Near Term Quantum Computation

by

Amara Katabarwa

(Under the Direction of Michael R Geller)

Abstract

Fault tolerant quantum computing will change fundamentally our conception of computing but is at the very least decades away. This is a consequence of the overhead brought about by the need to do Quantum Error Correction. In order to study quantum error correcting thresholds, assumptions about the noise model are made in order that the problem becomes tractable. In this thesis, we investigate and test the accuracy of one of these error models, *the Pauli Twirling Approximation* in the five qubit quantum error correcting code.

Secondly, we investigate two aspects of quantum computing before quantum fault tolerance. The first aspect is a proposed near term quantum processor that uses tune-able couplings in superconducting quantum devices and can naturally simulate real and symmetric Hamiltonians. We show that in fact this process can be turned into a universal computational device as we can prove that it can simulate in any complex Hermitian matrix. We do this by proving a new decomposition of lie groups. The second aspect of near term quantum computation is one that has been afforded to the quantum computing community as public quantum processors have been made available to the general public. We use these devices to study entanglement monotones of GHZ and cluster states.

Index words: Pauli Twirling Approximation, Quantum Error Correcting Codes, Entanglement Monotones, Lie Algebras, Lie Groups, Single Excitation Subspace Method, Qiskit

Near Term Quantum Computation

by

Amara Katabarwa

B.S., Emory University, 2011

A Dissertation Submitted to the Graduate Faculty
of The University of Georgia in Partial Fulfillment
of the
Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2019

Near Term Quantum Computation

by

Amara Katabarwa

Major Professor:	Michael R Geller
Committee:	Phillip Stancil
	K.K Mon

Electronic Version Approved:

Ron Walcott
Interim Dean of the Graduate School
The University of Georgia
December 2019

Acknowledgments

First and foremost I would like to thank Dr. Michael R. Geller, my advisor, from whom I learned most of what I know about what it means to do research. He has a knack for seeing what will be important on the two to three year time scale. As far as I know, excepting the research teams at D-wave systems, he was one of the first researchers to appreciate the concept and the necessity of doing research on Noisy Intermediate Scale Quantum Devices by proposing Single Excitation Subspace Processor at a time when all that mattered was exponential speed ups and quantum fault tolerance.

Next I would like to thank the team at Zapata, especially Peter D. Johnson and Tim Hirzel, who gave me the time, space and money to finish this daunting process. They provided me with a job when I could not get funding in my last year or find a part time job on short notice that would give me time and money to finish the thesis.

Last but certainly not least, I would like to thank my family, on whom I was a leech for longer than I should have been thus allowing me to be as carefree as I needed to be for as long as I shouldn't have been. A special thanks to my sister, Agasha, who drove 60 miles on a regular basis to see how I was doing.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The beginnings

It was 14 years after Feynman had conjectured that Quantum computers could efficiently simulate quantum systems that Seth Llyod proved it to be the case (1). This year had witnessed a dramatic increase in our understanding of quantum error correction with the work of Daniel Gottesmann, Andrew Steane and Raymond Laflamme. The first experimental architectures for qubits were given by David G.Cory *et. al* and independently Gershenfeld, Neil A. and Chuang, Isaac L. (2; 3) using NMR. The first, albeit very primitive, quantum processors were reported by(4). This simple two qubit processor implemented Grover's search algorithm. The experiments that had been done so far in NMR had involved states that were in fact mixed states. The viability of using NMR quantum processors was brought into question when S. L. Braunstein *et. al* (5) showed that quantum states within some neighborhood of maximally mixed states in fact had no entanglement. This point was brought home when Noah Linden and Sandu Popescu(6) showed that quantum entanglement was an absolute necessity in gaining the exponential power of quantum computers; further it was shown that the presence of entanglement was not only necessary but in fact was not sufficient if there was enough noise, in other words exponential resources where needed to combat the presence of noise. Efforts in other design architectures were also being pursued, for example E.Knill *et.al*(7) showed that contrary to belief universal quantum computation could be implemented with just linear optical elements. Experiments(8; 9) with four and five photons demonstrating teleportation. The superconducting architectural design was a bit late in the game which showed two qubit entanglement(10–12)

1.2 The Noisy Intermediate Scale Quantum (NISQ) Era

The concept of using quantum processors for near-term application was pioneered by D-wave systems(13). By compromising on the universality of the quantum processing unit and restricting oneself to Quadratic Unconstrained Binary Optimization problems (QUBO), D-wave systems claimed to provide evidence for a quantum speed up(14–16). These results were later contested as it was not clear that entanglement throughout the whole device was being used in an essential way(17–19). Another architecture that was proposed was the Single Excitation Subspace (SES) processor that was meant for near term universal quantum simulation. In it's original form it was only capable of simulating real and symmetric Hamiltonians. In this thesis, chapter 3, we prove a novel lie theory decomposition whose immediate result is that the SES processor can in fact simulate any complex Hamiltonian.

On the purely algorithmic side, the first viable near term quantum algorithm, Variational Quantum Eigensolver (VQE), was developed by Peruzzo, Alberto *et. al*(20). By preparing a trial state with random initial parameter on the quantum processor and classically optimizing the result, with the help of the variational principle of quantum mechanics one can find the ground state of the Hamiltonian. This sparked a great deal of research into various applications(21–26) of the variational principle that go beyond finding the ground from tasks like Quantum Error Correction numbers(27) to simulating various dynamical processes(28).

The second algorithm that was discovered and was quickly seen to be a viable near term quantum algorithm was the Quantum Approximate Optimization Algorithm (QAOA) developed Farhi *et.al*(29). One can approximately, think of QAOA as trotterized Adiabatic Quantum Computing with each time step being a variational parameter which is optimized. The initial wave function evolves alternatively under a forcing Hamiltonian and the problem Hamiltonian. One optimizes variational process so that one minimizes the energy to find the ground state of the problem hamiltonian. This has found applications in discrete optimizations(30–32). This algorithm has been studied as a way to get a quantum advantage(32–36).

1.3 Quantum Computing for the Masses

In 2016, IBM took an unprecedented step and publicly offered a 5 qubit quantum processor(37). This was device was later supplemented by another 5 qubit device and 16 qubit device. Suddenly, a flurry of work(38–47) was done implementing toy example of different quantum computing and information theory algorithms and concepts. In Chapter 4 of this thesis, we present work measuring entanglement monotones on publicly available quantum processors provided by IBM. We assume the state prepared on the quantum processor is pure and real then relax that assumption.

Chapter 2

Pauli Twirling Approximation and Quantum Error Correction

2.1 The Advent of Quantum Error Correction

In 1995, Rudolf Landauer(48) argued that goal of building a quantum computer would be impossible. He had three main arguments against this

1. To prevent the accumulation errors what he called *re-standardization* (resetting the computational unit to it's original value) but this would throw away information about the error

2. Hamiltonian evolution could not be carried out because of manufacturing defects.

3. Miniaturization of devices would require ever more reliable components but this would entail explosion in the cost of development, tooling, testing and fabrication.

Analyzing the work of Llyod(49; 50) he was skeptical of the claim that the sorts of errors that would occur in gate schemes of Lloyd would be of the kind that would be correctable.

Ironically, a few months before Peter Shor had just published a the first quantum error correcting code. This addressed the issue at least in principle of the re-standardization procedure by introducing the concept of ancilla that allowed one to detect and correct the position of errors without destroying the quantum state in which the information was stored. These ancilla qubits could be re-standardized. His scheme required increasing the number of qubits from k to $9k$, where k is the number of qubits that one wants to protect. A few months later Andrew Steane(51) and Raymond Laflamme *et.all* (52) produced quantum error correcting codes that used $7k$ and $5k$ total qubits respectively.

Daniel Gottesmann(53) in 1996 introduced the stabilizer formalism that encompassed the quantum error correcting codes that had been introduced before but also introduced an infinite class of codes that saturated the quantum hamming bound.

2.2 Stabilizer Group

The discussion, in this section is a summary of the discussion in Gaitan (54). A quantum error correcting code (QECC) t that encodes k qubits into n qubits is defined through an encoding map ζ from the k-qubit Hilbert space H^k onto a 2^k dimensional subspace C_q of the n-qubit Hilbert space H^n; it is required to be unitary. We choose the single-qubit computational basis states (CB) to be the eigenstates of σ_z^j i.e

$$\sigma_z^j \left|\delta_j\right\rangle = (-1)^{\delta_j} \left|\delta_j\right\rangle, \tag{2.1}$$

where j labels the qubits. The CB states for H^k are formed by talking all possible tensor products of the single-qubits CB states:

$$\left|\delta\right\rangle \equiv \left|\delta_1 \ldots \delta_k\right\rangle = \left|\delta_1\right\rangle \otimes \ldots \otimes \left|\delta_k\right\rangle. \tag{2.2}$$

This establishes a one to one correspondence between the unencoded states $\left|\delta\right\rangle = \left|\delta_1 \ldots \delta_k\right\rangle$ and encoded states $\overline{\left|\delta\right\rangle} = \overline{\left|\delta_1 \ldots \delta_k\right\rangle}$ so we have that:

$$\overline{\left|\delta_1 \ldots \delta_k\right\rangle} = \zeta \left|\delta_1 \ldots \delta_k\right\rangle. \tag{2.3}$$

Also we have that $\sigma_z^j \rightarrow Z_j = \zeta \sigma_z^j \zeta^\dagger$

Quantum stabilizer codes, C_q, is identified with a unique subspace that is fixed by elements of an abelian subgroup, S, known as the stabilizer group. More specifically we have that for all $\left|c\right\rangle \in C_q$

$$s \left|c\right\rangle = \left|c\right\rangle. \tag{2.4}$$

The stabilizer group is a subgroup of the Pauli group that consists of n-fold distinct tensor product of the Pauli operators $\sigma_z, \sigma_y, \sigma_x$ and the identity operator

The stabilizer group S is constructed from a set of $n-k$ operators $g_1, \ldots g_{n-k}$ known as the generators of S because each element in the stabilizer group can be each element can be written as a unique product of powers of the generators

$$s = g_1^{p_1} \ldots g_{n-k}^{p_{n-k}}. \tag{2.5}$$

Because the stabilizer group is abelian, the generators are mutually commuting operators and are in fact hermitian, unitary operators and of order 2. As a consequence of their order their eigenvalues are merely ± 1.

The parent space H^n is 2^n dimensional we need n commuting operators to specify a unique state in the Hilbert space. In fact these n operators can be chosen to be the following: $g_1 \ldots g_{n-k}; Z_1 \ldots Z_k$ and the 2^n simultaneous eigenstates of these operators can be chosen to be the basis for H^n. These eigenstates can be labeled by strings $l = l_1, \ldots l_{n-k}; \delta = \delta_1 \ldots \delta_k$ so that

$$g_i \left| l, \delta \right\rangle = (-1)^{l_i} \left| l, \delta \right\rangle, \tag{2.6}$$

$$Z_j \left| l, \delta \right\rangle = (-1)^{\delta_j} \left| l, \delta \right\rangle, \tag{2.7}$$

where $i = 1 \ldots n-k$, $j = 1 \ldots k$ and l_i and $\delta_j = 0, 1$

Note that for a given string $l = l_1 \ldots l_{n-k}$ the set of 2^k eigenstates $|l; \delta\rangle$ span a subspace of H^n which can be labeled by the string l i.e $C_q(l_1, l_2 \ldots l_{n-k}) \equiv C_q(l)$ and provide a partition of H^n. The subspace determined by the stabilizer group is labeled as $C_q(00 \ldots 0)$. In other words, the subspace determined by the stabilizer group are those elements in the Hilbert that are left invariant under the action of the stabilizer group elements so, $s \left| c \right\rangle = \left| c \right\rangle \forall c \in C_q$

Modeling the noise in a quantum setting is more complicated than in the classical case. It is known to be exponentially hard to exactly simulate the noise acting for example in a quantum circuit. Thus if we are to study the noise in a quantum circuit and how to apply QECC we have to assume a model of the errors that can be efficiently simulated on a classical computer. For this discussion we will model the noise as simply the random application of σ_x, σ_y and σ_z with probabilities p_x, p_y and p_z respectively. The σ_x operator flips a qubit, σ_z potentially changes the phase of the qubit and σ_y does a combination of both. To prove the above behavior apply the operators to the state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ which form a basis for the one qubit Hilbert space.

So what we have is that elements of the stabilizer group can be labeled by bit strings of length $n-k$ $p = p_1 \dots p_{n-k} \in F_2^{n-k}$, where F_2^{n-k} is the vector space of dimension $n-k$ over the field of characteristic 2.

Theorem 2.2.1 *Let E be an error operator and C_q a quantum stabilizer code with generators $g_1, \dots g_{n-k}$. The image $E(C_q)$ of C_q under E is $C(l)$ with $l = l_i \dots l_{n-k}$.*

$$l_i = \begin{cases} 0 \text{ if } [E, g_i] = 0 \\ 1 \text{ if } \{E, g_i\} = 0 \end{cases}$$

Proof We assume what will be proved later, that is E either commutes or anti-commutes with g_i. So $g_i E|c\rangle = (-1)^{l_i} E g_i |c\rangle = (-1)^{l_i} E|c\rangle$ where $l_i = \{0, 1\}$. This means that corrupted state is an eigenvector of the generators. Now $\{|l;\delta\rangle : l \in H_2^{n-k}, \delta \in H_2^k\}$ span H_2^n and so $E|c\rangle = \sum_l \sum_\delta a(l;\delta)|l;\delta\rangle$. Because E commutes or anti-commutes with all the generators and $|l;\delta\rangle$ are eigenvectors of the generator, this all implies that sum over l does not exist and we only have one particular value of l. Therefore $E|c\rangle = \sum_\delta a(l;\delta)|l;\delta\rangle$. So the error takes the element of the code space to a specific subspace of $C(l)$ as a consequence, $E(C_q) \subset C(l)$. But these vector spaces are the same dimension so they are in fact equal. $\square$

The lesson to take is that for each E we can attach a syndrome measurement $S(E) = l_1 \dots l_{n-k}$

Example

Quantum Stabilizer Code for Phase Flip Channel

$\eta : H_s^1 \longmapsto C_q \subset H_2^3$

We need 2 generators $\{g_1, g_2\}$. There 8 possible errors. We want to protect our state against a phase flip. Thus there are three possible errors $E_1 = \sigma_z^1, E_2 = \sigma_z^2, E_3 = \sigma_z^3$. Depending on whether we have $|0\rangle, |1\rangle$ we will have a phase flip or not. The phase flip will show up as a change in the relative phase. So we choose eigenstates of $\sigma_x^1\sigma_x^2$, $\sigma_x^2\sigma_x^3$ as elements of our code subspace.

Error Syndrome

$S(E) = (1, 0)$

$$\{\sigma_z^1, \sigma_x^1\sigma_x^2\} = 0$$
$$[\sigma_z^1, \sigma_x^2\sigma_x^3] = 0$$

$S(E) = (1, 1)$

$$\{\sigma_z^2, \sigma_x^1\sigma_x^2\} = 0$$
$$[\sigma_z^2, \sigma_x^2\sigma_x^3] = 0$$

$S(E) = (0, 1)$

$$[\sigma_z^3, \sigma_x^1 \sigma_x^2] = 0$$

$$\{\sigma_z^3, \sigma_x^2 \sigma_x^3\} = 0$$

Elements of the stabilizer group are of the form $s(p) = g_1^{p_1} g^{p_2}$ $p_1, p_2 \in \{0, 1\}$ So $S = \{I, \sigma_x^1 \sigma_x^3, \sigma_x^1 \sigma_x^2, \sigma_x^1 \sigma_x^3\}$

2.2.1 Deeper Study of Stabilizer Formalism

The Pauli group G_n has elements written as $e = i^\lambda \sigma_{j_1}^1 \otimes \cdots \otimes \sigma_{j_n}^n$ where $\lambda = \{0, 1, 2, 3\}, j_k = \{I, x, y, z\}$. Since $\sigma_y^k = -i\sigma_x^k \sigma_z^k$ we can always write the elements of the Pauli group as $e = i^\lambda \sigma_x(a)\sigma_z(b)$ where $a = a_1 \ldots a_n$ and $b = b_1 \ldots b_n$; a and b are n bit strings. We will work with the quotient group G_n/C, $C = \{\pm I, \pm iI\}$.

Theorem 2.2.2 *1. The order of G_n and G_n/C are 2^{2n+2} and 2^n respectively.*
2. $\forall e \in G_n$, $e^2 = \pm I, e^\dagger = \pm e$, $e^{-1} = e^\dagger$
3. $\forall e, f \in G_n$ either $[e, f] = 0$ or $\{e, f\} = 0$

Proof The order of the group is a trivial exercise in Combinatorics. We move on to the second claim. $e^2 = i^{2\lambda} \sigma_{j_1}^1 \otimes \cdots \otimes \sigma_{j_n}^n \sigma_{j_1}^1 \otimes \cdots \otimes \sigma_{j_n}^n = (-1)^\lambda (-1)^{a \cdot b} \sigma_x(a)^2 \sigma_x(b)^2 = (-1)^{\lambda + a \cdot b} I \implies \pm I$.
$e^\dagger = (-i)^\lambda \sigma_z(b)^\dagger \sigma_x(a)^\dagger = (-1)^{a \cdot b} (-1)^\lambda (i)^\lambda \sigma_x(a)\sigma_z(b) = \pm e$. Lastly, we prove that the elements either commute or anti-commute. Let $e = i^{\lambda_e} \sigma_x(a_e)\sigma_z(b_e)$, $f = i^{\lambda_f} \sigma_x(a_f)\sigma_z(b_f)$. The $ef = (i)^{\lambda_e + \lambda_f} \sigma_x(a_e)\sigma_z(b_e)\sigma_x(a_f)\sigma_z(b_f) = (i)^{\lambda_e + \lambda_f} (-1)^{b_e \cdot a_f} \sigma_x(a_e)\sigma_x(a_f)\sigma_z(b_e)\sigma_z(b_f) = (i)^{\lambda_e + \lambda_f} (-1)^{b_e \cdot a_f} (-1)^{a_e \cdot b_f} \sigma_x(a_f)\sigma_z(b_f)\sigma_x(a_e)\sigma_z(b_e) = (-1)^\Gamma fe$ with $\Gamma = b_e \cdot a_f + a_e \cdot b_f \in \mathbb{Z}$. If Γ is even then the elements commute but if Γ is odd then the elements anti-commute $\square$

2.2.2 Errors

Errors having a vanishing syndrome $S(E) = 0$ commute with all generators of the stabilizer group. Let $C(S)$ be the set of error $e \in G_n$ that commute with $c \in S$. $C(S)$ is called the centralizer.

Theorem 2.2.3 *The centralizer is a subgroup of* G_n

Proof Firstly, the centralizer has the identity since $[I, S] = 0 \, \forall s \in S$. Secondly, for $g \in C(S)$ we have $[g, s] = 0 \, \forall s \in S$ so $gs = sg$ so $g^{-1}ggs = sggg^{-1}$ but $g^2 = I \implies g^{-1}s = sg^{-1} \implies [g^{-1}, s = 0]$ so $g^{-1} \in C(S)$. Lastly, given two elements g_1, $g_2 \in C(S)$, their product $g_1 g_2$ in in $C(S)$ since $g_1 g_2 s = g_1 s g_2 = s g_1 g_2$ and therefore $[g_1 g_2, s] = 0$. $\square$

Since the stabilizer group is Abelian $S \subset C(S)$, if an error $e \in C(S)$ is in S the it needs no error correction but if it is in $C(S) - S$ it will not be detectable.The proof proceeds as follows.

Theorem 2.2.4 $C(S)$ *is a normal subgroup of* G_n

Proof Let $c \in C(S), s \in S, g \in G_n$. We have $csc^{-1} \in S$. Now look at $gc(g^{-1}sg)g^{-1} = g(g^{-1}sg)cg^{-1} = s(gcg^{-1}) \in C(S)$. Therefore $C(S)$ is normal. A slightly more abstract way is to notice that G_n acts on S by conjugation with the kernel being $C(S)$. Now a kernel is a group and moreover a normal subgroup. $\square$

This all means we can create the quotient group $G_n/C(S)$.

