*, 2, 11, 2012.

[6] W.K. Wootters and W.H. Zurek. A single quantum cannot be cloned. *Letters to nature*, 299:802–803, 1982.

[7] Daniel A Lidar and Todd A Brun. *Quantum error correction.* Cambridge: Cambridge University Press, 2013.

[8] Gelo Noel M. Tabia. Quantum computing with cluster states. 2011. University of Waterloo.

[9] Austin G. Fowler *et al.* Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A*, 86, 032324, 2012.

[10] J Pachos. Introduction to topological quantum computation. 2010. University of Leeds.

[11] A. Kitaev and C. Laumann. Topological phases and quantum computation, 2009. arXiv:0904.2771v1.

[12] William Cook and André Rohe. Computing minimum-weight perfect matchings. *INFORMS Journal on Computing*, 11, No.2, 1999.

[13] Dave Bacon. Fault tolerant quantum computation and the threshold theorem. n.d. University of Washington, lecture notes: http://courses.cs.washington.edu/courses/cse599d/06wi/lecturenotes19.pdf.

[14] Robert Raussendorf and Jim Harrington. Fault-tolerant quantum computation with high threshold in two-dimensions! *Phys. Rev. Lett.*, 98, 190504, 2007.

[15] W Hensinger *et al.* T-junction ion trap array for two-dimensional ion shuttling, storage and manipulation. *Applied physics letters*, 88, 034101, 2006.

[16] D Jaksch *et al.* Experimental aspects of quantum computing. *Phys. Rev. Lett.*, 82, 1975, 1975.

[17] Sergei Bravyi and Alexei Kitaev. Universal quantum computation with clifford gates and noisy ancillas. *Phys. Rev. A*, 71, 022316, 2005.

[18] Bryan Eastin. Distilling one-qubit magic states into toffoli states. *Phys. Rev. A.*, 87, 032321, 2013.

[19] Ashley M. Stephens. Fault-tolerant threshold for quantum error correction with the surface code. *Phys. Rev. A*, 89, 022321, 2014.

[20] Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American statistical association*, 44:335–341, 1949.

[21] Takuya Ohno *et al.* Phase structure of the random-plaquette $\mathbb{Z}_2$ gauge model:accuracy threshold for a toric quantum memory. *Nuclear physics B*, 697, 3:462–480, 2004.

[22] R Raussendorf *et al.* A fault-tolerant one-way quantum computer. *Annals of physics*, 321, 2242, 2006.

[23] R Raussendorf *et al.* Topological fault-tolerance in cluster state quantum computation. *New journal of physics*, 9, 199, 2007.

[24] S. D. Barrett and T. M. Stace. Fault tolerant quantum computation with very high threshold for loss errors. *Phys. Rev. Lett.*, 105, 200502, 2010.

[25] Ashley M. Stephens *et al.* High-threshold topological quantum error correction against biased noise. *Phys. Rev. A*, 88, 060301, 2013.

[26] Stefanie Barz *et al.* Demonstrating elements of measurement-based quantum computation. *Phys. Rev. A*, 90, 042302, 2014.

[27] M. Kagarian *et al.* Topological color codes and two-body quantum lattice hamiltonians. *New J. Phys.*, 12, 025018, 2010.

[28] Hector Bombin and Miguel Angel Martin-Delgado. Topological color codes. Departamento de Fsica Terica I Universidad Complutense de Madrid, http://qserver.usc.edu/qec07/QEC07/HectorBombin.pdf.