Imperial College London

MSc Individual Project

Imperial College London

Department of Computing

Tapering Qudits Using the Symmetries of $\mathbb{Z}_d$

Author:
George Hilton

Supervisor:
Dr. Roberto Bondesan

Second Marker:
Professor Abbas Edalat

Submitted in partial fulfilment of the requirements for the MSc Degree in Advanced Computing of Imperial College London

September 2024

Abstract

Quantum computers have the capability to solve problems which were previously considered intractable through the power of entanglement and superposition. Their primary limitation lies in the stability of the physical qubits controlling the system – interactions with the environment cause the entangled quantum states to collapse. This phenomenon, known as *decoherence*, results in quantum computers losing the advantage over their classical counterparts. Additionally, current quantum hardware is limited in the number of qubits available, rendering it essential to develop resource-efficient algorithms that minimise qubit usage while preserving performance. We aim to optimise the number of qudits required to encode a Hermitian Hamiltonian with $\mathbb{Z}_d$ symmetries through *tapering*. The tapering we refer to, involves block diagonalisation with no loss of symmetry. This work takes inspiration from a tapering approach with qubits and generalises to the d-dimensional case of qudits. We will construct a solid mathematical foundation to justify the tapering, describe the algorithm to do so, implement an example and analyse the impact on space and time complexity for a selection of quantum algorithms.

Acknowledgements

I would like to thank my supervisor - Dr. Roberto Bondesan - for his invaluable support and guidance. More so, I want to thank him for creating the environment which sparked my passion for quantum computing. I feel truly fortunate to have had the opportunity to work together.

I would also like to thank my brother Harry, who makes everything a little easier.

Contents

1 Introduction and Preliminaries **4**

1.1 Computers 4

1.2 Quantum Computers 4

1.3 Mathematics Background 7

1.4 Qubits 11

1.4.1 Qubit Operators 11

1.4.2 Qubit Stabilisers 12

1.5 Qudits 14

1.5.1 Information 14

1.5.2 Qudit Pauli Operators 14

1.5.3 Qudit Pauli Group 15

1.5.4 Clifford Group 15

1.5.5 Stabilisers 16

2 Inspiration **17**

2.1 Historical Qubit Approach 18

2.2 Overview of Qubit Methods 20

3 von Neumann Bicommutant Theorem **21**

3.1 Topologies on $\mathcal{B}(\mathcal{H})$ 21

3.2 Commutant 23

3.3 von Neumann Bicommutant Theorem 23

3.4 Simultaneous Diagonalisation 26

3.5 Application to Qudit Tapering 26

4 Qudit Tapering **29**

4.1 Commutation 29

4.2 Construction of a Minimal Generating Set 30

4.3 Constructing U in Prime Power Dimensions 31

4.3.1 Definitions 31

4.3.2 Some useful results . . . 37
4.3.3 Action of the Operators . . . 39
4.3.4 Algorithm . . . 40
4.3.5 Elucidation . . . 41
4.3.6 Example . . . 42
4.4 Overview . . . 43

5 Quantum Clock Model Example 44
5.1 Hamiltonian and Stabiliser Group . . . 44
5.2 Minimal Generating Set . . . 45
5.3 Constructing U . . . 45
5.4 Transformed Hamiltonian when $n = 2$ . . . 45
5.5 General n . . . 49
5.6 Simulation . . . 52

6 Space and Time Complexity 54
6.1 Complexity of the U construction . . . 54
6.2 Impact on Quantum Algorithms . . . 54
6.2.1 Quantum Linear Systems Solver (QLSS) . . . 55
6.2.2 Randomised Quantum Solver . . . 56

7 Conclusion 58
7.1 Overview . . . 58
7.2 The Future . . . 60

8 Ethical Considerations 61

Chapter 1

Introduction and Preliminaries

1.1 Computers

Computers have irreversibly altered human existence. An ominpresent entity which links humans from each end of the globe instantly, provides immediate access to almost any piece of information in recorded history, keeps planes in the sky and money safe in banks. Fundamentally, computers allow us to perform complex computations quickly and solve problems which were previously considered impossible. Despite relentless computational advances, some challenges remain simply too complex for conventional computers, these problems are known as *intractable*. NP-hard and NP-complete problems are considered intractable as a classical computer cannot solve them in polynomial time - they are commonly problems which scale exponentially. The Travelling Salesman Problem is one such example, the task is to find the shortest route visiting n cities once and returning to the starting city; a brute force algorithm would require $(n-1)!$ checks which grows exponentially as n increases. Indeed, the N-body problem [1], which aims to predict the motion of N interacting particles under mutual gravitational attraction, is also intractable. It is not difficult to imagine that simulating the interactions between millions of stars in a galaxy would take a long time. There is, of course, an infinitely long list of problems which are considered intractable but this is not the purpose of the introduction nor of the thesis. Instead, we wish to convey the potential limitations of traditional computing.

