Three Results in Quantum Physics

by

Jaron Meser Huq

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master's of Mathematics
in
Applied Mathematics (Quantum Information)

Waterloo, Ontario, Canada, 2019

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis is split into three disjoint sections. The first deals with two practical issues regarding the use of unitary 2-designs. A simplified description of how to generate elements of the smallest known unitary 2-design on qubits is given which should be usable even for people who do not have much experience with the mathematics of finite fields. The section also gives a new way to decompose an arbitrary element of the Clifford group into one and two qubit gates and is by far the simplest decomposition of its kind. The second section describes similarities and differences between a probabilistic formulation of classical mechanics and quantum mechanics, with the intention that it could become a resource for physics students to show that just because a physical phenomenon is strange it is not necessarily quantum. The third section is speculative and delves into the relationship between a highly theoretical field of quantum information science, Quantum Prover Interactive Proofs, and a highly practical area of quantum information science, error characterization. Previously unnoticed links are drawn between these fields with the intention that further research can provide fertile ground for both to flourish.

Acknowledgements

My thanks to my beloved wife Suzie, without her support I would not have been able to finish this program. Thanks also to my supervisor Joseph Emerson and to my colleagues Joel Wallman, Kristine Brayden, Arnaud Carignan-Dugas and Hammam Elyas.

Table of Contents

List of Figures

List of Tables

Chapter 1

Synthesizing Elements of a Small Unitary 2-Design and a New Decomposition of the Clifford Group

Unitary 2-designs have been useful tools in quantum information science, with applications in problems like quantum error correction [1], simulation of sub-theories of quantum mechanics [2], quantum key distribution [3], randomized benchmarking [4], simulation of scrambling in black holes [5], modeling entanglement in condensed matter systems [6], and decoupling [7]. Their many uses have led to many different yet equivalent definitions. Speaking loosely, a unitary 2-design is a finite set of unitary matrices such that sampling from it approximates sampling uniformly at random from the set of all unitary matrices. The '2' in the name describes the degree of approximation as there exist unitary 1-designs that do the approximation worse and unitary t-designs with $t > 2$ that are better approximations.

Speaking more accurately [8] a *unitary t-design* for D dimensions is a finite set of unitary operators $\{U_k\}_{k=1}^{K}$ on C^D such that for every polynomial $P_{(t,t)}(U)$ of degree at most t in the matrix elements of U and at most t in the complex conjugates of those matrix elements,

$$\frac{1}{K}\sum_{(k=1)}^{K} P_{(t,t)}(U_k) = \int dU P_{(t,t)}(U)$$

Where the integral is over the Haar measure on the group of unitary matrices. In other words averaging a degree t polynomial over the elements of a unitary t-design is the same as averaging the polynomial over the entire set of unitary matrices.

The most popular unitary 2-design in the quantum information community is the Clifford group which is the finite set of quantum operations that can be generated by controlled-NOT, Hadamard and phase gates acting on qubits. However, the Clifford group is in fact a unitary 3-design so it could be considered overkill in some situations. Of particular interest in some applications has been a subgroup of the Clifford group first given a useful description by H.F. Chau [3], so for the remainder of this article the subgroup will be referred to as the Chau subgroup. This subgroup is a unitary 2-design and it has $2^{2n}(2^{3n}-2^n)$ elements on n qubits which is significantly smaller than the total Clifford group that grows like $2^{2n^2+O(n)}$. This is close to a lower bound on the size of a 2-design of $2^{4n}-2^{2n+1}+2$ from [9]. The same paper conjectures that the Chau subgroup is the smallest unitary 2-design on two or more qubits. There are some other characterizations of this group that are potentially useful. One of the most useful descriptions of the subgroup is as a collection of matrices over a finite field, this will elaborated upon later in this thesis. Chau's original paper gives a method for finding the unitary matrix associated with an element of the group but it takes time exponential in the number of qubits.

Of critical importance to be able to use the Clifford and Chau groups is to know how to implement the elements to the group as sequences of experimentally available gates, otherwise knowledge of the groups' existence would not be useful in laboratory situations. There are a handful of existing results on ways to turn elements of the Clifford group into sequences of gates. One of the first major results was from Aaronson and Gottesman [10] where they find a decomposition of Clifford circuits into the gate library { CNOT,H,P } with gate complexity $O(\frac{n^2}{\log n})$ arranged to have depth $O(\frac{n}{\log n})$. [1] However, their decomposition uses many more CNOT gates than later decompositions. From there Maslov and Roetteler [11] developed a new decomposition of the Clifford group which uses significantly fewer CNOT gates than the Aaronson-Gottesman decomposition, but in terms of big-O notation has the same complexity scalings. There have also been suggestions on how to implement approximate unitary 2-designs, notably in [8] they define ϵ- approximate unitary 2-designs and find that it is possible to implement elements of the 2-design with total size $O(n \log 1/\epsilon)$ and depth $O(\log n \log 1/\epsilon)$. However it is believed that many applications would require $\epsilon \approx 2^{-n}$, thereby erasing the benefits of using an approximate method [12]. Specific to the Chau subgroup instead of the whole Clifford group Cleve et al [12] found an algorithm for a gate decomposition which has lower complexity and depth than the one presented here but it requires ancilla qubits to implement an element of the group, thus restricting its usefulness for many applications on near term devices. As for lower bounds on the size of a gate decomposition of a unitary 2-design [12] has a result that, omitting

[1] A proof that this scaling is optimal under more general circumstances is included in the next subsection.

some technical details, any gate decomposition of a unitary 2-design would have have total gate count $\Omega(n)$ and depth $\Omega(\log(n))$.

1.1 Lower Bound on Complexity of Clifford Group Decomposition

This section gives a proof of the claim that over a library of 1- and 2-qubit gates and without having any ancilla qubits available, there are some elements of the Clifford group which require $\Omega(\frac{n^2}{\ln n})$ gates to synthesize. The proof is a slightly altered version of the proof given in [13] to prove a similar result about the minimum number of CNOT gates required to make a CNOT circuit.

Let your particular system have $n >> 1$ qubits. Say that your gate library contains a different 2-qubit gates that can act on any pair of qubits and b 1-qubit gates that can act on any qubit. Then there are at most $(an(n-1) + bn + 1)$ different operations that can be performed at one time if an extra "do nothing" operation is included. Most existing decompositions set $a = 1$ and $b = 2$ for the CNOT-Hadamard-Phase gate set. Let $N = N(n)$ be the maximum number of 1- and 2-qubit operations that is necessary to produce any element of the Clifford group on n qubits. Then by a simple counting argument you can conclude that the total number of possible Clifford circuits generated with at most N gates and do-nothing operations must exceed the size of the Clifford group.

$$(an(n-1) + bn + 1)^N \geq |C_n| > 2^{2n^2}$$

Taking a natural logarithm of both sides and simplifying yields

$$N \ln \left[an^2 \left(1 + \frac{b-a}{an} + \frac{1}{an^2} \right) \right] \geq (2 \ln 2) n^2$$

$$N \geq \frac{(2 \ln 2) n^2}{\ln \left[an^2 \left(1 + \frac{b-a}{an} + \frac{1}{an^2} \right) \right]}$$

$$N \geq \frac{(2 \ln 2) n^2}{2 \ln n + \ln a + \ln \left(1 + \frac{b-a}{an} + \frac{1}{an^2} \right)} = \Omega \left(\frac{n^2}{\ln n} \right)$$

Some things to note, the number of distinct 1- and 2-qubit gates did not actually matter so the coefficients a and b were only included for added rigor. Even more rigour

can be added by accounting for the fact that some gates like the CZ gate are symmetric so they add $n(n-1)/2$ options at each time step instead of $n(n-1)$ options, but the result of the previous proof shows that complication would have been superfluous in the end. Furthermore, even if 3-, 4-, or k-qubit gates were allowed it would not have changed the asymptotic scaling at the end as long as the number of distinct gates does not scale with n. There is nothing in the argument that requires the gate library to be composed exclusively of Clifford gates, so leaving the Clifford group with something like a T gate or Toffoli gate and returning later would not decrease the asymptotic complexity of a decomposition, but could still be a useful trick in practice for reducing gate counts.

This lower-bound scaling cannot be decreased further for the Clifford group since multiple existing decompositions, including the one presented in this thesis, have a worst case asymptotic complexity of $O\left(\frac{n^2}{\ln(n)}\right)$. However there is still an open question about the Chau subgroup. Since the size of the Chau subgroup is $2^{5n} - 2^{3n}$ redoing the above proof leaves open the possibility that some property of the Chau subgroup could be exploited to find a decomposition with a lower asymptotic gate count. Redoing the above proof would suggest that the total gate count may be as low as $O\left(\frac{n}{\ln(n)}\right)$ but that conflicts with a lower bound by Cleve in [12] that with probability at least $1/3$ the gate count of an element of a unitary 2-design is at least $n/4$. So at best the gate count of Chau group element decomposition could match the known lower bound of $O(n)$

1.2 Interlude 1 - Finite Fields

The structure of the Clifford group and Chau subgroup is most easily described using the language of finite fields. This section is a very brief introduction to the concept of a finite field and to show what notations will be used henceforth.

A *field* is any collection of elements F with two binary operations called *addition* and *multiplication*, denoted by $+$ and $\times$, such that

1. For any $a, b \in F$ you have $a + b \in F$ and $a \times b \in F$
2. For any $a, b, c \in F$ you have $(a + b) + c = a + (b + c)$ and $(a \times b) \times c = a \times (b \times c)$
3. For any $a, b \in F$ you have $a + b = b + a$ and $a \times b = b \times a$
4. There exist elements $0, 1 \in F$ such that for any $a \in F$ you have $a + 0 = a$ and $a \times 1 = a$

5. For all $a \in F$ there exist elements $-a \in F$ such that $(-a) + a = 0$

6. For all $a \in F$ with $a \neq 0$ there exists $a^{-1} \in F$ such that $a \times a^{-1} = 1$.

7. For all $a, b, c \in F$ you have $a \times (b + c) = a \times b + a \times c$

As examples the rational numbers, real numbers, and complex numbers are all fields. If the field has only finitely many elements it is called a *finite field*. The simplest finite field is F_2, the integers modulo 2. $F_2 = \{0, 1\}$ with the operations you would expect except for having $1+1 = 0$, which is the same as saying $1 = -1$. It is a fact that finite fields have size p^n for some prime p and integer n. It is also true that all finite fields of the same size are isomorphic. This eventually leads to the conclusion that any finite field of size p^n, denoted F_{p^n} is also a vector space of dimension n over F_p. For this thesis we can restrict ourselves to the case $p = 2$. There is the question of how to represent the field F_{2^n} in a useful form. The fact that F_{2^n} is a vector space over F_2 gives an obvious choice to represent elements of F_{2^n} are vectors of length n where every entry in the vector is 1 or 0. This does not tell us how to do any non-trivial multiplications. It turns out that F_{2^n} can also be represented as the set of polynomials with coefficients in F_2 subject to the relation $I(x) = 0$, where $I(x)$ is an *irreducible polynomial* of degree n over F_2. A degree n irreducible polynomial over F_2 is a polynomial $I(x)$ such that if $I(x) = p(x)q(x)$ then either $p(x) = 1$ or $q(x) = 1$. This irreducibility property ensures that no two non-zero polynomials multiply to be the zero polynomial, which would destroy the existence of multiplicative inverses in the field. To relate these polynomials to n-tuples of numbers simply arrange the coefficients of the polynomial into an n-tuple. Henceforth it will be standard that the coefficient of the constant term is the first entry in the vector, the coefficient of the linear term is the second entry, etc.

