Surface Codes

Anosh Wasker and Hunter Hamby

Department of Physics, University of Wisconsin–Madison

May 8th, 2023

Abstract

In this paper we present an overview of surface codes and derive an error model motivated by the Ising Model to suppress error for arbitrarily large number of qubits. We show that two research groups demonstrate a small improvement when the surface code was implemented and when the distance of the code was increased.

1 Introduction

With the rising popularity of quantum computing, it is important to understand the underlying motivations and challenges of the field. Quantum computing is an attempt to realize the computation power offered by two main properties: superposition and entanglement. Unlike classical bits that only offer a binary state, quantum computing utilizes the superposition property that gives us access to some combination of the two states and thus opening up a door to a different type of computation. Some examples include factorizing large prime numbers, simulating molecules and securing communication.

Increasing the decoherence of qubits and making fault tolerant quantum gates with high fidelity are ongoing challenges and therefore quantum error-correction is needed. Quantum error-correction requires a different approach than classical error-correction not just because of a different architecture but because of the theoretical limit placed by what is known as the "no-cloning" theorem which does not allow for a quantum state to be copied.

This paper explores surface codes for quantum error correction. They are relevant to superconducting qubits because of the geometric layout of the physical qubits that can be corresponded and tuned with the surface code. Surface code is relevant to quantum computing in superconducting platforms as they offer a lattice and nearest neighbour coupling.

In this paper we derive the quantum gate fidelity threshold to suppress error for a large number of qubits. We then look at a few papers and compare their results for the threshold and present our conclusions.

2 Stabilizer Measurement

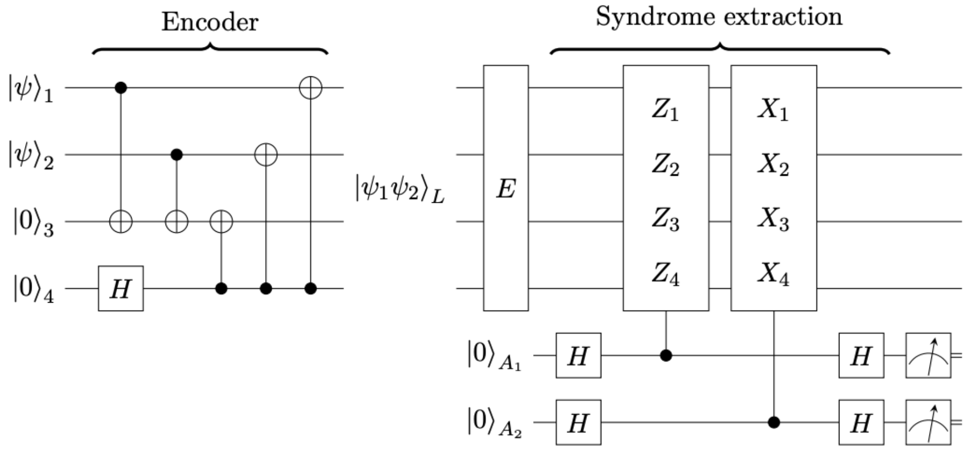
A stabilizer is an operator that tells us if the data qubit has experienced an error without measuring the data qubit.

The logical qubit state is an eigenstate of the stabilizer with an eigenvalue +1 for no-error and -1 for an error. [4]Stabilizers must have the following properties:

1. They must be Pauli group elements
2. They must stabilize all logical states.
3. All the stabilizers must commute with each other.

We need only to correct for bit-flips (X) and phase-flips (Z) that can be mapped to the X and Z gates. We do not need to worry about the Y errors since the Y gate can be decomposed into X and Z rotation errors.

As an example, let us consider the [[4,2,2]] detection code.