Theorem 2.2.5 *Two elements* e_1, e_2 *are in the same coset if and only if they have the same error syndrome.*

Proof If they are in the same coset then they differ by an element c of $C(S)$. Let $e_2 = e_1 c$. Consider $e_1 g_i \left|d\right\rangle$ with $\left|d\right\rangle$ being a code word. So $e_2 c^{-1} g_i \left|d\right\rangle = e_2 g_i c^{-1} \left|d\right\rangle = e_2 g_i \left|d\right\rangle = e_1 g_i \left|d\right\rangle$. In other direction, $e_2 e_1 g = g e_2 e_1$ so the product belongs in the centralizer. The product commutes because e_1, e_2 have the same syndrome measurement. Therefore $\exists c \in C(S)$ such that $e_2 e_1 = c \implies e_1 = c e_2^{-1}$, proving the theorem

The result is that different error syndromes are in 1-1 relation with the cosets of the centralizer.

Theorem 2.2.6 $|G_n : C(S)|$ = *number of cosets is* 2^{n-k}.

Proof Each coset corresponds to a unique syndrome measurement. The number of syndrome measurements is 2^{n-k}

Theorem 2.2.7 *The number of elements in* $C(S) = 2^{n+k+2}$

Proof By Lagrange's theorem $\frac{|G_n|}{C(S)} = |G_n : C(S)|$. So $\frac{G_n}{|G_n:C(S)|} = |C(S)| = \frac{2^{n+2}}{2^{n-k}} = 2^{n+k+2}$

Now $C(S)/S$ has elements that commute with the stabilizer generators but change the codeword. These turn out to be the logical operators.

2.3 Explicit production of the encoding circuit

2.3.1 Type 1 and Type 2 generators

The discussion in this section is a summary of material from(54; 55). Given a computational basis state $|\phi\rangle$ then a state in the codespace is $|\psi\rangle = \sum_{M \in S} M |\phi\rangle$, where M is a stabilizer and S is the stabilizer group. $|\phi\rangle$ will be referred to as the seed of the code word $|\psi\rangle$. Note that $M |\phi\rangle$ acts the seed for the same quantum code words as $|\phi\rangle$. Not every seed will produce a non-zero code word. We can produce a basis for the codespace in the following manner: pick a seed, and apply elements of the stabilizer group that do not produce a non-zero code; this will produce a state $|\psi_1\rangle$. Then pick another seed that is not in the linear expansion of $|\psi\rangle$ and apply elements in the stabilizer group that do not lead to a non-zero code to get $|\psi_2\rangle$. It is shown in (55) that this procedure leads to a basis for the vector space stabilized by the stabilizer group. The argument proves this by supposing you have a set of stabilizers $\mathcal{M} = (M_1, M_s \ldots M_r)$ and a set of classical states C_r and imagining the adding to the set $\mathcal{M}$ a stabilizer M_{r+1}. This stabilizer could do one of three things to a given seed ϕ in S_r

1. M_{r+1} could send $|\phi\rangle$ to a classical state not in S_r

2. M_{r+1} could return $\pm|\phi\rangle$

3. M_{r+1} could return $|\phi'\rangle \in S_r$

It turns out the third option can be ignored since M_{r+1} can be written in a form that acts like the second option. The first option leads to generators of *type 1* and second option leads to generators of *type 2*. Generators of *type 1* will eventually make the order of the set S_r, 2^b where b is the number of *type 1* generators. A generator of *type 2* must act on each qubit as the $I_i, -I_i$ or as Z_i where i is the label of the qubit. We can use $|\vec{0}\rangle = |00\ldots0\rangle$ as our first seed in our quest to find basis for the vector space stabilized by the stabilizer group. Any other seed can reached by operating some $N \in S$ that is a product of X. For $N|0\rangle$ to act as the seed for a non-trivial state, N must commute with every *type 2* generator. Assuming that our stabilzer group S is generated by $\langle M_1, M_2, \ldots M_a\rangle$, we can get a full set of 2^{n-a} seeds by taking products of $(n-a)$ operators $N_1, \ldots N_{n-a}$. The N_i are called the N_i are called *seed generators*.

Once we have determined *type 1* generators, M_i, of the stabilized vector space and the seed generators, N_i, we can define a unitary transformation to perform the encoding

$$\overbrace{|0\ldots0\rangle}^{d} \otimes |c_1 c_2 \ldots c_k\rangle \longmapsto \frac{1}{\sqrt{2^b}} \prod_{type\ 1} (1+M_i) N_1^{c_1} N_2^{c_2} \ldots N_k^{c_k} |\vec{0}\rangle\,, \tag{2.8}$$

where $|\vec{0}\rangle$ is has length n and $d = b + r$ where b is the number of *type 1* generators r is the number of *type 2* generators.

2.3.2 Standard Form of the Stabilizer code

In order to find the encoding scheme we use the language of binary vectors. Suppose we have a generator $G_i = G_{i_1} \otimes \dots G_{i_n}$. We introduce two vectors X_{G_i} and Z_{G_i}

$$(X_{G_i})_j = \begin{cases} 1 & \text{if } G_{ij} = X \text{ or } Y \\ 0 & \text{if } G_{ij} = Z \end{cases} \tag{2.9}$$

and

$$(Z_{G_i})_j = \begin{cases} 1 & \text{if } G_{ij} = Z \text{ or } Y \\ 0 & \text{if } G_{ij} = X \end{cases} \tag{2.10}$$

We also have X_G matrix of generators $G_1, \dots G_d$ is defined as the $n \times d$ matrix denotes as X_G where

$$(X_G)_{ji} = \begin{cases} 1 & \text{if } G_{ij} = X \text{ or } Y \\ 0 & \text{if } G_{ij} = Z \end{cases} \tag{2.11}$$

i.e a matrix whose columns are X_{G_i} and a similarly defined Z_G matrix whose columns are Z_{G_i}. In this new language the X-vectors that are linearly independent are the *type 1* generators and those that are null are the *type 2* generators.

It is useful to divide the set of qubits into three sets $\mathcal{A} = (1 \dots b)$, $\mathcal{B} = (b+1, \dots n-k)$ and $\mathcal{C} = (n-k+1 \dots n)$

We first summarize finding the encoding procedure before giving a detailed explanation

2.3.2.1 Summary of encoding procedure

I It is useful to divide the set of qubits into three sets $\mathcal{A} = (1 \dots b)$, $\mathcal{B} = (b+1, \dots n-k)$ and $\mathcal{C} = (n-k+1 \dots n)$

II Apply the seed generators (logical X operators) on the set $\mathcal{B}$ in a controlled fashion i.e apply the logical operator X_j (it was denoted as N_j before) conditioned on the state of qubit $n-k+j$ in set $\mathcal{C}$

III Apply Hadamard operators on the set of qubits in $\mathcal{A}$

IV Apply the projection operators created by stabilizer operators conditioned on the state of the j^{th} qubit where j is in set $\mathcal{B}$ on the qubits in $\mathcal{A}$

2.3.2.2 Careful exposition of encoding procedure

Re-writing $N_i^{c_i}$ as $X_k^{c_k}$ since the seed generators are in fact products of X operators. We can combine X_G and Z_G into a check matrix H like

$$H = (X_G^T | Z_G^T) \tag{2.12}$$

The check matrix can be put in the so called standard form which is the form

$$H = \left(\begin{array}{ccc|ccc} I_{b\times b} & A_{1(b\times(n-k-b))} & A_{2(b\times k)} & B_{b\times b} & C_{1(b\times(n-k-b)} & C_{2(b\times k)} \\ 0_{b\times b} & 0_{(b\times(n-k-b))} & 0_{(b\times k)} & D_{b\times b} & I_{(b\times(n-k-b))} & E_{(b\times k)} \end{array}\right) \tag{2.13}$$

The logical operators have $n-k$ degrees of freedom so that binary vector corresponding to them can chosen to be in the following form

$$v(X_l) = \left(\begin{array}{c} 0_{b\times 1} \\ u_{2(l)(n-k-b)\times 1} \\ u_{3(l)k\times 1} \\ \hline v_{b\times 1} \\ 0_{(n-k-b)\times 1} \\ 0_{(n-k-b)\times 1} \end{array}\right) \tag{2.14}$$

with u_3 have the special form $u_3(l) = (0 \dots 1_j \dots 0)$, where for example $u_{2(l)(n-k-b)\times 1}$ denotes a $(n-k-b)\times 1$ vector got by picking the l^{th} column from the matrix got by putting together the binary vectors of logical operators

$$X = \begin{pmatrix} | & \dots & | \\ v(X_1) & \dots & v(X_k) \\ | & \dots & | \end{pmatrix} \tag{2.15}$$

The above matrix takes the special *standard* form

$$X = \begin{pmatrix} 0 & E^T & I & | & (E^T C_1 + C_2^T & 0 & 0 \end{pmatrix} \tag{2.16}$$

With additional notation we write X_l as

$$X_l = S_x[u_2(l)] S_Z[v_1[l]] \sigma_x^{n-k+l} \tag{2.17}$$

where

$$S_x[u_2(l)] \equiv (\sigma_x^{b+1})^{u_{2,1(l)}} \dots (\sigma_x^{n-k})^{u_{2,n-k-b(l)}} \tag{2.18}$$

$$S_z[v_1(l)] \equiv (\sigma_z^{b+1})^{v_{1,1(l)}} \dots (\sigma_z^{r})^{v_{1,b(l)}} \tag{2.19}$$

We thus have the action of the logical operators being

$$X_k^{\delta_k} |0 \dots 0\rangle_n = \begin{cases} |0 \dots 0\rangle_n & \delta_k = 0 \\ S_x[u_2(k)] |0 \dots 01\rangle_n & \delta_k = 1 \end{cases} \tag{2.20}$$

Making our qubit sets more obvious we have the following action

$$X_k^{\delta_k} |0 \dots 0\rangle_n = \overbrace{|0 \dots 0\rangle}^{b} \otimes S_x[u_2(k)]^{\delta_k} \overbrace{|0 \dots 0\rangle}^{n-k-b} \otimes \overbrace{|0 \dots \delta_k\rangle}^{k} \tag{2.21}$$

Thus applying all the logical operators to get a computational basis state in the logical space is

$$X_1^{\delta_1} \dots X_k^{\delta_k} |0 \dots 0\rangle_n = \prod_{j=1}^{k} \overbrace{|0 \dots 0\rangle}^{b} \otimes S_x[u_2(j)]^{\delta_j} \overbrace{|0 \dots 0\rangle}^{n-k-b} \otimes \overbrace{|0 \dots \delta_k\rangle}^{k} \tag{2.22}$$

Next we need to apply the operator $(1 + g_i)$ where $j = (1 \dots b)$. First we define $U_j \equiv S_x[u_2(j)]^{\delta_j}$. These are controlled operations and their product is denoted as $U_T = U_1 \dots U_k$ Now using the standard form in 2.13 we have that the binary vector representing the stabilizer operator is

$$v_j^T(g_i) = \begin{pmatrix} 0 & \dots 1_j \dots & 0 & A_1(j) & A_2(j) & | & B(j) & C_1(j) & C_2(j) \end{pmatrix} \tag{2.23}$$

So in standard form $g_i = T_j \sigma_x^j (\sigma_z)^{B(j)}$ where T_j is the operator that identifies the operator that remains when we factor out all operators associated with qubit j. We have the operator that needs to be applied that projects the quantum state into the code subspace

$$\mathcal{G} = \prod_{j=1}^{n-k} (1 + g_i) = \mathcal{G}_1 \mathcal{G}_2 = \prod_{j=1}^{b} (1 + g_i) \prod_{j=b+1}^{n-k} (1 + g_j), \tag{2.24}$$

where the first product is over the *type 1* generators and the second product is over *type 2* generators. We define $|\psi\rangle = X_1^{\delta_1} \dots X_k^{\delta_k} |00 \dots 0\rangle_n$ and it is easy to see that $|\delta_1 \bar{\dots} \delta_k\rangle = \mathcal{G}_1 X_1^{\delta_1} \dots X_k^{\delta_k} \mathcal{G}_2 |00 \dots 0\rangle_n = \mathcal{G}_1 X_1^{\delta_1} \dots X_k^{\delta_k} |00 \dots 0\rangle_n$ where we used that the *type 2* generators act on a different set of qubits and therefore commute with the logical operators and that the *type 2* generators are product of Z operators and therefore act trivially act on our state.

Looking at 2.8 we see that the next step is explicate how to apply the non-unitary operation $(1 + g_j) |\psi\rangle$. We easily see that

$$(1 + g_j) |\psi\rangle = U_T |0 \dots 0 \delta_1 \dots \delta_k\rangle + T_j \sigma_x^j (\sigma_z)^{B(j)} U_T |0 \dots 0 \delta_1 \dots \delta_k\rangle . \tag{2.25}$$

The $\sigma_x^j(\sigma_z^j)^{B_j(j)}$ commutes with the U_T since they act on different sets of qubits and has the following action

$$\sigma_x^j(\sigma_z)^{B_j(j)}\,|0\ldots 0_j\ldots 0\;\delta_1\ldots\delta_k\rangle = |0\ldots 1_j\ldots 0\;\delta_1\ldots\delta_k\rangle\,. \tag{2.26}$$

Therefore the action of the projection operator is the following

$$(1+g_j)\,|\psi\rangle = U_T H_j\,|0\ldots 0_j\ldots 0\;\delta_1\ldots\delta_k\rangle = U_T\{|0\ldots 0_j\ldots 0\;\delta_1\ldots\delta_k\rangle + T_j\,|0\ldots 1_j\ldots 0\;\delta_1\ldots\delta_k\rangle\}, \tag{2.27}$$

where H_j is the hadamard gates on the j^{th} qubits in set $\mathcal{A}$.

Note: The generators are applied onto the quantum state by a controlled operation i.e T_j acts only if j in the set $\mathcal{A}$ is in the one state.

2.3.3 5 QUBIT CODE EXAMPLE

For the sake of concreteness we choose the 5 qubit code which will be studied later on. The stabilizers are $g_1 = \sigma_x^1\sigma_z^2\sigma_z^3\sigma_x^4\,, g_2 = \sigma_x^2\sigma_z^3\sigma_z^4\sigma_x^5,\,, g_3 = \sigma_x^1\sigma_x^3\sigma_z^4\sigma_z^5\,, g_4 = \sigma_z^1\sigma_x^2\sigma_x^4\sigma_z^5$ and the logical X and Z operators, respectively are $X = \sigma_x^1\sigma_x^2\sigma_x^3\sigma_x^4\sigma_x^5\;\; Z = \sigma_z^1\sigma_z^2\sigma_z^3\sigma_z^4\sigma_z^5$. In stand form we have we have the following check matrix

$$H = \left(\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right) \tag{2.28}$$

The 5 qubit code has no *type 2* generators. In standard form the logical operators are

$$X = \left(\begin{array}{ccccc|ccccc} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{array}\right) \tag{2.29}$$

$$Z = \left(\begin{array}{ccccc|ccccc} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array}\right) \tag{2.30}$$

which gives in pauli notation the following operators $X = \sigma_z^1\sigma_z^4\sigma_x^5 \;\; Z = \sigma_z^1\sigma_z^2\sigma_z^3\sigma_z^4\sigma_z^5$ The stabilizers in standard form are

$$\begin{aligned} g_1 &= (-i\sigma_y^1)\sigma_z^2\sigma_z^4(-i\sigma_y^5) \\ g_2 &= \sigma_x^2\sigma_z^3\sigma_z^4\sigma_x^5) \\ g_3 &= \sigma_z^1\sigma_z^2\sigma_x^3\sigma_x^5 \\ g_4 &= \sigma_z^1\sigma_z^3(-i\sigma_y^4)(-i\sigma_y^5) \end{aligned} \tag{2.31}$$

and the corresponding T operators are

$$\begin{aligned} T_1 &= \sigma_z^2\sigma_z^4(-i\sigma_y^5) \\ T_2 &= \sigma_z^3\sigma_z^4\sigma_x^5) \\ T_3 &= \sigma_z^1\sigma_z^2\sigma_x^5 \\ T_4 &= \sigma_z^1\sigma_z^3(-i\sigma_y^5) \end{aligned} \tag{2.32}$$

As a result of having no *type 2* generators we have $S_x[u_2] = I$ and our fiducial state is $|\overbrace{0000}^{b=4}\overbrace{\delta_1}^{k=1}\rangle$

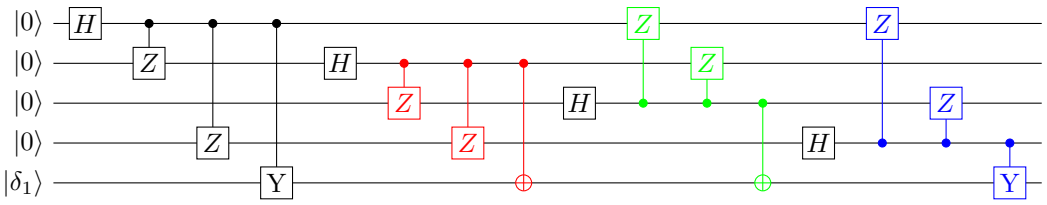


Figure 2.1: The operators T_1, T_2, T_3, T_4 are represented in black, red, green and blue respectively

The first two operations in T_1 and T_2 act trivially and can be ignored to give the following quantum circuit.

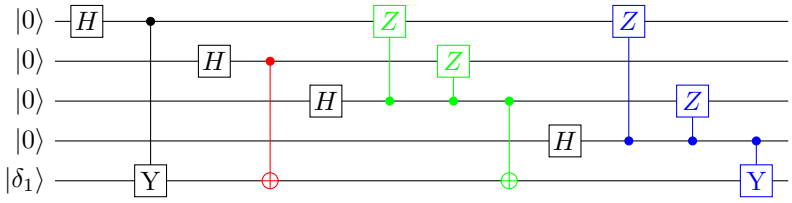


Figure 2.2: The operators T_1, T_2, T_3, T_4 are represented in black, red, green and blue respectively. The operators acting trivially have been removed

2.4 PTA AND 5 QUBIT CODE

Feynman introduced the idea of a universal quantum simulator,(56) noting that a classical Turing machine would require a time exponential in the number of particles to simulate quantum phenomena, while his proposed simulator made from quantum components would avoid such a scaling. A surge of interest grew around the nascent field of quantum information theory when Shor discovered his now famous algorithm, (57) which can provably factor numbers in polynomial time, in contrast to a classical machine which is believed to scale exponentially with the number of bits of the input. However, it was clear from the very beginning that the great power of quantum computing—using quantum superpositions and entanglement—also presented the greatest challenge to its realization; namely, the incredible delicacy of quantum states in the presence of unwanted environmental interactions.

The first major step to protect the delicacy was taken by Shor when he proposed a quantum circuit that could correct for any single-qubit error by encoding a logical qubit

into 9 physical qubits. (58) Shor's 9-qubit code and it's generalizations work perfectly if we make the unphysical assumption that all syndrome measurements are error-free. It is therefore necessary to understand the effect of decoherence and unitary gate errors on a complete fault-tolerant circuit. Unfortunately, the direct approach to this problem, namely a full Hilbert space simulation of the quantum circuit in the presence of errors and noise, is impractical because of the exponential relationship between amount of memory and time needed to simulate quantum circuits and the number of qubits. A way around this problem is to rely on the Gottesmann-Knill theorem, which shows that any circuit in which we prepare initial states in the computational basis, use only gates from the normalizer of the Pauli group (in this case the Clifford group), and measure operators from the Pauli group, can be efficiently simulated on a classical computer. (59) We are thus provided with a class of efficient error models that includes the Pauli and Clifford channels.

Simulation of a noisy quantum circuit is accomplished by performing each ideal operation followed by an error (a gate from Pauli or Clifford group) with some probability. It is then necessary to construct an error channel such that one approximates the true noise process as accurately as possible (with respect to some measure), and ideally with the additional property that the approximate channel upper-bounds the actual error. The first steps in this direction were taken by Magesan *et al.*(60) and Gutiérrez *et al.* (61) These investigations considered a single qubit density matrix and not a quantum error correcting circuit. Geller and Zhou (62) took a different approach and asked how well the Pauli twirling approximation (PTA), obtained by twirling the exact error channel over the Pauli basis, performed on a 4-qubit Bell-state preservation circuit, where an analog of the logical error rate can be defined. Despite its simplicity, the PTA was found to work surprising well over a large range of physical error rates, but did not always upper bound the exact error. A second test of the PTA was carried out by Tomita and Svore,(63) where the logical error rate was calculated for the distance-3 surface code. Although the PTA test was not the main focus of their work, these authors found excellent agreement for logical σ^z errors, but that the PTA

overestimated the logical σ^x error by a factor of 5 to 10, depending on the qubit T_1 time. The results of Tomita and Svore,(63) and the desire to extend the work of Ref. (62) to a test of the PTA on an actual logical error rate calculation, motivated the work reported here. In addition, two other related investigations have recently appeared: Puzzuoli *al.*(64) discussed the construction of efficient (Pauli and Clifford) error channels obtained by minimizing the diamond norm subject to the constraint that the approximate channel always upper bounds the error (an *honest* representation in the terminology of Refs. (60; 64)) and tested their accuracy when applied to error-correcting circuits. Gutiérrez and Brown (65) focused on Clifford channels and computed error thresholds for the Steane [[7,1,3]] code. The results of Refs. (62; 63; 65), together with the results reported below, suggest that the PTA is a reliable (and honest) predictor of the logical error rate, at least for low-distance codes.