As an example for $p = 2, n = 3$, $I(x) = x^3 + x + 1$ is an irreducible polynomial of degree 3 over F_2. So if it is seen as generating the field F_{2^3}, then to multiply the polynomials x^2 and $x + 1$ you multiply as usual to get $x^3 + x^2$ then apply the relation $x^3 = x + 1$ to find that $(x^2) \times (x + 1) = x^2 + x + 1$. In the form of 3-tuples the same calculation would be $(0, 0, 1) \times (1, 1, 0) = (1, 1, 1)$

1.3 Interlude 2 - Review of Symplectic Matrices and the Clifford Group

This section is intended to give a brief overview of the interplay between the how the Clifford group acts as a sets of unitary matrices and as a set of symplectic matrices. This

section contains no original research and is intended only as a review of certain facts about the Clifford group that will be frequently referenced in later sections. Readers are encouraged to see Section IV of [9] for the full details of the mathematical facts presented here.

First recall that the Clifford group on n qubits, C_n, has multiple convenient definitions. One referenced previously is that the Clifford group is the set of all gates generated by controlled-NOT, Hadamard and phase gates. Another is as the normalizer of the Pauli group P_n i.e.

$$C_n = \{U \mid \forall P \in P_n \implies UPU^\dagger \in P_n\}$$

Technically these definitions are different because the second definition allows for a continuous set of gates that would just apply an undetectable global phase to states. The definitions could be brought back into alignment by saying the Clifford group is either of the previously defined groups quotiented by the centre of the group, but those technical remarks will be glossed over.

Using the second definition as the normalizer of the Pauli group (with the centre of the group quotiented out) it becomes possible to uniquely describe elements of the Clifford group by their action on the Pauli group. First some ground work. Represent Pauli operators as a row-vector of length $2n$ filled with 1s and 0s. Multi-qubit Pauli operators are tensor products of single-qubit Pauli operators so say that $P = P_1 \otimes P_2 \otimes \dots P_n$. The vector representing P, call it a, will have $(a_i, a_{i+n}) = (0, 0)$ if $P_i = I$, $(a_i, a_{i+n}) = (1, 0)$ if $P_i = X$, $(a_i, a_{i+n}) = (1, 1)$ if $P_i = Y$, and $(a_i, a_{i+n}) = (0, 1)$ if $P_i = Z$. As an example on four qubits

$$I \otimes X \otimes Y \otimes Z \to (0, 1, 1, 0, 0, 0, 1, 1)$$

With this convention it will always be the case that for any element U of the Clifford group

$$UP_aU^\dagger = i^{\delta(U,a)}(-1)^{\epsilon(U,a)}P_{aS(U)}$$

The multiplications and additions in the Pauli operator's subscript happen in the finite field with two elements. Here δ and ϵ are bits. It is of course possible to instead store a number between 0 and 3 to represent the exponent of i, but it is useful to leave all numbers over the same field F_2. Thus, computing the action of the $2^n \times 2^n$ unitary matrix acting on

one of the 4^n elements of the Pauli group can be reduced to having a $2n \times 2n$ matrix act on a vector of length $2n$ and keeping track of the $2n$ phase bits. The matrix $S(U)$ depends only on the unitary matrix and do not depend on the Pauli operator P_a. The phase bits are the similar but their relationship to the unitary Clifford operator and Pauli operator is more complicated, so a notation which shows their functional dependencies explicitly was used.

Also notice that the structures of the Pauli and Clifford groups impose constraints on the form of S and the phase bits, because the relation

$$i^{\delta(VU,a)}(-1)^{\epsilon(VU,a)}P_{aS(VU)} = VUP_aU^\dagger V^\dagger = Vi^{\delta(U,a)}(-1)^{\epsilon(U,a)}P_{aS(U)}V^\dagger = i^{\delta(U,a)}i^{\delta(V,aS(U))}(-1)^{\epsilon(V,aS(U))}(-1)^{\epsilon(}$$

must hold. The constraints can be described succinctly. Define a $2n \times 2n$ matrix J as being composed of four square sub-matrices with the following form:

$$J = \begin{bmatrix} 0_n & 1_n \\ 1_n & 0_n \end{bmatrix}$$

Where 1_n is an $n \times n$ identity matrix and 0_n is an $n \times n$ zero matrix. Then S will satisfy $SJS^T = J$. Any matrix that satisfies this condition of leaving J invariant is called a *symplectic* matrix. There are some observations from this simple equations. Taking the determinant of this matrix equation it shows that S is invertible. If S satisfies the equation then so do S^T, S^{-1}, and S^{-T} where the superscript $-T$ is meant to indicate both an inverse and transpose. If S and R satisfy the equation then SR does too. Since the identity matrix clearly satisfies the relation and by the associativity of matrix multiplication the symplectic matrices form a group under matrix multiplication. It is clear that for the elements of the Clifford group, the symplectic matrix holds most of the interesting information about the operator and the extra phase bits hold a much smaller amount of information. Therefore for the remainder of this section, and indeed for most of this thesis, it will only be necessary to work with symplectic matrices, which is the same as focusing on $(Cliff_n/P_n)/U(1) \approx Sp(2n, F_2)$. Here Sp stands for symplectic, $2n$ means that the symplectic matrices in the set are $2n \times 2n$ matrices and F_2 means that the numbers in the matrix are elements of the field F_2.

The condition that a symplectic matrix leaves the matrix J invariant can be rephrased in another useful way. By writing the symplectic matrix as being made of four $n \times n$ sub-matrices

$$S = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

The condition for a matrix to be a symplectic matrix are as follows:

$$A^T C = C^T A$$
$$B^T D = D^T B$$
$$A^T D + C^T B = 1_n$$
$$B^T C + D^T A = 1_n$$

Furthermore there is a simple relationship for finding the inverse of a symplectic matrix

$$S^{-1} = \begin{bmatrix} D^T & -B^T \\ -C^T & A^T \end{bmatrix}$$

Since the blocks are permuted and now have transposes, the four commutation relations between the blocks show that the following four useful relations also hold

$$AB^T = BA^T$$
$$CD^T = DC^T$$
$$AD^T + BC^T = 1_n$$
$$CB^T + DA^T = 1_n$$

One of the papers that did the most to popularize the use of symplectic matrices is in Aaronson and Gottesman 2004 [10] where they are referred to as "tableaus". Readers should be aware that instead of associating each unitary operator to a symplectic as is done here they sometimes say that tableaus represent states and that they can contain redundant information. This interpretation did not become standard in subsequent works. As a final note Aaronson and Gottesman have a useful formula for counting the number of symplectic matrices as a function of the number of qubits which shows that $\mid Sp(2n, F_2) \mid \sim 2^{2n^2+n+o(n)}$

Matrix	Generators	Notes
$\begin{bmatrix} d_1 & d_2 \\ d_3 & d_4 \end{bmatrix}$	H,P	The group generated by single qubit gates. The four d_i matrices are diagonal
$\begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix}$	P	Group of Phase gates, d is diagonal.
$\begin{bmatrix} d_1 & d_2 \\ d_2 & d_1 \end{bmatrix}$	H	The group generated by Hadamard gates. The two d_i matrices are diagonal and $d_1 + d_2 = 1_n$
$\begin{bmatrix} 1 & 0 \\ d & 1 \end{bmatrix}$	$\sqrt{X}$	Group of$\sqrt{X}$ gates, d is diagonal.
$\begin{bmatrix} A & 0 \\ 0 & A^{-T} \end{bmatrix}$	CNOT	Group of CNOT gates. Isomorphic to $GL(n, F_2)$
$\begin{bmatrix} A & B \\ 0 & A^{-T]} \end{bmatrix}$	CNOT,P	Generated by CNOT and Phase gates. Can always be expressed as C-P-C-P
$\begin{bmatrix} 1 & B \\ 0 & 1 \end{bmatrix}$	CZ,P	It will always be the case that $B = B^T$. If the diagonal elements of B are 0 then only controlled-Z gates are needed. This subgroup is Abelian.

Table 1.1: Symplectic versions of subgroups of the Clifford group

1.3.1 Useful Subgroups of the Clifford Group

The Clifford group has many other useful subgroups besides the Chau subgroup. The table contains a list of some of the subgroups of the Clifford group that are generated by different gate libraries. A description of their form as a $2n \times 2n$ symplectic matrix is given as well as which physical gates generate the subgroup.

1.4 Generating a Symplectic Matrix

The simplest way to understand the overall structure of the Chau Subgroup is as a group of matrices over a certain extension field of the two element field F_2, in this case the relevant extension field for the case of n qubits is a field with 2^n elements here denoted by F_{2^n}. It is well known that in the language of symplectic matrices the Clifford group quotiented by the Pauli group (and the group of non-physical global phases) is isomorphic to the set of $2n \times 2n$

symplectic matrices (see previous subsection for details) over F_2, $C_n/P_n \approx Sp(2n, F_2)$. It is obvious that the Chau subgroup (with the Paulis quotiented out), here denoted $Chau_n$, has a representation as a set of $2n \times 2n$ symplectic matrices but there is another more convenient representation available. It turns out that $Chau_n \approx Sp(2, F_{2^n}) \approx SL(2, F_{2^n})$ [3]. The second isomorphism follows from a simple calculation showing that a 2×2 matrix satisfies the symplectic property if and only if its determinant is 1, so the 2×2 symplectic group is isomorphic to the 2×2 special linear group.

This isomorphism makes the generation of representatives of the Chau subgroup much more straight forward, it becomes as simple as choosing four elements of F_{2^n} which satisfy the relation $ad - bc = 1$. Therefore a procedure for generating an element of $SL(F_{2^n}, 2)$ is as follows:

1. Choose n bits uniformly at random and arrange them as an n-tuple. These are the standard basis coefficients of $\vec{a}$ where F_{2^n} is viewed as a vector space over F_2
2. If not all n bits are 0:
   (a) Choose two length n tuples of bits to represent $\vec{b}$ and $\vec{c}$
   (b) Set $\vec{d} = \frac{1+\vec{b}\vec{c}}{\vec{a}}$ where multiplication and division are defined for the field F_{2^n}
3. If all n bits are zero:
   (a) Choose a bit string for $\vec{b}$ uniformly at random from the set of bit strings which are not all 0
   (b) Set $\vec{c} = \vec{b}^{-1}$ where the inversion is defined in the field F_{2^n}
   (c) Choose $\vec{d}$ to be any length n bit string
4. Construct the matrix $\begin{bmatrix} \vec{a} & \vec{b} \\ \vec{c} & \vec{d} \end{bmatrix}$ This is an element of $SL(F_{2^n}, 2)$ which was chosen uniformly at random.

