Summer 2021 USRA Final Report

Rio Weil

October 24, 2021

Introduction:
The following is a summary of research work conducted with Dr. Robert Raussendorf, as a NSERC USRA Summer 2021 project.

Contents

1 Quantum Error Correction with the Shor Code

1.1 Problem Statement

The following problem is taken from the first homework set of Dr. Raussendorf's 2021W PHYS 523B course about Quantum Error Correction and Fault-Tolerant Quantum Computing.

Compute the concatenation threshold for the 9 -qubit Shor code, assuming the error correction procedure is as discussed in class and the decoherence is described by independent stochastic Pauli noise $\mathcal{T}$ on each qubit,

$$\mathcal{T} = (1-p)[I] + \frac{p}{3}([X]+[Y]+[Z]). \tag{1}$$

As a reminder, we recall that the Shor code is given by:

$$|0\rangle \mapsto \overline{|0\rangle} = \frac{(|000\rangle+|111\rangle)(|000\rangle+|111\rangle)(|000\rangle+|111\rangle)}{2\sqrt{2}} = \overline{|+\rangle}\,\overline{|+\rangle}\,\overline{|+\rangle} \tag{2}$$

$$|1\rangle \mapsto \overline{|1\rangle} = \frac{(|000\rangle-|111\rangle)(|000\rangle-|111\rangle)(|000\rangle-|111\rangle)}{2\sqrt{2}} = \overline{|-\rangle}\,\overline{|-\rangle}\,\overline{|-\rangle} \tag{3}$$

and the concatenation threshold is the value p_{th} such that if the physical error rate p is less than p_t, then the logical error rate goes to 0 in the limit of infinite concatenation.

1.2 First Solution - Second Order Approximation

We look at how the error channel behaves under concatenation and try to obtain an analytical solution up to second order in the error p. We start with:

$$\mathcal{T} = (1-p)[I] + \frac{p}{3}\left([X]+[Y]+[Z]\right)$$

We generalize this expression slightly to the case where X, Y, Z are not necessarily equal; we denote the indidual error probabilities as p_x, p_y, p_z with $p_{tot} = p_x + p_y + p_z$. We then have:

$$\mathcal{T} = (1-p_{tot})[I] + p_x[X] + p_y[Y] + p_z[Z] \tag{4}$$

Under concatenation (i.e. we apply this noise channel independently on all 9 qubits of the Shor code) we have:

$$\begin{aligned}
\mathcal{T}_9 = (1-p_{tot})^9[I^{\otimes 9}] &+ (1-p_{tot})^8 \sum_{i=1}^{9}\left(p_x[I^{\otimes 8}X_i] + p_y[I^{\otimes 8}Y_i] + p_z[I^{\otimes 8}Z_i]\right) \\
&+ (1-p_{tot})^7 \sum_{i=1}^{9}\sum_{j>i}^{9}\left(p_A p_B[I^{\otimes 7}A_iB_j]\right) && (A,B \in \{X,Y,Z\}) \\
&+ (1-p_{tot})^6 \sum_{i=1}^{9}\sum_{j>i}^{9}\sum_{k>j}^{9}\left(p_A p_B p_C[I^{\otimes 6}A_iB_jC_k]\right) && (A,B,C \in \{X,Y,Z\}) \\
&+ \dots
\end{aligned}$$

In the above expression, we note that the sums are taken over all possible combinations of $A, B, \dots$ for each term, and the $I^{\otimes n}$ indicates that there are no errors on the other qubits. Note that there are terms for 4, 5, 6, 7, 8, 9 errors in the expression for $\mathcal{T}_9$ above that we omit for the sake of brevity. We truncate this expression

to second order in p_i:

$$\mathcal{T}_9 \approx (1 - 9p_{tot} + 36p_{tot}^2)[I^{\otimes 9}] + (1-8p)\sum_{i=1}^{9}\Big(p_x[I^{\otimes 8}X_i] + p_y[I^{\otimes 8}Y_i] + p_z[I^{\otimes 8}Z_i]\Big) + \sum_{i=1}^{9}\sum_{j>i}^{9}\Big(p_A p_B[I^{\otimes 7}A_iB_j]\Big) \qquad (A,B \in \{X,Y,Z\})$$

Expanding out the third term into the 6 differnet possible pairs of X, Y, Z, we have:

$$\begin{aligned}\mathcal{T}_9 \approx (1 - 9p_{tot} + 36p_{tot}^2)[I^{\otimes 9}] &+ (1-8p)\sum_{i=1}^{9}\Big(p_x[I^{\otimes 8}X_i] + p_y[I^{\otimes 8}Y_i] + p_z[I^{\otimes 8}Z_i]\Big) \\ &+ \sum_{i=1}^{9}\sum_{j>i}^{9}\Big(p_x^2[I^{\otimes 7}X_iX_j]\Big) \\ &+ \sum_{i=1}^{9}\sum_{j>i}^{9}\Big(p_y^2[I^{\otimes 7}Y_iY_j]\Big) \\ &+ \sum_{i=1}^{9}\sum_{j>i}^{9}\Big(p_z^2[I^{\otimes 7}Z_iZ_j]\Big) \\ &+ \sum_{i=1}^{9}\sum_{j>i}^{9}\Big(p_xp_y[I^{\otimes 7}X_iY_j]\Big) + \sum_{i=1}^{9}\sum_{j>i}^{9}\Big(p_yp_x[I^{\otimes 7}Y_iX_j]\Big) \\ &+ \sum_{i=1}^{9}\sum_{j>i}^{9}\Big(p_xp_z[I^{\otimes 7}X_iZ_j]\Big) + \sum_{i=1}^{9}\sum_{j>i}^{9}\Big(p_zp_x[I^{\otimes 7}Z_iX_j]\Big) \\ &+ \sum_{i=1}^{9}\sum_{j>i}^{9}\Big(p_yp_z[I^{\otimes 7}Y_iZ_j]\Big) + \sum_{i=1}^{9}\sum_{j>i}^{9}\Big(p_zp_y[I^{\otimes 7}Z_iY_j]\Big)\end{aligned}$$

With our expression simplified such that it is more workable, we consider which of these terms contribute to a logical X error, a logical Y error, and a logical Z error. To start, the 9-fold tensor product of the identity (i.e. the $[I^{\otimes 9}]$ term) results in no error. Additionally, any 1-qubit errors can be corrected by the shor code, so all of the terms corresponding to single qubit errors do not contribute to any logical error.

Moving onto terms that can actually contribute error, we look at the 2 qubit errors.

For XX, we have that a logical X error occurs if we get two bit flip errors on the same block (e.g. qubits 1 and 2). There are 9 ways we can have two bit flips on the same block (leading to approximate error probability $9p_x^2$) and 27 ways we can have two bit flip errors on differnet blocks (e.g qubits 1 and 4, where no errors occur).

For YY, we have that a logical X error occurs if we get two Y errors on the same block (the phase flips cancel) and a logical Z error occurs if we get two Y errors on different blocks. Hence, there are 9 ways we can get a logical bit flip error (with error probability $9p_y^2$) and 27 ways we can get a logical phase flip error (with error probability $27p_y^2$).

For ZZ, we have that a logical Z error occurs if we get two Z errors on different blocks (if they are on the same block, the phase flip cancells out). Hence, we have that we have 27 ways we can get a logical phase flip error (with error probability $27p_z^2$) and 9 ways that two phase flips occur but we get no logical error.

For XY/YX, we have that a logical X error occurs if we get the X and Y errors occuring on the same block. There are 18 different permutations in which this can occur (the other permutations the errors are on different blocks, and are corrected independently) with error probability of $18p_xp_y$.

For XZ/XZ, no logical error occurs as the bit and phase flip errors are corrected on different levels independently.

For YZ/ZY, we have that a logical Z error occurs if we get the Z and Y errors occuring on differnet blocks. There are 54 ways in which this can occur (and 18 ways in which they can be on the same block, causing the phase flips to cancel and the bit flip error to be corrected). Therefore, there is a logical Z error with probability $54p_yp_z$.

Putting all of this together, we have that (to second order in the errors p_i) under concatenation the errors transform as:

$$\begin{aligned}
&X : p_x \mapsto 9p_x^2 + 18p_xp_y + 9p_y^2 = 9(p_x + p_y)^2 \\
&Y : p_y \mapsto 0 \\
&Z : p_z \mapsto 27p_y^2 + 54p_yp_z + 27p_z^2 = 27(p_y + p_z)^2
\end{aligned}$$

Indeed, we see that the Y errors become supressed (to second order, there are no logical Y errors as a result of Shor code concatenation). Although these recursion relations are initially coupled, we can consider adding the Y equation to the X and Z equations to get:

$$\begin{aligned}
&X + Y : p_x + p_y \mapsto 9p_x^2 + 18p_xp_y + 9p_y^2 = 9(p_x + p_y)^2 \\
&Y + Z : p_y + p_z \mapsto 27p_y^2 + 54p_yp_z + 27p_z^2 = 27(p_y + p_z)^2
\end{aligned}$$

Note that the RHS is unaffected as the Y error is supressed under concatenation. Then, we can introduce the parameters $p = p_x + p_y$ and $q = p_y + p_z$ to get:

$$\begin{aligned}
p &\mapsto 9p^2 \\
q &\mapsto 27q^2
\end{aligned}$$

In order to see the threshold values for p, q for the error to be reduced under concatenation, we solve the equations (taking the nonzero solution)

$$\begin{aligned}
9p^2 - p = 0 \implies p(9p - 1) = 0 \implies p = \frac{1}{9} \approx 0.111 \\
27q^2 - q = 0 \implies q(27q - 1) = 0 \implies q = \frac{1}{27} \approx 0.037
\end{aligned}$$

Hence for the error to go to zero under repeated concatenation, we require (approximately) that:

$$\begin{aligned}
p_x + p_y &< \frac{1}{9} \\
p_y + p_z &< \frac{1}{27}
\end{aligned}$$

We have that $p_x = p_y = p_z = \frac{p}{3}$ for the depolarizing channel, so we have that:

$$\begin{aligned}
\frac{2p}{3} &< \frac{1}{9} \\
\frac{2p}{3} &< \frac{1}{27}
\end{aligned}$$

The second equation provides a tighter bound on p so we may ignore the first. Solving for the (approximate)

threshold p, we have that:

$$p < \frac{1}{18} \approx 0.056$$

So we obtain that:

$$\boxed{p_t \approx 0.056}$$

Although this solution is extremely approximate, we will find that it agrees in order of magnitude to the proceeding numerical simulation and the analytical solution. This value turns out to be a slight underestimate.

1.3 Second Solution - Numerical Simulation with Qiskit

We simulate the Shor Code using IBM's "Qiskit", a Python quantum computing package. Doing so, we can extract the error concatenation threshold numerically. The detailed solution with code and explanations can be found in `HW1Q3Qiskit.ipynb` at `https://github.com/RioWeil/QI-USRA-2021`, so we do not go over it here. The final result was found to be:

$$\boxed{p_t = 0.075 \pm 0.01}$$

1.4 Third Solution - Analytic Solution (Inefficient)

We use a computer program to compute all 4^9 possible combinations of Pauli-operators on 9 qubits (there are 4 options, I, X, Y, Z) for each qubit. Note that the program for this classification (and all of the gritty symbolic computing) can be found in `HW1Q3Analytic1.ipynb`. Then, the computer program classifies the operators based on whether they would result in no logical error, a logical X error, a logical Y error, or a logical Z error. As a refresher, the Shor code is capable of correcting 1 qubit errors at each level of concatenation. The protocol for determining whether a Pauli string generates a logical X error is then as follows:

(1) Break up the 9-qubit code into three blocks (123, 456, 789).

(2) Count the number of X errors in each block (from either X_i or Y_i). It is labelled as erroneous if there are 2 or more such errors in the block.

(3) Count how many blocks are erroneous. If 1 or 3 blocks are erroneous, a logical X error occurs. If 0 or 2 blocks are erroneous, a logical X error does not occur.

Similarly, the protocol for determining whether a Pauli string generates a logical Z error is as follows:

(1) Break up the 9-qubit code into three blocks (123, 456, 789).

(2) Count the number of Z errors in each block. If there are 1 or 3 Z errors in a given block (from either Z_i or Y_i), then the block is labelled as erroneous. If there are 0 or 2 zero errors in a given block, then the block is not erroneous.

(3) Count how many blocks are erroneous. If 2 or more blocks are errnoneous, then a logical Z error occurs. Otherwise, a logical Z error does not occur.

Finally, a Y error occurs if both a logical X and logical Z error occurs. Repeating this process for each of the 4^9 Pauli strings leads to grouping into these 4 logical error types.