1.2 Quantum Computers

Quantum computers - leveraging the power of superposition, entanglement and quantum parallelism - have introduced the capability to speed up classically intractable problems (note, there are no known general solutions to NP-hard and NP-complete problems). A selection of the most iconic examples include Shor's groundbreaking discovery [2] of a number factorisation algorithm in polynomial time and Grover's algorithm [3], which finds an object in an unsorted array of n

objects in $\mathcal{O}(\sqrt{n})$ time versus the classical exhaustive approach in $\mathcal{O}(n)$. More modern advancements include the task of quantum phase estimation [4] which has been reduced from exponential time to polynomial time and solutions to instances of the satisfiability problem based on adiabatic evolution [5]. As exciting as these discoveries are let's consider some *real world* benefits of quantum computing.

The advancements made by quantum computers have the capability for profound social and economic impact. Quantum network medicine has unlocked "classically inaccessible molecular states, such as the often sought ground state (the lowest energy state of the molecule), which can then be encoded in the quantum machine, enabling us to extract the relevant quantities to predict its chemical behaviour" [6]. This is extremely useful for performing simulations which were previously too computationally intensive. It will allow exploration of the chemical compound space to search efficiently for new drugs and drug combinations. A few examples include [7] which demonstrated how quantum random walks can provide speed up in identifying effective drug-target interactions; [8] proposed a quantum solution to the NP-hard protein-folding problem and [9] investigated a number of machine learning approaches to identify target-binding compounds and to estimate the binding affinity between compounds and targets. It is clear that the intrinsic power of quantum algorithms has the potential to push science and research forward into new frontiers.

It is natural to think, *why aren't I reading this on a quantum computer?* Trying to build quantum computers and run quantum algorithms is significantly more challenging than producing the mathematics inter alia. Simply building qubits (the quantum bit defined in Section 1.4) produces its own set of complexities. Some of the most common physical realisations include

1. *Superconducting qubits* like the transmon qubits [10], they consist of Josephson junctions and capacitors which are arranged to form a circuit that behaves quantum mechanically at very low temperatures.

2. *Ion trap qubits* [11] are realised by trapping individual ions (like those of calcium, ytterbium, or beryllium) and manipulating them using electromagnetic fields in a vacuum. The internal energy levels of the ions are used to represent qubit states, and quantum gates are implemented using laser pulses.

3. *Photonic qubits* like polarisation qubits use the polarisation state of a photon e.g. vertical or horizontal to encode quantum information [12].

Apart from the evident physical and monetary challenges of implementing qubits (e.g. superconductors need temperatures near absolute zero) there are other issues. Qubits and therefore quantum computers are far more vulnerable to errors than their classical counterparts. As with most technology, there is the hope that advancements may reduce the magnitude of the errors but, due to

the nature of quantum mechanics, it is unlikely these errors can be totally corrected/prevented. Thus, it is not unreasonable to assume that quantum computers' reliability will remain permanently inferior to near faultless classical computers. Modern classical computers have an error rate of around 10^{-18} - at each timestep the hardware takes a measurement then converts the bit value, which varies between 0 and 1, to the nearer of the two. Unless a *cosmic ray* causes a rogue bit flip, classical computers are essentially faultless.