All that remains is to transform this 2×2 Special Linear matrix into a $2n\times2n$ Symplectic matrix to make available more familiar methods for dealing with Clifford circuits. The details of how this transformation works are available in [12] and are reviewed here for convenience.

Define the *field trace* function $T : F_{2^n} \to F_2$ as $T(a) = a^{2^0} + a^{2^1} + ... + a^{2^{n-1}}$. For any element of the extension field this function outputs 0 or 1 and it is a linear function in the sense that $T(a + b) = T(a) + T(b)$. Note that the high powers can be efficiently calculated via repeated squaring.

1. Construct an $n \times n$ matrix S such that $S_{i,j} = T(\vec{e_2}^{i+j-2})$ where $i, j = 1, ..., n$. Here $\vec{e_2}$ is the element of the field F_{2^n} that when viewed as an n-tuple over F_2 has a 1 in the second entry and zeros everywhere else.

2. Create a $2n \times 2n$ matrix such that the first n columns have column i as the concatenated bit strings for the output of the computation

$$\begin{bmatrix} \vec{a} & \vec{b} \\ \vec{c} & \vec{d} \end{bmatrix} \begin{bmatrix} \vec{e_i} \\ 0 \end{bmatrix} = \begin{bmatrix} \vec{a}\vec{e_i} \\ \vec{c}\vec{e_i} \end{bmatrix}$$

and column $n + i$ is the bit string for the output of the computation

$$\begin{bmatrix} \vec{a} & \vec{b} \\ \vec{c} & \vec{d} \end{bmatrix} \begin{bmatrix} 0 \\ \vec{e_i} \end{bmatrix} = \begin{bmatrix} \vec{b}\vec{e_i} \\ \vec{d}\vec{e_i} \end{bmatrix}$$

3. There is now a $2n \times 2n$ matrix, $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$, which is not symplectic. To make it symplectic apply the matrix S as follows.

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \rightarrow \begin{bmatrix} A & BS^{-1} \\ SC & SDS^{-1} \end{bmatrix}$$

1.5 Decomposing a Symplectic Matrix

Now there is the question of how to turn an element of $SL(2, F_{2^n})$ into a sequence of one- and two-qubit gates that can be implemented on a piece of hardware. This paper presents a new method for decomposing any symplectic matrix, not just elements of the Chau subgroup. This new method does not have any obvious relationship to previously existing decomposition methods. For now say that you have some $2n \times 2n$ symplectic matrix S that is composed of four $n \times n$ sub-matrices

$$S = \begin{bmatrix} A' & B' \\ C' & D' \end{bmatrix}$$

The primes on the sub-matrices are only there because the first step of the decomposition will swap some of the columns among the blocks, and it is those slightly altered blocks that will be used for the remainder of the algorithm.

1. As a first step, by slightly altering Lemma 6 of [10] it is possible to make A' have full rank by applying Hadamard gates to some of the qubits. In practice this can be done by row reducing A' and seeing which columns of A' are linearly dependent on the other columns. Applying Hadamard gates to the linearly dependent columns will make sure the upper-left block has full rank. Phrased another way, S has been factored so that

$$S = \begin{bmatrix} A' & B' \\ C' & D' \end{bmatrix} 1_{2n} = \begin{bmatrix} A' & B' \\ C' & D' \end{bmatrix} (H^I H^I) = \begin{bmatrix} A & B \\ C & D \end{bmatrix} H^I$$

Here the un-primed blocks A, B, C, D are by definition the blocks that are obtained by absorbing some of the Hadamard matrices into your symplectic matrix. The superscript I on the H indicates that Hadamard gates are applied to a set of indices $I = \{i_1, ..., i_k\}$ and A has full rank.

As a specific example, if you had a system of 3 qubits and A' had rank 2, it would require 1 Hadamard operator to make an A sub-matrix with rank 3. If, for example, performing Gaussian elimination on A' showed that column 2 was linearly dependent on columns 1 and 3 then the Hadamard operator would be on qubit 2. Furthermore you would have that the symplectic Hadamard matrix is equal to

$$H^{\{2\}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

This matrix would have the effect of swapping column 2 of A' with column 2 of B', as well as swapping column 2 of C' with column 2 of D'.

2. Now factor the matrix on the left as

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ CA^{-1} & 1 \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & A^{-T} \end{bmatrix} \begin{bmatrix} 1 & A^{-1}B \\ 0 & 1 \end{bmatrix}$$

The product of the three matrices on the left is symplectic by assumption and it turns out that each of the three factored matrices on the right are also symplectic. To prove that claim all that must be done is verify the four commutation relations

derived from $SJS^T = J$ are satisfied by all three matrices above, and that the three matrices multiply together to form the expected matrix. To begin, the relations are easy to verify for the middle matrix. For the left and right matrices the relations boil down to proving that CA^{-1} and $A^{-1}B$ are symmetric matrices. Using the fact that the original matrix is symplectic you can show

$$A^T C = C^T A \implies CA^{-1} = A^{-T}C^T = (CA^{-1})^T$$

and similarly for $A^{-1}B$ using the commutation relations for the inverse of a matrix

$$AB^T = BA^T \implies A^{-1}B = B^T A^{-T} = (A^{-1}B)^T$$

.

Thus, all of the commutation relations are satisfied so the three matrices are all symplectic. To show that the matrices multiply out to be the original matrix it suffices to show that $D = CA^{-1}B + A^{-T}$. Again, is most easily verified by using the commutation relations for the inverse of a symplectic matrix

$$CB^T + DA^T = 1_n \implies D = CB^T A^{-T} + A^{-T} \tag{1.1}$$

Substituting in the relation $B^T A^{-T} = (A^{-1}B)^T = A^{-1}B$

$$D = CB^T A^{-T} + A^{-T} = CA^{-1}B + A^{-T}$$

gives the required relation proving the claim.

3. Since CA^{-1} is a symmetric matrix, by an algorithm given in [10] or a slight variation given in [11] it is possible to find an invertible upper-triangular matrix U and a diagonal matrix L such that $CA^{-1} = UU^T + L$. Python code for doing this is given in the appendix. This decomposition leads to a further factorization of the symplectic matrix

$$\begin{bmatrix} 1_n & 0_n \\ CA^{-1} & 1_n \end{bmatrix} = \begin{bmatrix} 1_n & 0_n \\ UU^T + L & 1_n \end{bmatrix} = \begin{bmatrix} 1_n & 0_n \\ L & 1_n \end{bmatrix} \begin{bmatrix} U^{-T} & 0_n \\ 0_n & U \end{bmatrix} \begin{bmatrix} 1_n & 0_n \\ 1_n & 1_n \end{bmatrix} \begin{bmatrix} U^T & 0_n \\ 0_n & U^{-1} \end{bmatrix}$$

4. Similarly it is possible to decompose the third symplectic matrix as

$$\begin{bmatrix} 1_n & A^{-1}B \\ 0_n & 1_n \end{bmatrix} = \begin{bmatrix} 1_n & VV^T + K \\ 0_n & 1_n \end{bmatrix} = \begin{bmatrix} V & 0_n \\ 0_n & V^{-T} \end{bmatrix} \begin{bmatrix} 1_n & 1_n \\ 0_n & 1_n \end{bmatrix} \begin{bmatrix} V^{-1} & 0_n \\ 0_n & V^T \end{bmatrix} \begin{bmatrix} 1_n & K \\ 0_n & 1_n \end{bmatrix}$$

Where here K is a diagonal matrix and V is upper triangular.

5. Putting all of these together yields

$$S = \begin{bmatrix} 1_n & 0_n \\ L & 1_n \end{bmatrix} \begin{bmatrix} U^{-T} & 0_n \\ 0_n & U \end{bmatrix} \begin{bmatrix} 1_n & 0_n \\ 1_n & 1_n \end{bmatrix} \begin{bmatrix} U^T A V & 0_n \\ 0_n & U^{-1} A^{-T} V^{-T} \end{bmatrix} \begin{bmatrix} 1_n & 1_n \\ 0_n & 1_n \end{bmatrix} \begin{bmatrix} V^{-1} & 0_n \\ 0_n & V^T \end{bmatrix} \begin{bmatrix} 1_n & K \\ 0_n & 1_n \end{bmatrix} H^I$$

This can be interpreted as a chronological sequence of gates as follows:

1. If $L_{i,i} = 1$ apply a $\sqrt{X}$ gate on qubit i.
2. The block diagonal matrix containing U represents a set of CNOT gates which can be turned into a specific sequence of CNOT gates using an algorithm like the one given by Patel, Markov and Hayes [13].
3. Apply a $\sqrt{X}$ gate to every qubit.
4. This block diagonal matrix represents another set of CNOT gates.
5. Apply a $\sqrt{Z}$ gate to every qubit.
6. The block diagonal matrix containing V^{-1} is another set of CNOT gates.
7. If $K_{i,i} = 1$ apply a $\sqrt{Z}$ rotation to qubit i.
8. Finally, apply the Hadamard gates found in step 1 of the decomposition.

1.5.1 Generalization to Qudits

This algorithm can be easily extended to work for systems of odd-prime dimensional qudits, sometimes called quopits, as well. Keep in mind that if $d > 2$ the symplectic matrices are now over F_p and have the property that $SJS^T = J$ where

$$J = \begin{bmatrix} 0_n & -1_n \\ 1_n & 0_n \end{bmatrix}$$

.

Otherwise the algorithm is almost unchanged. The most important changes are detailed below.

1. On quopit systems the generalized versions of Hadamard gates have order 4 instead of order 2. Therefore it makes sense to absorb an H^3 into the symplectic to swap around the linearly dependent columns and physically implement only a single Hadamard at the end.

2. The factorization into three blocks is completely unchanged.

3. The factorizations of symmetric matrices CA^{-1} and $A^{-1}B$ into the forms $UU^T + L$ and $VV^T + K$ is easily done. The only subtlety is in remembering that over fields of characteristic $\neq 2$ you have $+1 \neq -1$ and $x^2 \neq x$ unless $x = 0, 1$.

   A more important difference is in the factorization of these blocks into CNOT stages and either $\sqrt{X}$ or $\sqrt{Z}$ gates. The factorization is still possible but care must be taken to understand what physical gates are needed. In the generalized case we can use (unitary) gates which are represented on one quopit by the symplectic matrices

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

   It is worth noting that these gates can be certain p^{th}-roots of the generalized Pauli X and Z operators respectively.