Methods and results

Consider the time evolution of a density matrix ρ represented by some superoperator Λ; then we have

$$\rho \to \Lambda(\rho) = \sum_m E_m \rho E_m^\dagger, \tag{2.33}$$

where the E_m are $N \times N$ Kraus matrices. Next consider a finite set of operations $\mathcal{B} = \{B_m\}$ with $m = 1, \ldots, K$. *Twirling* (66–69) the channel to obtain a new channel $\tilde{\Lambda}$ is to perform the operation

$$\tilde{\Lambda} = \frac{1}{K} \sum_{m=1}^{m=K} B_m^\dagger \Lambda(B_m \rho B_m^\dagger) B_m. \tag{2.34}$$

To arrive at the PTA, we simply consider the set $\mathcal{B}$ to be the n-qubit Pauli basis $\mathcal{P}_n$, defined as consisting of all possible tensor products

$$\mathcal{P}_n = \{I, X, Y, Z\}^{\otimes n}, \tag{2.35}$$

giving a total of 4^n distinct elements. Performing the PTA gives $\tilde{\Lambda}$ that is always diagonal in the Pauli basis, namely

$$\tilde{\Lambda} = \sum_{B_m \in \mathcal{P}_n} p_m B_m \rho B_m. \tag{2.36}$$

If Λ is trace-preserving then $\sum_m p_m = 1$, otherwise $\sum_m p_m < 1$. A detailed application of the PTA to single qubit decoherence models is carried out in Ref. [(62)].

In this work we apply the PTA to the calculation of the logical error rate for the 5-qubit code. The 5-qubit code is the smallest quantum error correcting code that can encode a logical qubit, and detect and correct a single one-qubit error. It is a distance 3 quantum error correcting circuit, meaning that with one error correcting cycle, 3 is the lowest number of single qubit errors that cannot be detected. This code can be implemented by measuring the stabilizers

$$X_1Z_2Z_3X_4,\ X_2Z_3Z_4X_5,\ X_1X_3Z_4Z_5,\ Z_1X_2X_4Z_5. \tag{2.37}$$

Note that by starting with the first stabilizer one can arrive at the other three by a cyclic permutation of the qubits. The logical $|0\rangle$ and logical $|1\rangle$ states for this code are

$$\begin{aligned} |0\rangle_L = \frac{1}{4}(&|00000\rangle + |11000\rangle + |01100\rangle + |00110\rangle + |00011\rangle + |10001\rangle \\ &- |10100\rangle - |01010\rangle - |00101\rangle - |10010\rangle - |01001\rangle \\ &- |11110\rangle - |01111\rangle - |10111\rangle - |11011\rangle - |11101\rangle) \end{aligned} \tag{2.38}$$

and

$$\begin{aligned} |1\rangle_L = \frac{1}{4}(&|11111\rangle + |00111\rangle + |10011\rangle + |11001\rangle + |11100\rangle + |01110\rangle \\ &- |01011\rangle - |10101\rangle - |11010\rangle - |01101\rangle - |10110\rangle \\ &- |00001\rangle - |10000\rangle - |01000\rangle - |00100\rangle - |00010\rangle). \end{aligned} \tag{2.39}$$

A logical state is prepared in the computational basis using the first five (data) qubits, as shown in Fig. 2.3. The next four qubits are used as ancilla qubits to measure the four

stabilizers after which four measurement outcomes (x_1, x_2, x_3, x_4) are obtained. There are 16 possible measurement outcomes which are in a one to one correspondence with the 16 possible errors that might occur [counting the outcome (0,0,0,0) as a trivial error]. In Table 2.1 we list all possible measurement outcomes and the corresponding single qubit errors. We call the implementation of the circuit and performing the measurement step a *cycle*, which is shown in Fig. 2.3. If one goes through a cycle and a single error occurs on any of the first 5 qubits, this might be reflected in the measurement result and thus detected. But instead suppose that no errors occur on the data qubits but right before the measurement step a bit-flip error occurs on the first syndrome qubit, giving the measurement outcome of (1,0,0,0). An incorrect interpretation of the result would be to conclude that one of the 16 possible errors on the data qubits has occurred, whereas in fact the fault lies with a syndrome qubit. We therefore require a protocol that is tolerant to a single syndrome-qubit (or readout) error. To this end, we note that for errors uncorrelated in time, it is likely that after readout and re-initialization that the syndrome qubit will return to its original "faithful" state at the end of the next cycle. The procedure followed in our simulations is therefore the following:

1. With the initial 9-qubit density matrix (representing data and syndrome qubits) perform the stabilizer measurements and the measurement step to complete one cycle. Record the measurement outcome.

2. For the next cycle, re-initialize the syndrome qubits but use the 5-qubit density matrix from the end of the last cycle.

3. Repeat step 2 until the same measurement outcome is obtained three times in succession. Call this event the completion of a *trial*.

After observing the same measurement outcome for three cycles in a row, we calculate

$$P = 1 - \mathrm{Tr}(\rho_{\mathrm{c}}\rho_{\mathrm{m}}), \tag{2.40}$$

measurement result	single-qubit error
0000	I
0001	Z_1
0010	X_3
0011	Z_0
0100	X_0
0101	X_2
0110	Z_4
0111	Y_0
1000	Z_2
1001	X_4
1010	X_1
1011	Y_1
1100	Z_3
1101	Y_2
1110	Y_3
1111	Y_4

Table 2.1: Syndrome measurement outcomes and their corresponding predicted single-qubit errors.

where ρ_{c} is the data-qubit density matrix obtained at the end of the final cycle and ρ_{m} is the data-qubit density matrix predicted by the stable measurement outcome. We then define the logical error rate as

$$P_{\mathrm{L}} = \frac{1}{N} \sum_{\text{trials}} P, \tag{2.41}$$

where N is the number of trials. " The value of N is chosen to make the sampling errors much smaller than the differences between the exact and PTA logical error rates we are interested in. Defining the logical error rate this way allows us to calculate an error rate that could be measured experimentally.

Our work here is done with surface code in mind where for example one deals with single syndrome qubits and the robustness of a syndrome measurement is achieved by comparing measurement results from a number of measurement cycles.

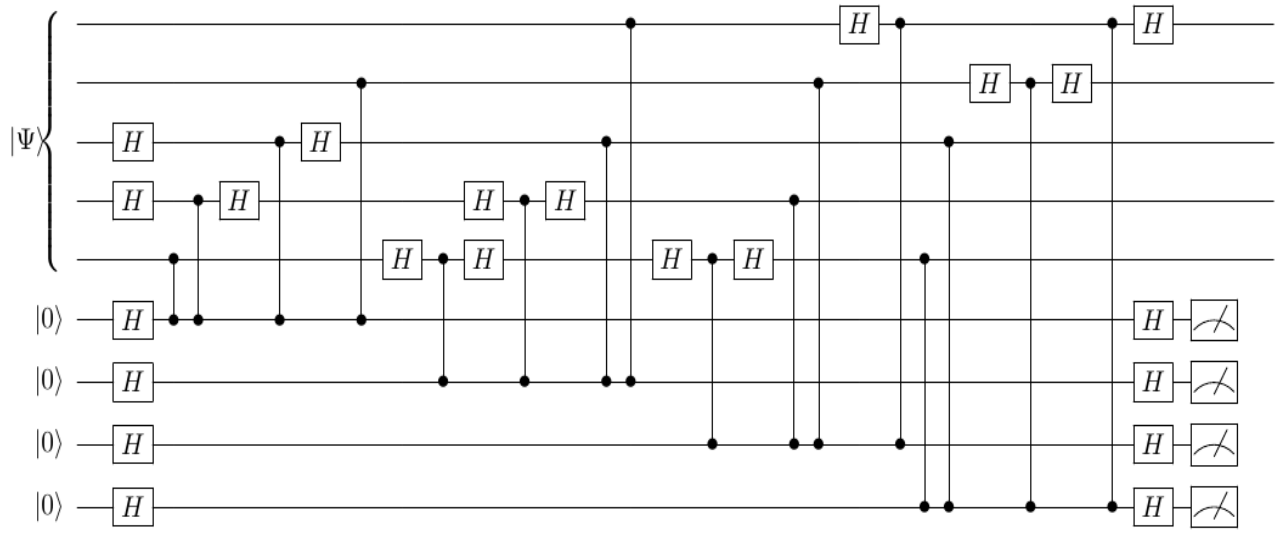


Figure 2.3: Stabilizer measurement circuit for the 5-qubit code written in terms of CZ gates (vertical lines with dots). A *cycle* is moving through this circuit once and performing the measurement step.

For the exact calculations of the average logical error rate P_{L}, decoherence was included by using Kraus matrices for amplitude damping and pure dephasing as defined in Ref. [(62)]. The gates are assumed to act instantaneously and the non-unitary evolution is implemented using the operator sum representation between the action of the gates for a time of 25×10^{-9} s. Unitary gate errors are introduced by using the non-ideal CZ gate(62)

$$V = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{1-E_1} & \sqrt{E_1}e^{i\phi} & 0 \\ 0 & -\sqrt{E_1}e^{-i\phi} & \sqrt{1-E_1} & 0 \\ 0 & 0 & 0 & e^{i\delta} \end{pmatrix}, \tag{2.42}$$

whereas the Hadamards are taken to be ideal. The form (2.42) reflects actual errors in a CZ gate implemented with superconducting qubits (neglecting leakage). (70) There are three parameters in (2.42) that can be changed, namely, E_1, ϕ and δ. In the simulations,

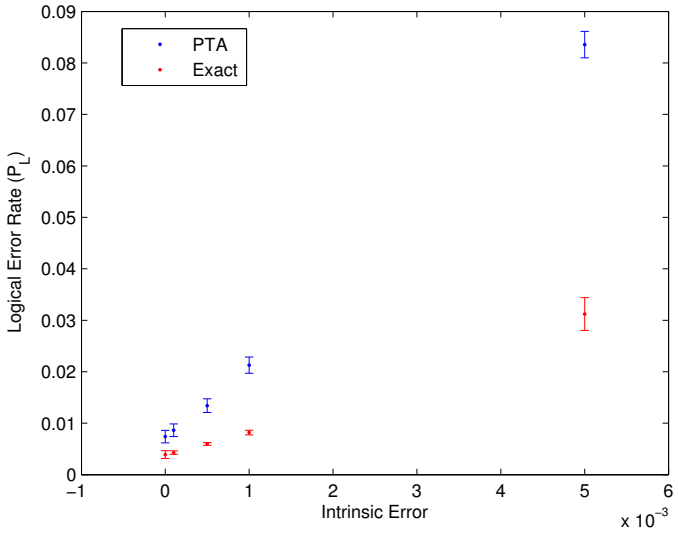


Figure 2.4: Logical error rate for the $|0\rangle_L$ state with $T_1 = T_2 = 100\mu$s.

we choose $\phi = 0$ and distribute the total intrinsic gate error E equally between E_1 and δ. By *intrinsic* or unitary gate error we mean the gate error in the absence of decoherence. The PTA applied to (2.42) yields 16 two-qubit Pauli error operators with probabilities given in Ref. (62). To obtain standard errors on the order 10^{-3} or smaller, about 20 trials were required, which took several days of runtime to complete. The PTA calculations of $P_{\rm L}$ were done using classical Monte Carlo, which introduces larger sampling errors. A total of 10000 trials were performed to get sampling errors down to around 10^{-3}.

Figures 2.4, 2.5, and 2.6 give the logical error rate $P_{\rm L}$ versus intrinsic error for three values of T_1, with $T_2 = T_1$. We find in these cases that the PTA overestimates the logical error rate by about a factor of 2 to 3. In Fig. 2.7, we fix the total intrinsic error to $E = 10^{-3}$ and test the PTA for five different states on the logical Bloch sphere: the eigenstates of σ^z, σ^x, and the $+1$ eigenstate of σ^y.

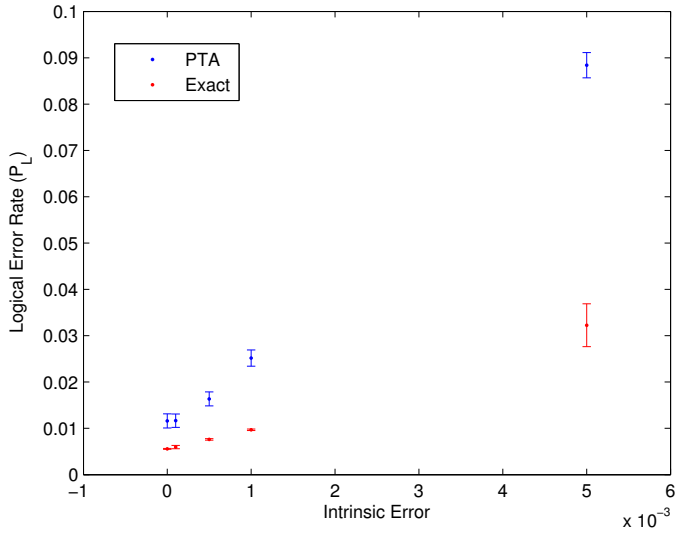


Figure 2.5: Logical error rate for the $|0\rangle_L$ state with $T_1 = T_2 = 70\mu$s.

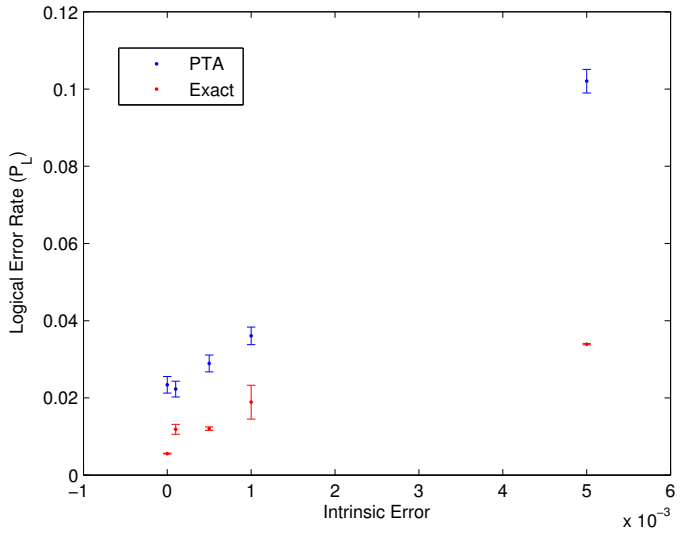


Figure 2.6: Logical error rate for the $|0\rangle_L$ state with $T_1 = T_2 = 40\mu$s.

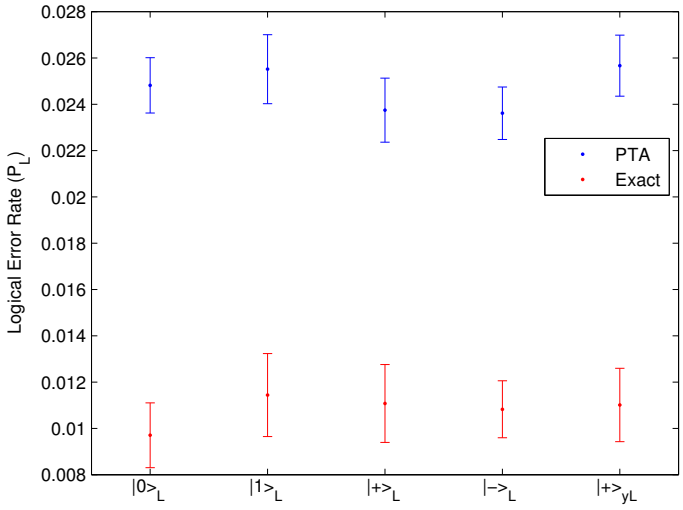


Figure 2.7: Logical error rate for different states on the Bloch sphere with $T_1 = T_2 = 70\mu$s and 10^{-3} intrinsic error.

Conclusions

We have studied the PTA logical error rate compared to an exact calculation that includes both decoherence (amplitude damping and pure dephasing) and unitary gate errors. The tests reported here include 20 different settings—physical error rates and/or initial logical states—with the PTA *overestimating* the logical error rate by a factor of 1.9 to 3.1, with a mean ratio of 2.35. In the language of Megasan *et al.*,(60) we find that the PTA is always honest (the ratio is > 1) for the parameter regimes considered. We find no significant difference between PTA's performance for bit-flip and phase-flip errors, which would be reflected in Fig. 2.7, in contrast to the results of Tomita and Svore. (63) The explanation for the difference between the results is currently not known. There are a number of significant differences between the two arenas in which PTA's performance is measured that could be the explanation : the fact that in surface code we have a two dimensional structure which allows for more interactions between the qubits,the fact that one is topological and the other is not, or it could be as simple as the work in Ref. (63) studied a considerably larger system.

This warrants future study. We also find that, as expected, the PTA is less accurate for unitary errors, in agreement with Refs. (62) and (64).

Chapter 3

The ABA Decomposition and Single Excitation Subspace Method

3.1 Introduction to Lie Algebras

Let F be a field. A *Lie Algebra* over F is a a F-vector space K, together with a bilinear map, the *lie bracket* :

$$L \times L \to L \qquad (x, y) \longmapsto [x, y] \tag{3.1}$$

satisfying the following properties

- .[x,x] =0 for all $x \in L$
- .[x,[y,z]] + [y,[z,x]] + [z,[x,y]]=0 for all $x, y, z \in L$

Some examples of lie algebras

1. Let F = $\mathbb{R}$. The vector product (x,y) $\longmapsto x \wedge y$
2. Any vector space V has a lie bracket defined by [x,y]=0 for all $x, y \in V$
3. Set of all linear maps from V $\to$ V. This is a vector space over F known as the general linear algebra [x,y] $\equiv x \circ y - y \circ x$ for all x, y $\in gl(V)$ where $\circ$ denote the composition of maps

3.2 Subalgebras

Lie Subalgebra of L is a vector space $K \subseteq L$ such that [x,y] $\in K$ for all x,y $\in K$.

Ideal of a lie algebra L is a subspace I of L such that [x,y]$\in I$ for all x $\in I$ and y $\in L$

An important example of an ideal is the centre of L defined by

$Z(L) \equiv \{x \in L : [x, y] = 0 \text{ for all } y \in L\}$

We also have a notion of a homomorphism. A *Lie Homomorphism* is defined as the map $\phi : L_1 \to L_2$ such that

$$\phi([x, y]) = [\phi(x), \phi(y)] \tag{3.2}$$

An important example of Lie homomorphism will be the *adjoint homomorphism* defined as such

$$ad : L \to gl(L) \text{ by } (adx)(y) \equiv [x, y] \tag{3.3}$$

If $\phi : L_1 \to L_2$ is a homomorphism, then $ker(\phi)$ is an ideal of L_1 and the image of ϕ, $im(\phi)$, is a lie subalgebra of L_2 because if $x, y \in L_1$ and $x, y \in ker(\phi)$ then $\phi([x, y]) = [\phi(x), \phi(y)] = 0$. $\phi(x) = \phi(y) = 0$.
Now let there be an element $z \in L$ then $\phi([z, y]) = [\phi(z), \phi(y)] = 0$. So $ker\phi$ is an ideal of L_1.

3.2.1 Ideals and Homomorphisms

Ideals play a similar role that normal subgroups play in group theory. In other words we can use ideals to construct other types of lie algebras in the same way that we used normal subgroups to construct quotient groups. Suppose I and J are ideals of a lie algebra L, Then the following are true:

1. $I \bigcap J$ is an ideal, since we already we that $I \bigcap J$ is a subspace of L. So we need to check that for $x \in L$ and $y \in L$, $[x, y] \in I \bigcap J$, but the result follows quickly since I and J are both independently ideals of L.

2. I + J $\equiv \{x + y : x \in I, y \in J\}$ is an ideal. Let z $\in L$, consider [z,(x+y)] = [z,x] + [z,y] = a + b with $a \in J$ and $b \in J$. So a +b $\in I + J$ making I +J an ideal.