After, the errors within each logical error type are grouped into equivalence classes based on the same probability of occuring; for example, both $IIXXXXXXX$ and $XXXXXXXII$ occur with probability $p_x^7(1 - p_x - p_y - p_z)^2$. This grouping is done by using a dictionary data structure. After this, p'_x, p'_y and p'_z (that is, the probability of getting a logical $X/Y/Z$ error after one level of concatenation of the code) are computed by summing over all the probabilities in each classification. We find that all three are polynomials that are functions of p_x, p_y, p_z; in other words, highly coupled (the full gory glory of these polynomials can be found in the Jupyter notebook). However, we can make (as we did for the approximate solution) the substitution of $p = p_x + p_y$, $q = p_y + p_z$ to find that the recursion relations decouple[1]. Computing $p' = p'_x + p'_y$ and $q' = p'_y + p'_z$ we get:

$$p' = -32p^9 + 144p^8 - 216p^7 + 84p^6 + 72p^5 - 54p^4 - 6p^3 + 9p^2 \tag{5}$$

$$q' = -128q^9 + 576q^8 - 1152q^7 + 1344q^6 - 1008q^5 + 504q^4 - 162q^3 + 27q^2 \tag{6}$$

To solve for the concatenation thresholds of p and q, we can solve for values of p, q that satisfy $p' = p, q' = q$. Doing this numerically, we obtain that this occurs at $p = 0.1341$, $q = 0.0499$. We take this threshold value for q as it is the tighter bound of the two. We now recall that the probabilities between the three Pauli errors were equally split, with $p/3$ for each (p now refers to the *physical* error rate, as it did originally). Hence, solving for the threshold probability p_t, we have that:

$$p_y + p_z = \frac{p_t}{3} + \frac{p_t}{3} = q_t = 0.0499$$

Which solving for p_t, we get:

$$\boxed{p_t = 0.0748}$$

However, we can even do slightly better, and obtain a completely closed form analytic solution. We start by observing what the polynomial q' looks like when plotted: Looking at this plot, we immediately notice

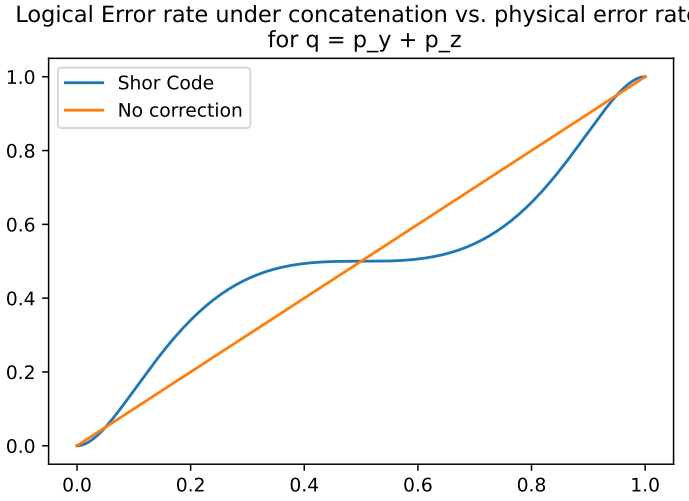

Figure 1: Plot of q' against q for the Shor code and for the case of no error correcting code.

[1]This is actually a phenomena that can be observed in general for the entire class of CSS codes, of which the Shor code is a member.

that there are five roots, three of which can be immediately obtained as $q = 0$, $q = 0.5$, $q = 1$. It might seem like our hopes end here, as even if we are able to factor these out, we are left with a degree 6 polynomial which (thanks to Galois) we know in general does not have a closed form rational expression. However, we make the observation that the plot of q' is symmetric about $q = 0.5$. Making the substitution of $w = q - 0.5$ to $-128q^9 + 576q^8 - 1152q^7 + 1344q^6 - 1008q^5 + 504q^4 - 162q^3 + 27q^2 - q$, we obtain:

$$-128w^9 + 6w^3 - w$$

which we can factor to obtain:

$$-w(2w-1)(2w+1)(32w^6 + 8w^4 + 2w^2 - 1)$$

Now, the last term is a cubic polynomial in w^2, so we can apply the cubic formula to obtain the roots for w^2. We can then take the square root of this, and take the real roots as our analytical solution for w_t. Undoing the shift by 0.5 to obtain back q_t, we get:

$$q_t = -\sqrt{-\frac{1}{12} - \frac{1}{72\left(61/3456 + \sqrt{417/1152}\right)^{1/3}} + \left(61/3456 + \sqrt{417/1152}\right)^{1/3}} + \frac{1}{2}$$

which we can multiply by $\frac{3}{2}$ (as before) to obtain the concantenation threshold p_t as:

$$\boxed{p_t = -\frac{3}{2}\sqrt{-\frac{1}{12} - \frac{1}{72\left(61/3456 + \sqrt{417/1152}\right)^{1/3}} + \left(61/3456 + \sqrt{417/1152}\right)^{1/3}} + \frac{3}{4}}$$

1.5 Fourth Solution - Analytic Solution (Efficient)

In this solution, we take a similar approach to the previous one but find a much more efficient, elegant, and motivated approach. The details can be found in `HW2Q3Analytic2.ipynb`. The prime insight is that the Shor code is the concatenation of the 3 qubit bit flip code with the 3 qubit phase flip code. Hence, if we study how the Pauli errors behave under application of these codes, we can then compose the two results to obtain the recursion relations for p'_x, p'_y, and p'_z as before. The gain here is that instead of checking $4^9 = 262144$ cases of Pauli operators, we instead only check $4^3 = 64$ cases; a 4 order of magnitude speedup!

The process for error classification for the bit flip code is the same as discussed in the previous solution. If there are 2 or more physical X errors (caused by X or Y), then an X logical error occurs. If there are 1 or 3 physical Z errors (caused by Z or Y), then a logical Z error occurs. If both an X and Z errors occur, then a Y logical error occurs. After this classification, we find $p'_{xbf}, p'_{ybf}, p'_{zbf}$ (the probabilities of logical $X/Y/Z$ errors after use of the bit flip code) under one level of concatenation to be:

$$p'_{xbf} = p_x^3 + 3p_x^2(-p_x - p_y - p_z + 1) + 3p_x p_y^2 + 6p_x p_y p_z + 3p_y^2(-p_x - p_y - p_z + 1) \tag{7}$$

$$p'_{ybf} = 3p_x^2 p_y + 3p_x^2 p_z - 6p_x p_y(p_x + p_y + p_z - 1) + p_y^3 + 3p_y^2 p_z \tag{8}$$

$$p'_{zbf} = -6p_x p_z(p_x + p_y + p_z - 1) + 3p_y p_z^2 + 3p_y(p_x + p_y + p_z - 1)^2 + p_z^3 + 3p_z(p_x + p_y + p_z - 1)^2 \tag{9}$$

In another stroke of insight, we realize that we do not have to repeat the same calculation for the 3-qubit phase flip code. The bit and phase flip codes are symmetric, so we can find $p'_{xpf}, p'_{ypf}, p'_{zpf}$ for the phase flip code simply by interchanging p'_{xbf} and p'_{zbf} in the bit flip case, and then replacing all instances of p_x

with p_z and p_z with p_x. This yields:

$$p'_{xpf} = -6p_z p_x(p_z + p_y + p_x - 1) + 3p_y p_x^2 + 3p_y(p_z + p_y + p_x - 1)^2 + p_x^3 + 3p_x(p_z + p_y + p_x - 1)^2 \tag{10}$$

$$p'_{ypf} = 3p_z^2 p_y + 3p_z^2 p_x - 6p_z p_y(p_z + p_y + p_x - 1) + p_y^3 + 3p_y^2 p_x \tag{11}$$

$$p'_{zpf} = p_z^3 + 3p_z^2(-p_z - p_y - p_x + 1) + 3p_z p_y^2 + 6p_z p_y p_x + 3p_y^2(-p_z - p_y - p_x + 1) \tag{12}$$

To obtain the logical errors under a level of concatenation of the Shor code (p'_x, p'_y, p'_z), we can substitue $p'_{xbf}, p'_{ybf}, p'_{zbf}$ into p_x, p_y, p_z in the expressions of $p'_{xpf}, p'_{ypf}, p'_{zpf}$. Doing so and making the the same substitution of $p = p_x + p_y, q = p_y + p_z$, we obtain the same recursion relatons for p, q as in the previous solution, from which the extraction of the concetenation threshold of the Shor code follows.

2 Jordan Wigner and SPT Phases

2.1 Background and Problem Statement

A fundamental open problem that currently persists the field of quantum computation and information is the source of the computational power/advantage that quantum computers have over classical computers. Results such as the super-polynomial speedup brought by Shor's factoring algorithm, as well as the exponentially-growing amount of information necessary for a classical simulation (2^N coefficients are required to keep track of N qubits, and $2^N \times 2^N$ unitary matrices to act on them) seem to point to the existence of this advantage, but the source of this power seems to be inconclusive. While answers such as entanglement may be intuitive (and while quantum entanglement is certainly *necessary* for the advantage, as without entanglement the quantum simulation would be easily classically tractable), there are several famous results in the field that argue against this being the only answer. These include the Gottesman-Knill theorem, which shows that Clifford gates (of which entangling CNOTs are a part) are efficiently classically simulable through the stabilizer formalism, as well as a result that shows that most states in the Hilbert space are "too entangled" to be useful for quantum computation.

To the end of adressing this problem, the paradigm of measurement based quantum computation turns out to be useful. Unlike in the gate-based model where the computational "power" is distributed throughout the various unitary gates that are used (with a trivial measurement in the computational basis at the end), in MBQC the computation follows via (computationally trivial) single-qubit measurements on an initial resource state. Hence, this initial resource state can be thought to contain all the "power" of the quantum computation, and hence study of these resource states are useful in the identification and classification of quantum computational resources.

In this line, past research has delved into the symmetry and symmetry-protected-topological-order of such resource states in order to learn more about their computational power. However, a problem arises in that computational phases are in general defined only in the thermodynamic limit, while in reality computation is necessarily carried out on systems of finite size. To this end, Arnab (a current Masters and soon-to-be PhD student in the group) has shifted the focus of study to that of the string order parameter of such finite states. Working with the 1 dimensional interpolated cluster state, defined as the ground state of:

$$H(\alpha) = -\cos\alpha \sum_j Z_{j-1} X_j Z_{j+1} - \sin\alpha \sum_j X_j \tag{13}$$

It has been shown that the expectation value of the string order parameter:

$$K_j = Z_j X_{j+1} I_{j+2} X_{j+3} \dots Z_N \tag{14}$$

equals 1 when $\alpha = 0$, where we have the cluster state (which is the standard resource state for MBQC) and drops to 0 as we pass $\alpha = \frac{\pi}{4}$ and we go into the cluster phase (at $\alpha = \frac{\pi}{2}$, the ground state is just the product state of $|+\rangle$s, which has no entanglement and hence no computational power). We will not go deep into the theory or the derivation of this string order parameter here (this discussion is left for Arnab's soon-to-be-submitted Master's thesis), but the main takeaway is that there exists an SOP defined for finite systems whose expectation value corresponds to the quantum computational power of the system for MBQC. This is clearly of interest as it gives us a tool to classify quantum computational resources in the finite regime.

Arnab's numerical results were obtained using Tensor Network/Matrix Product State methods, finding the ground states and the expectation value of the SOP through use of a DMRG algorithm (as an aside; although this computation is in general an exponentially hard problem, the localized entanglement of the cluster state and the interpolated cluster state makes it such that the computations can be efficiently carried out using MPS methods. We again defer to Arnab's thesis for the details). However, it would be nice to obtain a corresponding analytical result to match these numerics. This is where the summer project comes in.

There exists a standard condensed matter technqiue known as the Jordan-Wigner transform. The insight this transform provides is that there exists a map between N spin 1/2 particles and N spinless fermions. While the interpolated cluster state Hamiltonian may not be diagonalizable in the spin picture, the hope with the tranformation is that such a diagonalization will be possible in the fermion picture. Indeed, there have been results where it has been shown that the diagonalization is possible *if the Hamiltonian is quadratic in the fermionic operators* (it may be possible in other cases, but the known methods are for the quadratic case). After the eigenspectrum and ground state is obtained, one can transform back into the spin picture to attempt to calculate the expectation value of the string order parameter.

An excellent discussion of the Jordan-Wigner transform is provided by Michael Nielsen at `https://michaelnielsen.org/blog/archive/notes/fermions_and_jordan_wigner.pdf`. The bare necessary facts we will use for this problem are:

$$\{a_j, a_k\} = 0 \tag{15}$$

$$\{a_j, a_k^\dagger\} = \delta_{jk} I \tag{16}$$

$$Z_j = a_j a_j^\dagger - a_j^\dagger a_j \tag{17}$$

$$X_j = -(Z_1 \dots Z_{j-1})(a_j + a_j^\dagger) \tag{18}$$

where the first two relations are the fermionic canonical commutation relations (where $a_j, a_j^\dagger$ are the annhilation and creation operators respectively) and the second two relations are what result from the inversion of the JW transform. Note that the commutation relations along with the X_j transformation to obtain that:

$$X_j X_{j+1} = (a_j^\dagger - a_j)(a_{j+1} + a_{j+1}^\dagger) \tag{19}$$

We now proceed to the obtained results.