We cannot apply this method in quantum computers because measurements project quantum states into a reduced dimension - this is known as *collapsing* the state; continuous measurements would totally collapse the entangled states and leave us with a classical computer. Entanglement will be encountered briefly further on but, for this thesis no detailed knowledge is required. Fundamentally, it is important to be aware that qubits are highly sensitive to their environment; the interaction between quantum states and the environment causes partial-measurements which collapse the system. This phenomenon is known as *decoherence*. As we do not operate in an ideal world, quantum computers are *open* systems; this means when the quantum system collides with particles or photons there is some form of interaction which contributes to decoherence. Similarly, thermal noise can cause random fluctuations in the system's quantum state and in the case of trapped ions or superconducting qubits, stray electromagnetic fields can induce unwanted transitions between states, both leading to decoherence. There are ways to combat this, namely quantum error correction and error correcting codes. The idea is that we need multiple physical qubits to safely encode a logical qubit with error correction; this means for even simple quantum algorithms which require a few logical qubits, a large number of physical qubits are required. For a detailed explanation of quantum codes see [13].

Intuitively, we recognise that for a given quantum algorithm or Hamiltonian we want to keep the number of logical qubits to the absolute minimum. We will focus on the reduction process known as *tapering*, or more specifically efficiently block diagonalising a Hamiltonian with no loss of symmetry. In classical computation, symmetries in physical systems are routinely leveraged to reduce the size of matrices for diagonalisation, this is considered relatively *easy* by application of the von Neumann Bicommutant Theorem (see Chapter 3). However, on quantum computers, this process is far more complex. Efficiently implementing the Hamiltonian blocks corresponding to different symmetry sectors is non-trivial and remains an open question in general. Previous works [14, 15] have made strides in addressing this challenge but, they are restricted to qubits which are 2-dimensional. In this thesis, we construct an approach to taper qudits (defined in Section 1.5), which are the d-dimensional extension of a qubit, using the symmetries of the group $\mathbb{Z}_d$. Qudits offer a promising direction for both theoretical and practical reasons. Many physically interesting systems, such as the Quantum Clock Model [16], are naturally described in terms of qudits rather

than qubits. Furthermore, qudits can be natively implemented (or via multiple qubits) on emerging quantum hardware platforms, providing potential advantages in terms of hardware efficiency and resource scaling.

1.3 Mathematics Background

We include some foundational definitions to provide a reasonable background for the thesis. More complex definitions and ideas will be discussed in detail throughout. The definitions in Section 1.3 are standard however, for further reading or clarification please see [17, 18].

Firstly, some qualities and transformations of matrices.

Definition 1.3.1 *The **adjoint** of a matrix is given by taking its conjugate transpose. We use* $\dagger$ *for notation.*

Definition 1.3.2 *A matrix* U*, is **unitary**, if* $U^\dagger U = UU^\dagger = I$*.*

Definition 1.3.3 *A matrix* A *is **Hermitian** or **self-adjoint** if* $A = A^\dagger$*.*

Now, let's define some essential group theory from first principles.

Definition 1.3.4 *A **group** is a set of elements* G*, and a binary operation* $* : G \times G \to G$*, which satisfy the following conditions:*

G1: *(associativity)* $\forall\ a, b, c \in G$*,* $(a * b) * c = a * (b * c)$

G2: *(existence of identity)* $\exists\ e \in G$ *such that* $a * e = a = e * a$*,* $\forall\ a \in G$

G3: *(existence of inverses)* $\forall\ a \in G$*,* $\exists\ a^{-1} \in G$ *such that,* $a * a^{-1} = e = a^{-1} * a$

Definition 1.3.5 S *is a **subgroup** of* G*, denoted* $S \leq G$*, under its binary operation, if* S *is a nonempty subset of a group* G *and*

S1: *(closure)* $a, b \in S \implies a * b \in S$

S2: *(existence of inverses)* $a \in S \implies a^{-1} \in S$

Note, this implies $e \in S$*.*

Definition 1.3.6 *A group* G*, is **abelian**, if* $\forall\ a, b \in G$*,* $ab = ba$ *(shorthand for* $a * b = b * a$*).*

Definition 1.3.7 *The **multiplicative group of integers modulo*** d *is denoted* $\mathbb{Z}_d$*.*

Definition 1.3.8 $\mathbb{Z}_d^n$ *refers to* n *dimensional vectors where each of the elements belong to* $\mathbb{Z}_d$*.*

Definition 1.3.9 *If G is a group, the **center** of G is the set of elements that commute with every element of G*

$$Z(G) = \{z \in G \mid \forall g \in G, zg = gz\}.$$

Definition 1.3.10 *Let G be a group acting on a set X. The **orbit** of an element $x \in X$ is defined as all its possible destinations under the group action*

$$Orb(x) := \{y \in X \mid \exists g \in G, y = g * x\}$$

Definition 1.3.11 *The group $SL(N, \mathbb{Z}_d)$ is defined as the group, under matrix multiplication, of $N \times N$ matrices over $\mathbb{Z}_d$, having determinant 1.*