4. The handling of the new version of generalized versions of CNOT stages is also very different. First, CNOT is most generalized to be the SUM gate that has the effect $SUM(|a\rangle|b\rangle) = |a\rangle|b \oplus a\rangle$, where the addition is modulo p. Second, on quopits there is a new family of single quopit Clifford gate not available in the qubit case. Following [14] they will be denoted S_r for any $r \in \{1, \ldots, p-1\}$. As a $p \times p$ unitary matrix acting on computational basis states they have the effect $S_r|a\rangle = |ra\rangle$ and as a 2×2 symplectic matrix acting on vectors representing Pauli matrices has the form

$$S_r = \begin{bmatrix} r & 0 \\ 0 & r^{-1} \end{bmatrix}$$

   To see why both SUM and S gates are necessary recall how CNOT blocks are turned into sequences of gates in the qubit case. Applying a CNOT gate to two qubits a and b changes the symplectic matrix representing the Clifford gate by adding (modulo 2) column a to column b and adding column $b + n$ to column $a + n$. Therefore, it is possible to use this action of adding columns to each other to place 1s on the diagonals of the matrix and make all of the off-diagonal entries 0s. More specifically, start by taking looking at the transpose of the upper-left block, i.e. the A block, of the

symplectic. Perform Gaussian elimination on this matrix and record whenever one row is added to another row. Each of these additions corresponds to an application of a CNOT gate. So if all of those CNOTs are applied in order they would compile into a symplectic that computes A^{-1} since their action was to turn A into the identity matrix. To find the sequence of CNOTs that compiles to A simply reverse the order of gates.

Notice that since in the qubit case the only numbers that appear inside of the symplectic matrices are 0 and 1, so adding rows to each-other is sufficient to perform Gaussian elimination. In the quopit case numbers from 0 to $p-1$ would appear in the symplectic matrices. Therefore when performing a Gaussian elimination the steps where a row is multiplied by a constant mean an S_r gate is being applied for some value of r. As a final note making the gates compile to A instead of A^{-1} is slightly different in this case because $SUM \neq SUM^{-1}$ and for most r you will have $S_r \neq S_r^{-1}$ but there are of course ways around that.

1.6 Comparison of Decomposition Algorithms

This section compares three decompositions of the Clifford group to see how they perform based on total number of CNOT gates used and maximum depth of CNOT gates. [2] It is reasonable that only CNOT gates are compared because asymptotically the vast majority of the gates in a Clifford circuit over the {CNOT,Hadamard,Phase} gate library or the {CNOT,Hadamard,Phase,X_{90}} gate library will be CNOT gates. This is because in all the decompositions that are being compared the gates are arranged into layers containing just a single type of gate. A layer of CNOT gates may contain $O(n^2/\log n)$ gates, but any round of single qubit gates has at most n gates arranged to have depth 1. Thus, ensuring that CNOT circuits are handled in the same way is critical in determining if a comparison is fair, and it suffices to compare performance in terms of CNOTs to determine which decomposition is superior. To that end all of the CNOT rounds were decomposed into circuits by the same algorithm, which is a slightly improved version of the one given in [13].

[2] Readers should be aware that here CNOT depth is counted by tracking how many CNOT gates are applied to each qubit and taking the maximum of those numbers. Due to the non-commutative nature of CNOT gates this may be slightly lower than the depth of a circuit counted by other means.

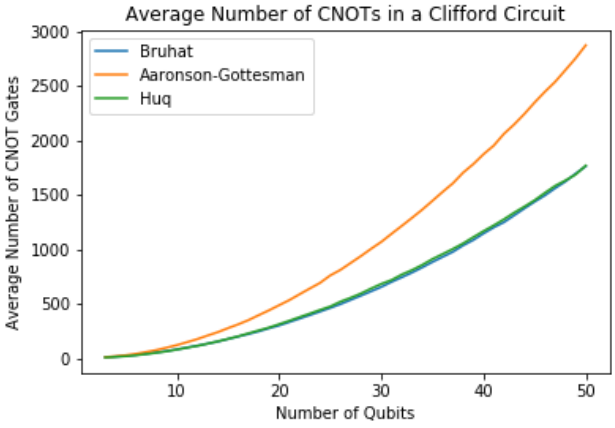

Figure 1.1: CNOT Gate Count vs Number of Qubits

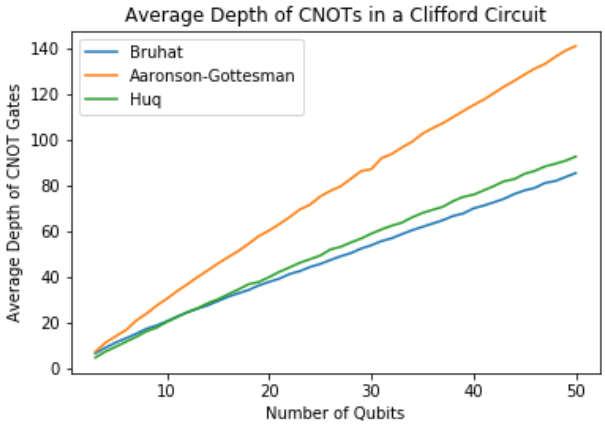

Figure 1.2: Maximum Gate Depth vs Number of Qubits

These plots were made by decomposing 100 randomly generated Symplectic matrices and taking their average number of CNOT gates and average maximum depth of CNOT gates. It is important to note that because the most prominent algorithm for generating symplectic matrices [15] requires users to input a random integer between 0 and the size of the symplectic group it quickly becomes difficult to use the algorithm because the size of the group exceeds the size of the seed of the random number generator. To get around this a suggestion from [11] was used, and the fact that the number of circuits with a Bruhat-decomposed form is almost exactly the number of Symplectic matrices. That allows you to make a Bruhat decomposed circuit that can be compiled to be re-decomposed using the other methods. As expected the average CNOT gate count scale like $\Theta(\frac{n^2}{\ln(n)})$ and the average maximum CNOT depths scale like $\Theta(\frac{n}{\ln(n)})$. The complexity of Maslov's Bruhat decomposition and the new decomposition have leading coefficients of $\approx 2.85 \pm 0.05$ while

the Aaronson-Gottesman decomposition has a leading coefficient of ≈ 4.5. The circuit depth scaling coefficients of the new, Bruhat and Aaronson-Gottesman decompositions are approximately 7.3,6.7, and 11 respectively. Readers may notice that there are no error bars to show the standard deviations of the averages. This is because the ratio of the standard deviation to the average of the total gate count goes like $\sigma_n/\mu_n \approx 1/2n$ and for the depth the ratio goes like $\sigma_n/\mu_n \approx 1/3n$. At this scale the error bars would be barely visible on the plots.

1.7 Computational Complexity

Even in its present form without much optimization for computational complexity the new algorithms provide some of the computationally cheapest methods for generating elements of a unitary 2-design and turning elements of the Clifford group into sequences of gates. To generate a 2×2 element of $SL(2, F_{2^n})$ which represents an element of the Chau subgroup requires at most $3n$ random bits for the symplectic part and $2n$ random bits for the Pauli part, which is significantly less than the approximately $2n^2 + 3n$ random bits needed to generate a uniformly random symplectic matrix using the algorithm of [15]. The finite field operations of adding, multiplying, dividing, and taking square roots can all be performed in time $O(n^2)$ or less [16]. If the matrix whose entries are the results of the field-trace function are pre-computed then the bottleneck in this algorithm is the matrix multiplications to turn the special-linear matrix into a symplectic matrix which can be performed in time $O(n^3)$. This matches the time-complexity of the algorithm in [15], however in practice the new decomposition presented here seems to have a smaller leading coefficient and is much more efficient on small inputs.

For the decomposition algorithms to turn symplectic matrices into sequences of gates, they all have roughly the same running time. All of them involve the use of linear algebra algorithms that have runtime $O(n^3)$ and the modified version of the Patel-Markov-Hayes algorithm that slightly optimizes the number of CNOT gates has runtime $O(n^4)$, making that the total worst case run-time of the algorithm.

1.8 Conclusion

A straight-forward method for generating elements of a group unitary-2 design, and a new and much simpler method for decomposing elements of the Clifford group was presented. The method for generating elements of the Chau subgroup is less computationally intensive

and requires fewer random bits than it takes to generate an element of the Clifford group using commonly known methods. The new decomposition of the Clifford group is much easier to understand than any previously existing decomposition and uses no math beyond basic linear algebra over a finite field, while also matching the previous best decomposition in terms of total gate count. It is now possible for researchers working on problems like randomized benchmarking and quantum key distribution to use the Chau subgroup in a plug and play manner.

Chapter 2

The Quantum and Classical Double Wells

2.1 Introduction

In any first course on quantum mechanics (QM) it is common to teach students about the Ehrenfest Theorem as a first example of the correspondence between Classical Mechanics (CM) and QM. On one hand this correspondence seems obvious because Ehrenfest Theorem says that for a 1D quantum particle

$$\frac{d\langle\hat{x}\rangle}{dt} = \frac{\langle\hat{p}\rangle}{m}$$
$$\frac{d\langle\hat{p}\rangle}{dt} = -\langle V'(\hat{x})\rangle \approx -V'(\langle\hat{x}\rangle)$$

Where the approximate equality holds in the case that the wave function is localized in a region where the potential function is approximately constant. In layman's terms quantum particles will follow Newton's laws if they are well localized. However, it is well known that the Ehrenfest Theorem is unnecessary and insufficient as a condition for quantum-classical transitions [17]. Other conditions like saying $\hbar$ is small with respect to the action of the system have some role in the path-integral formulation of QM but leads to other problems, particularly when a wave function contains expressions like $x/\hbar$ as in the case of a momentum operator eigenstate. Here we show that the failures of these correspondences between CM and QM are smaller if a different formulation of CM is used while simultaneously showing how intuition about simple quantum phenomena can be built out of knowledge of classical physics. Liouville mechanics is a generalization of CM that was developed in the late 19th century. Instead of representing the state of a system as a single point in a phase space, the state of the particle is described by a probability

distribution over the phase space variables. Examples are given of simple situations with surprising outcomes, and it turns out that the results of Liouville Mechanics (LM) can be closer to the results of QM than either one is to CM.

2.2 Liouville Mechanics

In 1D Classical Mechanics the position and momentum of a point particle can be described by a set of ODEs, Hamilton's equations:

$$\frac{dq}{dt} = \frac{\partial H}{\partial p}$$
$$\frac{dp}{dt} = -\frac{\partial H}{\partial q}$$

So it may seem strange to switch over to a view where classical systems are described by a less familiar and more complicated PDE, the Liouville equation

$$\frac{d\rho}{dt} = \Big(\frac{\partial H}{\partial p}\frac{\partial}{\partial q} - \frac{\partial H}{\partial q}\frac{\partial}{\partial p}\Big)\rho$$

It turns out this equation is just Hamilton's equations in disguise. This is a first order linear partial differential equation so the Method of Characteristics can be applied to find solutions. Once this is done you will find that the flow lines are just the solutions to Hamilton's equations applied at every infinitesimal region of phase space. Phrased differently, one can imagine the probability density function being like a cloud of infinitesimal particles scattered over the phase space. As they propagate in time they each flow around according to Hamilton's equations without interacting with each other.