Figure“1: Stabilizer Code Circuit [4]

2.1 [[4,2,2]] Detection Code

This code works for a two qubit state that that could be subject to both X- and Z-errors. Figure:1 shows how to encode these qubits to get a logical state $|\psi\rangle_L$

The stabilizer for the two-qubit logical state is

$$\mathcal{S} = \langle X_1X_2X_3X_4, Z_1Z_2Z_3Z_4\rangle \tag{1}$$

From the circuit Fig:1 we can see that our logical state becomes

$$\begin{aligned} |\psi_f\rangle =& \frac{1}{2}(I^{\bigotimes 4} + Z_1Z_2Z_3Z_4)\,|\psi\rangle_L\,|0\rangle_{A1} + \frac{1}{2}(I^{\bigotimes 4} - Z_1Z_2Z_3Z_4)\,|\psi\rangle_L\,|1\rangle_{A1} \\ &+ \frac{1}{2}(I^{\bigotimes 4} + X_1X_2X_3X_4)\,|\psi\rangle_L\,|0\rangle_{A2} + \frac{1}{2}(I^{\bigotimes 4} - X_1X_2X_3X_4)\,|\psi\rangle_L\,|1\rangle_{A2} \end{aligned} \tag{2}$$

If there is an error in the logical qubit we see that the stabilizers will return a value of -1 and therefore the coefficient of the $|0\rangle_A$ -ancilla qubit will be zero. If there is no error then the coefficient of the $|1\rangle)_A$ -ancilla will be zero. Using this knowledge we can measure the ancilla to determine if there was an error.

Measuring these ancillas is called a syndrome extraction where the output of the syndrome determines they type of error. Table in Fig:2 shows the errors and their syndromes.

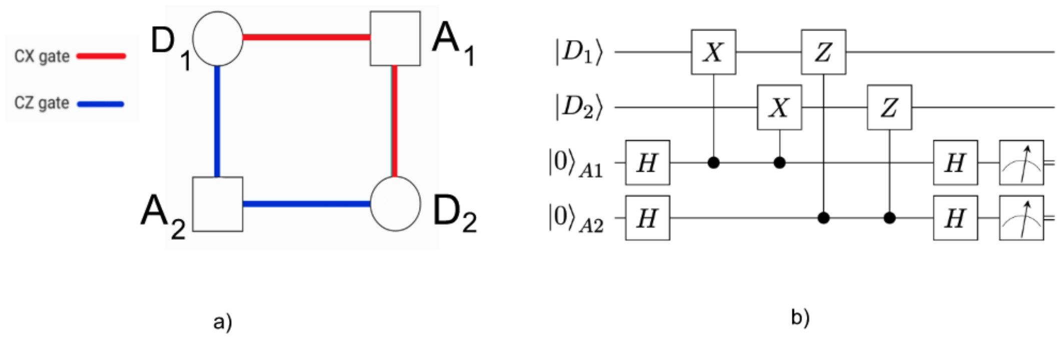
Error	Syndrome, S	Error	Syndrome, S	Error	Syndrome, S
X_1	10	Z_1	01	Y_1	11
X_2	10	Z_2	01	Y_2	11
X_3	10	Z_3	01	Y_3	11
X_4	10	Z_4	01	Y_4	11

Figure“2: Syndrome Extraction Values [4]

There are larger codes like the [[9,1,3]] Shor code that can not only detect but also correct error.

3 Surface Code

Surface codes are a class of topological codes meaning they are modular and a base element can be patched together to form arbitrarily large code sizes. The surface code requires only parity checks within sets of nearest-neighbor qubits in a two-dimensional qubit layout [4][5]. This is particularly useful for superconducting qubits as they exist on a 2D lattice and thus the nearest neighbour interaction of those qubits means the surface code is easily adaptable.

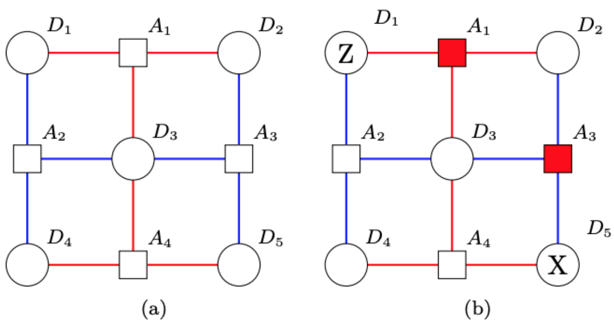


Figure“3: a) shows a four-cycle surface code with the data qubits and ancilla qubits. b) Circuit diagram of the four-cycle surface code. *Modified from [4] for clarity