Definition 1.3.12 *Let H be a subgroup of a group G. The subset $aH = \{ah \mid h \in H\}$ of G is the **left coset** of H containing a, while the subset $Ha = \{ha \mid h \in H\}$ is the **right coset** of H containing a.*

Definition 1.3.13 *A subgroup H of a group is a **normal subgroup** if $gH = Hg$ for $\forall g \in G$.*

Definition 1.3.14 *Let H be a normal subgroup of G. We define the **quotient group** $(G/H, *)$ as the set of cosets,*

$$G/H = \{gH \mid g \in G\},$$

with group action $(g_1H) * (g_2H) = (g_1g_2)H$.

Definition 1.3.15 *A subset S of a group G is a **generating set** for G if every element of G can be written as a finite product of elements of S and their inverses. When S is a generating set, we say S generates G and use the notation $G = \langle S \rangle$. Generating sets are not necessarily unique and are considered minimal if $\nexists T \subset S$ where $G = \langle T \rangle$.*

Now we transition to increasingly abstract algebra. The definitions are necessary to define the foundations of quantum computing theoretically and for some essential theorems later in the thesis.

Definition 1.3.16 *A **ring** is a set of elements $\mathcal{R}$ and two binary operations $+$ and $*$ which satisfy the following conditions:*

R1: *(additive associativity) $\forall x, y, z \in \mathcal{R}$, $(x + y) + z = x + (y + z)$*

R2: *(additive commutativity) $\forall x, y \in$, $x + y = y + x$*

R3: *(additive identity) $\exists 0 \in \mathcal{R}$ such that $\forall x \in \mathcal{R}$, $0 + x = x + 0 = x$*

R4: *(additive inverse) $\exists\ -x \in \mathcal{R}$ such that $x + (-x) = (-x) + x = 0$*

R5: *(left and right distributivity) For all $x, y, z \in \mathcal{R}$, $x * (y + z) = (x * y) + (x * z)$ and $(x + y) * z = (x * z) + (y * z)$*

It is said to be ***commutative*** *if* $\forall a, b \in \mathcal{R}\ a * b = b * a$.

Definition 1.3.17 *Let* $(\mathcal{R}, +_{\mathcal{R}}, *_{\mathcal{R}})$ *be a ring. Let* $(G, +_G)$ *be an abelian group. A* ***left module over*** $\mathcal{R}$ *is an* $\mathcal{R}$*-algebraic structure* $(G, +_G, \circ)_{\mathcal{R}}$ *with one operation* $\circ$*, the (left) ring action, which satisfies the following left module axioms*

M1: *(scalar multiplication (left) distributes over module addition)*

$$\forall \lambda \in \mathcal{R}, \forall x, y \in G: \quad \lambda \circ (x +_G y) = (\lambda \circ x) +_G (\lambda \circ y)$$

M2: *(scalar multiplication (right) distributes over scalar addition)*

$$\forall \lambda, \mu \in \mathcal{R}, \forall x \in G: \quad (\lambda +_{\mathcal{R}} \mu) \circ x = (\lambda \circ x) +_G (\mu \circ x)$$

M3: *(associativity of scalar multiplication)*

$$\forall \lambda, \mu \in \mathcal{R}, \forall x \in G: \quad (\lambda *_{\mathcal{R}} \mu) \circ x = \lambda \circ (\mu \circ x)$$

Note, we can similarly define a right module which is symmetric to the left module definition.

Definition 1.3.18 *Let* $(\mathcal{R}, +, *)$ *be a ring. Let* M *be a module over* $\mathcal{R}$*. A subset* $N \subseteq M$ *is a* ***submodule*** *of* M *if:*

S1: *(closed under addition)* $\forall n_1, n_2 \in N$*, the sum* $n_1 + n_2 \in N$.

S2: *(closed under scalar multiplication)* $\forall r \in \mathcal{R}$ *and* $n \in N$*, the product* $r \circ n \in N$.

S3: *(contains the zero element) the additive identity* $0_M \in N$.