2.2 A Note on Boundary Conditions

It turns out that the choice of boundary conditions in this problem is a subtle but crucial choice. There are three possibilities for the choice. The first is the open boundary condition case, where:

$$H(\alpha) = -\cos\alpha \sum_{j=2}^{N-1} Z_{j-1} X_j Z_{j+1} - \sin\alpha \sum_{j=1}^{N} X_j. \tag{20}$$

We see that the $i = 1$ and $i = N$ terms are excluded. We note that for $\alpha = 0$, the ground state of this Hamiltonian is in fact four-fold degenerate; there are only $N - 2$ stabilizers specified with the open BCs, and as a result, the ground state has a choice of ± 1 at each end; this results in a 4-fold degenereacy of the ground state (which can be problematic, as it is not clear which of these four ground states would correspond to the cluster state). The next option is the periodic boundary condition case, where:

$$H(\alpha) = -\cos\alpha \left[\left[\sum_{j=2}^{N-1} Z_{j-1} X_j Z_{j+1} \right] + Z_N X_1 Z_2 + Z_{N-1} X_N Z_1 \right] - \sin\alpha \sum_{j=1}^{N} X_j. \tag{21}$$

Here, we have added the $Z_N X_1 Z_2$ and $Z_{N-1} X_N Z_1$ terms/stabilizers at the end of the chain, turning it into a ring. While this case will turn out to be the most analytically clean, there are problems in that the numerics do not line up with the numerical results obtained by Arnab (as per Paul's past work on this topic). Finally, we have the closed boundary condition case, where:

$$H(\alpha) = -\cos\alpha \left[\left[\sum_{j=2}^{N-1} Z_{j-1} X_j Z_{j+1} \right] + X_1 Z_2 + Z_{N-1} X_N \right] - \sin\alpha \sum_{j=1}^{N} X_j. \tag{22}$$

We can see that we have added two stabilizers compared to (20) (and hence our ground state is unique and the cluster state as we desire), and this is the theoretical "correct" form of the Hamiltonian for the cluster state on a chain. Indeed, this is the Hamiltonian as used by Arnab in the process of the numerical diagonalization. However, this case has a large fundamental problem; it is not analytically tractable with the Jordan-Wigner transform method. To see this, we recall that we require that the Hamiltonian be quadratic in the fermionic operators in order for the diagonalization to be carried out in the fermionic picture using standard methods. Applying the Jordan-Wigner transform to the $X_1 Z_2$ term, we have:

$$X_1 Z_2 = -I(a_1 + a_1^\dagger)(a_2 a_2^\dagger - a_2^\dagger a_2) = a_1 a_2 a_2^\dagger - a_1 a_2^\dagger a_2 + a_1^\dagger a_2 a_2^\dagger - a_1^\dagger a_2^\dagger a_2. \tag{23}$$

From this, we see that the $X_1 Z_2$ term brings about terms which are of order 3 in the Hamiltonian, dashing any hope of an analytical diagonailization in this scenario (the $Z_{N-1} X_N$ term is similarly problematic). Hence, with our attempt, there exist only two possible routes; the open boundary case as given in (20), and the periodic boundary case as given in (21).

2.3 Open BCs - Jordan Wigner Transformation

Because pairs of X operators will be much easier to work with (as can be seen in (19)), we rotate our basis; namely, we map $X_j \mapsto Z_j$ and $Z_j \mapsto -X_j$. With this, our hamiltonian becomes:

$$H(\alpha) = -\cos\alpha \sum_{j=2}^{N-1} X_{j-1} Z_j X_{j+1} - \sin\alpha \sum_{j=1}^{N} Z_j \tag{24}$$

First working with the first term, we have:

$$X_{j-1} Z_j X_{j+1} = X_{j-1} X_{j+1} Z_j = (Z_1 \dots Z_{j-2})(a_{j-1}^\dagger + a_{j-1})(Z_1 \dots Z_j)(a_{j+1}^\dagger + a_{j+1}) Z_j$$

Note that as each of the Z_js (and their products) are even products of annhilation/creation operators (using the fact that $\{a_j, a_k^\dagger\} = \delta_{jk} I$, $\{a_j, a_k\} = 0$), we can move the Z_2 and the $Z_1 \dots Z_{j-2}$ to the center. To show this

is justified, consider the following Lemma:
Lemma. *Let A, B, C be operators. If $\{A, B\} = 0$ and $\{A, C\} = 0$, then $[A, BC] = 0$.*

Proof. If $\{A, B\} = 0$, then $AB = -BA$ and similarly if $\{A, C\} = 0$ then $AC = -CA$. It then follows that:

$$ABC = (AB)C = (-BA)C = -B(AC) = -B(-CA) = BCA$$

and hence $[A, BC] = 0$ as claimed. □

Hence we may move the Z_js to the center as convenient:

$$\begin{aligned}
X_{j-1}Z_jX_{j+1} &= (a^\dagger_{j-1} + a_{j-1})(Z_1 \dots Z_{j-2})(Z_1 \dots Z_{j-1})(a^\dagger_{j+1} + a_{j+1}) \\
&= (a^\dagger_{j-1} + a_{j-1})Z_{j-1}(a^\dagger_{j+1} + a_{j+1}) \\
&= (a^\dagger_{j-1} + a_{j-1})(a_{j-1}a^\dagger_{j-1} - a^\dagger_{j-1}a_{j-1})(a^\dagger_{j+1} + a_{j+1}) \\
&= (a^\dagger_{j-1}a_{j-1}a^\dagger_{j-1} - a_{j-1}a^\dagger_{j-1}a_{j-1})(a^\dagger_{j+1} + a_{j+1}) \\
&= (a^\dagger_{j-1}(I - a^\dagger_{j-1}a_{j-1}) - a_{j-1}(I - a_{j-1}a^\dagger_{j-1}))(a^\dagger_{j+1} + a_{j+1}) \\
&= (a^\dagger_{j-1} - a_{j-1})(a^\dagger_{j+1} + a_{j+1})
\end{aligned}$$

Where we use the commutation relations and the fact that creation/annhilation operators on the same site square to zero. We can also apply the transform on the $\sum Z_j$ term to conclude that our Hamiltonian in the fermionic picture is:

$$\boxed{H(\alpha) = -\cos\alpha \sum_{j=2}^{N-1}(a^\dagger_{j-1} - a_{j-1})(a^\dagger_{j+1} + a_{j+1}) - \sin\alpha \sum_{j=1}^{N}(a_j a^\dagger_j - a^\dagger_j a_j)} \tag{25}$$

This completes the transformation of the Hamiltonian.

2.4 String Order Parameter - Jordan Wigner Transformation

Next, we find the transform the string order parameter. We recall that the SOP is given by:

$$K_j = I_1 I_2 \dots Z_j X_{j+1} I_{j+2} \dots I_{N-2} X_{N-1} Z_N = Z_j X_{j+1} I_{j+2} \dots I_{N-2} X_{N-1} Z_N$$

Note that we assume that the length of the SOP is *odd*. After rotation, the SOP becomes:

$$K_j = X_j Z_{j+1} I_{j+2} \dots I_{N-2} Z_{N-1} X_N$$

Applying the Jordan-Wigner transformation (partially), we have:

$$K_j = (Z_1 \dots Z_{j-1})(a^\dagger_j + a_j)Z_{j+1} I_{j+2} \dots I_{N-2} Z_{N-1}(Z_1 \dots Z_{N-1})(a^\dagger_N + a_N)$$

Again applying the Lemma, we move the Z_j product at the front of the expression to the end of the expression (and then use the fact that $Z_j^2 = I$):

$$K_j = (a^\dagger_j + a_j)Z_{j+1} I_{j+2} \dots I_{N-2} Z_{N-1}(Z_j \dots Z_{N-1})(a^\dagger_N + a_N)$$

Cancelling out the intermediate Z_j terms as well, we obtain:

$$K_j = (a_j^\dagger + a_j)(Z_j Z_{j+2} \dots Z_{N-2})(a_N^\dagger + a_N)$$

We (finally) JW transform the Z terms:

$$\begin{aligned}
K_j &= (a_j^\dagger + a_j)(a_j a_j^\dagger - a_j^\dagger a_j)(a_{j+2}a_{j+2}^\dagger - a_{j+2}^\dagger a_{j+2}) \dots (a_{N-2}a_{N-2}^\dagger - a_{N-2}^\dagger a_{N-2})(a_N^\dagger + a_N) \\
&= (a_j^\dagger a_j a_j^\dagger - a_j a_j^\dagger a_j)(a_{j+2}a_{j+2}^\dagger - a_{j+2}^\dagger a_{j+2}) \dots (a_{N-2}a_{N-2}^\dagger - a_{N-2}^\dagger a_{N-2})(a_N^\dagger + a_N) \\
&= (a_j^\dagger (I - a_j^\dagger a_j) - a_j (I - a_j a_j^\dagger))(a_{j+2}a_{j+2}^\dagger - a_{j+2}^\dagger a_{j+2}) \dots (a_{N-2}a_{N-2}^\dagger - a_{N-2}^\dagger a_{N-2})(a_N^\dagger + a_N) \\
&= (a_j^\dagger - a_j)(a_{j+2}a_{j+2}^\dagger - a_{j+2}^\dagger a_{j+2}) \dots (a_{N-2}a_{N-2}^\dagger - a_{N-2}^\dagger a_{N-2})(a_N^\dagger + a_N)
\end{aligned}$$

Hence our SOP in the fermionic picture is:

$$\boxed{K_j = (a_j^\dagger - a_j)(a_{j+2}a_{j+2}^\dagger - a_{j+2}^\dagger a_{j+2}) \dots (a_{N-2}a_{N-2}^\dagger - a_{N-2}^\dagger a_{N-2})(a_N^\dagger + a_N)} \tag{26}$$

2.5 Open BCs - Numerical Diagonalization

Since the momentum k is a "good" quantum number (in the sense that there exists a basis in which k and the energy are simultaneously diagonal), for an analytical closed form solution, it would be ideal to first Fourier transform (25) to the momentum basis, and then collapse the resulting double sums using the relation:

$$\frac{1}{N}\sum_{j=1}^{n} e^{i(k-k')j} = \delta_{k,k'}. \tag{27}$$

However, in the open boundary condition case, things are not so convenient. In paricular, since the cluster state term only runs from $j = 2$ to $j = N - 1$, the above cancellation cannot be applied in a nice way. Even if we were to add/substract the $j = 1$ and $j = N$ terms as to be able to make the substitution and simplification, the resulting expression will turn out to contain many terms and not be particularly analytically tractable for diagonalization (we will find that the periodic case will be more approachable). Hence, we pursue a numerical diagonalization with computer software instead.

To set this up, we first make the observation that general quadratic fermionic Hamiltonians can be written in the form:

$$H = \sum_{jl}\left(\alpha_{jl}a_j^\dagger a_l - \alpha_{jl}^* a_j a_l^\dagger + \beta_{jl}a_j a_l - \beta_{jl}^* a_j^\dagger a_l^\dagger\right). \tag{28}$$

We can write this as:

$$H = \begin{bmatrix} a_1^\dagger \dots a_n^\dagger a_1 \dots a_n \end{bmatrix} \begin{bmatrix} \alpha & -\beta^* \\ \beta & -\alpha^* \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ a_1^\dagger \\ \vdots \\ a_n^\dagger \end{bmatrix} \tag{29}$$

where α, β are block matrices such that α is Hermitian and β is anti-symmetric. It then suffices to consider

the diagonalization of:

$$\begin{bmatrix} \alpha & -\beta^* \\ \beta & -\alpha^* \end{bmatrix}.$$

We can diagonalize this via a transformation matrix P:

$$D = P \begin{bmatrix} \alpha & -\beta^* \\ \beta & -\alpha^* \end{bmatrix} P^\dagger \tag{30}$$

where D is diagonal and written with respect to some new fermionic basis $\left\{b_1^\dagger, b_2^\dagger, \ldots, b_n^\dagger, b_1, b_2, \ldots, b_n\right\}$. The form of α, β make it such that the b operators will also be fermionic after the diagonalization. To obtain back the original fermionic a operators, we can undo the transformation with:

$$\left[a_1^\dagger \ldots a_n^\dagger a_1 \ldots a_n\right] = \left[b_1^\dagger \ldots b_n^\dagger b_1 \ldots b_n\right] P. \tag{31}$$

Hence, we can use numerics (in this case python) to carry out the diagonalization of the matrix as in (30). From this we obtain the eigenvectors and eigenvalues of the matrix. In particular, we note that the matrices we diagonalize here are of size $2N \times 2N$, and will therefore have $2N$ eigenvalues and eigenvectors. This at first seems like an apparent contradiction as there should only be N eigenvalues; this is resolved in that the $2N$ eigenvalues we obtain come in $\pm\lambda$ pairs; hence, obtaining an eigenenergy of the actual system involves picking one of the two of $\pm\lambda_i$s for $i \in \{1, \ldots, N\}$ and then summing up each choice. The eigenstate of the system is then obtained by applying the eigenvectors corresponding to each of the chosen $\pm\lambda_i$s to the vacuum state $|0\rangle$.

To obtain the ground state of the system, we therefore choose the $-\lambda_i$ outcome for each i. The lowest energy state is obtained by summing up $\sum_{i=1}^{N} -\lambda_i$, and the ground state obtained by applying each of the corresponding eigenvector (fermionic operators) to the vaccum state $|0\rangle$. From here, we can use (31) to express the ground state in terms of the original set of fermionic operators, and then use the resulting expression (which is $|0\rangle$ with some sequence of $a_j^\dagger, a_j$ fermionic operators applied) with (26) (the transformed string order parameter) to calculate the expectation value (at which point we may require consideration of Wick contractions in order to make this computation possible).