As a quick and well-known application to demonstrate the power of the Liouville equation is in identifying solutions to $\frac{d\rho}{dt} = 0$. Assuming the solution has the form $\rho(q,p) = A(q)B(p)$ and separating variables will give $\rho \propto constant$ or $\rho \propto e^{-H/T}$, where T is a constant. These are closely related to the microcanonical ensemble and the canonical ensemble solutions from statistical mechanics [18].

2.3 Hamiltonian, Initial Conditions, and Numerical Methods

The setup for the numerics is as follows. The potential function is a simple quatric polynomial: $V(x) = x^4 - 32x^2$. This is the simplest potential function with a double-well possible.

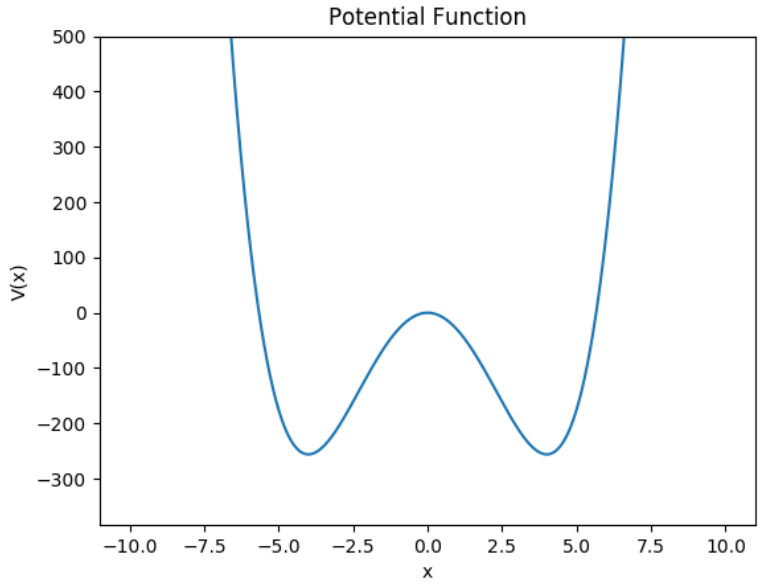

Figure 2.1: The potential function used in the simulations

The well is symmetric about $x = 0$ where it has a local maximum, the minimum values of the well are at $x = \pm 4$ and the other roots of the polynomial are at $x = 4\sqrt{2} \approx 5.66$.

In the CM case the initial conditions are chosen so that the particle starts at some point $x = \mu$ with its momentum $p = 0$. The QM initial state is a Gaussian wave-packet starting with $\langle x \rangle = \mu$ and $\langle p \rangle = 0$. The variance of the distribution is varied to compare and contrast with the other cases. The LM case is more subtle because the initial probability distribution is over both the x and p variables, so $\rho = \rho(x, p)$, so it was chosen to be a double Gaussian function with the initial variances being the same as the variances of the quantum operators. It is worth noting that this is one of the major differences between LM and QM: LM states can be arbitrarily sharply centered on regions of phase space while QM states automatically respect uncertainty relations like $\sigma_x \sigma_p \geq \hbar/2$.

The CM case uses a Runge-Kutta 4 algorithm to calculate the trajectory of a single classical particle. The LM case is only a small variation on the CM case. As stated earlier, when applying the method of characteristics the flow lines are just the trajectories of single particles. It becomes easy to sample initial conditions according to the initial condition of the PDE, which is a probability distribution, and simulate the trajectory of that particle. The data from all the trajectories can then be analyzed to determine how the moments of the position and momentum distributions are changing. The QM case uses the DVR method of [19].

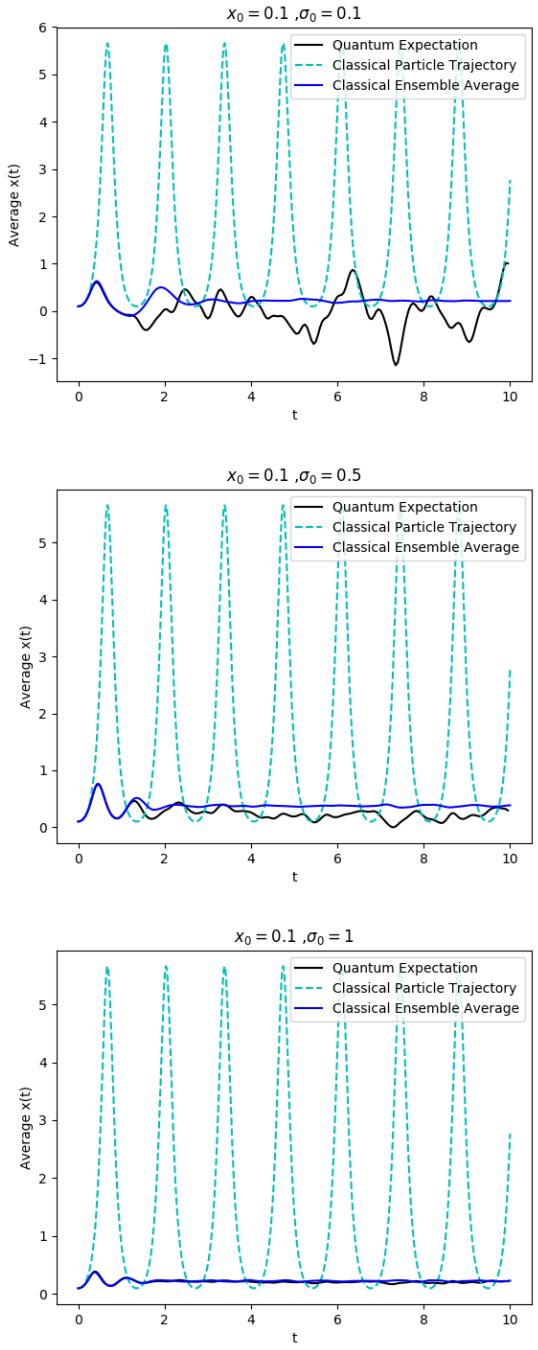

Figure 2.2: $\langle x(t) \rangle$ vs t for CM, LM and QM particles in a double well with $x_0 = 0.1$ and various initial widths of wave-packets

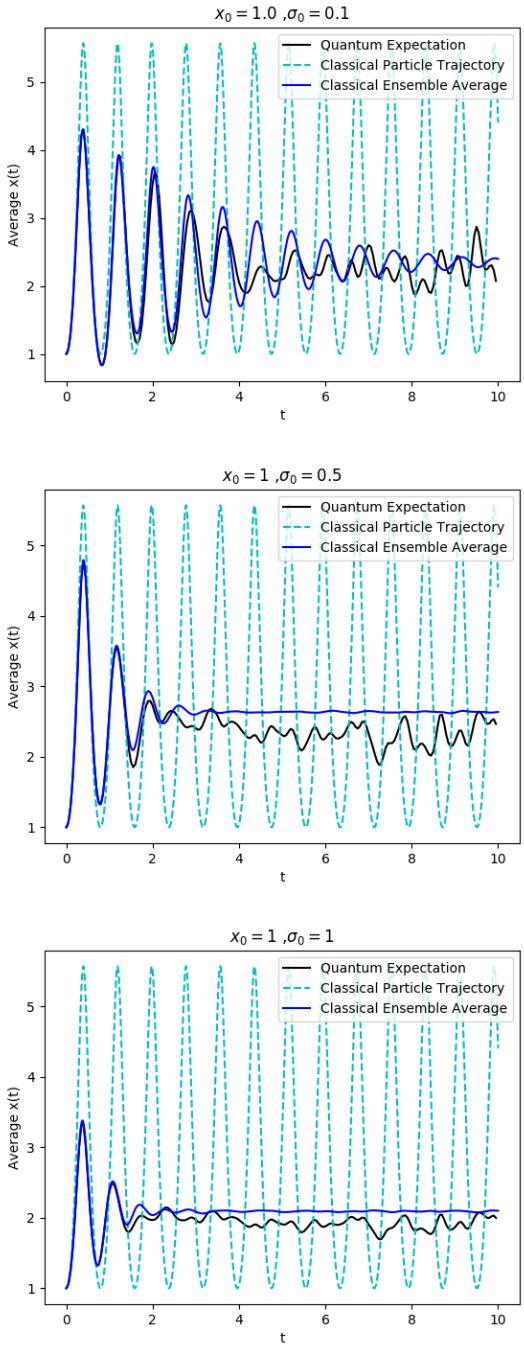

Figure 2.3: $\langle x(t) \rangle$ vs t for CM, LM and QM particles in a double well with $x_0 = 1.0$ and various initial widths of wave-packets

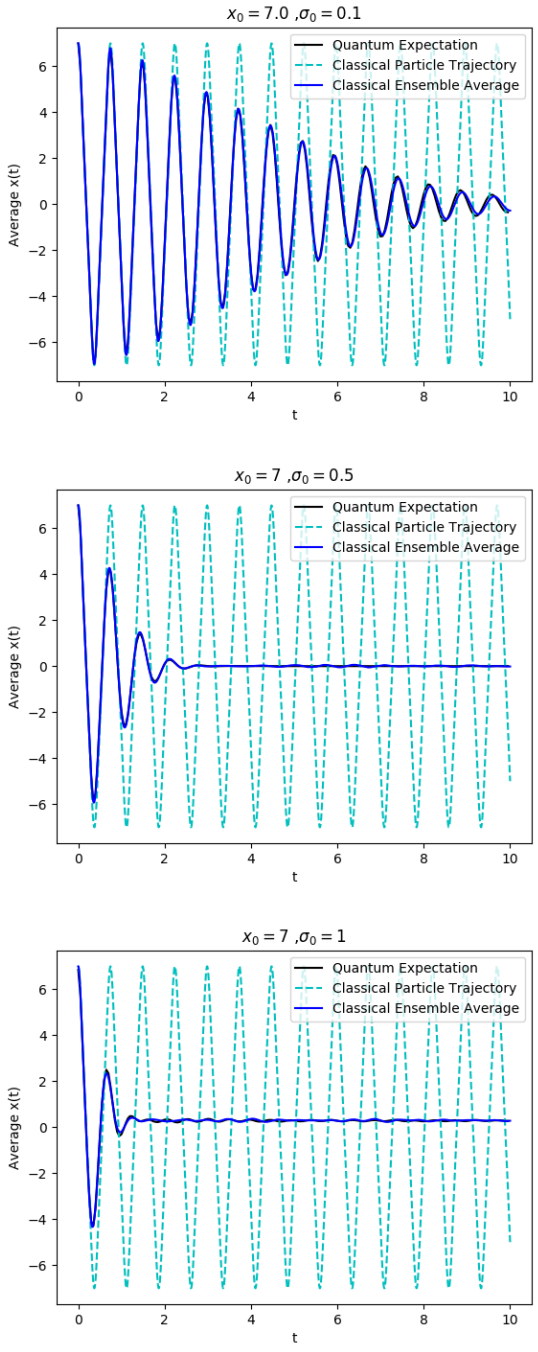

Figure 2.4: $\langle x(t) \rangle$ vs t for CM, LM and QM particles in a double well with $x_0 = 7.0$ and various initial widths of wave-packets

It is obvious from these plots that the QM and LM results correlate with each other much more strongly than either correlates with the CM particle solution. This should draw contrast with previous works in the field like [20] which focus on when the differences between LM and QM grow the fastest. As for the specific situations, the erratic behaviour in the first row is because the wave-packet begins by straddling the central local maximum of the potential. It seems like interference effects in the quantum case cause the erratic behaviour of the mean position to undergo strong fluctuations that do not occur in the Liouville case. In the second row the oscillation rate of the LM and QM case are slightly different until the interference effects take over and the motions become essentially uncorrelated. The author does not have an explanation for these different oscillation rates. The third row shows essentially perfect correlation between the LM and QM cases, along with the damping effects that are given a theoretical explanation in a following section.