Definition 1.3.19 *A* ***field*** *is a set* $\mathbb{F}$*, containing at least two elements, on which multiplication and addition are defined and the following conditions hold for all elements* $x, y, z \in \mathbb{F}$*:*

F1: *(commutativity of addition)* $x + y = y + x$

F2: *(associativity of addition)* $(x + y) + z = x + (y + z)$

F3: *(existence of an additive identity)* $\exists\ 0 \in \mathbb{F}$*, called zero, such that* $x + 0 = x$

F4: *(existence of additive inverses)* $\forall\ x$, $\exists\ -x \in \mathbb{F}$ *such that* $x + (-x) = 0$

F5: *(commutativity of multiplication)* $xy = yx$

F6: *(associativity of multiplication)* $(x * y) * z = x * (y * z)$

F7: *(distributivity)* $x * (y + z) = x * y + x * z$ *and* $(x + y) * z = x * z + y * z$

F8: *(existence of a multiplicative identity) There is an element* $1 \in \mathbb{F}$*, such that* $1 \neq 0$ *and* $x * 1 = x$

F9: *(existence of multiplicative inverses) If* $x \neq 0$*, then* $\exists\ x^{-1} \in \mathbb{F}$ *such that* $x * x^{-1} = 1$

Definition 1.3.20 *A* ***vector space*** *consists of a set* V*, a field* $\mathbb{F}$*, and two operations: vector addition and scalar multiplication which satisfy the following conditions:*

V1: *(associativity of vector addition)* $(u + v) + w = u + (v + w)\ \forall\ u, v, w \in V$

V2: *(existence of a zero vector)* $\exists\ 0$ *such that* $u + 0 = u,\ \forall\ u \in V$

V3: *(additive inverse)* $\forall\ u \in V,\ \exists\ -u \in V$*, such that* $u + (-u) = 0$

V4: *(associativity of multiplication)* $(ab)u = a(bu)\ \forall\ a, b \in \mathbb{F}$ *and* $u \in V$

V5: *(distributivity)* $(a + b)u = au + bu$ *and* $a(u + v) = au + av\ \forall\ a, b \in \mathbb{F}$ *and* $u, v \in V$

V6: *(unitarity)* $\exists\ 1 \in \mathbb{F}$ *such that* $1u = u\ \forall\ u \in V$

Definition 1.3.21 *Let* V *be a vector space over a complex subfield* $\mathbb{F}$*. An* ***inner product*** *is a map* $\langle *, * \rangle : V \times V \to \mathbb{F}$ *that satisfies the following conditions*

I1: *(conjugate symmetry)* $\langle x, y \rangle = \overline{\langle y, x \rangle}, \quad \forall x, y \in V$

I2: *(linearity in the first argument)* $\langle ax + y, z \rangle = a\langle x, z \rangle + \langle y, z \rangle, \quad \forall x, y, z \in V,\ a \in \mathbb{F}$

I3: *(non-negative definiteness)* $\langle x, x \rangle \in \mathbb{R}_{\geq 0}, \quad \forall x \in V$

I4: *(positiveness)* $\langle x, x \rangle = 0 \implies x = 0_V, \quad \forall x \in V$

Definition 1.3.22 *A* ***Hilbert space*** *is a vector space* $\mathcal{H}$ *over* $\mathbb{C}$ *that is equipped with a complete inner product. Complete simply means that all Cauchy sequences converge i.e. there are no missing points from the vector space.*

Definition 1.3.23 *A set of vectors,* V *are* ***orthonormal*** *if the inner products between any* $v_i, v_j \in V$ *is* $\delta_{i,j}$*.*

Definition 1.3.24 *A set of vectors* $\{v_1, \ldots, v_n\}$ *are linearly independent if* $\sum_{i=1}^{n} \alpha_i v_i = 0 \implies \alpha_i = 0\ \forall i$*.*

Definition 1.3.25 *A set of vectors* $\{v_1, \ldots v_n\}$ *span a vector space* V *if* $\forall v \in V$*, there is a set of scalars* $\{\alpha_1, \ldots, \alpha_n\}$ *such that* $v = \sum_{i=1}^{n} \alpha_i v_i$*.*

Definition 1.3.26 *A set of orthonormal vectors forms a* ***basis*** *is they are linearly independent and span the vector space.*