Having laid down the process in general, let us now apply it to our Hamiltonian at hand. We start with (25):

$$H(\alpha) = -\cos\alpha \sum_{j=2}^{N-1} (a_{j-1}^\dagger - a_{j-1})(a_{j+1}^\dagger + a_{j+1}) - \sin\alpha \sum_{j=1}^{N} (a_j a_j^\dagger - a_j^\dagger a_j)$$

Expanding out the product we have:

$$H(\alpha) = -\cos\alpha \sum_{j=2}^{N-1} \left(a_{j-1}^\dagger a_{j+1}^\dagger + a_{j-1}^\dagger a_{j+1} - a_{j-1} a_{j+1}^\dagger - a_{j-1} a_{j+1}\right) - \sin\alpha \sum_{j=1}^{N} (a_j a_j^\dagger - a_j^\dagger a_j)$$

Now, recall that we want to express this Hamiltonian as in equations (28), (29). In other words, we need to express this such that the coefficients form block matrices with the desired (Hermitian/anti-symmetric) properties. In order to accomplish this, we use the CCRs (15), (16) to consider that $[a_j, a_k^\dagger] = 0$ if $j \neq k$ and

$[a_j, a_k] = 0$ and hence:

$$a^\dagger_{j-1}a^\dagger_{j+1} = \frac{a^\dagger_{j-1}a^\dagger_{j+1} - a^\dagger_{j+1}a^\dagger_{j-1}}{2}$$
$$a^\dagger_{j-1}a_{j+1} = \frac{a^\dagger_{j-1}a_{j+1} - a_{j+1}a^\dagger_{j-1}}{2}$$

Therefore we can write the Hamiltonian as:

$$H = -\frac{\cos\alpha}{2}\sum_{j=2}^{N-1}\left(a^\dagger_{j-1}a^\dagger_{j+1} - a^\dagger_{j+1}a^\dagger_{j-1} + a^\dagger_{j-1}a_{j+1} - a_{j+1}a^\dagger_{j-1} - a_{j-1}a^\dagger_{j+1} + a^\dagger_{j+1}a_{j-1} - a_{j-1}a_{j+1} + a_{j+1}a_{j-1}\right) \\ - \sin\alpha\sum_{j=1}^{N}\left(a_j a^\dagger_j - a^\dagger_j a_j\right) \tag{32}$$

which we can verify has coefficients such that the block matrix α in (29) is Hermitian and β is anti-symmetric. We can now pass this to sympy to perform the diaognalization for us. Performing this diagonalization, we obtain the following figures:

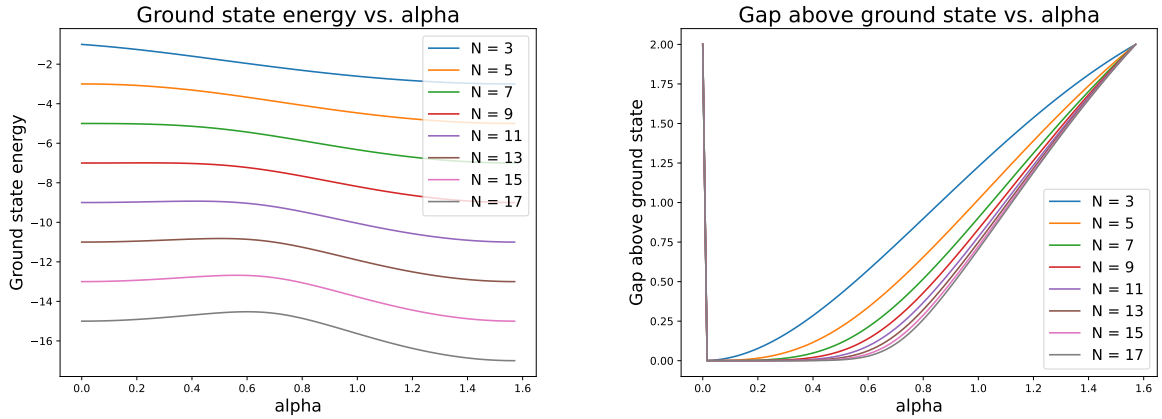

Figure 2: Plots of the Ground state energy and gap above the ground state as a function of α (and chain length N) for the interpolated cluster state chain with open boundary conditions.

Looking at these figures, we see results consistent with our expectations; for the ground state energies, the cluster state has energy $-N+2$ (as is expected as we are missing two stablizers, each which contribute -1) and for the product state we have energy $-N$. We have a smooth interpolation between the two energies as we vary α. Note that if we were dealing with the closed boundary case (as given in Equation (22)), we would expect to see $-N$ at both $\alpha = 0, \frac{\pi}{2}$. For the plot of the gap above the ground state, we again see results that line up with our expectations. At $\alpha = 0$, we have a 4-fold degenerate ground state as two (pairs) of our eigenvalues are 0. This is to be expected as we are missing two stabilizers in the open boundary condition case. We hence have a gap above the GS of 2 at $\alpha = 0$ which drops steeply to ~ 0 as soon as we move away from the $\alpha = 0$ point. From there, we have a smooth rise to a gap of 2 for the product state as we increase α. Note that as we increase α, this rise becomes sharper and sharper.

It is worth noting that these results conflict slightly with the results obtained from Arnab's numericcs on the closed chain. There, the ground state is never degenerate, and we have a gap above the ground state

of 2 at both $\alpha = 0$ and $\alpha = \frac{\pi}{2}$. The gap dips as we approach the center of the plot ($\alpha = \frac{\pi}{4}$) and vanishes completely in the thermodynamic limit.

It is also worth noting that the lack of a continuous gap as we go away from $\alpha = 0$ indicates that we are not dealing with an SPT phase here.

Now that the ground state has been obtained numerically in terms of the vacuum state $|0\rangle$ and the fermionic operators $b_1^\dagger, \ldots, b_n^\dagger, b_1, \ldots, b_n$, the next step is to use equation (31) to obtain the ground state in terms of the original fermionic operators. We can then combine this with the expression for the string order parameter 26 to try to extract the expectation value of the SOP for this ground state as a function of N and α. This is certainly a nontrivial calculation and would require us to create some program that can systematically apply Wick contractions to move around the fermionic operators and calculate this expectation value. While this was not eventually achieved, future work could revisit this step/calculation to see if the results match up to expectations/the previous numerical results obtained.

The notebook containing all of the numerical calculations can be found in `NumericalDiag.ipynb`.

2.6 Periodic BCs - Jordan Wigner Transformation

For the periodic case, we start with the Hamiltonian in (21). The resulting Hamiltonian will be the same as in (25), but will have two extra terms corresponding to $-\cos\alpha X_N Z_1 X_2$ and $-\cos\alpha X_{N-1} Z_N X_1$. Hence, let us calculate the Jordan Wigner transform of these terms (and add them to the previous result).

Let us start with the first term. Note that it is questionable whether the JW transform is even meaningful in the current context, but let us try to apply it and see where it takes us. Appplying the JW transform to the Xs, we have:

$$-\cos\alpha(Z_1 \ldots Z_{N-1}(a_N + a_N^\dagger)Z_1 Z_1(a_2 + a_2^\dagger)) = -\cos\alpha(Z_1 \ldots Z_{N-1}(a_N + a_N^\dagger)(a_2 + a_2^\dagger))$$

Using that $Z_N^2 = I$, we have:

$$-\cos\alpha(X_N Z_1 X_2) = -\cos\alpha(Z_1 \ldots Z_N Z_N(a_N + a_N^\dagger)(a_2 + a_2^\dagger)).$$

Since $Z_1 \ldots Z_N$ is a symmetry of both the cluster state term and the product state term, we have that:

$$Z_1 Z_2 \ldots Z_N = \pm 1 \tag{33}$$

Hence we have that:

$$-\cos\alpha(X_N Z_1 X_2) = \mp\cos\alpha Z_N(a_N + a_N^\dagger)(a_2 + a_2^\dagger).$$

Transforming Z_N, we have:

$$-\cos\alpha(X_N Z_1 X_2) = \mp\cos\alpha(a_N a_N^\dagger - a_N^\dagger a_N)(a_N + a_N^\dagger)(a_2 + a_2^\dagger)$$

using that $a_N^2 = (a_N^\dagger)^2 = 0$, this reduces to:

$$-\cos\alpha(X_N Z_1 X_2) = \mp\cos\alpha(a_N a_N^\dagger a_N - a_N^\dagger a_N a_N^\dagger)(a_2 + a_2^\dagger)$$

using that $a_N a_N^\dagger = I - a_N^\dagger a_N$, and $a_N^\dagger a_N = I - a_N a_N^\dagger$ we have that:

$$-\cos\alpha(X_N Z_1 X_2) = \mp\cos\alpha((I - a_N^\dagger a_N)a_N - (I - a_N a_N^\dagger)a_N^\dagger)(a_2 + a_2^\dagger) = \mp\cos\alpha(a_N - a_N^\dagger)(a_2 + a_2^\dagger)$$

So we therefore have that:

$$-\cos\alpha(X_N Z_1 X_2) = \mp\cos\alpha(a_N^\dagger - a_N)(a_2^\dagger + a_2) \tag{34}$$

Repeating the same process for $-\cos\alpha(X_{N-1}Z_N X_1)$, we have:

$$\begin{aligned}
-\cos\alpha(X_{N-1}Z_N X_1) &= -\cos\alpha(Z_1 \dots Z_{N-2}(a_{N-1} + a_{N-1}^\dagger)Z_N(a_1 + a_1^\dagger)) \\
&= -\cos\alpha(Z_1 \dots Z_{N-2}Z_N(a_{N-1} + a_{N-1}^\dagger)(a_1 + a_1^\dagger)) \\
&= -\cos\alpha(Z_1 \dots Z_{N-2}IZ_N(a_{N-1} + a_{N-1}^\dagger)(a_1 + a_1^\dagger)) \\
&= -\cos\alpha(Z_1 \dots Z_{N-2}Z_{N-1}Z_{N-1}Z_N(a_{N-1} + a_{N-1}^\dagger)(a_1 + a_1^\dagger)) \\
&= -\cos\alpha(Z_1 \dots Z_{N-2}Z_{N-1}Z_N Z_{N-1}(a_{N-1} + a_{N-1}^\dagger)(a_1 + a_1^\dagger)) \\
&= \mp\cos\alpha(Z_{N-1}(a_{N-1} + a_{N-1}^\dagger)(a_1 + a_1^\dagger)) \\
&= \mp\cos\alpha(a_{N-1}a_{N-1}^\dagger - a_{N-1}^\dagger a_{N-1})(a_{N-1} + a_{N-1}^\dagger)(a_1 + a_1^\dagger) \\
&= \mp\cos\alpha(a_{N-1}a_{N-1}^\dagger a_{N-1} - a_{N-1}^\dagger a_{N-1}a_{N-1}^\dagger)(a_1 + a_1^\dagger) \\
&= \mp\cos\alpha((I - a_{N-1}^\dagger a_{N-1})a_{N-1} - (I - a_{N-1}a_{N-1}^\dagger)a_{N-1}^\dagger)(a_1 + a_1^\dagger) \\
&= \mp\cos\alpha(a_{N-1} - a_{N-1}^\dagger)(a_1^\dagger + a_1) \\
&= \mp\cos\alpha(a_{N-1}^\dagger - a_{N-1})(a_1^\dagger + a_1).
\end{aligned}$$

So the second boundary term can be expressed as:

$$-\cos\alpha(X_{N-1}Z_N X_1) = \mp\cos\alpha(a_{N-1}^\dagger - a_{N-1})(a_1^\dagger + a_1). \tag{35}$$

Therefore, after adding the two boundary terms, we end up with the Hamiltonian:

$$\begin{aligned}
H = &-\cos\alpha \sum_{j=2}^{N-1} \Big((a_{j-1}^\dagger - a_{j-1})(a_{j+1}^\dagger + a_{j+1})\Big) \mp \cos\alpha(a_N^\dagger - a_N)(a_2^\dagger + a_2) \mp \cos\alpha(a_{N-1}^\dagger - a_{N-1})(a_1^\dagger + a_1) \\
&- \sin\alpha \sum_{j=1}^{N}(a_j a_j^\dagger - a_j^\dagger a_j)
\end{aligned} \tag{36}$$

Note that there is now a bit of ambuguity left; depending on our choice of taking the value of the symmetry to be ± 1, the eigenspectrum and ground state we will obtain will differ. However, (while this choice is not sufficiently mathematically motivated), we do observe that picking the +1 value for the two symmetries results in a homogenous (and hence analytically tractable) form for the Hamiltonian:

$$H = -\cos\alpha \sum_{j=1}^{N} \Big((a_{j-1}^\dagger - a_{j-1})(a_{j+1}^\dagger + a_{j+1})\Big) - \sin\alpha \sum_{j=1}^{N}(a_j a_j^\dagger - a_j^\dagger a_j) \tag{37}$$

where $j = -1$ corresponds to $j = N$ and $j = N + 1$ corresponds to $j = 1$.