3.1 Surface code four-cycle

The fundamental building block for surface code is called the surface code four-cycle. It is easier to visualize the surface code with the diagram in Figure:3(a) rather than the (b) circuit diagram. Here the red arms are the controlled-X operations and likewise the blue arms are the controlled-Z operations on the data qubits D1 and D2. The Ancillas are A1 and A2. The A1 is responsible for the XX stabilizer while the A2 responsible for the ZZ stabilizer. The four-cycle surface is just a building block, and it not useful because it does not code any logical qubit. It has two qubits so $n = 2$ and two stabilizers so $m = 2$. The number of logical qubits $k = n - m = 0$. Therefore, we explore an example of the [[5,1,2]] code.

3.2 [[5,1,2]] Code



Figure“4: a) The [[5,1,2]] code b) Ancilla A_1 detects the Z error in data qubit D_1 while Ancilla A_3 detects the X error in data qubit D_5 [4]

In Figure:[4] we can see that the four-cycle code is tiled to form a bigger lattice. Here we would find that we have the following stabilizers.

$$\mathcal{S}_{[[5,1,2]]} = \langle X_{D_1}X_{D_2}X_{D_3}, Z_{D_1}Z_{D_3}Z_{D_4}, X_{D_3}X_{D_4}X_{D_5}, Z_{D_2}Z_{D_3}Z_{D_5},\rangle \tag{3}$$

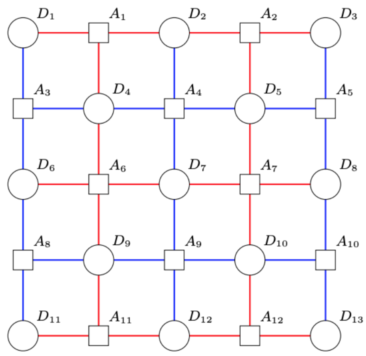
These stabilizers will help detect errors on the data qubits. If there is an error then the ancilla qubit will measure a "1".

In Figure:4 (b) we can see that there is a Z error on qubit D_1 this means that ancilla qubit A_1 will measure a "1". Likewise there is an X error on qubit D_5 and so ancilla qubit A_3 will detect it by giving a "1" measurement. The measurement of the ancillas then tell us what gates (X or Z) to apply to the data qubits to correct the errors.

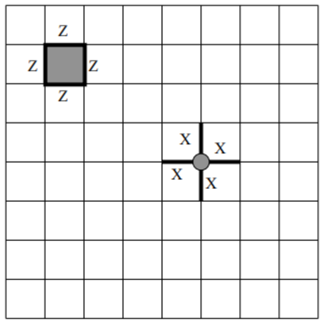
The notation for the stabilizers can be a little confusing. X_{D_1} implies that there it detects the X error on data qubit D_1 on ancilla A_2 and so the blue edge connecting D_1 and A_2 is in fact a controlled-Z gate.

3.3 What would scaling look like?

To scale the surface code would mean to increase the lattice. The [[13,1,3]] is the smallest surface code that does detection and correction and it is modular and can be stacked to scale up.



Figure“5: The [[13,1,3]] code can be visualised as 4 sets of the [[5,1,2]] code combined to make a bigger lattice



Figure“6: Toric code representation showing the X stabilizer acting on a site, and the Z stabilizer acting on a plaquette.

If the distance $d = \lambda$ then the

$$[[n = \lambda^2 + (\lambda - 1)^2, k = 1, d = \lambda]] \tag{4}$$

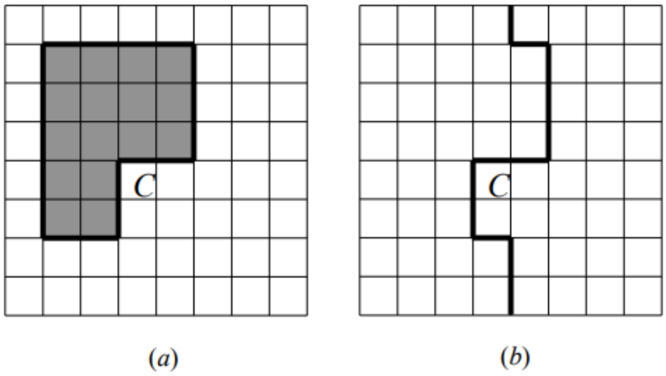
Here n = number of qubits and k = number of logical qubits. The distance of a code means the number of operations required to get from one logical state to another. The [[13,1,3]] surface code is the smallest surface code that can detect as well as correct error. The repeated measurements of the ancillas tell you what gates to implement on the data qubit to correct it.