3. Product of ideals, [I,J] : Span $\{[x, y] : x \in I, y \in J\}$ is an ideal. To prove we start with the Jacobi Identity. $[u, [x, y]] + [x, [y, u]] + [y, [u, x]] = 0 \implies [u, [x, y]] = [x, [u, y]] + [[u, x], y] x \in I, y \in J$ and $u \in L$. So $[u, y] \in J$ since J is an ideal therefore $[x, [u, y]] \in [I, J]$ by the definition, similar argument applies for $[[u, x], y]$. Now a general element,t, of [I,J] is a linear combination i.e $t = \sum c_i [x_i, y_i]$ where c_i are scalars from the field and $x_i \in I, y_i \in J$. Now pick and element $u \in L$ and consider $[u, t] = [u, \sum c_i [x_i, y_i]] = \sum c_i [u, [x_i, y_i]]$. But $[u, [x_i, y_i]] \in [I, J]$ and so the sum is . So in summary we have that $[u, t] \in [I, J]$

A special construction occurs if we take I=J=L. We write [L,L] = L' and call it the derived algebra of L.

3.2.2 Quotient Algebras

We may consider the cosets of the ideal defined as follows $z + I = \{z + x : x \in I\}$ for $z \in L$ so the quotient vector space is $L/I = \{z + I : z \in L\}$. The lie bracket on L/I may be defined by $[w + I, z + J] := [w, z] + I$ for $w, z \in L$. The lie bracket above is bilinear i.e $[w + I, (u + I) + (v + I)] = [w + I, u + I] + [w + I, v + I]$ and $[w + I, (u + I) + (v + I)] = [w, u] + I + [w, u] + I$. Same argument applies for the left side. We also have that $[w + I, w + I] = [w, w] + I = I$. Jacobi Identity is also is satisfied i.e :

$$
\begin{aligned}
[[u + I, [v + I, w + I]]] + [(v + I), [(w + I), (u + I)]] + [(w + I), [(u + I), (v + I)]] = \\
[(u + I), [v, w] + I] + [(v + I), [w, u] + I] + [(w + I), [u, v] + I] = \\
[u, [v, w]] + I + [v, [w, u]] + I + [w, [u, v]] + I = \\
[u, [v, w]] + [v, [w, u]] + [w, [u, v]] + I = I
\end{aligned}
$$

We can motivate why we consider the cosets of ideals rather than any old sub-algebra. The main motivation is that we want the lie bracket defined on the quotient algebra to be well-defined in other words we do not want the answer to depend on the representative of the coset we chose. Consider the following: We have defined $[x+I, y+I]$ to be $[x,y]+I$. But suppose we choose another representative for each coset and perform the bracket i.e $[(x+j)+I, (y+k)+I]$ for $j,k \in I$, we still want the answer to be $[x,y]+I$ since x and $x+j, y$ and $y+k$ are in the same cosets respectively. We now do the computation

$$\begin{aligned}[(x+j)+I, (y+k)+I] &= [(x+j),(y+k)]+I \\ &= [x,y]+[j,y]+[x,k]+[j,k]+I\end{aligned}$$

It should be clear that the last commutator is in I and just the first commutator is the result we want. It then follows that $[y,j],[x,k] \in I$ but this makes I an ideal.

We now see the analogue of the three isomorphism theorems we saw for groups in the context of lie algebras.

3.2.3 Isomorphism Theorems

1. Let $\phi : L_1 \to L_2$ be a homomorphism of Lie algebras. Then $ker\phi$ is an ideal of L_1 and $im\phi$ is a subalgebra of L_2 and $L_1/ker\phi \simeq im\phi$

2. If I and J are ideals of a lie algebra, then $(I+J)/J \simeq I/(I \bigcap J)$

3. Suppose that I and J are ideals of a lie algebra L such that $I \subseteq J$. Then J/I is an ideal of L/I and $(L/I)/(J/I) \simeq L/J$

3.2.4 Solvable Lie Algebras

We take an ideal of I of a lie algebra L and ask when the factor or quotient algebra L/I is abelian.

Lemma 3.2.1 *Suppose I is an ideal of L. Then L/I is abelian iff I contains the derived algebra* $L^{'}$

Proof The algebra L/I is abelian iff for all x,y $\in L$ we have [x+I, y+I]= [x,y] + I = I or $\forall x, y \in L$ we have $[x, y] \in I$. Since I is a subspace of L, this holds iff the space spanned by the brackets [x,y] is contained in I, $L^{'} \subseteq I$.

This argument says that the derived algebra $L^{'}$ is the smallest ideal of L that has an abelian quotient. By the same argument then derived algebra $L^{'}$ has a smallest ideal whose quotient is abelian. We denote this smaller derived algebra as $L^{(2)}$. The argument goes on iteratively. We can define the derived series of L to be the series with the terms $L^{(1)} = L^{'}$ and $L^{(k)} = [L^{(k-1)}, L^{(k-1)}]$. Then $L \supseteq L^{(1)} \supseteq L^{(2)} \supseteq L^{(3)} \ldots$

Definition The lie algebra L is said to be solvable if for some m ≥ 1 we have $L^{(m)} = 0$

As a consequence the Heisenberg algebra is solvable but $\mathbf{sl}(2, C)$ is not solvable.

If L is solvable, then the derived series of L provides us with an "approximation" of L by a finite series of ideals with abelian quotients. This works the other way round.

Lemma 3.2.2 *If L is a lie algebra with ideals* $L = I_0 \supseteq I_1 \supseteq I_2 \supseteq I_3 \ldots I_{m-1} \supseteq I_m = 0$ *such that* I_{k-1}/I_k *is abelian for 1* $1 \leq k \leq m$*, then L is solvable.*

Proof Key idea: Show that $L^{(k)}$ is contained in I_k for k between 1 and m. Putting k=m will then give $L^{(m)} = 0$. L/I_1 is abelian, we know that $L^{'} \subseteq I_1$ For the inductive step, we suppose $L^{(k-1)} \subseteq I_{k-1}$ where $k \geq 2$. By construction I_{k-1}/I_k is abelian, this means that $[I_{k-1}, I_{k-1}]$ must be contained in I_k(By Lemma 3.5.1). But $L^{(k-1)}$ is contained in I_{k-1} by our inductive hypothesis so we deduce that $L^{(k)} = [L^{(k-1)}, L^{(k-1)}] \subseteq [I_{k-1}, I_{k-1}]$ and hence $L^{(k)} \subseteq I_k$. QED

This proves that if $L^{(k)}$ is non-zero then I_k is also non-zero. Hence the derived series may be thought of as the fastest descending series whose successive quotients are abelian.

Lie algebra homomorphism are linear maps that preserve Lie Brackets, and so one would expect that they preserve the derived series. Suppose that $\phi : L_1 \to L_2$ is a surjective homomorphism of a lie algebras show that $\phi(L_1^{(k)}) = (L_2)^{(k)}$, we proceed by induction on k. We already have that $\phi(L_1) = L_2$.

So $\phi([L_1, L_1]) = [\phi(L_1), \phi(L_1)]$ by the property of homomorphisms and by assumption we now have that $[\phi(L_1), \phi(L_1)] = [L_2, L_2]$ $\therefore$ we have that $\phi : L_1' \to L_2'$. The inductive step is to assume that $\phi(L_1^{(k-1)}) = (L_2^{(k-1)})$ and consider the derived algebra of $L_1^{(k-1)}$. $\phi([L_1^{(k-1)}, L_1^{(k-1)}]) = [\phi(L_1^{(k-1)}), \phi(L_1^{(k-1)})]$. This is equal to $[L_2^{(k-1)}, L_2^{(k-1)}]$ by the inductive step. Our desired result therefore follows since $\phi([L_1^{(k-1)}, L_1^{(k-1)}) = \phi(L_1^k) = [L_2^{(k-1)}, L_2^{(k-1)}] = (L_2)^k$. QED.

It turns out that if L is a lie algebra then

1. if L solvable, then every subalgebra and every homomorphic image of L is solvable
2. Suppose that L has an ideal I such that I and L/I are solvable. Then L is solvable.
3. If I and J are solvable ideals of L then I+J is a solvable ideal of L.

Theorem 3.2.3 *Let L be a finite dimensional Lie algebra. There is a unique solvable ideal of L containing every solvable ideal of L.*

Proof Let R be a solvable ideal of the largest possible dimension. We know that if I and J are solvable ideals then I+J is solvable. Let I be a solvable ideal. We have that R+I as solvable this means that R $\subseteq R + I$ and therefore dim(R) $\leq$ dim(R+I). But we chose R to have the largest possible dimension and therefore dim(R)= dim(R+I) and hence R = R+I so I $\subseteq R$. The largest solvable ideal is called the *radical of L* and is denoted as *rad* L QED.

The notion of a radical of L suggests the following definition.

Definition A non-zero finite dimensional Lie Algebra L is said to be *semisimple* if it has no nonzero solvable ideals for equivalently rad L =0.

An example is $\mathbf{sl}(2, C)$ has non-trivial ideals so it is semisimple. If L is a lie algebra, then the factor algebra L/(rad L) is semisimple. This makes sense since rad +L is the unique solvable ideal that contains all other solvable ideals. Therefore if we mod out by that we are left with a factor algebra that has no non-zero solvable ideals.

3.3 Some Representation Theory

Purpose : Examine the ways in which an abstract Lie Algebra can be viewed concretely as a subalgebra of the endomorphism algebra of a finite dimensional vector space.

Definition Let L be a lie algebra over a field F. A representation of L is a lie algebra homomorphism $\phi : L \to \mathbf{gl}(V)$ where V is a finite dimensional vector space over F.

Suppose $\phi : L \to \mathbf{gl}(V)$ is a representation. The image of ϕ is a lie subalgbera of $\mathbf{gl}(V)$ and the kernel of ϕ is an ideal of L.

Thus in general we loose some information when we work with ϕ. But when the kernel is zero then the map is one to one and information is not lost. The representation is then said to be **faithful**.

Examples:

1. $ad : L \to \mathbf{gl}(L) : (ad_x)y = [x, y]$. This provides a representation of L with V=L. This is known as the adjoint representation. The kernel of the adjoint representation is Z(L). Hence the adjoint representation is faithful when the center of L is zero.

Consider the adjoint representation of $\mathbf{sl}(2, C)$. Show that with respect to basis (h,e,f) ad_h is the matrix $\left(\begin{smallmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -2 \end{smallmatrix}\right)$

The basis for $\mathbf{sl}(2, C)$ is $e = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right), f = \left(\begin{smallmatrix} 0 & 0 \\ 1 & 0 \end{smallmatrix}\right), h = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$ and note the following commutation relations $[h, f] = -f, [h, e] = 2e, [e, f] = h$. From these the result follows.

2. Suppose that L is a lie subalgebra of $\mathbf{gl}(V)$. The inclusion map $L \to \mathbf{gl}(V)$ is trivially a lie algebra homomorphism. The corresponding representation is knows as the **natural representation**

3. Every Lie algebra has a trivial representation. To define this representation, take V=F and define $\phi = 0$ for all $x \in L$

3.3.1 Modules for Lie Algebras

Suppose that L is a lie algebra over a field F. A *lie module* for L, or alternatively an L-module is a finite dimensional F-vector space V together with a map

$$L \times V \to V \qquad (x, v) \longmapsto x.v \tag{3.4}$$

satisfying the conditions

1. $(\lambda x + \mu y) = \lambda(x.v) + \mu(y.v)$
2. $x.(\lambda u + \mu w) = \lambda(x.u) + \mu(x.w)$
3. $[x, y].v = x(y.v) - y.(x.v)$

$\forall x, y \in L, v, w \in V$ and $\lambda, \mu \in F$

The first and second properties imply that the map $(x, v) \longmapsto x.v$ is a bilinear map and the second implies that the map $v \longmapsto (x.v)$ is a linear endomorphism of V, so elements of L act on V by linear maps.

3.3.2 Submodules and Factor Modules

Suppose that V is a lie module for the Lie Algebra L. A submodule of V is a subspace of W of V which is invariant under the action of L. i.e for each $x \in L, w \in W$, we have $x.w \in W$.

In the language of representation, submodules are known as sub-representations.

Examples:

1. Let L be a lie algebra, we may make L into an L-module via the adjoint representation. The submodules of L are exactly the ideals of L.

Proof $x \in L, y \in L$, our map $L \times V \to V$ will be defined here as $L \times L \to L$ and will be the lie bracket. L is playing both the role of a lie algebra and a vector space so that $(x, y) \longmapsto [x, y] \in L$ by $(ad_x)y$. The subspaces that are invariant under this map are the ideals

2. Let $L = b(n, F)$ be the lie algebra of $n \times n$ upper triangular matrices and let V be the natural L-module, so by definition $V = F^n$ and the action of L is given by applying matrices to column vectors. Let $e_1, e_2 \dots e_n$ be the standard basis for F^n. For $1 \leq r \leq n$. Let $W_r = span\{e_1, \dots e_r\}$, W_r is a submodule of V.

3. $L \to$ complex solvable Lie algebra. Suppose $\phi : L \to \mathbf{gl}(V)$ is a representation of L. As ϕ is a homomorphism, $im\phi$ is a solvable sub-algebra of $\mathbf{gl}(V) \implies$ V has a one dimensional sub-representation.

Suppose that W is a submodule of the L-module V. We can give the quotient vector space V/W the structure of an L-module by setting

$$x.(v + W) := (x.v) + W \qquad \text{for } x \in L \text{ and } v \in V$$

We call this module the quotient or factor module V/W. For an example of a quotient module, suppose I is an ideal of the Lie Algebra L. The factor module L/I becomes an L-module via

$$x.(y + I) := (ad_x)y + I = [x, y] + I$$

Looking at it differently, L/I is a lie algbera with Lie bracket given by $[x+I, y+I] = [x,y]+I$. So regarded as a L/I-module, the factor module L/I is the adjoint representation of L/I on itself.

3.3.2.1 L-module Homomorphisms

Let L be a lie algebra and let V, W be L-modules. An L-module homomorphism from $V \to W$ is a linear map $\theta : V \to W$ such that $\theta(x.v) = x.\theta(v), \forall v \in V, x \in L$.
Let $\phi_V : L \to \mathbf{gl}(V)$ and $\phi_W : L \to \mathbf{gl}(W)$ be representation corresponding to V and W. In the language of representation theory, the condition becomes $\theta \circ \phi_V = \phi_W \circ \theta$. Because we have vector spaces and homomorphisms lying around there also analogues for the three isomorphism theorems for lie modules.

Concretely, we can give an example. Suppose we have a one dimensional abelian lie algebra L, spanned by x. We can find a representation for it $f \in \mathbf{gl}(V)$. But suppose we find another representation of it $g \in \mathbf{gl}(W)$ and we further suppose that there is a homomorphism θ from V to W. If this homomorphism turns out to be an isomorphism then we know from the way lie module homomorphisms work that f and g will be equivalent iff $\theta f = g\theta$. An explicit example is diagonalizing a matrix.

3.3.2.2 Schur's Lemma

A lie module V is said to be irreducible, or simple, it it is non-zero and t has no sub-modules other than $\{0\}$ and V. The L-module V is completely reducible if it can be written as a direct sum of irreducible L-modules; i,e $V = S_1 \bigoplus S_2 \ldots \bigoplus S_k$ where each S_i is an irreducible L-module. Suppose that S and J are irreducible Lie modules and that $\theta : S \to T$ is a non-zero sub-module homomorphism. Then $im\theta = J$. Similarly $\ker\theta$ is a proper sub-module of S, so $\ker\theta = 0$.It follows that θ is an isomorphism from S to J, so there are no

non-zero homomorphisms between non-isomorphic irreducible modules. We now consider a homomorphism from a lie-module to itself.

Theorem 3.3.1 *(Schur's Lemma:) Let L be a complex Lie algebra and let S be a finite-dimensional irreducible L-module. A map $\theta : S \to S$ is an L-module homomorphism iff θ is a scalar multiple of the identity transformation i.e $\theta = \lambda I$ for some $\lambda \in \mathbb{C}$*

Proof "If" direction is simple. The "only if" direction is non-trivial. Suppose $\theta : S \to S$ is a L-module homomorphism, then θ is a linear map of a complex vector space, and so it must have an eigenvalue, say λ. Now $\theta - \lambda I$ is also L-module homomorphism. The kernel of this map contains the $\lambda-$ eigenvector for θ, and so it is a non-zero submodule of S. As S is irreducible, S = $\ker(\theta - \lambda I)$; that is $\theta = \lambda I$

3.4 The Cartan Decomposition and KAK decomposition

3.4.1 Cartan Decomposition

This section summarizes material from (71). We state the following main theorem for this section without proof. We have the following notation: $ad_{\mathcal{K}}(\mathcal{P}) = [\mathcal{K}, \mathcal{P}]$ and $Ad_g(\mathcal{K}) = g\mathcal{K}g^{-1}$

Theorem 3.4.1 *$\mathcal{G}$ has a direct-sum decomposition of the form $\mathcal{G} = \mathcal{K} \oplus \mathcal{P}$ satisfying:*

1. *$\mathcal{K}$ is a subalgebra, $ad_{\mathcal{K}}(\mathcal{P}) \subset \mathcal{P}$ and $[\mathcal{P}, \mathcal{P}] \subset \mathcal{K}$*
2. *The killing form B of $\mathcal{G}$ restricted to $\mathcal{K}$ is negative definite, while the restriction to $\mathcal{P}$ is positive definite.*
3. *If $\mathcal{K}'$ is any other sub algebra of $\mathcal{G}$such that the killing form restricted to $\mathcal{K}'$ then there is some element $g \in G$ such that $Ad_g(\mathcal{K}') \subset K$*

So the Cartan decomposition is unique up to inner-automorphisms. Define a linear map $s : \mathcal{G} \to \mathcal{G}$ by condition

$$s(X) = \begin{cases} X \text{ for } X \in \mathcal{K} \\ -X \text{ for } \in \mathcal{P} \end{cases} \tag{3.5}$$

Thinking of $X \in \mathcal{K}$ or $X \in \mathcal{P}$ as eigenvectors of s then clearly we must have $\mathcal{G} = \mathcal{K} \oplus \mathcal{P}$ with $\mathcal{K}$ being the eigenspace with eigenvalue 1 and $\mathcal{P}$ being eigenspace with eigenvalue -1 hence

$$s([\mathcal{K}, \mathcal{K}]) = [s(\mathcal{K}), s(\mathcal{K})]) = [\mathcal{K}, \mathcal{K}] \tag{3.6}$$

$$s([\mathcal{P}, \mathcal{P}]) = [s(\mathcal{P}), s(\mathcal{P})] = [-\mathcal{P}, -\mathcal{P}] = [\mathcal{P}, \mathcal{P}] \tag{3.7}$$

$$s([\mathcal{K}, \mathcal{P}]) = [s(\mathcal{K}), s(\mathcal{P})] = [\mathcal{K}, -\mathcal{P}] = -[\mathcal{K}, \mathcal{P}] \tag{3.8}$$

Thus, we clearly see that $[\mathcal{K}, \mathcal{K}] \subset \mathcal{K}$, $[\mathcal{P}, \mathcal{P}] \subset \mathcal{K}$ and $[\mathcal{K}, \mathcal{P}] \subset \mathcal{P}$ since $[\mathcal{K}, \mathcal{K}]$, $[\mathcal{P}, \mathcal{P}]$ are eigenvectors with eigenvalue 1 and $[\mathcal{K}, \mathcal{P}]$ is an eigenvector with eigenvalue -1. This means the existence of this automorphism s of $\mathcal{G}$ is equivalent to the conditions in the theorem.

Let G be a semi-simple Lie Group, K a compact symmetric subgroup i.e $\mathcal{G} = \mathcal{K} \oplus \mathcal{P}$ with $Ad\mathcal{K}(\mathcal{P}) \subset \mathcal{P}$ and $[\mathcal{P}, \mathcal{P}] \subset K$

Definition A Cartan sub algebera of the space $\frac{\mathcal{G}}{\mathcal{K}}$ is a maximal abelian sub-algebra of $\mathcal{P}$ which we shall denote as $\mathcal{A}$

Definition A regular element $X \in \mathcal{P}$ if the centralizer of X in P is a cartan sub algebra so X belongs to only one cartan sub algebra.