2.7 Periodic BCs - Numerical Diagonalization

Performing a numerical diagonalization as in the previous section for the periodic boundary condition case (Where we use the Hamiltonian given in Equation (32), with the $i = 0$ and $i = N$ terms added to the Cluster state term), we get the following plots:

We note some fairly intriguing behavior as a result of this diagonalization. First, we note that the $N = 4$ case looks wildly off from the other cases, with behavior that is more akin to the open boundary conditions

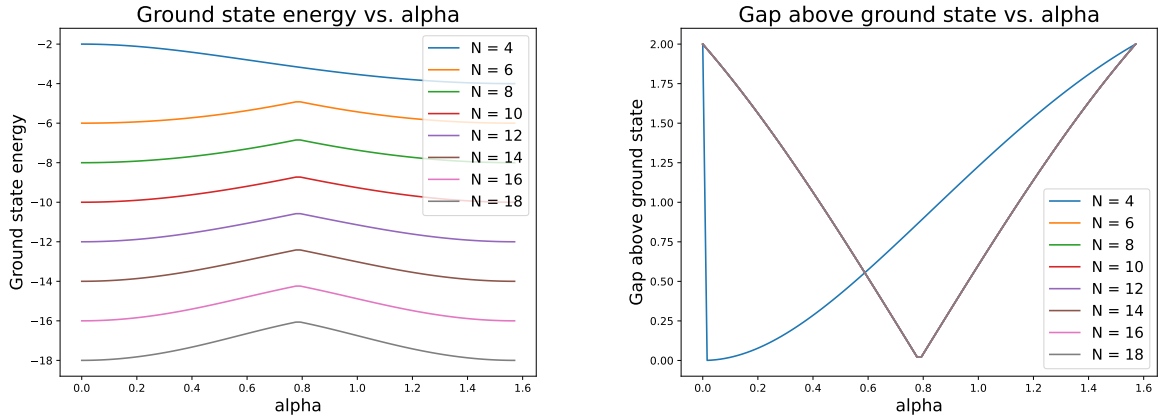

Figure 3: Plots of the Ground state energy and gap above the ground state as a function of α (and chain length N) for the interpolated cluster state chain with periodic boundary conditions. Note that the chain is only meaningfully defined for even N and these are reflected in the choices of chain length for the plots.

rather than the periodic ones. Indeed, looking at the eigenvalues (eigenenergies) for the cluster state ($\alpha = 0$) we observe that there is a bizarre phenomenon where there are 0 eigenvalues for the $N = 4$ case and for none of the other cases. Somehow, the ground state is degenerate for the $N = 4$ ring but not the others. Additionally, another strange phenomenon we observe is the lack of N dependence (excepting the $N = 4$ anomoly) in the plots. We would expect that the gap as a function of α will dip further down as we increase the chain length, but here we instead observe that there is no chain length dependence on the plot whatsoever.

These two odd phenomena point to either an error in the code, or some point of miscalculation. As a sanity check, a numerical diagonalization in the spin basis was attempted for the $N = 4$ ring, and yielded the expected results. A plot of the gap above the ground state for this exact spin basis diagonalization is given below:

Given the discreptancy between the gaps obtained via the fermionic calculation vs. the direct calculation in the spin picture,a possible option may be to explore the values of the symmetries; recall in the previous sections that we made the choice to choose the +1 value for each of the symmetries in the JW transformation. Perhaps changing the parity of these symmetries will yield different/more physical results. Or perhaps these plots indicate some failing of the JW transform when generalized to the periodic case, though the success of other research papers in this realm would seem to indicate an error on my part is more likely.

2.8 Periodic BCs - Fourier Transform

Given the Hamiltonian in (37), we can apply the Fourier transform to each term:

$$a_j^\dagger = \sum_k \frac{e^{-ikj}}{\sqrt{N}} c_k^\dagger, \quad a_j = \sum_k \frac{e^{ikj}}{\sqrt{N}} c_k. \tag{38}$$

also recall the wavevector identity given in (27):

$$\frac{1}{N} \sum_{j=1}^{n} e^{i(k-k')j} = \delta_{k,k'}$$

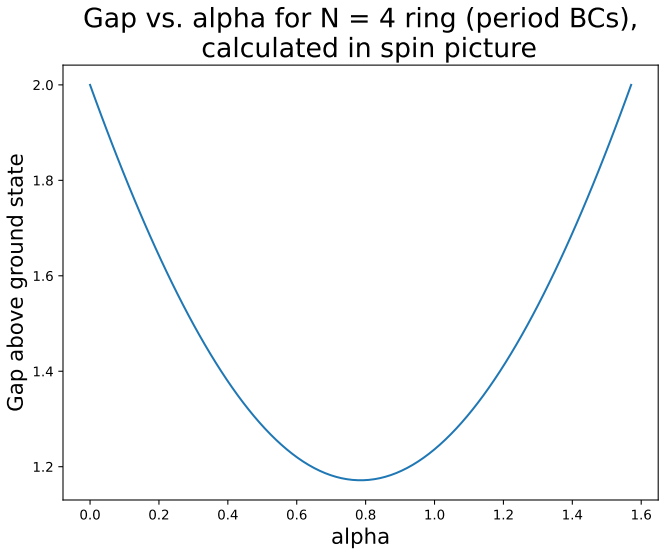

Figure 4: Gap above the Ground state for the interpolated Hamiltonian as a function of α for the $N = 4$ chain. The gap was found by taking the difference between the lowest eigenvalue and the second lowest eigenvalue. We see the expected result that the gap is 2 at the two values of $\alpha = 0, \frac{\pi}{2}$, and smoothly decreases as the $\alpha = \frac{\pi}{4}$ is approached. In the $N \to \infty$ limit, the gap closes completely.

First expanding out the Hamiltonian, we have:

$$H = -\cos\alpha \sum_{j=1}^{N} (a_{j-1}^\dagger a_{j+1}^\dagger + a_{j-1}^\dagger a_{j+1} - a_{j-1} a_{j+1}^\dagger - a_{j-1} a_{j+1}) - \sin\alpha \sum_{j=1}^{N} (a_j a_j^\dagger - a_j^\dagger a_j)$$

Applying the Fourier transform, we obtain:

$$\begin{aligned} H = &-\cos\alpha \frac{1}{N} \sum_{j=1}^{N} \sum_{k} \sum_{k'} e^{-ik(j-1)} e^{-ik'(j+1)} c_k^\dagger c_{k'}^\dagger - \cos\alpha \frac{1}{N} \sum_{j=1}^{N} \sum_{k} \sum_{k'} e^{-ik(j-1)} e^{ik'(j+1)} c_k^\dagger c_{k'} \\ &+ \cos\alpha \frac{1}{N} \sum_{j=1}^{N} \sum_{k} \sum_{k'} e^{ik(j-1)} e^{-ik'(j+1)} c_k c_{k'}^\dagger + \cos\alpha \frac{1}{N} \sum_{j=1}^{N} \sum_{k} \sum_{k'} e^{ik(j-1)} e^{ik'(j+1)} c_k c_{k'} \\ &- \sin\alpha \frac{1}{N} \sum_{j=1}^{N} \sum_{k} \sum_{k'} e^{ikj} e^{-ik'j} c_k c_{k'}^\dagger + \sin\alpha \frac{1}{N} \sum_{j=1}^{N} \sum_{k} \sum_{k'} e^{-ikj} e^{ik'j} c_k^\dagger c_{k'} \end{aligned}$$

Grouping terms in a convenient way to apply the wavevector identity:

$$\begin{aligned}H = &-\cos\alpha\frac{1}{N}\sum_{j=1}^{N}\sum_{k}\sum_{k'}e^{-ij(k+k')}e^{ik-ik'}c_k^\dagger c_{k'}^\dagger - \cos\alpha\frac{1}{N}\sum_{j=1}^{N}\sum_{k}\sum_{k'}e^{-ij(k-k')}e^{ik+ik'}c_k^\dagger c_{k'}\\ &+\cos\alpha\frac{1}{N}\sum_{j=1}^{N}\sum_{k}\sum_{k'}e^{-ij(-k+k')}e^{-ik-ik'}c_k c_{k'}^\dagger + \cos\alpha\frac{1}{N}\sum_{j=1}^{N}\sum_{k}\sum_{k'}e^{-ij(-k-k')}e^{-ik+ik'}c_k c_{k'}\\ &-\sin\alpha\frac{1}{N}\sum_{j=1}^{N}\sum_{k}\sum_{k'}e^{-ij(-k+k')}c_k c_{k'}^\dagger + \sin\alpha\frac{1}{N}\sum_{j=1}^{N}\sum_{k}\sum_{k'}e^{-ij(k-k')}c_k^\dagger c_{k'}\end{aligned}$$

Applying the wavevector identity:

$$\begin{aligned}H = &-\cos\alpha\sum_{k}\sum_{k'}\delta_{k,-k'}e^{ik-ik'}c_k^\dagger c_{k'}^\dagger - \cos\alpha\sum_{k}\sum_{k'}\delta_{k,k'}e^{ik+ik'}c_k^\dagger c_{k'}\\ &+\cos\alpha\sum_{k}\sum_{k'}\delta_{-k,-k'}e^{-ik-ik'}c_k c_{k'}^\dagger + \cos\alpha\frac{1}{N}\sum_{j=1}^{N}\sum_{k}\sum_{k'}\delta_{k,-k'}e^{-ik+ik'}c_k c_{k'}\\ &-\sin\alpha\sum_{k}\sum_{k'}\delta_{k,k'}c_k c_{k'}^\dagger + \sin\alpha\sum_{k}\sum_{k'}\delta_{-k,-k'}c_k^\dagger c_{k'}.\end{aligned}$$

Using the dirac deltas to collapse the sums, we obtain:

$$\boxed{H = -\cos\alpha\sum_{k}(e^{2ik}c_k^\dagger c_{-k}^\dagger + e^{2ik}c_k^\dagger c_k - e^{-2ik}c_{-k}c_{-k}^\dagger - e^{-2ik}c_k c_{-k}) - \sin\alpha\sum_{k}(c_k c_k^\dagger - c_{-k}^\dagger c_{-k})} \tag{39}$$

2.9 Periodic BCs - Bugolibov Transform/Analytic Diagonalization

Let us attempt a Bugolibov transformation on the simplified hamiltonian. We consider a transformation of the form:

$$\gamma_k = u_k c_k + iv_k c_{-k}^\dagger \tag{40}$$

Where the u_k, v_ks are chosen such that two conditions are satisfied; namely, that the γ_ks follow the fermionic commutation relations, and that H is diagonal in the γ_k basis. The above transformation has matrix form:

$$\begin{bmatrix}\gamma_k\\ \gamma_{-k}\\ \gamma_k^\dagger\\ \gamma_{-k}^\dagger\end{bmatrix} = \begin{bmatrix}u_k & 0 & 0 & iv_k\\ 0 & u_{-k} & iv_{-k} & 0\\ 0 & -iv_k & u_k & 0\\ -iv_{-k} & 0 & 0 & u_{-k}\end{bmatrix}\begin{bmatrix}c_k\\ c_{-k}\\ c_k^\dagger\\ c_{-k}^\dagger\end{bmatrix} \tag{41}$$

Before we substitute this into the Hamiltonian, let us place down some contraints on u_k, v_k such that the fermionic commutation relations must be satisfied. Namely, we have the constraints that $\{\gamma_j, \gamma_k^\dagger\} = \delta_{jk}I$ and $\{\gamma_j, \gamma_k\} = 0$. We then have three relations to check, namely:

(1) $\{\gamma_k, \gamma_k^\dagger\} = I$

(2) $\{\gamma_k, \gamma_{-k}^\dagger\} = 0$

(3) $\{\gamma_k, \gamma_{-k}\} = 0$

For the first relation, we have that:

$$\{\gamma_k, \gamma_k^\dagger\} = \{u_k c_k + i v_k c_{-k}^\dagger, -i v_k c_{-k} + u_k c_k^\dagger\} = I$$

Using the linearity of the anti-commutator:

$$-u_k v_k \{c_k, c_{-k}\} + v_k^2 \{c_{-k}^\dagger, c_{-k}\} + u_k^2 \{c_k, c_k^\dagger\} + i v_k u_k \{c_{-k}^\dagger c_k^\dagger\} = I$$

Using the known commutation relations for the c_ks, we have:

$$u_k^2 I + v_k^2 I = I$$

Therefore:

$$\boxed{u_k^2 + v_k^2 = 1} \tag{42}$$

For the second relation, we have:

$$\begin{aligned}\{\gamma_k, \gamma_{-k}^\dagger\} &= \{u_k c_k + i v_k c_{-k}^\dagger, -i v_{-k} c_k + u_{-k} c_{-k}^\dagger\} = 0 \\ &= -u_k v_{-k} \{c_k, c_k\} + u_k u_{-k} \{c_k, c_{-k}\} v_k v_{-k} \{c_{-k}^\dagger, c_k\} + i v_k u_{-k} \{c_{-k}^\dagger, c_{-k}^\dagger\} \\ &= 0 + 0 + 0 + 0\end{aligned}$$

So the second relation tells us nothing. For the third relation, we have:

$$\begin{aligned}\{\gamma_k, \gamma_{-k}\} &= \{u_k c_k + i v_k c_{-k}^\dagger, u_{-k} c_{-k} + i v_{-k} c_k^\dagger\} = 0 \\ &= u_k u_{-k} \{c_k, c_{-k}\} + i u_k v_{-k} \{c_k, c_k^\dagger\} + i v_k u_{-k} \{c_{-k}^\dagger, c_{-k}\} - v_k v_{-k} \{c_{-k}, c_k^\dagger\} \\ &= i u_k v_{-k} I + i v_k u_{-k} I\end{aligned}$$

So we obtain that:

$$u_k v_{-k} + u_{-k} v_k = 0$$

Let us then choose:

$$\boxed{u_k = u_{-k}, v_k = -v_{-k}} \tag{43}$$

so the above constraint is satisfied. With these contraints, the transformation takes the form:

$$\begin{bmatrix} \gamma_k \\ \gamma_{-k} \\ \gamma_k^\dagger \\ \gamma_{-k}^\dagger \end{bmatrix} = \begin{bmatrix} u_k & 0 & 0 & i v_k \\ 0 & u_k & -i v_k & 0 \\ 0 & -i v_k & u_k & 0 \\ i v_k & 0 & 0 & u_k \end{bmatrix} \begin{bmatrix} c_k \\ c_{-k} \\ c_k^\dagger \\ c_{-k}^\dagger \end{bmatrix}$$