4 Error Model for the Surface Code

It is theorised that if quantum gates have a probability of error below a certain threshold, then any error can be corrected with probability 1 by scaling the lattice arbitrarily large. This is ideal for surface code since it can be scaled easily using the 4-cycle. To start we consider a class of surface codes called toric codes. Toric code is a square lattice with periodic boundary conditions. The links connecting points on the lattice represent qubits. Each “site” has an X stabilizer acting on it, and each “plaquette” has a Z stabilizer acting on it as Figure: 6 shows.

To derive this threshold of error we consider chains of error occurring on the lattice. We are particularly interested in what are called homologically trivial chains, chains that are the boundary of plaquettes tiled in such a way that the chain forms a loop as shown in Figure:7. These are of interest as these error chains can be corrected without changing the information encoded in the qubits. This is because the chain encloses plaquettes that act with Z stabilizers and the product of these does not effect the quantum state. The same idea applies to X errors since sites become plaquettes in the dual lattice. Homologically nontrivial chains allow for error to be corrected but since the correction is not a stabilizer it changes the encoded information and thus is of less interest.

As we can see from Table 2, different errors can have the same syndrome. In the language of algebraic topology, this is saying that a particular error chain can share a boundary with several different error chains. This means we do not have to explicitly find the error chain to correct the error, we just have to find a chain that shares the error chain’s boundary. This is the case given that applying the correction gate to each link will correct the error without changing the encoded information, meaning we only need to find the homology class this error chain belongs.



Figure“7: Homological Cycles. (a) Shows a homologically trivial cycle. (b) Shows a homologically nontrivial cycle.

For a toric code with lattice length L and probability of error p occurring in a qubit, if p is small enough then the errors in the lattice should be dilute. That is there should not be error chains that are significant in size with respect to the lattice. This implies we can correct multiple error chains with one homology class since small errors distributed throughout the lattice are easier to connect in a trivial way than multiple long error chains. More importantly though this means if p is small enough then as L grows the probability of guessing the homology class of the error approaches 1. This implies the error is guaranteed to be fixed if L is large enough.

Now to derive the connection between surface code and the random bond Ising model.

4.1 Motivations from the Ising-Model

In statistical-mechanics the Ising Model analyses the spin states for a given lattice. Ising solved it for 1-dimension in his thesis. The model has discrete variables that represents site spins in the -1 or +1. Since this model uses discrete binary variables for the site states and involves a lattice it is used as a motivation to model errors in the surface code. We derive an error model following the work in [2].

Given a lattice we assume that the probability of error occurs with probability p and assume the errors in qubits are independent of each other. We now suppose an error chain E occurs that can be defined by the function $n_E(l) = 1$ if $l \in E$ and $n_E(l) = 0$ if $l \notin E$, where l is a link in the lattice. The probability of chain E occurring is given by

$$\text{prob}(E) = \prod_l (1-p)^{1-n_E(l)} p^{n_E(l)} \tag{5}$$

Given this error chain we want to find the probability for all other chains E' that share a boundary with E. We start by writing $E' = E + C$, where C is a cycle and $+$ represents the disjoint union of sets. The probability of $n_C(l) = 1$ and $n_E(l) = 0$ is proportional to

$$\left(\frac{p}{1-p}\right)^{n_C(l)} \tag{6}$$

This represents the probability of the link being in E' but not E. If $n_C(l) = 1$ and $n_E(l) = 1$ then this is the chance the link is in E but not E'. This occurs with probability proportional to