It's a theorem which we shall not prove here that if $\mathcal{A}$ is a maximal abelian sub algebra of $\mathcal{P}$ then $\mathcal{A}$ contains a regular element.

Theorem 3.4.2 *Let A be a fixed cartan sub algebra of $\mathcal{P}$ and let $\mathcal{A}'$ be any maximal abelian sub algebra of $\mathcal{P}$. Then there exists an element $k \in K$ such that $Ad_k(\mathcal{A}') \subset \mathcal{A}$*

Proof Let B be the killing form of $\mathcal{G}$ and let X, X' be regular elements of $\mathcal{A}$ and $\mathcal{A}'$. We consider the following function $k \longmapsto B(Ad_k(X'), X)$ and let k be an element of K for which it takes on a minimum value. Since K is compact this value must exist. Let $Z \in \mathcal{K}$ then $t \longmapsto B(Ad_{(exp(tZ)}Ad_k(X'), X)$ has a minimum value for $t = 0$

$$0 = \frac{d}{dt}|_{t=0} B(Ad_{exp(tZ)}Ad_k(X'), X)$$
$$0 = B([Z, Ad_k(X')], X)$$
$$0 = -B(Z, [Ad_k(X'), X])$$

Now, $[Ad_k(X'), X] \in \mathcal{K}$ but it's orthogonal to every element of $\mathcal{K}$ with respect to the killing form. We know the killing form is negative definite so $[Ad_k(X'), X] = 0$ since X is a regular then $Ad_k(X') \in \mathcal{A}$. $Ad_k(X')$ is also a regular element of $Ad_k(\mathcal{A})$ so $Ad_k(\mathcal{A}') = \mathcal{A}$

3.4.2 KAK DECOMPOSITION

Let G be a connected lie group with a finite center with the lie algebra $\mathcal{G}$. Let $\mathcal{K}$ be the connected subgroup with lie algebra $\mathcal{K}$. Suppose P is the image of $\mathcal{P}$ in $\mathcal{P}$ in G under the exponential map. Then it's a theorem that we shall not prove since it requires long excursion into Riemannian Differential Geometry that $G = KP$. Let $\tau = exp(X)$ for $X \in \mathcal{P}$ $s(\tau) = s(exp(X)) = exp(s(X)) = exp(-X) = \tau^{-1}$ We state another theorem without proof

Theorem 3.4.3 *Consider the set of all $g \in G$ such that $s(g) = g^{-1}$. P consists of the set of all such elements that are in the identity connected component*

Theorem 3.4.4 *If $\tau \in P$ $g \in G$, then $g\tau s(g^{-1})$ belongs to P*

Proof $s(g\tau s(g^{-1})) = s(g)s(\tau)s^2(g^{-1}) = s(g)\tau^{-1}g^{-1} = (g\tau s(g^{-1}))^{-1}$ Next we need to check that it's connected to identity component. Now G is connectedm so g can be connected to the identity by a curve $t \to g(t)$. By definition of P, τ is joined to the identity by curve $t \to \tau(t)$ so $t \to g(t)\tau(t)s(g(t)^{-1})$ can be used to join $g\tau s(g^{-1})$ to the identity. $\square$

The isotropy subgroup of G at the identity has characteristic $ges(g^{-1}) = e \implies g = s(g)$ where e is the identity element. This group contains K but may be larger; call it $K(s)$. The following theorem is stated without proof.

Theorem 3.4.5 *The orbit of G acting on P at e fills all of P and P is isomorphic to $\frac{G}{K(s)}$.*

We have $G = PK$. Let $\mathcal{A}$ be a cartan sub algebra of $\mathcal{P}$. Let $A = exp(\mathcal{A})$, it can be shown that A consists of elements that commute with $\tau = exp(X)$ can be joined to the identity by a curve consisting of elements in P that commute with τ. Now every element in $\mathcal{P}$ can be written $Ad_k(X)$ for $x \in \mathcal{A}$, $k \in K$ so at the lie group level we have that every element has form $Ad_k(\tau) = k\tau k^{-1} = k\tau s(k^{-1})$ so $\mathcal{P} = Ad_K(A)$ so we have $\tau k = g\ \tau \in P$, $k \in K$ and $g \in G$ but $\tau = k_1 a k_!^{-1}$ for $a \in A$, $k_1 \in K$ so $g = k_1 a k_1^{-1} k$. We have thus arrived at the KAK decomposition, namely that $G = KAK$

3.5 Pre-threshold Quantum Computation (Near Term Quantum Computation)

There is currently great interest in the development of special-purpose quantum computing devices and methodologies that do not require full error correction and which are practical now. For example, D-Wave Systems produces commercial quantum annealers based on superconducting circuits that solve an important class of binary optimization problems (15). However it is not known whether the D-Wave annealers can outperform conventional classical supercomputers (16; 72). An optical approach (73) that solves an arguably less important problem—sampling from the distribution of bosons scattered by a unitary network—but which is likely capable of quantum speedup has also been investigated (74–76). An approach called the single-excitation-subspace (SES) method, also based on supercondonducting circuits, has been proposed (77). Here computations are performed in the n-dimensional SES of a complete graph of n qubits. We call these examples *prethreshold*, referring to the thresh-

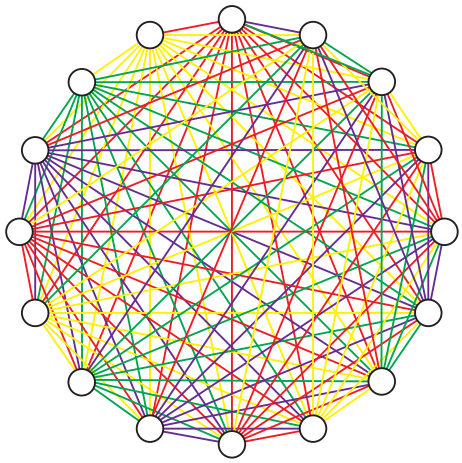


Figure 3.1: Complete graph with $n=16$. The vertices (open circles) are qubits and the edges (colored lines) are tunable couplers.

old theorem of fault-tolerant quantum computation, because they do not require exceeding fidelity and qubit-number thresholds before being applicable.

A quantum computer chip implementing the SES method consists of a fully connected array of superconducting qubits with tunable frequencies and tunable pairwise $\sigma^x \otimes \sigma^x$ couplings; an abstract representation is given in Fig. 3.1. It works by operating in a subspace of the full 2^n-dimensional Hilbert space where the Hamiltonian can be directly programmed. This programmability eliminates the need to decompose operations into elementary one- and two-qubit gates, enabling larger computations to be performed within the available coherence time. The price for this high degree of controllability is that the approach is not scalable. However, a technically unscalable quantum computer is still useful for prethreshold quantum computation and might even be able to achieve speedup relative to a classical supercomputer for certain tasks. The SES approach trades physical qubits and high connectivity for, in effect, longer coherence. This is a sensible trade for quantum computing architectures such as superconducting circuits, whose largest prethreshold problem sizes are limited by coherence time, not by the difficulty of introducing additional qubits. A realistic

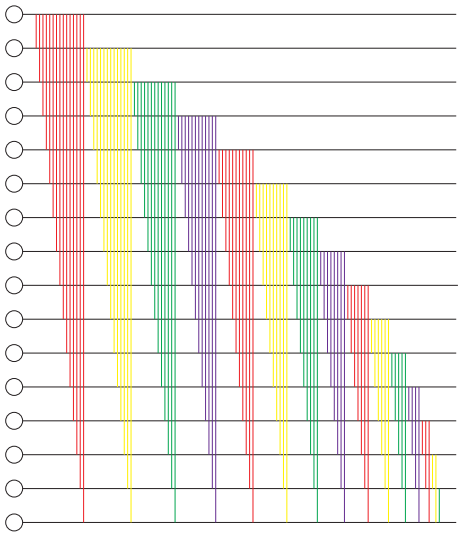


Figure 3.2: Possible layout for the 16-qubit chip.

chip layout that provides space for the coupler circuits and avoids the crossovers of Fig. 3.1 is shown in Fig. 3.2.

3.6 SES Method

A significant restriction of the SES method presented in Ref. (77) is that the Hamiltonian programmed into the hardware is real and symmetric, whereas the most general Hamiltonian is complex Hermitian. If a target operation has the form e^{-iA}, where A is a known real symmetric generator matrix, then the unitary can be implemented in one step. This is the case when the unitary is *symmetric* ($U = U^\top$) and is reviewed in Sec. 3.6.1.1. In that section we also provide an improved procedure for constructing the time-optimal SES Hamiltonian $\mathcal{H}$ corresponding to a given generator A.

However, a general element of the unitary group $\mathrm{U}(n)$ has the form e^{-iM} with M complex Hermitian. This is the *nonsymmetric* unitary case ($U \neq U^\top$) discussed in Sec. 3.6.1.2. We show there that any nonsymmetric element $U \in \mathrm{U}(n)$ can be implemented in three steps, for any n.

3.6.1 SES IMPLEMENTATION OF UNITARY OPERATORS

3.6.1.1 Single-step implementation of symmetric unitaries

The basic single-step operation in SES quantum computing is the implementation of symmetric unitaries of the form $U = e^{-iA}$, with A real and symmetric (77). Therefore, a standard task in SES algorithm design and implementation is the construction of an optimal protocol—an SES Hamiltonian $\mathcal{H}$ and evolution time $t_{\rm qc}$—to implement that unitary. We assume here that the generator matrix A is *known*; if it is not then the classical overhead for obtaining A from U must be included in the quantum runtime. (We also note that the generator $A = i \log U$ is not unique.) The optimal protocol for implementing a symmetric unitary depends on the functionality assumed of the chip, especially of the tunable coupler circuits. Here we assume that the experimentally controlled SES Hamiltonian can be written, apart from an additive constant, as

$$\mathcal{H} = g_{\rm max} K \quad \text{with} \quad -1 \leq K_{ii'} \leq 1, \tag{3.9}$$

which we call the *standard form*. In this case we are assuming that the couplings can be tuned continuously between $-g_{\rm max}$ and $g_{\rm max}$, and that the qubit frequencies can be varied within a window of width $2g_{\rm max}$ about some parking frequency. Because we are free to change the overall phase of an SES state, we write the symmetric unitary as

$$U = e^{-i(A-cI)} e^{-ic}, \tag{3.10}$$

where I is the $n \times n$ identity matrix, and then ignore the global phase e^{-ic}. The value of c is chosen to minimize the evolution time $t_{\rm qc}$, which is proportional to the angle

$$\theta_A \equiv \max_{ii'} |A_{ii'} - c\delta_{ii'}|. \tag{3.11}$$

The K matrix in (3.9) is then given by

$$K = \frac{A - cI}{\theta_A}, \tag{3.12}$$

and the evolution time is

$$t_{\rm qc} = \frac{\hbar\theta_A}{g_{\max}}. \tag{3.13}$$

Note that θ_A is not bounded by 2π and can become arbitrarily large. The global phase angle that minimizes θ_A is

$$c = \frac{\min_i A_{ii} + \max_i A_{ii}}{2}, \tag{3.14}$$

which is proved below. Although we have assumed that the SES Hamiltonian $\mathcal{H} = g_{\max}K$ is abruptly switched on for a time $t_{\rm qc}$ before being abruptly switched off—which is the fastest protocol—any SES Hamiltonian of the form $\mathcal{H} = g(t)K$ such that $\int (g/\hbar)\, dt = \theta_A$ may be used instead.

To minimize (3.11) over c we consider two cases: In the first case $\max_{ii'} |A_{ii'}|$ occurs for an *off-diagonal* element of A, in which case the minimum value of θ_A is independent of c (because c only affects the diagonal elements of the shifted matrix $A - cI$). Therefore we only need to consider the second case where $\max_{ii'} |A_{ii'}|$ occurs for a *diagonal* element. The diagonal elements consist of points

$$x \in \left\{A_{11}, A_{22}, \cdots, A_{nn}\right\} \tag{3.15}$$

on the real number line, bounded between $\min_i A_{ii}$ and $\max_i A_{ii}$. Placing c at the midpoint of the smallest region containing all the points in (3.15) minimizes the largest distance $|A_{ii} - c|$.

3.6.1.2 Three-step implementation of nonsymmetric unitaries: ABA decomposition

Our protocol relies on the matrix decomposition

$$U = O_1 e^{-iD} O_2^\top, \tag{3.16}$$

where D is a real diagonal matrix and the $O_i \in O(n)$ are real orthogonal matrices. This identity follows from the KAK decomposition of the Lie group $\mathrm{U}(n)$ (78). To obtain the O_i and D from U, we first compute

$$\chi \equiv UU^\top = O_1 e^{-2iD} O_1^\top, \tag{3.17}$$

which is both symmetric and unitary. The real and imaginary parts of χ are also separately symmetric. Then the unitarity condition

$$\big(\mathrm{Re}\,\chi - i\,\mathrm{Im}\,\chi\big)\big(\mathrm{Re}\,\chi + i\,\mathrm{Im}\,\chi\big) = I \tag{3.18}$$

shows that $\mathrm{Re}\,\chi$ and $\mathrm{Im}\,\chi$ commute and can be simultaneously diagonalized. O_1 is determined by a Schur decomposition of $\mathrm{Re}\,\chi$, which always produces a real O_1 (unlike the decomposition of χ itself). Then e^{-2iD} and O_2 are obtained from $O_1^\top \chi O_1$ and $U^\top O_1 e^{iD}$, respectively.

The three-step implementation for a nonsymmetric $U \in \mathrm{U}(n)$ follows from the identity

$$U = e^{-iA} e^{-iB} e^{iA}, \tag{3.19}$$

which we call the *ABA decomposition*. Here A and B are real symmetric $n \times n$ matrices. To derive (3.19) we express the target unitary in the spectral form $U = V e^{-i\Lambda} V^\dagger$, where V is

complex unitary and Λ is real and diagonal. Decomposing V using (3.16) we have

$$\begin{aligned} U &= O_1 e^{-iD} O_2^\top e^{-i\Lambda} O_2\, e^{iD} O_1^\top, \\ &= e^{-iO_1 D O_1^\top} (O_1 O_2^\top)\, e^{-i\Lambda}\, (O_1 O_2^\top)^\top e^{iO_1 D O_1^\top}, \end{aligned} \tag{3.20}$$

which leads to (3.19) with generators

$$A = O_1 D O_1^\top, \tag{3.21}$$

$$B = O_1 O_2^\top \Lambda O_2 O_1^\top, \tag{3.22}$$

which are both real and symmetric. The classical run-time to obtain A and B is about

$$1.4 \times n^{2.3}\, \mu\text{s} \tag{3.23}$$

on a laptop computer[1].The quantum run time to implement a non-symmetric unitary is

$$t_{\text{qc}} = \frac{\hbar(2\theta_A + \theta_B)}{g_{\max}}, \tag{3.24}$$

with θ defined in (3.11). The generator matrices A and B in (3.19) are not unique.

The ABA decomposition allows for the possibility of implementing highly complex operations in three steps. But this does not imply that an entire algorithm, compiled into a single unitary, can be implemented in constant time, because the compiled unitary might not be known a priori, and there is classical overhead (3.23) for computing A and B. More importantly, evaluating A and B for an entire algorithm would presumably be prohibitive when one is attempting to outperform classical computers. Furthermore, algorithms might include measurement steps that cannot be postponed to the end.

[1]Computations were performed running 64-bit MATLAB R2015a on an Apple Mac-book Pro with a $2.5GHz$ Intel Core $i7$ quad-core processor. Classical run-times were determined by averaging the computation times over 1000 random instances of U for n between 50 and 500. The classical run time for a 100100 matrix is about 60 ms

3.7 APPLICATIONS

3.7.1 Hamiltonian simulation

A useful application of (3.19) is to $U = e^{-iHt/\hbar}$, where H is a given complex Hamiltonian. In this case we have

$$e^{-iHt/\hbar} = e^{-iA}e^{-iB}e^{iA}, \tag{3.25}$$

with A and B given by (3.21) and (3.22), where Λ is a diagonal matrix containing $t/\hbar$ times the spectrum of H. This enables the fast simulation of any time-independent Hamiltonian with an SES chip[2].

3.7.2 SES pure state preparation in 3 steps

In some cases it is possible to compile an entire algorithm down to only three steps. As an example we give an algorithm for preparing any (normalized) pure SES state of the form

$$|\psi\rangle = \sum_{i=1}^{n} a_i \,|i), \qquad a_i = |a_i|e^{i\theta_i}, \qquad 0 \le \theta_i < 2\pi. \tag{3.26}$$

Here $|i) \equiv |0\cdots 1_i \cdots 0\rangle$ is the ith SES *basis state* of the n-qubit processor. We proceed by giving a protocol with linear depth that is practical for small n, which is then subsequently compiled down to three steps.

We start with the basis state $|1)$, which is prepared from the system ground state $|00\cdots 0\rangle$ by a microwave pulse, and then apply the standard-form SES Hamiltonian $\mathcal{H} = g_{\max}K_{\text{star}}$

[2]In principle it is possible to simulate time-dependent complex Hamiltonians as well. However, this would require decomposing the evolution into a sequence of short time steps such that $H(t)$ is approximately constant within each time step and applying the ABA decomposition at each time step. Given the classical overhead required at each step, this does not seem useful for outperforming classical computers

for a time $t_{\rm qc} = \pi\hbar/\sqrt{n}g_{\rm max}$, with

$$K_{\rm star} \equiv \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & \cdots & 0 \\ \frac{1}{2} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2} & 0 & 0 & \cdots & 0 \end{pmatrix} \tag{3.27}$$

the adjacency matrix for a star graph with qubit 1 at the center (see Sec. IIIA of (77)). This produces the *uniform* state

$$\big|{\rm unif}\big\rangle \equiv \frac{|1) + |2) + \cdots + |n)}{\sqrt{n}}, \tag{3.28}$$

apart from a phase.

If the occupation probabilities in the target state (3.26) are uniform,

$$|a_i|^2 = \frac{1}{n}, \tag{3.29}$$

we call it a *uniform weight* state and represent it by the bar graph in Fig. 3.3. In this case we would apply the diagonal Hamiltonian $\mathcal{H} = g_{\rm max}K$, where

$$K = -\begin{pmatrix} \frac{\theta_1}{2\pi} & 0 & \cdots & 0 \\ 0 & \frac{\theta_2}{2\pi} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\theta_n}{2\pi} \end{pmatrix} \tag{3.30}$$

to the uniform state $|{\rm unif}\rangle$ for a time $t_{\rm qc} = 2\pi\hbar/g_{\rm max}$, which gives the desired target.

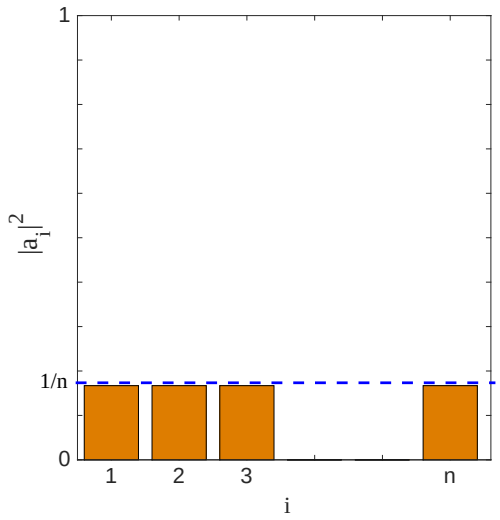


Figure 3.3: Occupation probabilities for a uniform weight state. Phases of the probability amplitudes a_i are not represented in this figure.