Inverting this transformation, we get:

$$\begin{bmatrix} c_k \\ c_{-k} \\ c_k^\dagger \\ c_{-k}^\dagger \end{bmatrix} = \begin{bmatrix} u_k & 0 & 0 & -i v_k \\ 0 & u_k & i v_k & 0 \\ 0 & i v_k & u_k & 0 \\ -i v_k & 0 & 0 & u_k \end{bmatrix} \begin{bmatrix} \gamma_k \\ \gamma_{-k} \\ \gamma_k^\dagger \\ \gamma_{-k}^\dagger \end{bmatrix}$$

Now, we can actually simplify this relation; staring at the relations a little more closely, we can make the

observation that the relations are still satisfied if we identify:

$$\boxed{u_k = \cos\theta_k, v_k = \sin\theta_k}. \tag{44}$$

Hence our transformation becomes:

$$\boxed{\begin{bmatrix} \gamma_k \\ \gamma_{-k} \\ \gamma_k^\dagger \\ \gamma_{-k}^\dagger \end{bmatrix} = \begin{bmatrix} \cos\theta_k & 0 & 0 & i\sin\theta_k \\ 0 & \cos\theta_k & -i\sin\theta_k & 0 \\ 0 & -i\sin\theta_k & \cos\theta_k & 0 \\ i\sin\theta_k & 0 & 0 & \cos\theta_k \end{bmatrix} \begin{bmatrix} c_k \\ c_{-k} \\ c_k^\dagger \\ c_{-k}^\dagger \end{bmatrix}}. \tag{45}$$

Inverting this, we obtain:

$$\boxed{\begin{bmatrix} c_k \\ c_{-k} \\ c_k^\dagger \\ c_{-k}^\dagger \end{bmatrix} = \begin{bmatrix} \cos\theta_k & 0 & 0 & -i\sin\theta_k \\ 0 & \cos\theta_k & i\sin\theta_k & 0 \\ 0 & i\sin\theta_k & \cos\theta_k & 0 \\ -i\sin\theta_k & 0 & 0 & \cos\theta_k \end{bmatrix} \begin{bmatrix} \gamma_k \\ \gamma_{-k} \\ \gamma_k^\dagger \\ \gamma_{-k}^\dagger \end{bmatrix}}. \tag{46}$$

Now, let us substitute this into H, and hopefully we will find an expression for θ_k such that the off diagonal terms will all cancel. Unfortunately, when this was attempted, the resulting expression was extremely unwieldy and not tractable. Likely, there is some additional simplification of (39) still to be done before things can be worked out in a nice form, or there was some error in the Fourier transform process. Indeed, `https://arxiv.org/abs/1112.4414` performs a similar calculation to identical results. The paper considers more general Hamiltonians of the form:

$$H = -\sum_{i=1}^{N} X_{i-1}Z_iX_i - h\sum_{i=1}^{N} Z_i + \lambda_y \sum_{i=1}^{N} Y_iY_{i+1} + \lambda_x \sum_{i=1}^{N} X_iX_{i+1} \tag{47}$$

And after the process of the JW and fourier transforms, ends up with the expression:

$$H = 2\sum_{0\le k\le\pi} \left[\epsilon_k(c_k^\dagger c_k + c_{-k}^\dagger c_{-k}) + i\delta_k(c_k^\dagger c_{-k}^\dagger + c_k c_{-k})\right] \tag{48}$$

Where:

$$\begin{aligned} k &= \frac{\pi}{N}(2m + 1 - q) \\ \epsilon_k &= \cos(2k) - (\lambda_x + \lambda_y)\cos(k) - h \\ \delta_k &= \sin(2k) - (\lambda_x - \lambda_y)\sin(k) \end{aligned} \tag{49}$$

Four our case, we can set $\lambda_x = \lambda_y = 0$, $h = \sin\alpha$, and tack on a factor of $\cos\alpha$ to the cluster term. We therefore have that the above expression for the Hamiltonian holds, and our expressions for ϵ_k, δ_k becomes:

$$\begin{aligned} k &= \frac{\pi}{N}(2m + 1 - q) \\ \epsilon_k &= \cos\alpha\cos(2k) - \sin\alpha \\ \delta_k &= \cos\alpha\sin(2k) \end{aligned} \tag{50}$$

Furthermore, in the paper they successfully apply the Bugolibov transform/diagonalization to obtain:

$$H = 2\sum_k \omega_k \left(\gamma_k^\dagger \gamma_k + \gamma_{-k}^\dagger \gamma_{-k} - 1\right) + C. \tag{51}$$

Here, ω_k is given as:

$$\omega_k = \sqrt{\epsilon_k^2 + \delta_k^2} \tag{52}$$

and θ_k in the transformation as given in (46) is chosen to be:

$$\theta_k = -\frac{1}{2}\arctan\left(\frac{\delta_k}{\epsilon_k}\right) \tag{53}$$

to cancel out the off-diagonal terms. Given more time, it would be worthwhile to see if we could reach (51) from (39) and the transformation (46). From there, we could extract the analytical expression ground state $|\text{GS}\rangle$ of the interpolated clutster state, and (after transforming back into the fermionic j basis) compute the expectation value of the SOP as given in (26). It is notable that Paul's past work with trying this numerically gave results that did not agree with Arnab's results (which involved a tensor network calculation using the closed BC Hamiltonian as given in (22)), but it may be an interesting future comparison nontheless.

3 IBMQ SPT

3.1 On Interpolation and Equivalence Classes

In this project, our goal is to carry out experimental tests of theoretical results about measurement based quantum computing and symmetry-protected order. The motivations and theoretical results are discussed at length in the prohect logfile. The first step of the experiment is to construct the ground state of the interpolated cluster state hamiltonian (as seen in the previous section):

$$H(\alpha) = -\cos(\alpha)\sum_i Z_{i-1}X_iZ_{i+1} - \sin(\alpha)\sum_i Z_i \tag{54}$$

For $\alpha = 0$, we have the (known) task of constructing the cluster state (H on every $|0\rangle$ qubit, then CZ gates between neighbours). For $\alpha = \frac{\pi}{2}$, we have the (also easily known) task of constructing the product state (H on every $|0\rangle$ qubit). Since the cluster state ground state (which we can call $|\mathcal{C}\rangle$) and the ground state of $H(\alpha)$ (which we can call $|\mathcal{C}(\alpha)\rangle$) can be connected smoothly, there exists some unitary U such that:

$$|\mathcal{C}(\alpha)\rangle = U|\mathcal{C}\rangle \tag{55}$$

where U is short-rangle entangling and symmetric, that is, it commutes with the symmetries of the cluster state Hamiltonian:

$$[U, XIXIXI\ldots] = [U, IXIXIX\ldots] = 0. \tag{56}$$

Therefore, we can expand U as a sum of Pauli operators:

$$U = \sum_i b_i P_i \tag{57}$$

where P_i are pauli operators that are symmetric/obey the commutation relations in (56). While there are 4^n Pauli operators in total for n qubits, due to this symmetry condition, this gets reduced to $4^n/4$. However,

we note that operators such as ZXZ and $ZXIXZ$ act like the identity (as these are stabilizers of the cluster state)! Hence, we don't care about *all* the Pauli operators P_i, but rather, the Pauli operators modulo the cluster stabilizers. The claim is that for every P_i that satisfies the commutation relations in (56), there exists a representaiton modulo the cluster stabilizers such that:

$$P_i \sim (\text{Product of } X \text{ and } I \text{ operators only}) = X[i] \tag{58}$$

Hence, instead of U, we can consider the transformation T where:

$$T = \sum_i c_i X[i] \tag{59}$$

Where $X[i]$ is the representation of the equivalence class, consisting of only X and Is. As a result, there are only 2^n terms in the sum now (rather than the original $4^n/4$) as we consider all possible strings of operators of X and Is on each of the n sites. Note that in going from U to T, we do lose unitarity as we multiply the P_is with another operator.

As an interlude, for the actual implementation of the project, we will try to extract the coefficients c_i in (59) (Dmytro is currently working on Tensor Network methods for accomplishing this). T is not unitary, so it cannot be implemented on the machines, and U is unitary, but will still be hard to implement on the machines without having everything drown in decoherence. Hence, we will push T past the measurements, and use the coefficients to post-process the measurement outcomes on a cluster state to obtain the measuremnet outcomes on the interpolated cluster state (more details in the logfile).

However, the current problem does not concern the coefficient extractions, or this post-processing. Instead, we want to show that a switch from U to T is indeed possible; that is, a P that commutes with the cluster state symmetries has a representation modulo the cluster state stabilizers that consists solely of Pauli X and Is. We give the full problem statement below:

Suppose we have a Pauli operator P for a string of qubits of length N. Further suppose that P commutes with the symmetries of the cluster state Hamiltonian H, that is:

$$[P, XIXIXI\ldots] = [P, IXIXIX\ldots] = 0.$$

Form equivalence classes of such stabilizers modulo the cluster state operators $Z_{i-1}X_iZ_{i+1}$. Prove that the equivalence class for any such P above has representation that consists of solely X and I operators. For example, for the $N = 3$ case, we have that the operator:

$$P = Z_1X_2Z_3$$

satisfies the above commutation relations, but we can express this purely in terms of Is by modding out the stabilizer $Z_1X_2Z_3$:

$$P \cdot Z_1X_2Z_3 = (Z_1X_2Z_3)^2 = I_1I_2I_3.$$

Solution. Suppose we have a Pauli operator P that satisfies the above relation. If all sites of P commute with all sites of the two cluster hamiltonian symmetries $XIXIXI\ldots$ and $IXIXIX\ldots$, then P consists solely of X and I operators already, and we are done.

Suppose then that P does not consist of solely X and Is. Then, P contains Zs or Ys (or both). However, since P commutes with the symmetries, the following conditions must hold:

(1) There are in total an even number of Zs/Ys on odd sites.

(2) There are in total an even number of Zs/Ys on even sites.

To see why this must be the case, consider that $XZ = -ZX$ and $XY = -YX$. Hence, in order for P to commute with $XIXIXI\ldots$, there must be an even number of anticommutations on the odd sites (and hence an even number of Zs and Ys on odd sites) in order for the negative signs to cancel (the Is in the symmetry of course commute with everything). Identically, for P to commute with $IXIXIX\ldots$, there must be an even number of anticommutations on the even sites (and hence an even number of Zs and Ys on even sites).

To this end, let us pair up Z/Ys on odd sites (pairing up those that are the closest to each other) and let us pair up Z/Ys on even sites (again pairing up those that are the closest to each other). We now introduce an algorithm to deal with each pair and get rid of the Z/Ys.

For each pair on odd sites (one of $Z_{2n+1}Z_{2n+1+2k}$, $Y_{2n+1}Z_{2n+1+2k}$, $Z_{2n+1}Y_{2n+1+2k}$, $Y_{2n+1}Y_{2n+1+2k}$) we multiply P by the stabilizers $Z_{2n+1}X_{2n+2}Z_{2n+3}$, $Z_{2n+3}X_{2n+4}Z_{2n+5}$, $\cdots Z_{2n-1+2k}X_{2n+2k}Z_{2n+1+2k}$. In other words, multiply P by all the cluster stabilizers centered on even sites that fall in between the two offending Z or Y operators. This has the net effect of multiplying P by:

$$Z_{2n+1}X_{2n+2}I_{2n+3}X_{2n+4}\ldots I_{2n+2k-1}X_{2n+2k}Z_{2n+2k+1}.$$

This has the effect of removing the offending Z/Y terms at the $2n+1$ and $2n+2k+1$ sites. In addition, it introduces no new Z/Y operators in between. Repeating this process for every pair of odd sites with offending Z/Y operators, we remove all Zs/Ys on odd sites.

Similarly, for each pair on even sites (one of $Z_{2n}Z_{2n+2k}$, $Y_{2n}Z_{2n+2k}$, $Z_{2n}Y_{2n+2k}$, $Y_{2n}Y_{2n+2k}$), we multiply P by the stabilizers $Z_{2n}X_{2n+1}Z_{2n+2}$, $Z_{2n+2}X_{2n+3}Z_{2n+4}$, $\cdots Z_{2n-2+2k}X_{2n-1+2k}Z_{2n+2k}$. In other words, multiply P be all the cluster stabilizers centered on odd sites that fall in between the two offending Z or Y operators. This has the net effect of multiplying P by:

$$Z_{2n}X_{2n+1}I_{2n+2}X_{2n+3}\ldots I_{2n+2k-2}X_{2n+2k-1}Z_{2n+2k}.$$

This has the effect of removing the offending Z/Y operators at the $2n$ and $2n+2k$ sites. In addition, it introduces no new Z/Y operators in between.Repeating this process for every pair of even sites with offending Z/Y operators, we remove all Zs/Ys on even sites.

Hence, we have given an algorithm such that our Pauli operator P mod the cluster state stabilizers can be represented by purely X and I operators (as we have removed all the Z and Ys by the above algorithm). This concludes the proof. □

3.2 Debugging Qiskit and IBMQ Experiments

Before moving onto the more complex (and interesting) projects for experimental verification (on the IBM quantum computers) of:

(1) The splitting of rotations across multiple qubits to manage logical decoherence in SPT MBQC

(2) The smallest possible choice of rotation separations (■ = 2) is the most resource efficient, counter to the textbook regime of ■ much larger than the correlation length

we first investigate a simpler scenario. We implement a rotation operation on two qubits that amounts to an identity operation in the cluster state MBQC picture:

$$R_z(\beta)R_z(-\beta) = I$$

Where we start with an initial input state $|+\rangle$, apply the rotation and then the counter rotation later down the chain, and end with a measurement in the X eigenbasis. The computational state we perform these calculations on is not just the cluster state however; as discussed in the previous section, we carry out the operations on the ground state of the interpolated cluster state hamiltonian, as given by (54). As discussed previously, we do not actually implement the preparation of this state on the devices (being that this would be an arduous and noisy process), but instead use the coefficients c_i in (59) to post-process the measurement outcomes; the actual computation can then be carried out on a cluster state which can be prepared in a standard fashion.