2.4 The Origin of the Similarities

In essence the similarities between the QM and LM cases can be traced back to a (slightly generalized) version of the Ehrenfest theorem [20]. Both cases satisfy the equation

$$\frac{d\langle\hat{x}\rangle}{dt} = \frac{\langle\hat{p}\rangle}{m}$$
$$\frac{d\langle\hat{p}\rangle}{dt} = -\langle V'(\hat{x})\rangle$$

Where one must be careful to distinguish how the expectation values are calculated in the QM and LM cases. Then if V is expanded as a power series about the expectation value of the position operator you find

$$\begin{aligned}\frac{d\langle\hat{p}\rangle}{dt} &= -\langle V'(\langle\hat{x}\rangle) + V''(\langle\hat{x}\rangle)(x - \langle\hat{x}\rangle) + \frac{1}{2}V'''(\langle\hat{x}\rangle)(x - \langle\hat{x}\rangle)^2 + \cdots\rangle \\ &= -V'(\langle\hat{x}\rangle) - \frac{1}{2}V'''(\langle\hat{x}\rangle)\langle(x - \langle\hat{x}\rangle)^2\rangle - \cdots\end{aligned}$$

The Ehrenfest theorem in QM derived by assuming the wavefunction is well localized around the expectation value and only keeping the first term based on that justification. But notice that these equations are equally true in LM as in QM, so the Ehrenfest theorem and its next order perturbation are both true QM and LM. This does leave the question, where do the theories diverge? First recall that although the same symbols can be used for the two theories they do have different meanings when performing calculations. In LM since the state is represented by a probability distribution the expectations are calculated

as $\langle\hat{x}\rangle = \int\int x\rho(x,p)dxdp$ and $\langle\hat{p}\rangle = \int\int p\rho(x,p)dxdp$ whereas in QM they are $\langle\hat{x}\rangle = \int x|\psi(x)|^2dx$ and $\langle\hat{p}\rangle = \int \psi^*(x)(-i\frac{d}{dx})\psi(x)dx$. These differences accumulate over time, but there is a more fundamental issue at play. Notice that because there is no formula for how the higher order moments of the distributions evolve in time these formulas only describe what happens very close to the starting time. However, it is possible to find those higher order formulas and replace the PDEs of QM and LM with infinite sets of ODEs. In both cases the ODEs couple the moments of the position and momentum distributions $\langle\hat{x}^n\rangle$, $\langle\hat{p}^m\rangle$, with mixed terms like $\langle\hat{x}^k\hat{p}^l + \hat{p}^l\hat{x}^k\rangle$ and $\langle i(\hat{x}^k\hat{p}^l - \hat{p}^l\hat{x}^k)\rangle$. In the quantum case the position and momentum operators do not commute and factors of $\hbar$ appear explicitly, first in the $\langle p - \hat{p}\rangle^3$ term [20] and the sets of equations look more different the higher the order of the moment. It is interesting to note that these moments of the position and momentum distribution are the actual observable quantities in an experiment and the equations of motion for a system can be written in terms of them.

2.5 Equilibration, Settling location

It is quite obvious that the CM solution will never stop oscillating, and very obvious that the oscillations of $\langle x\rangle$ quickly die out in the QM and LM cases based on the numerics. However one should keep in mind that those are just the expectation values of the position operator that are settling down, and in fact the reversible nature of the flows stops the quantum wave function and classical state distribution from reaching a time independent state. Nevertheless, it is obvious from the numerics that $\langle\hat{x}\rangle$ does approach a constant in the LM and QM cases, but the constant value is not the bottom of the well. There is a simple explanation in the case of a double well: there is a probability that the particle can be found in the other well, so that moves the expectation value closer to the other well. In fact, there is a more general explanation to both the fact that equilibration happens in wells and the equilibration point is not the bottom of the well. Since

$$m\frac{d^2\langle\hat{x}\rangle}{dt^2} \approx -V'(\langle\hat{x}\rangle) - \frac{1}{2}V'''(\langle\hat{x}\rangle)\langle(x - \langle\hat{x}\rangle)^2\rangle = F(\mu) - \frac{\sigma}{2}F''(\mu)$$

Where for simplicity the substitutions $\langle\hat{x}\rangle = \mu$ and $\langle(x - \langle\hat{x}\rangle)^2\rangle = \sigma$ are used.

Consider the case that the particle is in a potential well centered around $x = 0$ with steep sides. From basic physics we know that a CM particle released on a steep side of the well would undergo oscillations indefinitely in the well. Thus a plot of the force on a localized particle would show that the force points towards the centre of the well and

there is nothing else to consider. For the LM and QM particles the next order correction to the Ehrenfest theorem shows that the motion of the expectation value has a correction factor that is proportional to the product the concavity of the force function at the mean position and the variance of the position distribution. More specifically since $\sigma > 0$ and the steep walls condition implies that $F''(x) > 0$ on the left wall and $F''(x) < 0$ on the right wall. The correction term will always impede the LM and QM motions compared to the CM motion while the particle is localized on a steep wall of a potential well. Therefore, this impedance is much like a frictional force and will cause the mean positions of LM and QM particles to equilibrate in potential wells. Furthermore the equilibration will happen faster the more the wave-packet is spread out. This is well demonstrated in the plot from the third row of Figure 2.1.

As to the location of the equilibrium position consider what would happen to the position expectation value of a particle placed at a point in an asymmetric potential well such that $F(\mu) = 0$. Then, the first order term corresponding to the Ehrenfest theorem would vanish and only the higher order terms would remain. The well is specified to be asymmetric to imply that the concavity of the force function is non-zero at the bottom of the well. Thus, the mean value of the position would move slightly away from whichever side of the well is steeper. This is easier to understand intuitively in the LM case than the QM case. Imagining the LM wave-packet as a swarm of non-interacting classical particles you can predict that particles will be traveling faster on the steep side of an asymmetric well and slower on the shallower side of the well. That would imply that each particle spends more time on the shallow side so that on average the wave-packet is moved slightly towards the shallow side. This matches perfectly with what the mathematics of the extended Ehrenfest theorem predicts.

2.6 Unitarity and its Consequences

Although there are obvious similarities in the plots of the QM and LM particles a small mystery remains: these physical theories have different mathematical substructures but produce similar results. Non-relativistic quantum theory is based on the Schrdinger equation, a linear PDE that is first order in time and second order in position, describing the propagation of a complex 2-norm normalized wave-function by a unitary operator. On the other hand Liouville Mechanics is based on the Liouville equation, a linear PDE that is first order in time, position, and momentum, acting on a 1-norm normalized probability distribution in a manner which seems more complicated. There is a way to bring them a

little closer in form, by describing Liouville equation in unitary form [21]. It turns out that the formulas for the propagation of ρ and $\sqrt{\rho}$ are the same.

$$\frac{d\sqrt{\rho}}{dt} = \Big(\frac{\partial H}{\partial p}\frac{\partial}{\partial q} - \frac{\partial H}{\partial q}\frac{\partial}{\partial p}\Big)\sqrt{\rho}$$

This means the Liouvillian now acts on a square-integrable function and preserves the square-integrable nature of the initial condition. A calculation shows that the Liouvillian is currently an anti-Hermitian operator, so by multiplying both sides of the equation by $-i$ you find:

$$i\frac{d\sqrt{\rho}}{dt} = i\Big(\frac{\partial H}{\partial p}\frac{\partial}{\partial q} - \frac{\partial H}{\partial q}\frac{\partial}{\partial p}\Big)\sqrt{\rho}$$

So now classical phenomena in LM can be described by a unitary operator propagating forward a complex square-integrable function, much closer to the quantum case. This can also provide an answer to questions that many students must have, if physical transformations are inherently continuous on the initial conditions of the system how can it be that particles with only tiny differences in their initial conditions can undergo very different evolutions over long time periods? The double well provides a simple example. Since the roots of the potential function are at $-4\sqrt{2}$,0,$4\sqrt{2}$ if a CM particle with initial momentum $p_0 = 0$ has an initial position $0 < q_0 < 4\sqrt{2}$ it will be trapped in the right well. If the particle has $q_0 > 4\sqrt{2}$ it will be able to jump over the centre of the double well and undergo a more complex oscillation. So in CM it is possible that two particles with arbitrarily close initial conditions will still undergo very different long term behavior. It is worth noting that Chaos Theory is not at work here because the CM particle's state is described by a time-independent ODE on a two dimensional phase space and proper chaotic behavior requires at least a three dimensional phase space or a two dimensional space with time-dependence.

So what happens with the QM and LM cases? For the QM case it is well known that unitary evolution forces $\langle\psi(0)|\phi(0)\rangle = \langle\psi(t)|\phi(t)\rangle$ for all times t and initial conditions $|\psi(0)\rangle$ and $|\phi(0)\rangle$. This can be interpreted as saying that states which begin very close together remain very close together for all time.The unitary version of the Liouville equation forces a similar constraint in the LM picture. The situation can be seen in the plots below.