Typically the target is not a uniform weight state, as represented in Fig. 3.4. In this case we use the solution

$$|\text{unif}\rangle = W_{\text{diag}}\,(U_{\text{swap}}U_{\text{diag}})_M \cdots (U_{\text{swap}}U_{\text{diag}})_1 |\psi\rangle \tag{3.31}$$

to the inverse problem of constructing the uniform state $|\text{unif}\rangle$ from the target (79; 80). Each of the M steps in (3.31) consists of a pair of operations U_{diag} and U_{swap} that move weight between a pair of components. After $M = O(n)$ steps a uniform weight state is created. The final operation W_{diag} shifts the phases of the uniform weight state to that of (3.28). The first step is:

1. Find the components $i_{\min}$ and $i_{\max}$ with the smallest and largest weights, respectively (if not unique, any solution is sufficient). These satisfy

$$|a_{i_{\min}}|^2 \leq \frac{1}{n} \leq |a_{i_{\max}}|^2, \tag{3.32}$$

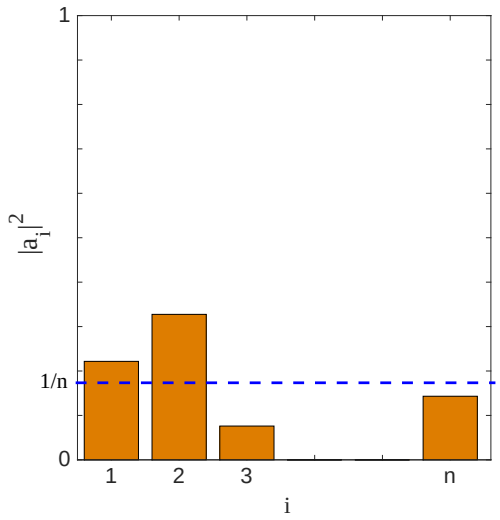


Figure 3.4: Occupation probabilities for a typical target state. In this example the components of maximum and minimum weight have indices $i_{\max}=2$ and $i_{\min}=3$.

excluding the case where both $\leq$ signs are identities (which would violate the assumption that the target is nonuniform). Therefore $|a_{i_{\min}}|^2 < |a_{i_{\max}}|^2$.

2. Perform a phase shift $U_{\text{diag}} = e^{-i\mathcal{H}t_{\text{qc}}/\hbar}$ that brings the probability amplitudes $a_{i_{\min}}$ and $a_{i_{\max}}$ to the form $a_{i_{\min}} = |a_{i_{\min}}|$ and $a_{i_{\max}} = i|a_{i_{\max}}|$, with $|a_{i_{\min}}| < |a_{i_{\max}}|$. Apply SES Hamiltonian (3.9), where K is a diagonal matrix with $K_{i_{\min},i_{\min}} = \theta_{i_{\min}}/3\pi$ and $K_{i_{\max},i_{\max}} = (\theta_{i_{\max}}/3\pi) - \frac{1}{6}$, the other elements zero, and $t_{\text{qc}} = 3\pi\hbar/g_{\max}$. This phase shift is necessary to prepare the state for the next operation.

3. Implement a partial iSWAP $U_{\text{swap}} = e^{-i\mathcal{H}t_{\text{qc}}/\hbar}$ from component $i_{\max}$ to $i_{\min}$ to bring the weight of $i_{\min}$ to the uniform value,

$$|a_{i_{\min}}|^2 \to \frac{1}{n}, \tag{3.33}$$

and leaving component $i_{\max}$ with weight

$$|a_{i_{\max}}|^2 \to |a_{i_{\min}}|^2 + |a_{i_{\max}}|^2 - \frac{1}{n}. \tag{3.34}$$

Apply SES Hamiltonian (3.9) with $K_{i_{\min},i_{\max}} = K_{i_{\max},i_{\min}} = 1$ and all other elements zero, and $t_{\rm qc} = \varphi\hbar/g_{\max}$ with φ given by

$$|a_{i_{\min}}| \cos\varphi + |a_{i_{\max}}| \sin\varphi = \sqrt{1/n}. \tag{3.35}$$

There is always a solution with $0 < \varphi < \pi/2$.

This completes the first step.

If after the first step $(U_{\rm swap}U_{\rm diag})_1|\psi\rangle$ is a uniform weight state, it can be written in the form

$$\frac{e^{i\alpha_1}|1) + e^{i\alpha_2}|2) + \cdots + e^{i\alpha_n}|n)}{\sqrt{n}}, \tag{3.36}$$

and we apply the final operation $W_{\rm diag} = e^{-i\mathcal{H}t_{\rm qc}/\hbar}$ to produce (3.28). Here we use SES Hamiltonian (3.9) with

$$K = \begin{pmatrix} \frac{\alpha_1}{2\pi} & 0 & \cdots & 0 \\ 0 & \frac{\alpha_2}{2\pi} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\alpha_n}{2\pi} \end{pmatrix} \tag{3.37}$$

and $t_{\rm qc} = 2\pi\hbar/g_{\max}$. If $(U_{\rm swap}U_{\rm diag})_1|\psi\rangle$ is not a uniform weight state, we again find the minimum and maximum weight components $i_{\min}$ and $i_{\max}$, and follow the above protocol to generate $(U_{\rm swap}U_{\rm diag})_2(U_{\rm swap}U_{\rm diag})_1|\psi\rangle$. The procedure is repeated until

$$(U_{\rm swap}U_{\rm diag})_M \cdots (U_{\rm swap}U_{\rm diag})_2(U_{\rm swap}U_{\rm diag})_1|\psi\rangle \tag{3.38}$$

is a uniform weight state, after which $W_{\rm diag}$ is applied. The number of iterations required satisfies

$$M \leq n-1. \tag{3.39}$$

This completes the solution to the inverse problem (3.31).

We now use (3.31) to obtain

$$|\psi\rangle = (U_{\rm diag}^{\dagger} U_{\rm swap}^{\dagger})_1 \cdots (U_{\rm diag}^{\dagger} U_{\rm swap}^{\dagger})_M \, W_{\rm diag}^{\dagger} |{\rm unif}\rangle, \tag{3.40}$$

which solves the general state-preparation problem in $O(n)$ steps. Hermitian conjugations are implemented by changing the signs of the K matrices given above. The protocol given in (3.40) is, by itself, practical for small n.

The complete state preparation operation can be summarized as

$$|\psi\rangle = U\,|1), \tag{3.41}$$

where

$$U \equiv (U_{\rm diag}^{\dagger} U_{\rm swap}^{\dagger})_1 \cdots (U_{\rm diag}^{\dagger} U_{\rm swap}^{\dagger})_M \, W_{\rm diag}^{\dagger} \, e^{-i\frac{\pi}{\sqrt{n}} K_{\rm star}} \tag{3.42}$$

is the *compiled* unitary of the state-preparation algorithm. The three-step state preparation protocol uses the ABA decomposition to implement (3.42). The total state preparation time, not including the $|1)$ state initialization time, is given in (3.24).

For example, suppose we wish to prepare the randomly chosen target

$$\begin{aligned} |\psi\rangle = & 0.4829\,|1) + (-0.5478 - 0.0575i)\,|2) + (0.1142 + 0.2387i)\,|3) \\ & + (0.4095 + 0.2400i)\,|4) + (-0.3215 + 0.2545i)\,|5), \end{aligned} \tag{3.43}$$

in the $n=5$ graph, where for convenience the first component has been chosen to be real. Following the state-preparation protocol leads to the compiled unitary

$$U = \begin{pmatrix} 0.4829 & 0.4499 - 0.0158i & 0.4499 - 0.0158i & 0.4478 - 0.0133i & 0.3984 + 0.0450i \\ -0.5478 - 0.0575i & 0.5855 - 0.4153i & 0.1778 - 0.0249i & -0.1305 + 0.2703i & -0.0855 + 0.2273i \\ 0.1142 + 0.2387i & 0.4664 + 0.0700i & -0.7862 - 0.2582i & 0.0910 - 0.0284i & 0.1145 - 0.0222i \\ 0.4095 + 0.2400i & 0.0841 - 0.1271i & 0.1471 - 0.1492i & -0.7941 + 0.1818i & 0.1471 - 0.1492i \\ -0.3215 + 0.2545i & 0.1071 + 0.1577i & 0.1071 + 0.1577i & 0.1071 + 0.1580i & 0.1399 - 0.8386i \end{pmatrix} \tag{3.44}$$

up to a phase factor. The first column of (3.44) is the target state. The ABA decomposition (3.19) then leads to

$$A = \begin{pmatrix} -1.1145 & 0.1981 & 0.3247 & -0.0776 & -0.1888 \\ 0.1981 & -2.6988 & 0.0219 & -0.2069 & -0.0249 \\ 0.3247 & 0.0219 & -1.9798 & -0.5623 & 0.1052 \\ -0.0776 & -0.2069 & -0.5623 & -0.5291 & -0.0747 \\ -0.1888 & -0.0249 & 0.1052 & -0.0747 & -1.7104 \end{pmatrix} \tag{3.45}$$

and

$$B = \begin{pmatrix} -3.0826 & 1.8972 & 0.3983 & 0.8753 & 0.5934 \\ 1.8972 & -3.7784 & 0.5761 & 0.3537 & 0.5581 \\ 0.3983 & 0.5761 & -3.2370 & 0.1664 & 0.2327 \\ 0.8753 & 0.3537 & 0.1664 & -2.6191 & 0.1488 \\ 0.5934 & 0.5581 & 0.2327 & 0.1488 & -4.6171 \end{pmatrix}. \tag{3.46}$$

The associated K matrices and evolution times are determined from the procedure given in Sec. 3.6.1.1:

$$\begin{aligned} K_A &= \begin{pmatrix} 0.4604 & 0.1826 & 0.2993 & -0.0715 & -0.1741 \\ 0.1826 & -1 & 0.0202 & -0.1907 & -0.0229 \\ 0.2993 & 0.0202 & -0.3373 & -0.5183 & 0.0970 \\ -0.0715 & -0.1907 & -0.5183 & 1 & -0.0689 \\ -0.1741 & -0.0229 & 0.0970 & -0.0689 & -0.0889 \end{pmatrix}, \\ \theta_A &= 1.0848, \end{aligned} \tag{3.47}$$

and

$$\begin{aligned} K_B &= \begin{pmatrix} 0.2822 & 1 & 0.2100 & 0.4614 & 0.3128 \\ 1 & -0.0845 & 0.3037 & 0.1864 & 0.2942 \\ 0.2100 & 0.3037 & 0.2009 & 0.0877 & 0.1226 \\ 0.4614 & 0.1864 & 0.0877 & 0.5266 & 0.0785 \\ 0.3128 & 0.2942 & 0.1226 & 0.0785 & -0.5266 \end{pmatrix}, \\ \theta_B &= 1.8972. \end{aligned} \tag{3.48}$$

The total state preparation time, not counting the $|1)$ state initialization, is given by (3.24). This is about 13 ns for the target state (3.43) in an SES chip with $g_{\max}/2\pi = 50\,\text{MHz}$.

Although state preparation is implemented in three steps for any n, the runtime does have a weak n-dependence, because θ_A and θ_B do. Averaged over random targets we find that

$$\overline{2\theta_A + \theta_B} \approx 4.0 \times n^{0.06}. \tag{3.49}$$

For small n, either the linear-depth protocol (3.40) or the three-step protocol based on (3.42) can be used. However for large n, only the three-step protocol is practical.

3.7.3 Computation of expectation values

Finally, we show how to compute the expectation value

$$\langle O \rangle \equiv \mathrm{Tr}(\rho O) \tag{3.50}$$

of any $n \times n$ Hermitian observable O, by implementing the protocol of Reck *et al.* (81). Here ρ is any pure or mixed SES state provided as an input to the procedure.

Standard readout of an SES processor consists of the simultaneous measurement of each qubit in the diagonal basis. The SES condition means that a single qubit will be found in the state $|1\rangle$, with the remaining $n-1$ qubits in $|0\rangle$. Let i be the qubit observed in it's excited state. The probability of observing the excitation in qubit i is $p_i = (i|\rho|i)$. Therefore, if we have access to multiple copies of ρ we can repeat the readout N times to obtain estimates of the occupation probabilities p_i with sampling errors no larger than $(2\sqrt{N})^{-1}$.

To compute $\langle O \rangle$, perform a (classical) spectral decomposition to a unitary V containing the eigenvectors of O as columns, and a real diagonal matrix D: $O = VDV^\dagger$. Then we have

$$\langle O \rangle = \mathrm{Tr}(\rho V D V^\dagger) = \mathrm{Tr}(\rho' D), \tag{3.51}$$

where

$$\rho' \equiv V^\dagger \rho V. \tag{3.52}$$

Therefore we can compute $\langle O \rangle$ by applying the unitary operator $V^\dagger$ using the ABA decomposition, measuring the resulting occupation probabilities, which we denote by $p_i^{[V^\dagger]}$ to indicate the application of $V^\dagger$, and then classically evaluating the quantity

$$\langle O \rangle = \sum_{i=1}^{n} D_{ii}\, p_i^{[V^\dagger]}. \tag{3.53}$$

Chapter 4

Measuring Entanglement Monotones on near term quantum devices

4.1 INTRODUCTION

Entanglement captures the intrinsic non-locality of quantum systems and is a critical resource for quantum speedup. The crudest measure of entanglement is whether or not it is present: If the actual density matrix is a tensor product of single-qubit density matrices, it's not entangled. Witnesses (82; 83) have been used to establish entanglement in multi-qubit systems with up to 20 qubits (84–88). When present, however, it is also interesting to quantify the degree of entanglement and its robustness to noise and decoherence (84; 85; 89–91). In this work we experimentally estimate entanglement monotones (92–94) for the GHZ

$$\frac{|0\rangle^{\otimes n} + |1\rangle^{\otimes n}}{\sqrt{2}} \tag{4.1}$$

and linear (one-dimensional) cluster state

$$\prod_{i=1}^{n-1} \mathrm{CZ}_{i,i+1} |+\rangle^{\otimes n} \tag{4.2}$$

on the IBM Rüeschlikon (ibmqx5) chip (**?**). Here $\mathrm{CZ}_{ii'}$ is the gate $\mathrm{diag}(1,1,1,-1)$ on qubits i and i', and we have mapped the physical qubits to chains of length $n = 3, 4$. Specifically,

we measure the three-qubit monotone.

$$\begin{aligned}\mathcal{E}_3 = \frac{1}{3}\Big|&\langle XYY\mathcal{C}\rangle^2 + \langle ZYY\mathcal{C}\rangle^2 - \langle IYY\mathcal{C}\rangle^2 \\ &+\langle YXY\mathcal{C}\rangle^2 + \langle YZY\mathcal{C}\rangle^2 - \langle YIY\mathcal{C}\rangle^2 \\ &+\langle YYX\mathcal{C}\rangle^2 + \langle YYZ\mathcal{C}\rangle^2 - \langle YYI\mathcal{C}\rangle^2\Big|,\end{aligned} \tag{4.3}$$

where $\mathcal{C}$ is the complex conjugation operator, and the four-qubit monotones

$$\mathcal{E}_{4a} = \langle YYYY\mathcal{C}\rangle^2 \tag{4.4}$$

and

$$\begin{aligned}\mathcal{E}_{4b} = \Big|&\langle XYXY\mathcal{C}\rangle^2 + \langle ZYZY\mathcal{C}\rangle^2 - \langle XYIY\mathcal{C}\rangle^2 \\ &+\langle ZYXY\mathcal{C}\rangle^2 + \langle ZYZY\mathcal{C}\rangle^2 - \langle ZYIY\mathcal{C}\rangle^2 \\ &-\langle IYXY\mathcal{C}\rangle^2 - \langle IYZY\mathcal{C}\rangle^2 + \langle IYIY\mathcal{C}\rangle^2\Big|.\end{aligned} \tag{4.5}$$

In this work we ignore the complex conjugation operator because our states are assumed to be real.

Entanglement monotones have several features that make them ideal for quantifying entanglement. First, they are strictly non-increasing under local (single-qubit) CPTP maps, making them less sensitive to decoherence than entanglement entropies and related measures based on reduced density matrices. Measuring their decay with time allows us to quantify and hopefully understand the environment of a noisy superconducting quantum computer and its effects on large-scale entanglement. Second, they allow for a direct comparison between different families of states and, if known, the maximum possible (95). The monotone $\mathcal{E}_3$, a symmetrized 3-tangle (96), is ideally equal to 1 (the maximum value) for both the GHZ and cluster states. (This is expected because the $n=3$ GHZ and linear cluster states are in the same entanglement class.) The monotone $\mathcal{E}_{4a}$ is the square of the 4-concurrence (97–99), and

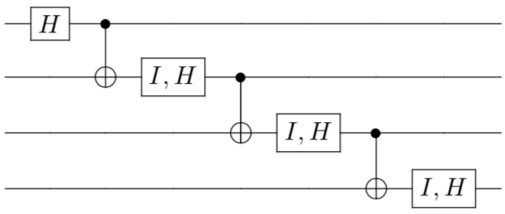


Figure 4.1: State preparation circuits for $n = 4$. Single qubit gates 2 through n are either identities (I) for GHZ or Hadamards (H) for the cluster state. The vertical gates are CNOTs.

is ideally equal to 1 for the GHZ state but vanishes for the cluster (showing that for $n = 4$ they are in different classes). Therefore we cannot use $\mathcal{E}_{4a}$ to quantify relative entanglement. We still measure it, however, because it's a simple generalization of the squared two-qubit concurrence $\langle YY\mathcal{C}\rangle^2$ (100). The monotone $\mathcal{E}_{4b}$ is ideally equal to 1 for both GHZ and cluster states, and vanishes not only on every four-qubit mixed product state

$$\rho_1 \otimes \rho_2 \otimes \rho_3 \otimes \rho_4 \tag{4.6}$$

but also on partially entangled states of the form

$$\rho_{12} \otimes \rho_{34}, \tag{4.7}$$

where ρ_{12} and ρ_{34} are entangled two-qubit states. In other words, $\mathcal{E}_{4b}$ measures genuine four-qubit entanglement. The monotones $\mathcal{E}_3$ and $\mathcal{E}_{4a}$ also measure genuine multi-qubit entanglement in this same sense.

A challenge of the monotone approach is that it relies on the anti-linear conjugation operation $\mathcal{C}$. In this work we will attempt to bypass the complex conjugation step on the basis that the ideal GHZ and cluster states are real. This is an approximation that will limit the accuracy of our technique, and may also cause the monotones to have unphysical oscillations in time, unless the imaginary part is zeroed. (We note that the approach of Di

Candia *et al.* (89) circumvents this limitation, at the expense of additional qubit and gate overhead.)

Table 4.1: State preparation error averaged over 32 independent circuit implementations, each estimated with 16 random readout-corrected Pauli expectation values. The preparation errors are written as the sample mean ± the standard error.

Error (%)	$n = 3$	$n = 4$
GHZ	11.83 ± 0.51	21.77 ± 0.22
Cluster	9.05 ± 0.34	21.17 ± 0.29

4.2 STATE PREPARATION CIRCUITS AND FIDILITIES

The state preparation circuits are shown in Fig. 4.1. The GHZ and cluster preparation circuits are the same except for $n-1$ Hadamards, and we confirm that their state preparation fidelities are very similar. This is important because it allows us to assume that the entangled states are prepared with similar fidelity. To quantify this we prepare the $n = 3, 4$ GHZ and cluster states and measure their state preparation error (fidelity loss) by Flammia-Liu fidelity estimation (101). The results are summarized in Table 4.1.

We find that the state preparation errors are quite noisy and (after readout correction) dominated by CNOT errors, which explains why the extra Hadamards in the cluster state preparation circuit do not, on average, result in a larger circuit error. In fact Table 4.1 shows that the cluster states have slightly smaller state preparation errors. Histograms showing the distribution of state preparation errors are provided in Appendix 4.6.1.

4.3 MONOTONES

The measured monotone (4.3) is shown in Fig. 4.2. Here $\mathcal{E}_3$ is measured after waiting for a time t. The delay is implemented by a sequence of 80ns identity gates. In addition to the GHZ and cluster states, we also measure the uniform state $|+\rangle^{\otimes n}$ to include a nonentangled control subject. The observed oscillations in the data appear to violate the important non-increasing property of an entanglement monotone under local noise. The monotones (4.4)

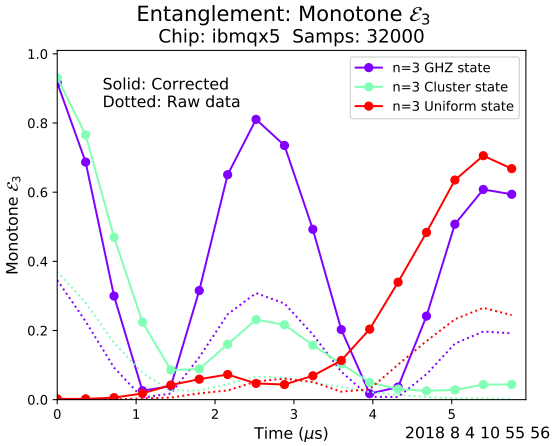


Figure 4.2: $\mathcal{E}_3$ versus time. The solid lines interpolate between readout-corrected measurements.

and (4.5), shown in Figs. 4.3 and 4.4, are also non-monotonic. To confirm that the observed oscillations are not an artifact of the particular chip or qubits used, we also measured $\mathcal{E}_3$ on the 5-qubit IBM Tenerife (ibmqx4) chip, and find oscillations with a similar frequency, as shown in Fig. 4.5.