We start with a simple 5-qubit quantum machine (As these are available for free, and will provide good benchmarking). The circuit we will perfom on the machines is given below, and it simulates the MBQC procedure on the gate based machine.

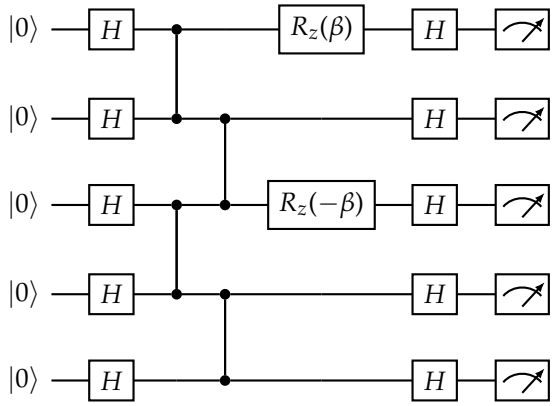

Figure 5: Circuit that can be carried out on a gate-based machine to simulate the 2-qubit rotation protocol. The H and CZs at the beginning turn the register into a cluster state, the $R_z(\beta)$ and $R_z(-\beta)$ rotate the qubit, and the H and measurements at the end result in a measuremnet in the X eigenbasis. All in all, this simulates the teleportation of a $|+\rangle$ eigenstate from one end of the register to another (after undergoing a rotation and counterrotation)

What we predict theoretically is that when we perform this protocol on the cluster state (that is, $\alpha = 0$ in the interpolated Hamiltonian), the $R_z(\beta)$ and $R_z(-\beta)$ cancel out perfectly, leaving us with a $|+\rangle$ state on the last qubit, and leading to a measurement outcome of $\langle X \rangle = 1$. However, due to effects of logical decoherence, for $\alpha \neq 0$ the expectation value is reduced. The expectation value indeed depedends on the rotation angle β (being that for $\beta = 0$ we still expect $\langle X \rangle = 1$ due to symmetry protection) so we can choose $\beta = \frac{\pi}{2}$ as the choice furthest away from this to best study these effects of the logical decoherence. The theoretical details of this are outlined in the project logfile as well as Arnab's work.

In order to obtain these values from the experiment, a little more work is required. In particular, there is some post-processing that is required; some native to the MBQC protocol, and some post-selection that additionally arises from the lack of adaptive measurements on the IBMQ machines. Let us describe this for the 5-qubit chain. When we perform MBQC, we have to consider byproduct operators that we pull through to the end of the measurement, as well as adaptive measurements that are a consequence of us pulling these operators through. In our current 5 qubit experiment, the protocol will be:

(1) Start with $|+\rangle$ on qubit 1

(2) Measure qubit 1 in the $R_z(\beta)$ rotated basis

(3) Measure qubit 2 in the X eigenbasis

(4) Measure qubit 3 in the $R_z(-\beta)$ rotated basis

(5) Measure qubit 4 in the X eigenbasis

(6) Measure qubit 5 in the X eigenbasis (to read off the measurement outcome).

however, this is not the full extent of the protocol. Indeed, depending on the measurement outcomes we obtain, we have to apply a biproduct correction operator on the state; here, this correction operator is given as $Z^{s_1+s_3}X^{s_2+s_4}$ where the addition is taken modulo 2. Since the state we are transporting is $|+\rangle$, the X correction operator does nothing in the final post-processing, and we are left with just a $Z^{s_1+s_3}$ operator we must consider. This can be easily implemented in Qiskit as a correction operation we apply.

However, a less trivial problem is the fact that when we pull the X^{s_2} across to the end, it anti-commutes with the $R_z(-\beta)$ measurement on the third qubit; in the adaptive MBQC protocol, we would combat this by performing a measurement $R_z((-1)^{1+s_2}\beta)$ on the third qubit (indeed, the temporal order of MBQC falls out as a consequence of the fact that the pullthrough of biproduct measuremnet operators requires us to change our measurement bases). However, there is a fundamental problem here; on the Qiskit gate machines, we cannot do a measuremnet at an arbitrary point in the circuit; measurements are performed on all qubits at once at the end. This obviously poses a problem as we cannot appropriately modify our measuremnet bases based on the measurement outcome at the second qubit (as the protocol requires). The way to handle this in our experiment is to post-select the data based on the measuremnet outcomes. We reject all counts which have a measurement of 1 on the second qubit, as these would result in an incorrect rotation on the third qubit. For a longer chain, we would need to more heavily post-select, but for the 5-qubit chain, we only need to throw out half of our measuremnet shots as a result of this necessary post-processing.

Credits to Amrit for most of coding for the calculation of $\langle X\rangle$ with the necessary post-processing and post-selection. I put in a lot of time refining this code and correcting errors to make sure the code was actually processing the data correctly; there was a 2-week stretch where the code would give us unexpected results even for simulated experiments (where we should have expected nothing but perfect outcomes). Fortunately, after much time I was able to locate the issues with this processing, and our end product is now fully-functioning with the processing we require. The full function to calculate $\langle X\rangle$ can be found in the `process_data.py` file.

3.3 Circuit Optimization

As discussed in the above section, when the original circuit (given in (5)) is performed on a machine with no error (in this case, the circuit is simulated on a quantum computer simulator with no error model), then the $\langle X\rangle$ measurement on the last qubit (with proper post-selection/processing) should just yield 1 (as after post-selection/processing, the final state on the fifth qubit is $|\psi_5\rangle = |+\rangle$; hence, $\langle X\rangle = 1$). This is because the rotation by angle β on the first qubit and by angle $-\beta$ on the second qubit mathematically cancel, leaving us just with the initial $|+\rangle$ input state. This is indeed what is observed in simulating this circuit.

However, in executing this circuit on a real quantum computer with errors, the situation is much more bleak; instead of the angle-independent straight line at $\langle X\rangle = 1$, we see an angle-dependent curve which starts at $\langle X\rangle \approx 0.75$ for $\beta = 0, \pi$ and dips to $\langle X\rangle \approx 0.55$ for $\beta = \frac{\pi}{2}$. After accounting for measurement error and correcting accordingly, we improve slightly (with $\langle X\rangle \approx 0.95$ for $\beta = 0, \pi$ and $\langle X\rangle = 0.75$ for $\beta = \frac{\pi}{2}$), but this still seems to spell out doom for the project, as the severe angle dependence as well as the low $\langle X\rangle$ for such a small circuit (both in depth and the number of qubits) makes a more technical experiment as we would like to do later seemingly impossible. However, this is not the end! It turns out that the Qiskit transpiler creates a *highly inefficient* circuit when transpiling the above original circuit into the universal

gateset available on the IBMQ machines. For completeness, we note that the universal gateset is given by:

$$\left\{I, X, \sqrt{X}, CX, R_z(\beta)\right\}.$$

Because there are no H or CZ gates available in the gateset, with the transpilation, the circuit that ends up running on the machine turns out to be extremely long and (therefore) more error-prone than intended. We can therefore use circuit identities to write the above circuit in a more efficient way and such that it only uses the gates in the IBMQ universal gate set (such that we will not get garbage when we transpile a circuit that contains other elements).

The circuit identities we will invoke are that $H^2 = I$ and that $HZH = X$. Adding HH in between the two CZs on the second rail, we obtain the following (first iteration) of the optimized circuit:

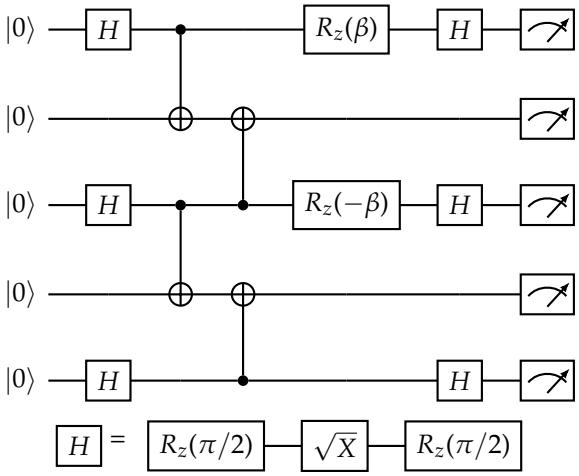

Figure 6: First iteration of optimized circuit, that only uses gates in the universal gate set of the IBMQ machine. The Hs are realized on the machine as $R_z(\pi/2)\sqrt{X}R_z(\pi/2)$.

So after this first round of optimization, we are left with a circuit of depth 9. Not bad (our original transpiled circuit had depth of over 30!) but not fantastic. Amrit's results of running this on the IBM machines are as follows:

However, we can do even better with our circuit optimization. First, we can realize that $R_z(\beta)|0\rangle = |0\rangle$ as the $|0\rangle$ state is left invariant by rotations about the z-axis. Hence, the three $R_z(\pi/2)$ rotations in the beginning can be absorbed into the initial state/removed from the circuit. In addition, the second/third set of $R_z(\pi/2)$s can be absorbed into the $R_z(\beta)$ rotation angles as to do the rotation simultaneously (not three at a time). Finally, the last rotation of $R_z(\pi/2)$ can be absorbed into the measurement, as it commutes with the measurement in the computational basis. This yields the (most) optimized circuit for the protocol:

The results of running this on the IBM (Santiago) 5 qubit machine are as follows:

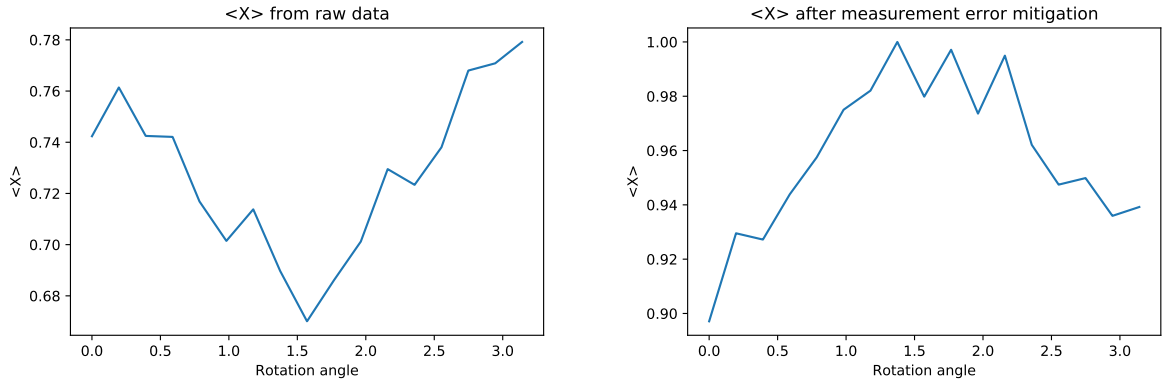

Figure 7: Results (before and after measurement error mitigation) of $\langle X \rangle$ measurement on the final qubit as a function of the rotation angle β in the above two-qubit rotation protocol.

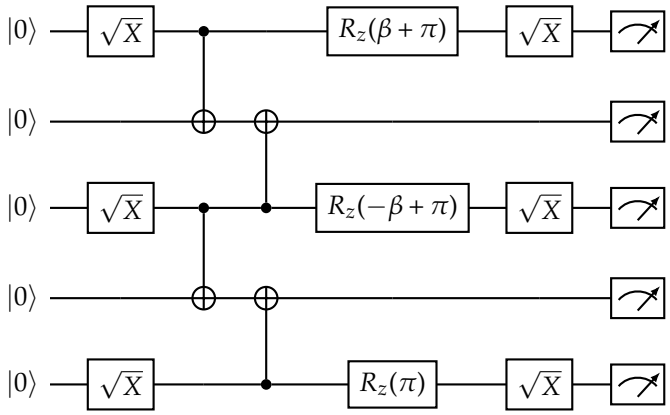

Figure 8: Maximally optimized circuit for the two qubit rotation protocol. This circuit has depth 6.

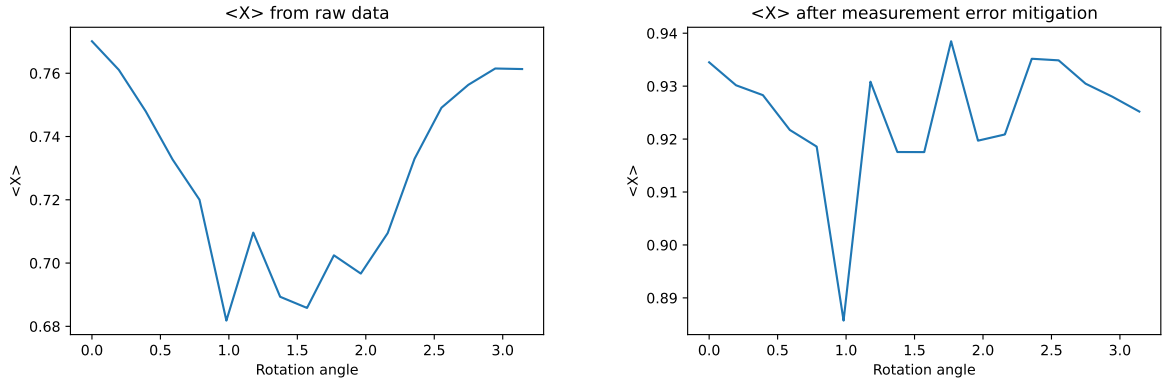

Figure 9: Results (before and after measurement error mitigation) of $\langle X \rangle$ measurement on the final qubit as a function of the rotation angle β in the above two-qubit rotation protocol. The optimized circuit depicted in Figure (8) was used for the experiment.