$$\left(\frac{1-p}{p}\right)^{n_C(l)} \tag{7}$$

This implies the probability of chain E' occurring given E is

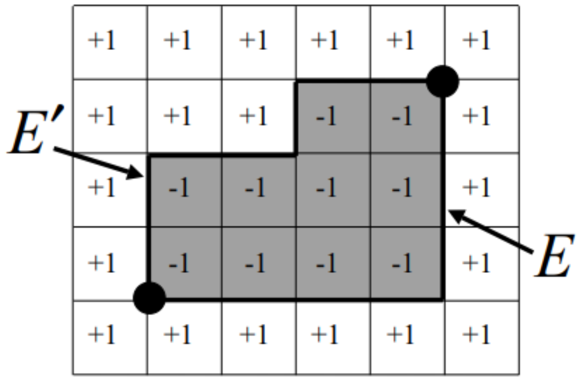
$$\text{prob}(E'|E) \propto \prod_l e^{J_l} e^{-2J_l n_C(l)} \tag{8}$$

where

$$e^{-2J_l} = \begin{cases} \frac{p}{1-p}, & \text{if } l \notin E \\ \frac{1-p}{p}, & \text{if } l \in E \end{cases} \tag{9}$$

This probability shown here is dependent only on the boundary of E. This probability implies that the partition function describing this model in the dual lattice is

$$Z(J,\eta) = \sum_{\{\sigma_i\}} \exp\left(J \sum_{\langle ij \rangle} \eta_{ij}\sigma_i\sigma_j\right) \tag{10}$$



Figure“8: Representation of error in the lattice using the partition function for the dual lattice.

with $\eta = 1$ if $l \in E^*$ and -1 if $l \notin E^*$, E^* being the error chain in the dual lattice. By describing this partition function in the dual lattice we can picture it in the actual lattice with Figure 8. If we assume that η's are chosen at random then $\eta = 1$ with probability p and $\eta = -1$ with probability -1, then this partition function then describes the $2D$ random bond Ising model. Particularly the equation

$$e^{-2J} = \frac{p}{1-p} \tag{11}$$

describes the coupling strength and bond probability that describes the Nishimori line. The Nishimori line crosses a phase transition at a fixed point. The value of p that is this fixed point represents the quantum threshold.

5 Numerical Analysis

The $2D$ random bond Ising model is not solvable analytically. This means we must resort to numeric means to solve for the threshold probability. In this section we present two papers that solve for this value. These papers use different methods for finding the value of the threshold probability. [3] reports a value of $p_c = 0.1094 \pm 0.00002$ while [5] reports a value of $p_c = 0.1033 \pm 0.00002$. We then briefly discuss the discrepancy.

5.1 Methods and Results

We first discuss the methods and results for [3].

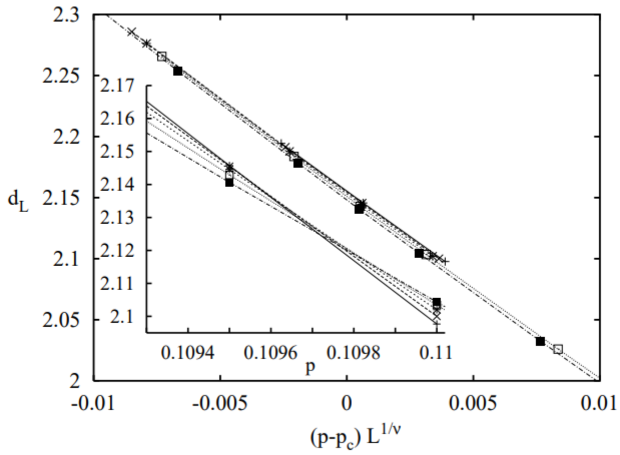
The free energy of a domain wall for a lattice with length L is given by