Considering first the GHZ states, we interpret these oscillations as resulting from a nonzero relative phase ϕ between the $|0\rangle^{\otimes n}$ and $|1\rangle^{\otimes n}$ components, as defined in

$$\frac{|0\rangle^{\otimes n} + e^{i\phi}|1\rangle^{\otimes n}}{\sqrt{2}}. \tag{4.8}$$

Note that a nonzero $\phi \mod \pi$ results in an imaginary component in (4.8), invalidating our approach of neglecting the complex conjugations in the monotone definitions. To support this interpretation we measure the Pauli expectation value $\langle X^{\otimes n}\rangle$, which can be used to extract the value of ϕ. The data are shown in Fig. 4.6. In the modified GHZ state (4.8),

$$\langle X^{\otimes n}\rangle = \cos\phi. \tag{4.9}$$

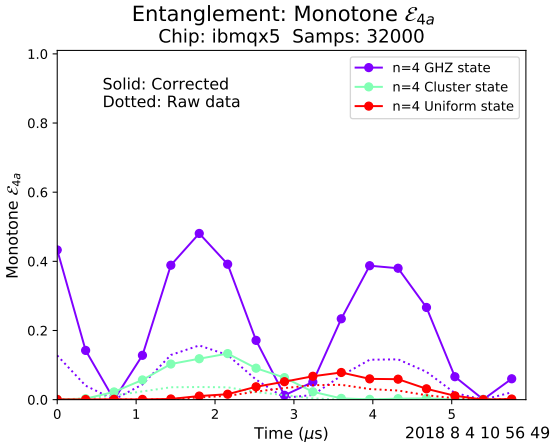


Figure 4.3: $\mathcal{E}_{4a}$ versus time.

Fitting to the data we find that

$$\phi = 2\pi f_z\, t \quad \text{with} \quad f_z \approx \begin{cases} 167\,\text{kHz} & \text{for} \quad \text{n=3} \\ 238\,\text{kHz} & \text{for} \quad \text{n=4.} \end{cases} \tag{4.10}$$

The n dependence of f_z, easily visible in Fig. 4.6, suggests that each qubit has an unexpected 55 to 60 kHz splitting in the rotating frame. Interestingly, however, we did not observe oscillations in single-qubit Ramsey scans $\left(|0\rangle \to e^{i\pi Y/4}\; I_{\text{delay}\,t}\; e^{-i\pi Y/4}|0\rangle\right)$ versus t on any of the individual qubits, but did in the concurrence $\langle YY\mathcal{C}\rangle$ of a Bell state, suggesting that the phase drift (4.10) is intrinsically a multi-qubit effect.

Similar oscillations occur for the cluster state, but for this family the oscillation amplitude is much smaller and the oscillations are sometimes only barely visible before the entanglement has vanished. We do not have a simple model for the perturbed state in this case.

4.4 Phase drift compensation

Having identified the phase drift (4.10) as the origin of the oscillations in the GHZ monotones, we attempt to cancel it by applying compensating virtual (software) Z rotations with a phase

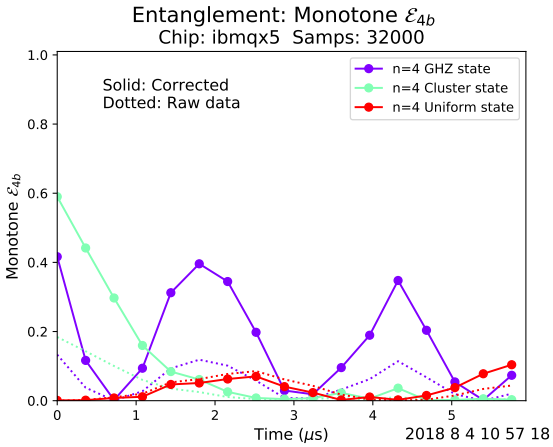


Figure 4.4: $\mathcal{E}_{4b}$ versus time.

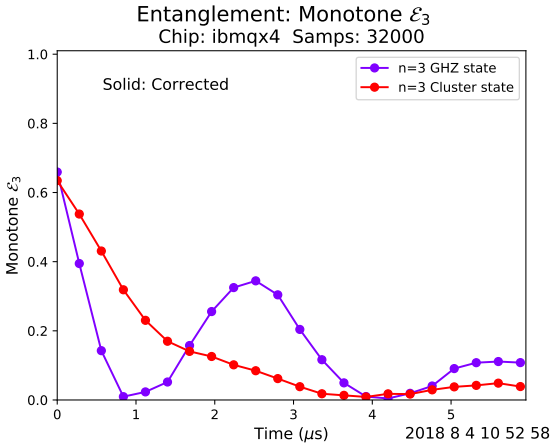


Figure 4.5: Oscillating $\mathcal{E}_3$ versus time on the IBM Tenerife (ibmqx4) chip.

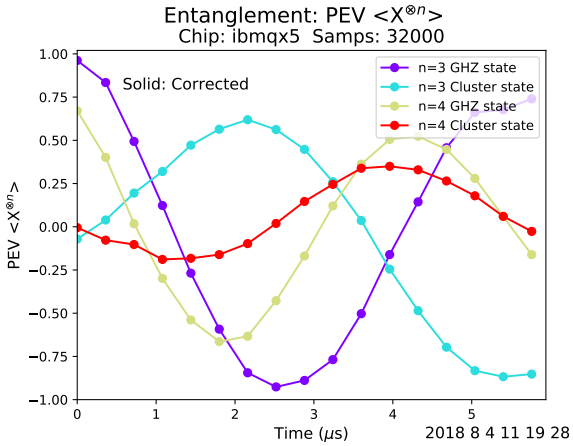


Figure 4.6: Pauli expectation values versus time.

opposite opposite to (4.10). Identical phase shifts of $-\phi/n$ are applied to each qubit. We apply this same correction to the GHZ, cluster, and uniform states. The results are shown in Figs. 4.7 and 4.8. We refer to these corrected measurement results as *phase compensated* data.

The oscillations in $\mathcal{E}_3$ and $\mathcal{E}_{4b}$ are no longer present after phase compensation; they are now properly monotonic (apart from measurement noise). And after this correction, the GHZ states appear to be significantly more robust than the cluster states, which is the most interesting conclusion from this work.

The behavior of the uniform state $|+\rangle^{\otimes n}$, which we included as a (nominally) non-entangled control, is also interesting. In Fig. 4.2, the $n=3$ uniform state (red curve) appears to spontaneously develop a high degree of entanglement. However after phase compensation, Fig. 4.7, entanglement is absent. This is consistent with our interpretation that the states in Fig. 4.2 develop imaginary components, invalidating our $\mathcal{E}_3$ measurement technique, which assumes purely real components. But the behavior of the $n=4$ uniform state is different: In Fig. 4.4, the uniform state (red curve) shows no significant entanglement, but the phase compensated data does.

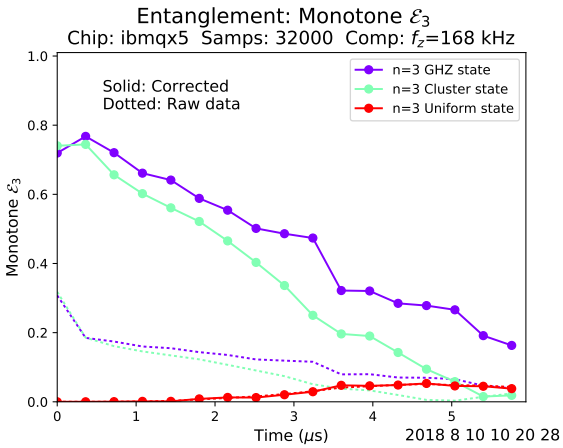


Figure 4.7: Same as Fig. 4.2, except that identical phase shifts of $-\phi/n$ are now applied to the qubits during the delay, to compensate for the phase drift. The GHZ state is clearly more robust than the cluster state at long times.

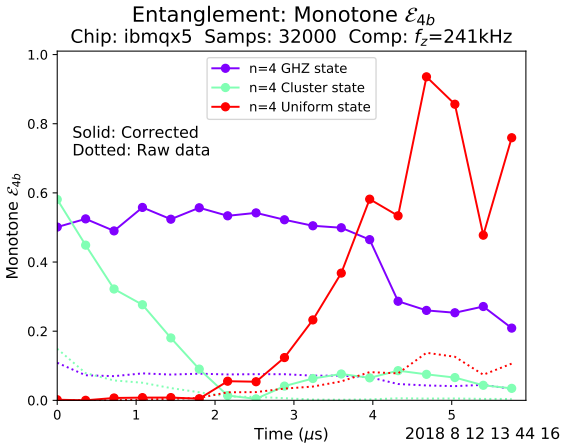


Figure 4.8: Same as Fig. 4.4, but with phase compensation. Again the GHZ state appears to be much more robust.

4.4.1 A Simple Model for Relative Phases

We present the simplest model that reproduces the gross features of these results i.e the source of these extra relative phases. The general idea is that there are a stray magnetic fields acting on each qubit such that the Hamiltonian is the following

$$H = \sum_{i=1}^{i=4} g_i \sigma_z^i \tag{4.11}$$

where g_i is the strength of the stray magnetic field. The precise strength, g_i, in (2) is a free parameter in the model and in order that it reproduces the gross features of the dynamics i.e that entanglement collapse and revival occurs roughly on the time scale of the data, the couplings have to be on the order of 0.1 MHZ. This prediction does not change even if one uses unique T_1 and T_2 for each qubit.

We can gain more evidence for the model introduced in (2) by calculating the entanglement monotones. Again the couplings were randomly chosen to be around 0.1 MHZ.

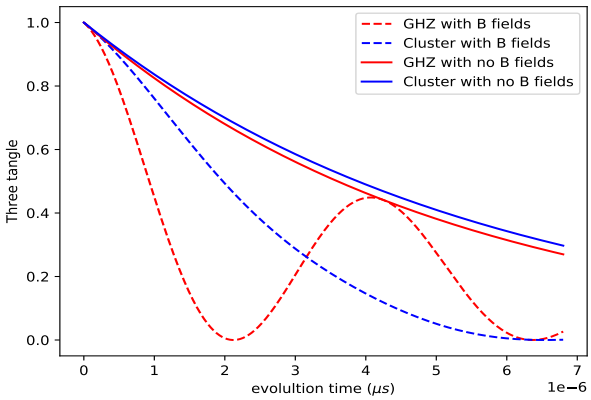


Figure 4.9: Simulation of three qubits in a uniform bath $T_1 = 40\mu s$, $T_2 = 51\mu s$

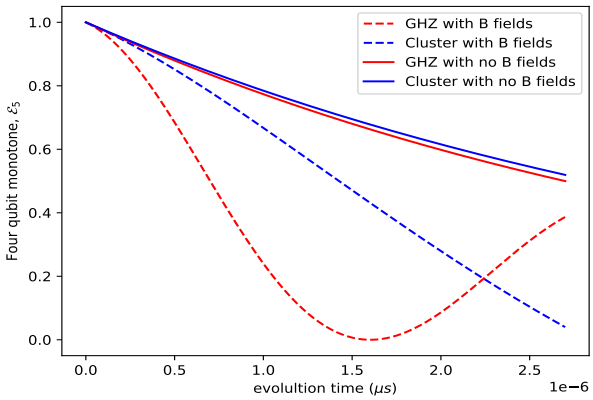
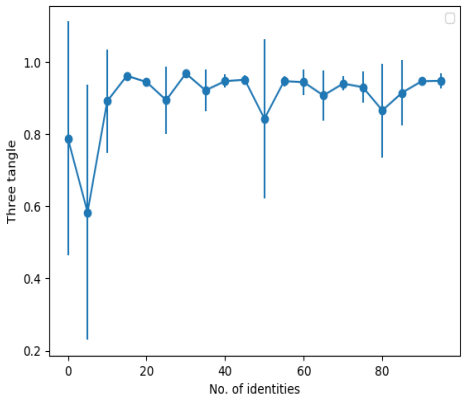


Figure 4.10: Simulation of four qubits in a uniform bath $T_1 = 40\mu s$, $T_2 = 51\mu s$

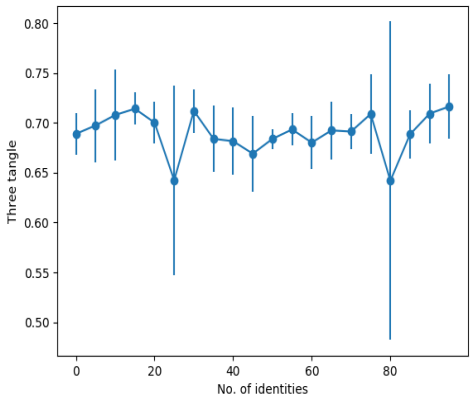
4.5 Convex Roof Extensions

To remove the assumption of real pure states, we next perform experiments where we remove this assumption. First we introduce the concept of a *roof extension*. Let Ω be a compact convex set with G being a real valued function. A point $\omega \in \Omega$ is a called a *roof point* if there is at least one extremal convex decomposition $\omega = \sum_j p_j \pi_j$, $\pi_j \in \Omega^{pure}$ such that $G(\omega) = \sum p_j G(\pi_j)$. If this takes place we call the decomposition *optimal* with respect to G or *G-optimal*. Let $\pi \to g(\pi)$ be a real function on Ω^{pure} A roof of G is called a *roof extension* of g if $G(\pi) = g(\pi)$ for pure states. A *convex roof extension of g*, G, is convex and coincides on Ω^{pure} with g.

We do full tomography on three qubits and calculate the convex roof extension of the three tangle using the methods detailed in the following reference(102).



(a) Three tangle of GHZ, averaged over 4 trial runs



(b) Three tangle of cluster, averaged over 4 trial runs

Figure 4.11: The experiments were done on ibmqx2

As can be seen, the oscillations observed have disappeared giving more evidence that was observed was the quantum states picking up relative phases that made the wave-function complex as opposed to remaining real. By doing full state tomography, we have also removed the assumption of the states being pure. The results shown include readout correction to the results.

4.6 CONCLUSIONS

We have studied the relative strength and robustness of entanglement of the three- and four-qubit GHZ and linear cluster states on the IBM Rüeschlikon (ibmqx5) superconducting quantum computer by measuring entanglement monotones $\mathcal{E}_3$ and $\mathcal{E}_{4b}$, defined in (4.3) and (4.5). These entanglement measures have the property that they are ideally equal to one for the GHZ and cluster states, allowing for a meaningful cross-comparison. However we find that $\mathcal{E}_3$ and $\mathcal{E}_{4b}$ are in fact non-monotonic, which we ascribe to the development of imaginary components of the states. After proposing a simple phase-drift model for this effect, we attempt to correct it, and after the correction find the GHZ states to be significantly

more robust than the cluster states, exhibiting higher fidelity (data not shown here) and entanglement at later times.

What is expected theoretically? There is a commonly stated expectation that cluster states are more robust, due to their property that they remain partially entangled after measurement of a subset of $k < n$ qubits. (This property is essential for their use in measurement-based quantum computation.) We note, however, that this form of entanglement is not of the genuine multi-qubit type measured by $\mathcal{E}_3$ and $\mathcal{E}_{4b}$. Furthermore, our results contradict the predictions of a Markovian $T_{1,2}$ model and a non-Markovian dephasing model with realistic (but spatially uncorrelated) $1/f$ flux noise.

A simple explanation for the robustness of GHZ states observed here is that they only have two components (one relative phase) to acquire pure dephasing errors. And our compensation technique, which applies identical phase shifts $-\phi/n$ to each qubit, may be non-optimal for cluster states, whereas for GHZ states any distribution of phase shifts adding up to $-\phi$ is acceptable.

We note that the data presented here was acquired over more than 9 months, during which there were drifts in system parameters and gate fidelities, leading to small inconsistencies between some of the figures. However the principal observations, that GHZ and cluster state preparation fidelities and initial entanglement are similar, that $\mathcal{E}_3$ and $\mathcal{E}_{4b}$ oscillate in time, and that after phase compensation the GHZ states remain more entangled, were generally observed.

Finally, by doing full state tomography, we were able to truly measure the three tangle and observed the GHZ state containing more entanglement than the cluster state.

4.6.1 STATE PREPARATION ERRORS

Here we provide the data summarized in Table 4.1.

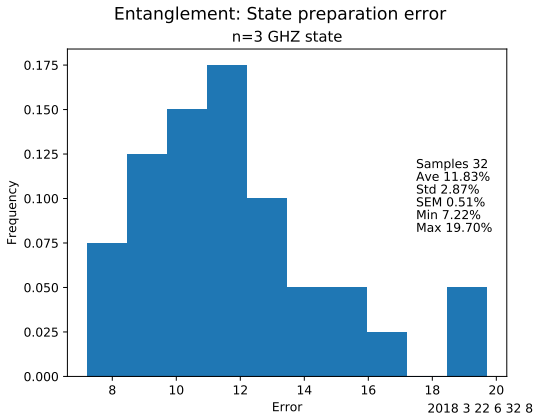


Figure 4.12: Histogram of 32 independent state preparation errors for the $n=3$ GHZ state.

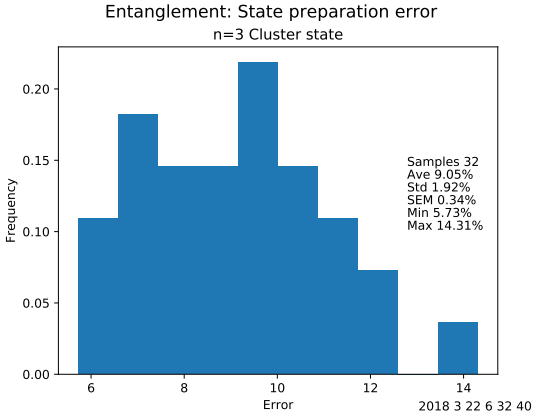


Figure 4.13: Histogram of 32 independent state preparation errors for the $n=3$ cluster state.

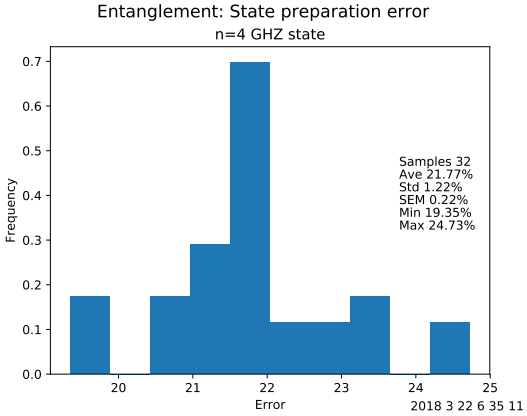


Figure 4.14: State preparation errors for the $n=4$ GHZ state.

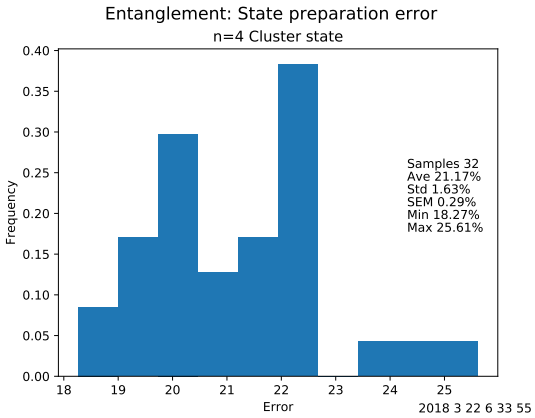


Figure 4.15: State preparation errors for the $n=4$ cluster state.

Bibliography

[1] Seth Lloyd. Universal Quantum Simulators. *Science*, 273(5278):1073–1078, aug 1996.

[2] David G Cory, Amr F Fahmy, and Timothy F Havel. Ensemble quantum computing by NMR spectroscopy. *Proceedings of the National Academy of Sciences of the United States of America*, 94(5):1634–9, mar 1997.

[3] Neil A. Gershenfeld and Isaac L. Chuang. Bulk Spin-Resonance Quantum Computation. *Science (New York, N.Y.)*, 275(5298):350–6, jan 1997.

[4] Isaac L. Chuang, Neil Gershenfeld, and Mark Kubinec. Experimental Implementation of Fast Quantum Searching. *Physical Review Letters*, 80(15):3408–3411, apr 1998.

[5] Samuel. L. Braunstein, Carlton. M. Caves, Richard. Jozsa, Noah. Linden, Sandu. Popescu, and Rüdiger. Schack. Separability of Very Noisy Mixed States and Implications for NMR Quantum Computing. *Physical Review Letters*, 83(5):1054–1057, aug 1999.