We note that this is a fairly nfortunate result; the fidelity of the machine, even for such a small circuit size, is not good at all (though the angular dependence does fortunately vanish with measurement error mitigation). The optimization did not seem to provide any significant gain (perhaps to be expected, as the jump from a garbage transpiled circuit to a decent one would be larger than a decent one to a slightly more optimized one). We do note that the measurement error mitigation does consistently push the fidelity to the $0.9 \sim 0.95$ range, but this is just at the boundary where we will be able to find a measureable effect when we go to the logical decoherence measurements. Moving forwards, we will really have to think about the errors in the machines and how we will deal with them; else, we will not be able to extract a useful result/signal from the noise as we wish to do for the later experiments.

3.4 The Threat of Measurement Errors

As part of benchmarking the machines, it is useful to determine where the largest source of error is. The following results give an indication that measurment error is the most significant source of the error (and moreover, that certain outcomes have a higher error rate of measurement than others). We start by looking at the measurement statistics for two very simple circuits; one with a preparation of the $|0\rangle^{\otimes n}$ initial state and then an immediate computational basis measurement, and another with a preparation of the $|0\rangle^{\otimes n}$ initial state, then an application of a Pauli-X to each qubit, and then a computational basis measurement.

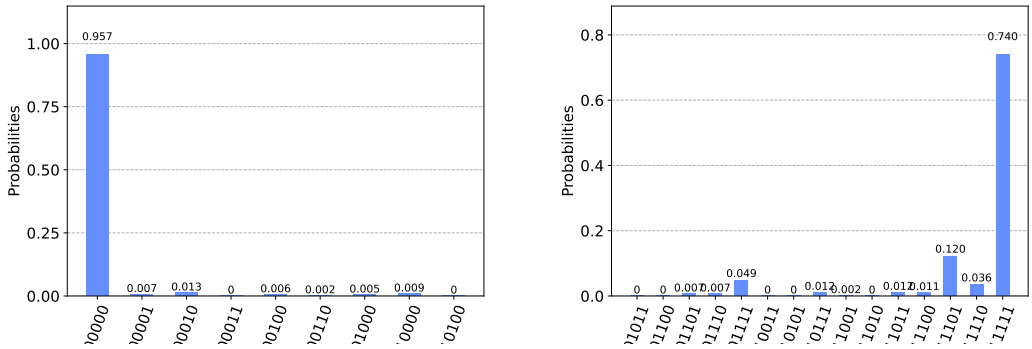

Figure 10: Measurement statistics (8000 shots each) of computational basis measurement for 5-qubit circuit with no gates (left) and circuit with Pauli X gates applied to each qubit (right). Measurements were carried out on the 5-qubit IBMQ Santiago machine on August 4th.

We see that while the fidelity of the all 0s measurement is quite high, the fidelity of the all 1s measurement is abysmal. In particular, there are certain qubits for which measuring the 1 outcome seems to be more error prone (qubit 5 and qubit 2). One possible explanation is that the Pauli-X gate is more erroneous; however, if this was the case, then we would observe that applying two sets of Pauli X gates would return a very errorneous outcome as well. Let us then run this circuit experimentally:

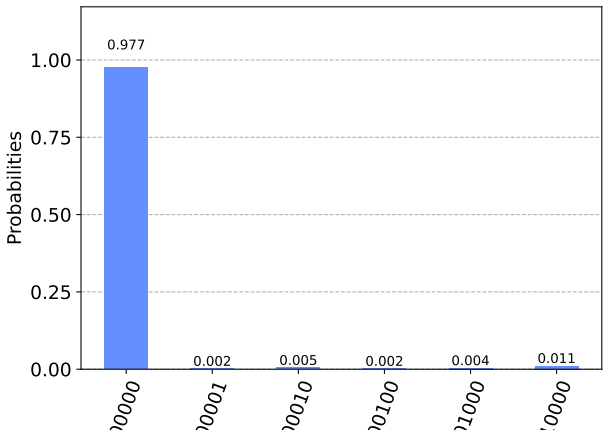

Figure 11: Measurement statistics (8000 shots) of computational basis measurement for 5-qubit circuit where Pauli X gates are applied twice to each qubit before measurement. Measurements were carried out on the 5-qubit IBMQ Santiago machine on August 4th.

We see that the measurement statistics are as good as they were for the circuit where we apply no gates to the qubits whatsoever. The conclusion is that the single-qubit gates are not the problem, but rather the measurement of certain (1/excited state) outcomes on the qubits is more error prone.

A more full picture of the above tests can be obtained by looking at the measurement calibration circuits, which test each of the 2^n possible computational basis measurement outcome circuits.

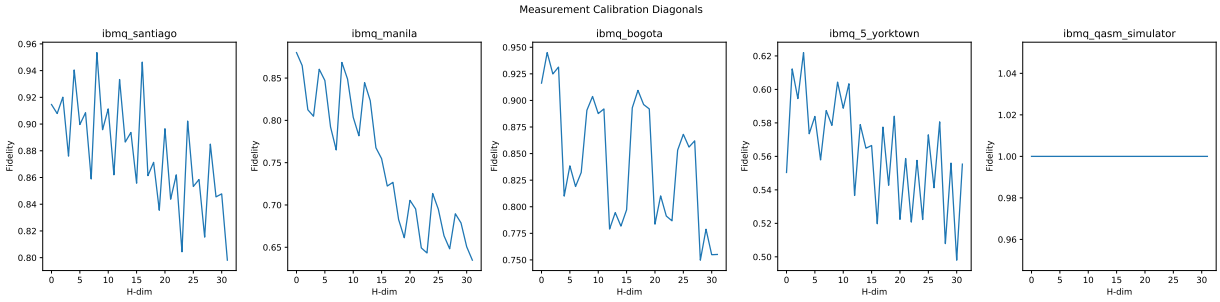

Figure 12: Fidelity measurements for error calibration circuits. On the X-axis are different circuits corresponding to the preparation of the given state in binary (e.g. the data point 31 on the X-axis corresponds to a circuit that prepares the $|11111\rangle$ state, with Pauli X gates on each qubit). On the y-axis is the fidelity/probability of measuring the expected outcome corresponding to that calibration circuit (e.g. for the data point 31, what is the probability that the 11111 outcome is actually measured). Measurements were carreid out on the 5-qubit Santiago, Manila, Bogota, and Yorktown machines (along with the simulator/error free-result for comparison).

Looking at these plots we can indeed confirm that the measurement fidelity depends on the state being measured. The measurement error mitigation seeks to mitigate this, but is not a perfect procedure. There is likely some physical mechanism that makes measuring certain outcomes on certain qubits more error-prone, but this is still to be hypothesized/investigated. Some possible next steps are to fit a measuremnet error model to the above plots to obtain a clearer picture of the measurement errors per qubit, and to compare this with 2-qubit gate error benchmarks to see how the order of the error stacks up.

3.5 Measurement Error Modeling

Given the high measurement error as investigated in the previous section, a worthwhile task is a characterization of the form of the error. The most ideal scenario is if the measurements can be described by a local measurement model; that is, we can define two measurement fidelity parameters f_{i0} and f_{i1} to each qubit, that tells us the probability that we will measure the state $0/1$ (respectively) on qubit i when the qubit is prepared in the state $|0\rangle/|1\rangle$ (respectively). In the 5-qubit case, our model consists of the 10 parameters $f_{a0}, f_{a1}, f_{b0}, f_{b1}, \ldots f_{e0}, f_{e1}$ and the fidelity of the $|00000\rangle$ state (that is, the probability in which one would measure 00000 after a $|00000\rangle$ state is prepared and measured in the computational basis) is given by $f_{a0} \cdot f_{b0} \cdot f_{c0} \cdot f_{d0} \cdot f_{e0}$.

To obtain these parameters through numerical fitting, we can take the plots in 12 and fit them with a kind of superposition of 5-square waves, with period given by 2^{i+1} for the ith qubit, the wave amplitude given by $f_{i0} - f_{i1}$ and the wave offset/centerpoint given by f_{i1}. The overall curve is given by multiplying the 5 curves together. Using a curve-fitting routine allows us to extract the optimal values of these 5 parameters.

This fitting seemed to be successful. However, it is also inherently problematic; in this routine, we first *assume* that the global measurement errors can be described by a local measurement error model, and then fit such a model to the overall global measurement fidelity curve. However, this does not allow us to conclude that the measurement error is indeed well described by a local model. Indeed, a 10-parameter fit would be easy to fit to almost anything to some degree of error. Instead, the more sound option is to measure these $2N$ $f_{i0/1}$ parameters individually (via local measurements) and then multiplying them together to see if we can obtain the global fidelity curve as given by 12.

In order to measure these parameters individually, we generate and conduct measurements on $2N$ differnet circuits. Each circuit consists of either the preparation of the $|0\rangle$ state on the ith qubit (to extract the f_{i0} parameter) or the preparation of the $|1\rangle$ state on the ith qubit via a single qubit Pauli-X gate (to extract the f_{i1} parameters). Each circuit consists of a single qubit preparation and subsequent measurement. The subsequent measurmenet statistics (dividing the number of correct outcome shots by the total number of shots) should give $f_{i0/1}$, and multiplying the $2N$ obtained values together in the correct respective ways (i.e. $f_{aj_1} f_{bj_1} f_{cj_3} f_{dj_4} f_{ej_5}$ for $|j_1 j_2 j_3 j_4 j_5\rangle$) should recover the global fidelity curve as given by the measurement calibration circuits. The results of this are given below:

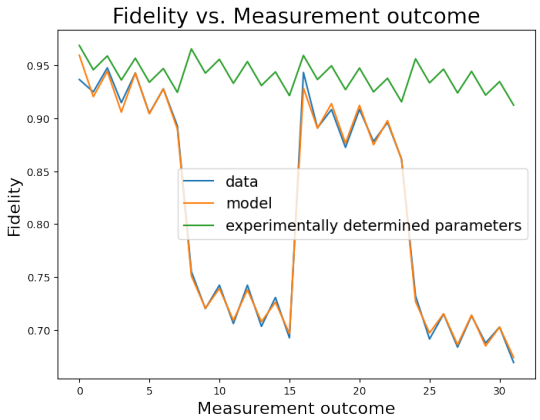

Figure 13: Fidelity results as obtained from global measurements on states $|j_1 j_2 j_3 j_4 j_5\rangle$, from a 10-parameter fit to a 5-square wave (local error) model, and from extraction of the error parameters using local measurements (single-qubit measurement circuits on each qubit). Measurements were carried out on the 5-qubit IBMQ Santiago machine on August 16th. Poor resolution is my fault for not saving the data in the correct format.

We can evidently see from the above figure that the global measurement fidelity curve does not match up at all with the (green) curve obtained by assuming a local measurement error model and multiplying the experimentally determined local measurement error parameters together. While the orange curve seems to indicate a good fit, the green curve shows us that the assumption of a local measurement error model is not physical.

However, there is some definite tension upon re-running this entire routine (on the same machine on a different day). The most recent results are below:

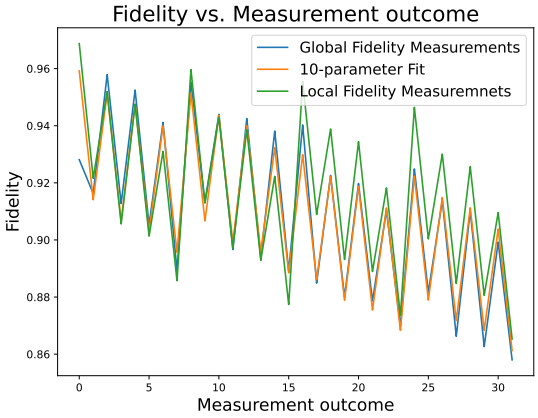

Figure 14: Fidelity results as obtained from global measurements on states $|j_1 j_2 j_3 j_4 j_5\rangle$, from a 10-parameter fit to a 5-square wave (local error) model, and from extraction of the error parameters using local measurements (single-qubit measurement circuits on each qubit). Measurements were carried out on the 5-qubit IBMQ Santiago machine on August 22nd.

This second figure seems to indicate that while the local measurement model is imperfect, to good first approximation it does well characterize the global fidelity curve. Evidently, there is some tension as to

whether the local measurement error model is a good approximation for the global measurement fidelities or not depending on when the routine is ran.

Given these results, we are somewhat hesistant to make a definite conclusion one way or another. Considering the physical implementation of the measurement, it is very possible that measurements of neighbouring qubits can introduce a further error term. Future investigations to benchmark this error will have to look at these second-order/two qubit measurement error terms, which hopefully (when combined with the single-qubit measurement errors) provides a full characterization of the measurement errors in the IBMQ machines. Fortunately, if this measurement model does provide a full characterization, then this will mean a good amount of scaling when we look at larger IBM quantum machines (as such errors are localized to single qubits and their neighbours, for which each qubit only has two of; essentially a 3-local model. Instead of $2N$ parameters, we will have $4N$ parameters, much less than is needed for the 2^N full set of parameters).

All the results obtained in this section can be investigated in the `ErrorModelFit.ipynb` notebook. Note that work on project is ongoing, even though the summer term has concluded!