As expected the CM particle undergoes a very different oscillation in the two cases even though the difference in their starting point is 0.002. In contrast the LM and QM particles

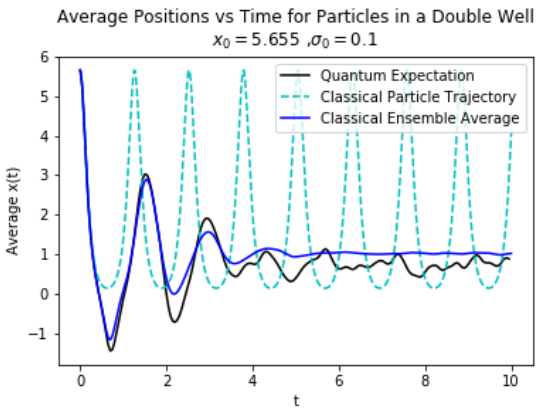

Figure 2.5: $\langle x(t) \rangle$ vs t for particle below a critical point

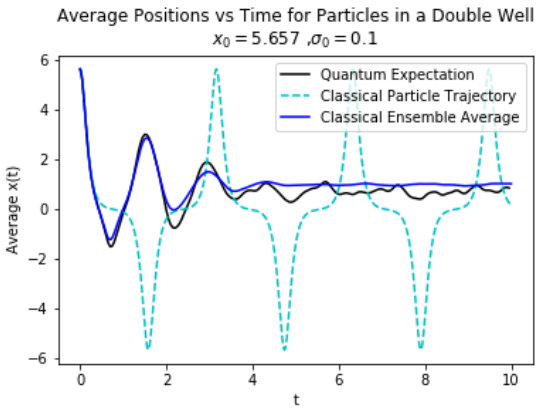

Figure 2.6: $\langle x(t) \rangle$ vs t for particle above a critical point

undergo essentially identical motion in both cases because their initial conditions are so close to each other.

Before readers think that QM and LM are practically the same there are some caveats. The Liouvillian operator itself is not an observable of the system so it cannot be given a simple physical interpretation. Also, the set of eigenvalues of the Liouvillian may extend to $-\infty$, whereas the set of eigenvalues of a quantum Hamiltonian is always bounded below. These mathematical differences become important for more complicated systems. The allowable initial conditions for the Liouville equation are always probability distributions, but it is known that there are quantum wavefunctions for which the moments of position and momentum operators cannot be described as moments of a classical probability distributions.

2.7 Conclusion

For many decades now physicists have been struggling to figure out which "quantum" phenomena truly have roots in QM and which have alternative, classical descriptions. The body of knowledge on these subjects is now large enough that it can be daunting to students of physics, many of whom will not explore the topic to learn that probabilistic outcomes and unitary evolution have places in classical mechanics. This section's intent is to elucidate how classical and quantum phenomena can be more difficult to distinguish once Liouville mechanics is taken into account. Furthermore some examples were given to show why it can be useful to study Liouville mechanics to better understand the structure of classical mechanics.

Chapter 3

A Discussion on Connections Between Quantum Prover Interactive Proofs and Randomized Benchmarking

3.1 Introduction

With the current size and growth rate of the fields under the umbrella of Quantum Information Science with their disparate goals and techniques it should come as no surprise that there are when distinct sub-fields can face similar issues and resolve them in similar ways. This of course leads to the possibility of a cross pollination between sub-fields lending new techniques to researchers on both sides. This speculative section elucidates some connections between Quantum Prover Interactive Proofs (QPIPs) and error characterization (EC), in particular Randomized Benchmarking (RB). However, this section will not be a fully rigorous introduction to either of those fields.

The motivation behind RB and its family of protocols is as follows: Given a quantum information processing device which may or may not be influenced by noise processes, how can users detect the presence of noise in polynomial time, and what information can users gain about the noise? It is possible to phrase the motivation for QPIPs in a similar way: If one person (called the Prover) claims to have a quantum information processing device which can run polynomial-time quantum algorithms, how can another person (called the Verifier) determine that the computer works as advertised without alterations, whether they be intentional (deceit) or not (errors). Furthermore QPIPs are often assosciated with Verifiable/Blind Quantum Computing, meaning the Verifier wishes to detect if the Prover

is eavesdropping on the computation and to ensure that the Prover does not gain any information about the computation.

These descriptions of course gloss over some differences in the design philosophies of the techniques. QPIPs have strong ties to blind quantum computing because researchers in that field prepare for the case that the Prover is malicious. More importantly, most QPIPs are designed on a highly theoretical level so certain practical concerns are ignored which leads to expensive protocols that are guranteed to work. In contrast, EC protocols are almost exclusively created to test for faults on currently available or near-term hardware so there is has been a great effort to find protocols that have feasible resource requirements, with the possibility some slightly more expensive protocols never see serious consideration in the community.

3.2 Closest Points of Contact

The simplest way to see the connection between RB and QPIPs is in the standard RB protocol [4] and A. Broadbent's QPIP based on repeated runs [22]. The protocol for RB can be summarized as follows:

1. Choose M circuit lengths $l_1, \ldots, l_M$
2. For each circuit length l_i choose $N > 30$ circuits composed of l_i elements of a group unitary 2-design that compose to an identity circuit. Most typically, these are elements of the Clifford group.
3. Run each circuit on an initial state $|0\rangle^{\otimes n}$ enough times to estimate the probability that at the end of the identity circuit the state is still $|0\rangle^{\otimes n}$
4. Fit all of the return probabilities for each value of l_i to an exponential decay $Pr(success) = Ap^l + B$. Under a certain set of reasonable assumptions the decay curve is guaranteed to have this form.
5. The decay parameter p can then be interpreted as giving an estimate of the strength of the noise in the system.

This may seem rather strange when viewed as an algorithm because RB would always return a string of zeros when run on a perfect machine. The point is that on an imperfect machine mistakes will be made. Thus, checking to see how long it takes for those mistakes

to be made on some sort of standard problem can allow you to estimate the size of the largest computation an imperfect machine can perform.

With this in mind Broadbent's Repeated-Run QPIP can be summarized as:

For the duration of the protocol implement certain gates through gate gadgets provided in the original source [22]. The gadgets involve having the Verifier send a quantum state to the Prover who then performs a certain entangling operation on the sent qubit and one of his own qubits and reports the result of a measurement on one of the qubits to the Verifier. The state of the sent qubit determines if a T-gate or a Pauli operator is applied to the Prover's state. The outcome of the measurement is deterministic and depends on the sent state and the Prover's state. The purpose of the gate gadgets is to make it so that the Prover does not always know if Hadamard or T gates are being applied to the state, while also allowing the Verifier to detect deviations from the protocol.

Now implementing certain gates via the gate gadgets, randomly choose to do one of the following three operations:

1. Perform a long Clifford circuit that composes to be an identity circuit on the standard input $|0\rangle^{\otimes n}$ and check that the state remains in the state $|0\rangle^{\otimes n}$ at the end of the circuit.

2. Perform a long Clifford circuit that composes to be an identity circuit on the input $|+\rangle^{\otimes n}$ and measure the state at the end.

3. Perform the circuit for some desired computation.

Repeat the previous step as necessary.

If any deviations are detected the Verifier can abort and hope that the random outcomes of RB run on $|+\rangle^{\otimes n}$ states will obfuscate the results of the computation.

When written in these terms the connection between the Broadbent QPIP and RB is unmistakable. Broadbent did not use the term Randomized Benchmarking in her original paper, or attempt to draw connections between QPIPs and Randomized Benchmarking. Therefore the following two subsections will give descriptions of various error characterization protocols and how they may be adapted into QPIPs, and some other QPIPs that may be adapted into error characterization protocols.

3.3 QPIPs with Applications of Error Characterization

3.3.1 The Reichardt, Unger, Vazirani Protocol

There is a QPIP protocol developed in [23] based on the rigidity of the CHSH game. Essentially, the rigidity of the CHSH game means that if two people are playing the CHSH game and their win rate is sufficiently close to the maximum possible win rate of $\cos^2(\pi/8)$ then the strategy they are using must be approximating the well known optimal strategy. This could be useful because it is still difficult to detect SPAM errors in sub-exponential time and the rigidity of the CHSH game could provide a way to ensure state preparations and measurements are aligned. This QPIP uses one Verifier interacting classically with two (or more) Provers who are not allowed to interact except for being able to perform operations on shared halves of Bell states. Say the two Provers are Alice and Bob, they share many copies of entangled Bell singlet states. From there the Prover sends out instructions to randomly perform one of four options:

1. Perform a round of the CHSH game. The Prover does this to verify that the states and measurements used by Alice and Bob are approximately the proper ones (up to local isometries)

2. Bob is told to perform a certain set of measurements that collapse Alice's states into a set of resource states. Alice is told to perform the measurements that would be used to play the CHSH game. Thus, Alice's measurements in the X and Z bases can be used to perform state tomography on the resource states. This ensures that Alice's state preparations are correct.

3. Alice is told to perform the measurements that collapse the Bell pairs into resource states. Bob is told to perform Bell basis measurements on pairs of qubits. Bob's measurement data can be analyzed to perform process tomography on Alice's measurements. This ensures that the measurements are correct.

4. Alice is told to perform the resource preparation measurements and Bob is told to perform the Bell basis measurements. This makes Alice perform a desired computation through repeated gate teleportation.

As long as Alice and Bob cannot communicate, they cannot be sure which of the four cases has been assigned, and as the Verifier can check the outcomes of their results in three of the four cases there is little to no room for Alice and Bob to diverge from the

protocol. It should be noted that the complexity of this QPIP increases very quickly as the tomography stages are very expensive and the CHSH game rigidity results are somewhat weak and require many repetitions. This could be adapted into an error characterization protocol as follows. First, rethink the foundation of the protocol. Instead of having two seperate quantum computers and one Verifier, say you have one computer with the qubits partitioned into two sets of two qubits. From there play the CHSH game to verify that the win rate is sufficiently close to the maximal value. If that test passes have one set of qubits perform Alice's role and the other Bob's. Perform the the second test and if that passes perform the third. This should verify that the preparations, operations, and measurements are aligned and functioning properly.

3.3.2 The Mahadev QPIP

Readers will probably notice that some of the previous QPIPs all involve multiple parties with access to quantum information processing devices. It was an open question for multiple years whether a Verifier with only a **BPP** computer could send and receive classical messages with a quantum Prover to perform and certify the correctness of a quantum computation. The problem was of enough interest that a cash prize being offered by S. Aaronson for a result one way or the other. A result strong enough for Aaronson to award the prize money was discovered by U. Mahadev [24]. Hence, it is worth asking if the Mahadev QPIP can be re-purposed into a new type of error characterization protocol which is significantly different from all that have come before. However, upon reading her paper you can see that Mahadev *started* by creating an error characterization protocol and extending that to be a QPIP.

A concise description of the QPIP protocol is difficult as the protocol depends techniques from cryptography that are difficult to summarize. At its heart is the idea that the Verifier has access to two different families of functions which the Prover cannot distinguish. The functions have the "trap-door" property which means they are difficult for the Verifier to invert but the Prover has access to a "key" which makes it simple to invert the functions. The Verifier can then ask for the Prover to perform measurements on quantum states and send back the results. One family of functions can detect deviations from the protocol and allow the Verifier to abort. The other family can be used to make the Prover perform measurements of a local Hamiltonian. Through a well known result in complexity theory, any problem in BQP can be reduced to determining if the ground state energy of a local Hamiltonian is above or below certain values. Thus, by sending and receiving classical information a Verifier can interact with a Prover to perform a quantum computation and

abort if there are any deviations from the protocol.