$$d_L = L^2(f_L^{(p)} - f_L^{(a)}) \tag{12}$$

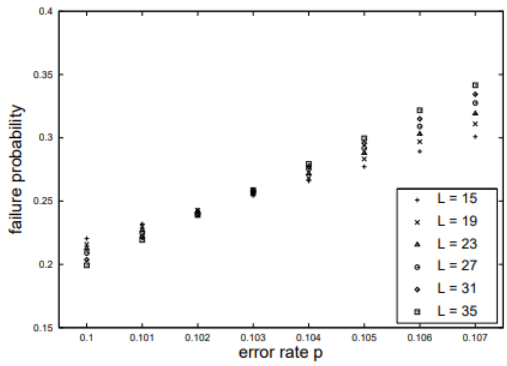
where $f_L^{(p)}$ represents the free energy per site with periodic boundary conditions and $f_L^{(a)}$ represents the free energy per site with antiperiodic boundary conditions. Using a transfer matrix technique, this is the standard way to solve the Ising model, they found $f_L^{(p)} = \frac{\ln\left(Z^{(p)}\right)}{LN}$ and $f_L^{(a)} = \frac{\ln\left(Z^{(a)}\right)}{LN}$. Since the partition function is known, they fix p and calculate d_L for that value. The critical probability is that of the Nishimori line which intersects the phase transition line. They then calculate d_L for different values of L. At the crossing of these lines must be the critical probability since it is a fixed point. They find this fixed point value to be $p_c = .1094 \pm 0.0002$. Figure 9 is the graph representing their findings.

We now discuss the findings of [5].

The method of approach in [5] was different than that of [3]. Assuming a lattice of dimension $L \times L$, they then generate a sample by assigning 1 and -1 to sites with probability $1-p$ and p respectively. This generates an error chain E on the dual lattice containing all the -1 bonds. They then find the chain E' in the dual lattice that closes the loop with the minimum number of bonds. Figure 8 also happens to demonstrate this process. This is done with Edmond's matching algorithm, a well known algorithm in graph theory. They can then find if $D = E + E'$ is homologically trivial. If so then it is a success and if not it is a failure. They define a function $P_{\text{fail}}(p)$ that determines the probability of D being nontrivial given a value of p. Since $P_{\text{fail}}(p)$ should go to 0 for $p < p_c$ and should increase dramatically for $p > p_c$, they plot $P_{\text{fail}}(p)$ vs. p for different L values and their crossing should be p_c, a fixed point once again. Figure 10 shows their findings. They find $p_c = 0.1031 \pm 0.00001$. This is a significant difference compared to [3].



Figure“9: Graph from [3]. The y-axis is the domain wall free energy and the x-axis is normalized probability centered at the critical probability. They plot the energy for 5 different lattice sizes. Their intersection is a fixed point and represents the critical probability.



Figure“10: Graph from [5]. The y-axis represents the probability of failure in recovery and the x-axis represents the error rate. They plot data points for 6 different lattice sizes. The intersection of these plots is a fixed point and represents the critical probability.

5.2 Discussion of Results

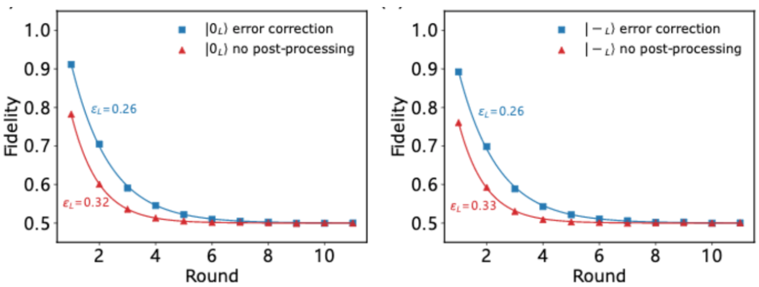
While the difference in the value of p_c is not orders of magnitude apart, it is different enough to bring up discussion. [5] reports values from other papers with a few agreeing with the findings of [4] and a few agreeing with the findings of [3]. It reports one paper found a value roughly in between these values at 0.105. The largest reason for the difference in threshold is that they calculated different physical properties to find p_c. According to [5] the domain wall free energy is more sensitive to parameters. This would seem to imply that the result of [5] is more accurate. We believe that the method used by [5] is more in line with the original idea discussed in [2] and thus is probably a more accurate threshold.

While the results are different, we note that they are close enough that a threshold can likely still be established. If the probability of an error is below 0.1, then both papers agree that an arbitrarily large lattice can correct any error.

6 Implementation

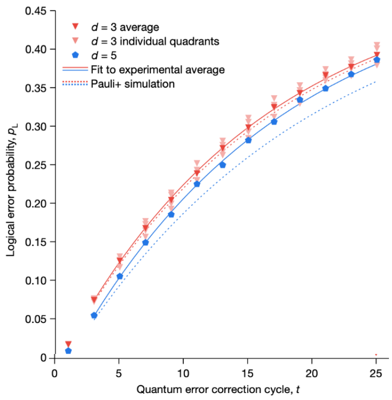
We summarize the results from two from this past year.