[6] Noah Linden and Sandu Popescu. Good Dynamics versus Bad Kinematics: Is Entanglement Needed for Quantum Computation? *Physical Review Letters*, 87(4):047901, jul 2001.

[7] Emanuel. Knill, Raymond. Laflamme, and Gerard. J. Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, 409(6816):46–52, jan 2001.

[8] Zhi Zhao, Yu-Ao Chen, An-Ning Zhang, Tao Yang, Hans J. Briegel, and Jian-Wei Pan. Experimental demonstration of five-photon entanglement and open-destination teleportation. *Nature*, 430(6995):54–58, jul 2004.

[9] Jian-Wei Pan, Matthew Daniell, Sara Gasparoni, Gregor Weihs, and Anton Zeilinger. Experimental Demonstration of Four-Photon Entanglement and High-Fidelity Teleportation. *Physical Review Letters*, 86(20):4435–4438, may 2001.

[10] Matthias Steffen, Markus Ansmann, Radoslaw C Bialczak, Nadav Katz, Erik Lucero, Robert McDermott, Matthew Neeley, Eva M Weig, Andrew N Cleland, and John M. Martinis. Measurement of the entanglement of two superconducting qubits via state tomography. *Science (New York, N.Y.)*, 313(5792):1423–5, sep 2006.

[11] Peter. J. Leek, Stefan. Filipp, Peter. Maurer, Manuel. Baur, Raechel. Bianchetti, Johannes M. Fink, Martin. Göppl, Lars. Steffen, and Andreas. Wallraff. Using sideband transitions for two-qubit operations in superconducting circuits. *Physical Review B*, 79(18):180511, may 2009.

[12] Leonardo DiCarlo, Jerry. M. Chow, Jay. M. Gambetta, Lev S. Bishop, Blake R. Johnson, I. Schuster, David, Johannes Majer, Alexandre. Blais, Luigi. Frunzio, Steven. M. Girvin, and Robert J. Schoelkopf. Demonstration of two-qubit algorithms with a superconducting quantum processor. *Nature*, 460(7252):240–244, jul 2009.

[13] Richard. Harris, Jonas. Johansson, Andrew J. Berkley, Mark W. Johnson, Trevor Lanting, Siyuan Han, Paul Bunyk, Eric. Ladizinsky, T. Oh, Ilya. Perminov, E. Tolkacheva, Sergey Uchaikin, E. M. Chapple, C. Enderud, C. Rich, M. Thom, J. Wang, B. Wilson, and Geordie. Rose. Experimental demonstration of a robust and scalable flux qubit. *Physical Review B - Condensed Matter and Materials Physics*, 81(13):1–19, 2010.

[14] Sergio Boixo, Tameem Albash, Federico M. Spedalieri, Nicholas Chancellor, and Daniel A. Lidar. Experimental signature of programmable quantum annealing. *Nature Communications*, 4:1–12, 2013.

[15] M. W. Johnson, M. H.S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J.S. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose. Quantum annealing with manufactured spins. *Nature*, 473(7346):194–198, 2011.

[16] Sergio Boixo, Troels F. Rønnow, Sergei V. Isakov, Zhihui Wang, David Wecker, Daniel A. Lidar, John M. Martinis, and Matthias Troyer. Evidence for quantum annealing with more than one hundred qubits. *Nature Physics*, 10(3):218–224, 2014.

[17] Graeme Smith and John Smolin. Putting "Quantumness" to the Test. *Physics*, 6(c):1101–1103, 2013.

[18] Lei Wang, Troels F. Rønnow, Sergio Boixo, Sergei V. Isakov, Zhihui Wang, David Wecker, Daniel A. Lidar, John M. Martinis, and Matthias Troyer. Comment on: "Classical signature of quantum annealing". pages 1–8, 2013.

[19] John A. Smolin and Graeme Smith. Classical signature of quantum annealing. *Frontiers in Physics*, 2:52, sep 2014.

[20] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man Hong Yung, Xiao Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5:4213, 2014.

[21] Dave Wecker, Matthew B. Hastings, and Matthias Troyer. Progress towards practical quantum variational algorithms. *Physical Review A*, 92(4):042303, oct 2015.

[22] Jarrod R. McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2), 2016.

[23] C Hempel and Et al. Scaling and realtime implementation of a quantum-classical variational algorithm. *In preparation*, 2016.

[24] Jarrod R. McClean, Mollie E. Kimchi-Schwartz, Jonathan Carter, and Wibe A. De Jong. Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states. *Physical Review A*, 95(4):1–10, 2017.

[25] Zhi Cheng Yang, Armin Rahmani, Alireza Shabani, Hartmut Neven, and Claudio Chamon. Optimizing variational quantum algorithms using pontryagin's minimum principle. *Physical Review X*, 7(2):1–10, 2017.

[26] Ying Li and Simon C Benjamin. Efficient variational quantum simulator incorporating active error minimization. *Physical Review X*, 7(2):21050, jun 2017.

[27] Peter D. Johnson, Jonathan Romero, Jonathan Olson, Yudong Cao, and Alán Aspuru-Guzik. QVECTOR: an algorithm for device-tailored quantum error correction. nov 2017.

[28] Suguru Endo, Ying Li, Simon Benjamin, and Xiao Yuan. Variational quantum simulation of general processes. dec 2018.

[29] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm. nov 2014.

[30] Eric R. Anschuetz, Jonathan P. Olson, Alán Aspuru-Guzik, and Yudong Cao. Variational Quantum Factoring. aug 2018.

[31] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem. dec 2014.

[32] Cedric Yen-Yu Lin and Yechao Zhu. Performance of QAOA on Typical Instances of Constraint Satisfaction Problems with Bounded Degree. jan 2016.

[33] Aniruddha Bapat and Stephen Jordan. Bang-bang control as a design principle for classical and quantum optimization algorithms. page arXiv:1812.02746, 2018.

[34] G G. Guerreschi and A Y. Matsuura. QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. pages 1–14, 2018.

[35] Fernando G. S. L. Brandão, Michael Broughton, Edward Farhi, Sam Gutmann, and Hartmut Neven. For Fixed Control Parameters the Quantum Approximate Optimization Algorithm's Objective Function Value Concentrates for Typical Instances. page arXiv:1812.04170, 2018.

[36] Edward Farhi and Aram W. Harrow. Quantum Supremacy through the Quantum Approximate Optimization Algorithm. feb 2016.

[37] IBM Quantum Experience, 2017.

[38] Asier Ozaeta and Peter L. McMahon. Decoherence of up to 8-qubit entangled states in a 16-qubit superconducting quantum processor. pages 1–5, 2017.

[39] U. Alvarez-Rodriguez, M. Sanz, L. Lamata, and E. Solano. Quantum Artificial Life in an IBM Quantum Computer. *Scientific Reports*, 8(1):1–9, 2018.

[40] James R. Wootton and Daniel Loss. Repetition code of 15 qubits. *Physical Review A*, 97(5):1–5, 2018.

[41] Kenneth Rudinger, Timothy Proctor, Dylan Langharst, Mohan Sarovar, Kevin Young, and Robin Blume-Kohout. Probing context-dependent errors in quantum processors. pages 1–11, 2018.

[42] Ana Martin, Bruno Candelas, Ángel Rodríguez-Rozas, José D. Martín-Guerrero, Xi Chen, Lucas Lamata, Román Orús, Enrique Solano, and Mikel Sanz. Towards Pricing Financial Derivatives with an IBM Quantum Computer. pages 1–8, 2019.

[43] Emilie Huffman and Ari Mizel. Leggett-Garg test of superconducting qubit addressing the clumsiness loophole. (2):1–6, 2016.

[44] Narendra N. Hegade, Bikash K. Behera, and Prasanta K. Panigrahi. Experimental Demonstration of Quantum Tunneling in IBM Quantum Computer. 2017.

[45] Amolak Ratan Kalra, Navya Gupta, Bikash K. Behera, Shiroman Prakash, and Prasanta K. Panigrahi. Demonstration of the no-hiding theorem on the 5-Qubit IBM quantum computer in a category-theoretic framework. *Quantum Information Processing*, 18(6), 2019.

[46] A. A. Zhukov, E. O. Kiktenko, A. A. Elistratov, W. V. Pogosov, and Yu E. Lozovik. Quantum communication protocols as a benchmark for programmable quantum computers. *Quantum Information Processing*, 18(1):1–23, 2019.

[47] Mitali Sisodia, Abhishek Shukla, and Anirban Pathak. Experimental realization of nondestructive discrimination of Bell states using a five-qubit quantum computer. *Physics Letters, Section A: General, Atomic and Solid State Physics*, 381(46):3860–3874, 2017.

[48] Rolf Landauer. Is Quantum Mechaniccs Useful? *Philosophical Transcation: Physical Sciences and Engineering*, 353(1703):367–376, dec 1995.

[49] Seth Lloyd. A Potentially Realizable Quantum Computer. *Science*, 261(5128):1569–1571, 1993.

[50] Seth Lloyd. Envisioning a Quantum Supercomputer. *Science*, 263(5147):695, 1994.

[51] Andrew Steane. Multiple-particle interference and quantum error correction. *Proceedings of the Royal Society A.*, 452, nov 1996.

[52] Raymond Laflamme, Cesar Miquel, Juan Pablo Paz, and Wojciech Hubert Zurek. Perfect Quantum Error Correcting Code. *Phys. Rev. Lett.*, 77:198–201, jul 1996.

[53] Daniel Gottesmann. Class of quantum error-correcting codes saturating the quantum Hamming bound. *Phys.Rev.A*, 54:1862–1868, 1996.

[54] Frank Gaitan. *Quantum Error Correction and Fault Tolerant Quantum Computing.* CRC Press, first edition, feb 2008.

[55] Richard Cleve and Daniel Gottesman. Efficient computations of encodings for quantum error correction. *Physical Review A*, 56(1):76–82, jul 1997.

[56] Feynman Richard. Simulating Physics With Computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, 1982.

[57] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Computing*, 26:1484–1509, 1997.

[58] Peter W Shor. Scheme for reducing decoherence in quantum computer memory. *Physical Review A*, 52:R2493–R2496, oct 1995.

[59] Daniel Gottesmann. The Heisenberg Representation of Quantum Computers. *arXiv:quant-ph/9807006v1*, 1998.

[60] Christopher E Granade David G Cory Easwar Magesan Daniel Puzzuoli. Modeling quantum noise for efficient testing of fault-tolerant circuits. *Phys.Rev. A*, 87:12324, 2013.

[61] Mauricio Gutiérrez, Lukas Svec, Alexander Vargo, and Kenneth R Brown. Approximation of real errors by Clifford channels and Pauli measurements. *Physical Review A*, 87:030302(R), 2013.

[62] Michael R. Geller and Zhou Zhongyuan. Efficient error models for fault-tolerant architectures and the Pauli twirling approximation. *Physical Review A*, 88:12314, 2013.

[63] Tomita Yu and Krysta M. Svore. Low-distance Surface Codes under Realistic Quantum Noise. *Physical Review A*, 90:62320, dec 2014.

[64] Daniel Puzzuoli, Christopher Granande, Holger Haas, Ben Criger, Easwar Magesan, and Cory David. Tractable Simulation of Error Correction with Honest Approximations to Realistic Fault Models. *Physical Review A*, 89:22306, feb 2014.

[65] Mauricio Gutiérrez and Kenneth R. Brown. Comparison of a quantum error-correction threshold for exact and approximate errors. *Phys. Rev. A*, 91:22335, feb 2015.

[66] Joseph Emerson, Marcus Silva, Osama Moussa, Colm Ryan, Jonathan Baugh, Martin Laforest, David G Cory, and Raymond Laflamme. Symmetrized Characterization of Noisy Quantum Processes. *Science*, 317(5846):1893–1896, 2007.

[67] C. López, Cecilia, A Bendersky, and Cory G. David. Progress toward scalable tomography of quantum maps using twirling-based methods and information hierarchies. *Physical Review A*, 81:62113, 2010.

[68] Cecilia C. López, Lévi Benjamin, and David G. Cory. Error characterization in quantum information processing: A protocol for analyzing spatial correlations and its experimental implementation. *Physical .Review A*, 79:42328, 2009.

[69] Pastawski F Bendersky A. and Juan Pablo Paz. Selective and Efficient Estimation of Parameters for Quantum Process Tomography. *Physical Review Letters*, 100:90403, 2008.

[70] Joydip Ghosh, Andrei Galiautdinov, Zhongyuan Zhou, Alexander N Korotkov, John M Martinis, and Michael R Geller. High-fidelity controlled-σ^Z gate for resonator-based superconducting quantum computers. *Physical Review A*, 87:22309, feb 2013.

[71] Robert Hernmann. *Lie Groups for Physicists*. Addison-Wesley Pub Co.

[72] Troels F. Rønnow, Zhihui Wang, Joshua Job, Sergio Boixo, Sergei V. Isakov, David Wecker, John M. Martinis, Daniel A. Lidar, and Matthias Troyer. Defining and detecting quantum speedup. *Science*, 345(6195):420–424, 2014.

[73] Scott Aaronson and Alex Arkhipov. The Computational Complexity of Linear Optics. *The Theory of Computing*, 9:143, feb 2013.

[74] Matthew A. Broome, Alessandro Fedrizzi, Saleh Rahimi-Keshari, Justin Dove, Scott Aaronson, Timothy C. Ralph, and Andrew G. White. Photonic Boson Sampling in a Tunable Circuit. *Science*, 339:794, feb 2013.

[75] Justin B. Spring, Benjamin J. Metcalf, Peter C. Humphreys, W. Steven Kolthammer, Xian-Min Jin, Marco Barbieri, Animesh Datta, Nicholas Thomas-Peter, Nathan K. Langford, Dmytro Kundys, James C. Gates, Brian J. Smith, Peter G. R. Smith, and Ian A. Walmsley. Boson Sampling on a Photonic Chip. *Science*, 339:798, feb 2013.

[76] Max Tillmann, Borivoje Dakić, René Heilmann, Stefan Nolte, Alexander Szameit, and Philip Walther. Experimental boson sampling. *Nature Photonics*, 7:540, may 2013.

[77] Michael R. Geller, John M. Martinis, Phillip C. Sornborger, Andrew T. Stancil, You Hao, Emily J. Pritchett, and Andrei Galiautdinov. Universal quantum simulation with prethreshold superconducting qubits: Single-excitation subspace method. *Physical Review A*, 91:62309, jun 2015.

[78] Anthony W. Knapp. *Lie Groups Beyond an Introduction*. Birkhauser,, 1996.

[79] Chi K. Law and Joseph H. Eberly. Arbitrary Control of a Quantum Electromagnetic Field. *Physical Review Letters*, 76:1055, feb 1996.

[80] Max Hofheinz, H. Wang, Bialczak C. Ansmann, Markus Radoslaw, Erik Lucero, M Neeley, A D O'Connell, D Sank, J Wenner, John M. Martinis, and Andrew N.

Cleland. Synthesizing arbitrary quantum states in a superconducting resonator. *Nature*, 459:546, may 2009.

[81] Reck Michael, Anton Zeilinger, Herbert J. Bernstein, and Philip Bertani. Experimental realization of any discrete unitary operator. *Physical Review Letters*, 73:58, jul 1994.

[82] Ryszard Horodecki, Paweł Horodecki, Michał Horodecki, and Karol Horodecki. Quantum entanglement. *Reviews of Modern Physics*, 81(2):865–942, jun 2009.

[83] Otfried Gühne and Géza Tóth. Entanglement detection. *Physics Reports*, 474(1-6):1–75, apr 2009.

[84] Matthew Neeley, Radoslaw C. Bialczak, M. Lenander, E. Lucero, Matteo Mariantoni, A. D. O'Connell, D. Sank, H. Wang, M. Weides, J. Wenner, Y. Yin, T. Yamamoto, A. N. Cleland, and John M. Martinis. Generation of three-qubit entangled states using superconducting phase qubits. *Nature*, 467(7315):570–573, sep 2010.

[85] L. DiCarlo, M. D. Reed, L. Sun, B. R. Johnson, J. M. Chow, J. M. Gambetta, L. Frunzio, S. M. Girvin, M. H. Devoret, and R. J. Schoelkopf. Preparation and measurement of three-qubit entanglement in a superconducting circuit. *Nature*, 467(7315):574–578, sep 2010.

[86] Chao Song, Kai Xu, Wuxin Liu, Chui-ping Yang, Shi-Biao Zheng, Hui Deng, Qiwei Xie, Keqiang Huang, Qiujiang Guo, Libo Zhang, Pengfei Zhang, Da Xu, Dongning Zheng, Xiaobo Zhu, H Wang, Y.-A. Chen, C.-Y. Lu, Siyuan Han, and Jian-Wei Pan. 10-Qubit Entanglement and Parallel Logic Operations with a Superconducting Circuit. *Phys. Rev. Lett.*, 119:6, nov 2017.

[87] Nicolai Friis, Oliver Marty, Christine Maier, Cornelius Hempel, Milan Holzäpfel, Petar Jurcevic, Martin B. Plenio, Marcus Huber, Christian Roos, Rainer Blatt, and Ben

Lanyon. Observation of Entangled States of a Fully Controlled 20-Qubit System. *Physical Review X*, 8(2):021012, apr 2018.

[88] Yuanhao Wang, Ying Li, Zhang-qi Yin, and Bei Zeng. 16-qubit IBM universal quantum computer can be fully entangled. *npj Quantum Information*, 4(1):46, dec 2018.

[89] R. Di Candia, B. Mejia, H. Castillo, J. S. Pedernales, J. Casanova, and E. Solano. Embedding quantum simulators for quantum computation of entanglement. *Physical Review Letters*, 111(24):1–5, 2013.

[90] Ming Cheng Chen, Dian Wu, Zu En Su, Xin Dong Cai, Xi Lin Wang, Tao Yang, Li Li, Nai Le Liu, Chao Yang Lu, and Jian Wei Pan. Efficient Measurement of Multiparticle Entanglement with Embedding Quantum Simulator. *Physical Review Letters*, 116(7):2–5, 2016.

[91] Diogo Cruz, Romain Fournier, Fabien Gremion, Alix Jeannerot, Kenichi Komagata, Tara Tosic, Jarla Thiesbrummel, Chun Lam Chan, Nicolas Macris, Marc-André Dupertuis, and Clément Javerzac-Galy. Efficient Quantum Algorithms for GHZ and W States, and Implementation on the IBM Quantum Computer. *Advanced Quantum Technologies*, 2(5-6):1900015, jun 2019.

[92] Guifré Vidal. Entanglement monotones. *Journal of Modern Optics*, 47(2-3):355–376, feb 2000.

[93] Andreas Osterloh and Jens Siewert. Constructing N-qubit entanglement monotones from antilinear operators. *Phys. Rev. A*, 72:12337, jul 2005.

[94] Christopher Eltschka, Thierry Bastin, Andreas Osterloh, and Jens Siewert. Multipartite-entanglement monotones and polynomial invariants. *Physical Review A*, 85(2):022301, feb 2012.

[95] Patricia Contreras-Tejada, Carlos Palazuelos, and Julio I. de Vicente. Resource Theory of Entanglement with a Unique Multipartite Maximally Entangled State. *Physical Review Letters*, 122(12):120503, mar 2019.

[96] Valerie Coffman, Joydip Kundu, and William K. Wootters. Distributed entanglement. *Physical Review A*, 61(5):052306, apr 2000.

[97] Armin Uhlmann. Fidelity and concurrence of conjugated states. *Physical Review A*, 62(3):032307, aug 2000.

[98] Alexander Wong and Nelson Christensen. Potential multiparticle entanglement measure. *Physical Review A*, 63(4):044301, mar 2001.

[99] Andreas Osterloh and Ralf Schützhold. Four-concurrence in the transverse XY spin-1/2 chain. *Phys. Rev. A*, 96:12331, jul 2017.

[100] William K. Wootters. Entanglement of Formation of an Arbitrary State of Two Qubits. *Physical Review Letters*, 80(10):2245–2248, mar 1998.

[101] Steven T. Flammia and Yi-Kai Liu. Direct Fidelity Estimation from Few Pauli Measurements. *Physical Review Letters*, 106(23):230501, jun 2011.

[102] Beat Röthlisberger, Jörg Lehmann, and Daniel Loss. libCreme: An optimization library for evaluating convex-roof entanglement measures. *Computer Physics Communications*, 183(1):155–165, jan 2012.