There is a catch in that there are currently no exact constructions of the families of cryptographic functions. In their place Mahadev proves that there are approximate constructions based on the Learning With Errors problem that suffice and would allow the protocol to be carried out. Furthermore, there are no estimates of what computational resources would be needed to perform this protocol. Mahadev proves that the quantum resources scale polynomially in the size of the quantum computation but there are no detailed estimates.

3.4 Error Characterization Methods with Applications to QPIPs

The most straight-forward way to apply error detection techniques is to use variants of Randomized Benchmarking within Broadbent's QPIP protocol. The apparent point of view in the QPIP community is that errors occur because of malicious Provers who will interfere with unsuspecting Verifiers, and thus any interference is unacceptable. On the other hand error characterization techniques are made with the idea that any number of things can go wrong in a laboratory, some of which are more problematic than others. Thus, it becomes worthwhile to study ways of detecting specific types of noise. Combining these ideas Verifiers who wish to perform quantum computations but find themselves with few options on the free market may need to determine if the outcomes of their computations are trustworthy by checking the amount of noise in the system and performing tests to see if the noise is likely to be caused by malicious eavesdroppers or natural sources. Some of the most useful variants of Randomized Benchmarking to include in the Broadbent protocol could be:

1. Unitarity RB [25]. Estimates the coherence of the quantum noise. Could help to distinguish things like dephasing and control error noise. Depending on how technology advances in the future the coherence of noise may give important clues as to the source of the noise.

2. Leakage/Loss RB[26]. Detects if information is leaving the system into other degrees of freedom. Users can then decide if the amount of information leakage is acceptable then decide to abort if necessary.

3. Interleaved RB [27]. Tests to see if one particular gate in the set of gates available has a different error rate from the rest. May indicate control errors on one specific gate or may indicate that one particular gate is a target for eavesdropping.

4. Logical RB [28]. Randomized benchmarking performed on error corrected systems. Discrepancies between the results of logical RB and other error detection methods may hint that the error correction operations are being attacked. It should be noted that between the first two versions of their paper Broadbent et al retracted their claim that their QPIP is fault tolerant. It is not clear that LRB in its current form remedies their concerns.

There are of course many other variants of RB and many other error characterization protocols which could be used within the Broadbent protocol. Admittedly it is probably too preemptive to worry about quantum eavesdroppers and defenses against them while still in the early stages of the NISQ era when it is still impossible to guess what techniques for eavesdropping on computations will be used in the future.

3.5 Conclusion

With the size and growth rate of the field of quantum information it will be important to keep the distantly related parts of the community in contact so that there is a minimal amount of time and energy spent running over the same ground. It is the belief of the author that the tools and techniques of the error characterization community and the QPIP community are of value to each other and will be of great mutual benefit. The error characterization community could benefit from more rigorous mathematical analysis of protocols, while the QPIP community could benefit from taking more concern over the practical side of running various protocols.

References

[1] D. Gottesman. An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation. *ArXiv e-prints*, April 2009.

[2] Simon Anders and Hans J. Briegel. Fast simulation of stabilizer circuits using a graph-state representation. *Phys. Rev. A*, 73:022334, Feb 2006.

[3] H.f. Chau. Unconditionally secure key distribution in higher dimensions by depolarization. *IEEE Transactions on Information Theory*, 51(4):14511468, 2005.

[4] Easwar Magesan, Jay M. Gambetta, and Joseph Emerson. Characterizing quantum gates via randomized benchmarking. *Phys. Rev. A*, 85:042311, Apr 2012.

[5] Leonard Susskind. Computational complexity and black hole horizons. *Fortschritte der Physik*, 64(1):2443, 2016.

[6] Adam Nahum, Jonathan Ruhman, Sagar Vijay, and Jeongwan Haah. Quantum entanglement growth under random unitary dynamics. *Phys. Rev. X*, 7:031016, Jul 2017.

[7] Winton Brown and Omar Fawzi. Decoupling with random quantum circuits. *Communications in Mathematical Physics*, 340(3):867900, 2015.

[8] Christoph Dankert, Richard Cleve, Joseph Emerson, and Etera Livine. Exact and approximate unitary 2-designs and their application to fidelity estimation. *Physical Review A*, 80(1), Jun 2009.

[9] D. Gross, K. Audenaert, and J. Eisert. Evenly distributed unitaries: On the structure of unitary designs. *Journal of Mathematical Physics*, 48(5):052104, 2007.

[10] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70:052328, Nov 2004.

[11] Dmitri Maslov and Martin Roetteler. Shorter stabilizer circuits via bruhat decomposition and quantum circuit transformations. *IEEE Transactions on Information Theory*, 64(7):47294738, 2018.

[12] Richard Cleve, Debbie Leung, Li Liu, and Chunhao Wang. Near-linear constructions of exact unitary 2-designs. *Quantum Info. Comput.*, 16(9-10):721–756, July 2016.

[13] Ketan N. Patel, Igor L. Markov, and John P. Hayes. Optimal synthesis of linear reversible circuits. *Quantum Info. Comput.*, 8(3):282–294, March 2008.

[14] Bernhard Keller. On the cyclic homology of ringed spaces and schemes. 1998.

[15] Robert Koenig and John A. Smolin. How to efficiently select an arbitrary clifford group element. *Journal of Mathematical Physics*, 55(12):122202, 2014.

[16] Darrel R. Hankerson, Alfred J. Menezes, and Scott Vanstone. *Guide to elliptic curve cryptography*. Springer, 2010.

[17] L. E. Ballentine, Yumin Yang, and J. P. Zibin. Inadequacy of ehrenfest's theorem to characterize the classical regime. *Phys. Rev. A*, 50:2854–2859, Oct 1994.

[18] R. K. Pathria and Paul D. Beale. *Statistical mechanics*. Elsevier BH, 2011.

[19] Daniel T. Colbert and William H. Miller. A novel discrete variable representation for quantum mechanical reactive scattering via the smatrix kohn method. *The Journal of Chemical Physics*, 96(3):1982–1991, 1992.

[20] L. E. Ballentine and S. M. McRae. Moment equations for probability distributions in classical and quantum mechanics. *Phys. Rev. A*, 58:1799–1809, Sep 1998.

[21] Asher Peres and Daniel R. Terno. Hybrid classical-quantum dynamics. *Phys. Rev. A*, 63:022101, Jan 2001.

[22] Anne Broadbent. How to verify a quantum computation. 2015.

[23] Ben W. Reichardt, Falk Unger, and Umesh Vazirani. A classical leash for a quantum system: Command of quantum systems via rigidity of chsh games, 2012.

[24] Urmila Mahadev. Classical verification of quantum computations, 2018.

[25] Joel Wallman, Christopher Granade, Robin Harper, and Steven T. Flammia. Estimating the coherence of noise. 2015.

[26] Joel J. Wallman, Marie Barnhill, and Joseph Emerson. Characterization of leakage errors via randomized benchmarking. 2014.

[27] Easwar Magesan, Jay M. Gambetta, B. R. Johnson, Colm A. Ryan, Jerry M. Chow, Seth T. Merkel, Marcus P. da Silva, George A. Keefe, Mary B. Rothwell, Thomas A. Ohki, Mark B. Ketchen, and M. Steffen. Efficient measurement of quantum gate error by interleaved randomized benchmarking. *Phys. Rev. Lett.*, 109:080505, Aug 2012.

[28] Joshua Combes, Christopher Granade, Christopher Ferrie, and Steven T. Flammia. Logical randomized benchmarking, 2017.

Appendix A

Appendix

A.1 Subroutines

In this section some extra information for some of the operations used in this paper will be provided.

A.2 Decomposing a Symmetric Matrix

Let A be a symmetric $n \times n$ matrix over the two-element field. Then there exists a unique triangular matrix M and a unique diagonal matrix L such that $A = MM^t + L$. This is lemma 10 of [11] which is a minor variation on lemma 7 of [10]. This pseudo-code is based on Python.

```
1  def FindMLdecomp(A):
2      """
3      Takes in a square matrix A over F_2 and returns a lower ...
           triangular matrix M
4      and a diagonal matrix L such A = M M^T + L
5
6      """
7      n = len(A)
8      M = eye(n,dtype=int) #create identity matrix of size n
9      L=zeros([n,n],dtype=int) #create an nxn zero matrix
10
11     for i in range(1,n):
```

```
12          M[i,0]=A[i,0]
13
14      for i in range(1,n):
15          for j in range(1,i):
16              x=0
17              for k in range(j):
18                  x = (x + M[i,k]*M[j,k])%2
19              M[i,j] = (A[i,j]+x)%2
20
21      Aprime = (M@M.transpose())%2
22
23      for i in range(n):
24          if Aprime[i,i] != A[i,i]:
25              L[i,i]=1
26
27      return M,L
```

A.3 Decomposing a CNOT Circuit

At multiple stages in all the of the circuit decompositions mentioned in this article there are multiple points where users are told to turn a symplectic matrix into a sequence of CNOT gates. Say that the symplectic matrix you wish to decompose is $\begin{bmatrix} A^T & 0 \\ 0 & A^{-1} \end{bmatrix}$. Then recalling from [10] how a CNOT gate transforms the top left block of a symplectic matrix representing a Clifford circuit, it essentially adds one column of A^T to another column of A^T based on which qubit is the control and which the target. Recall also that Gaussian elimination algorithms add rows of a matrix to each other until the matrix is an identity matrix. Since the matrix in question is over F_2 there is no need to worry about multiplying rows by scalars. So, to turn the symplectic into a sequence of CNOTs enter the matrix A into a Gaussian Elimination algorithm and keep track of which row operations are performed, adding row i to row j corresponds to performing a CNOT with control qubit i to target qubit j. Note that since the sequence of gates found if applied in the order they were found turns the matrix A into an identity, that means the circuit for A^{-1} was found, therefore the circuit for A is obtained by reversing the order of the gates.

Using Gaussian elimination like this would generate circuit with $O(n^2)$ CNOT gates and it is possible to do better. In [13] an algorithm is given which in the average case uses $\Theta(n^2/\log n)$ CNOT gates by taking in an $n \times n$ matrix and an integer parameter m with $1 \leq m \leq n$. It was shown that taking $m \approx \log_2(n)$ is optimal. The decomposition

algorithm used for this paper's plots simply start with $m = 1$ and increment it until the best circuit which uses the fewest gates is found, sacrificing classical computational time to find shorter sequences of quantum gates. The original paper by Patel, Markov, and Hayes suggests that $m \approx log_2 n$ is optimal so the search does not take very long.