- In 2022, [6] show a 17-qubit surface code implementation that is able to correct logical errors. The device they use is the Zuchongzhi 2.1 superconducting quantum processor with 9 data qubits and 8 measurement qubits. The measurement qubits are $4Z$-type ancilla qubits and $4X$-type ancilla qubits.
- Earlier in 2023, Google Quantum AI [1] have tried to simulate the surface code on their Sycamore device with 72 transmon qubits and 121 tunable couplers. They use a subset of these consisting of 25 data qubits and 24 measurement (ancilla) qubits. They also use a variant of the surface code and thus use a $ZXXZ$



Figure“11: Shows the fidelity of logical $|0\rangle_L$ and the $|-L\rangle$ state is higher with the number of surface code cycles with error correction (blue line with squares) compared to the one without (red line with triangles). [6]

stabilizer unlike the the ZZ and XX as we are familiar with. Fig 12 shows that the distance 5 surface code does slightly better than the distance 3 code to suppress logical error.



Figure“12: Google Quantum AI show that the distance-5 performs slightly better than the distance-3 surface code. [1]

7 Conclusion and Outlook

Surface codes offer a solution to to resolve error correction in lattice based quantum computing platforms like superconducitng qubits. Stabilizers are operators that anti-commute with themselves but commute with the rest of the lattice help detect an error without measuring the data qubit via a syndrome extraction. Scaling a surface code means increasing the lattice and distance. The Ising Model motivates an error model for the surface code. The model makes the hypothesis that as you scale the lattice you can correct more errors. This was put to the test by the [6] and [1], and it was found that there is a slight increase in error suppression with an increase in distance of the surface code compared to when no surface code was applied.

In the future, we would like to understand the threshold accuracy of the surface code and its limits. Further work would involve seeing at what limit the surface code breaks down while scaling. We would like to collaborate with our collegues Preetham Tikkireddi and Atharva Vidwans to simulate these codes and find the threshold limit.

7.1 Acknowledgements

We would like to acknowledge Prof. Alex Levchencko for helping us understand the Ising Model and emphasizing the need to take Statistical Mechanics to understand where the Nishimori point comes from.

References

[1] Google Quantum AI. “Suppressing quantum errors by scaling a surface code logical qubit”. In: *Nature* 614.9 (2023), pp. 676–681. DOI: 10.1038/s41586-022-05434-1. URL: https://doi.org/10.1038/s41586-022-05434-1.

[2] Eric Dennis et al. “Topological quantum memory.” In: *J. Mth. Phys.* 43.9 (2002), pp. 4452–4505. DOI: 10.1063/1.1499754. URL: https://doi.org/10.1063/1.1499754.

[3] A. Honecker, M. Picco, and P. Pujol. “Universality Class of the Nishimori Point in the 2D $\pm J$ Random-Bond Ising Model”. In: *Phys. Rev. Lett.* 87 (4 Aug. 2001), p. 047201. DOI: 10.1103/PhysRevLett.87.047201. URL: https://link.aps.org/doi/10.1103/PhysRevLett.87.047201.

[4] Joschka Roffe. “Quantum error correction: an introductory guide”. In: *Contemporary Physics* 60.3 (2019), pp. 226–245. DOI: 10.1080/00107514.2019.1667078. URL: https://doi.org/10.1080/00107514.2019.1667078.

[5] Chenyang Wang, Jim Harrington, and John Preskill. “Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory”. In: *Annals of Physics* 303.1 (2003), pp. 31–58. ISSN: 0003-4916. DOI: https://doi.org/10.1016/S0003-4916(02)00019-2. URL: https://www.sciencedirect.com/science/article/pii/S0003491602000192.

[6] Youwei Zhao et al. “Realization of an Error-Correcting Surface Code with Superconducting Qubits”. In: *Phys. Rev. Lett.* 129 (3 Aug. 2022), p. 030501. DOI: 10.1103/PhysRevLett.129.030501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.129.030501.