Definition 1.3.27 $\mathbb{F}_d$ *denotes the finite field (also referred to as Galois Field) with* d *elements.* d *is always a prime power, i.e.* $d = p^n$ *where* p *is a prime number and* n *is a positive integer.*

Definition 1.3.28 *$\mathbb{F}_d^*$ denotes the multiplicative group of the Galois Field $\mathbb{F}_d$. This can be understood as $\mathbb{F}_d \backslash \{0\}$. It forms a group under the field's multiplicative operation.*

Definition 1.3.29 *A **ket** represents a vector in a complex vector space, often referred to as a state vector. It is denoted by the symbol $|\cdot\rangle$. A **bra** represents the dual vector (or covector) corresponding to a ket. It is denoted by the symbol $\langle\cdot|$.*

Definition 1.3.30 *The **spectrum** of a matrix is the set of its eigenvalues. More generally, if $T : V \to V$ is a linear operator on any finite-dimensional vector space, its spectrum is the set of scalars λ such that $T - \lambda I$ is not invertible.*

Definition 1.3.31 *An **algebra** over a field $\mathbb{F}$ is an ordered pair $(A, *)$ where A is a vector space over $\mathbb{F}$ and $* : A \times A \to A$ is a bilinear mapping. An algebra is known as unital if it has an identity.*

Definition 1.3.32 *Let $\mathcal{R}$ be a commutative ring. Let $(\mathcal{A}, *)$ be an algebra over $\mathcal{R}$. A **subalgebra** of $\mathcal{A}$ is a submodule $\mathcal{B} \subseteq \mathcal{A}$ such that: $\forall x, y \in \mathcal{B} : x * y \in \mathcal{B}$. That is, such that $\mathcal{B}$ is closed under $*$.*

1.4 Qubits

In classical computer systems information is transferred through bits; they take the form of a string of 0s and 1s which are elements from $\mathbb{Z}_2^n$, for arbitrary n. However, in quantum systems, quantum states are elements of a Hilbert space whose dimension varies based on context. Analagous to the traditional bit, we define a qubit or, quantum bit, to be the basic unit of information in quantum computing. They are commonly realised by manipulating and measuring subatomic particles like photons and electrons as discussed in Section 1.2. We can write the general form of a qubit as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{1.1}$$

where $\alpha, \beta \in \mathbb{C}$, $|\alpha|^2 + |\beta|^2 = 1$ and $\{|0\rangle, |1\rangle\}$ forms the canonical basis of $\mathbb{R}^2$. We see this is a superposition of the classical bit states with amplitudes α and β.

Quantum states can exist where multiple qubits cannot be decomposed as a product of single qubit states - this is known as entanglement. Entanglement is what provides quantum computers with their advantage of classical computers.

1.4.1 Qubit Operators

We define several classical unitary gates, known as Pauli operators.

$$X = |0\rangle\langle 1| + |1\rangle\langle 0| = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = |0\rangle\langle 0| - |1\rangle\langle 1| = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{1.2}$$

These operators form the basis of most qubit operations and will be essential for understanding the background context. Subscripts on the operators refers to the index of the qubit it acts on e.g. X_1 applies X to the first qubit.

1.4.2 Qubit Stabilisers

Definition 1.4.1 *We can define the* ***qubit Pauli Group*** *as the tensor product of X and Z operators up to multiplication by $\pm I$ and $\pm i$, under multiplication. It is more formally defined as*

$$\mathcal{P}_n^b \coloneqq \{\langle \pm I, \pm iI, X, Z\rangle^{\otimes n}, *\}. \tag{1.3}$$

The qubit Pauli operators satisfy a significant property that they *anti-commute* with one another i.e. $\sigma_i\sigma_j + \sigma_j\sigma_i = 0$ for $\sigma_i, \sigma_j \in \mathcal{P}_n^b$ and $i \neq j$.

Stabilisers are at the core of this work - to thoroughly understand them, we follow [13, 19] who provide unequivocal explanations.

Let's start with an example, specifically the EPR state of two qubits

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{1.4}$$

It is easy to see that $X_1X_2|\psi\rangle = |\psi\rangle$ and $Z_1Z_2|\psi\rangle = |\psi\rangle$. We say that $|\psi\rangle$ is stabilised by X_1X_2 and Z_1Z_2. The intrinsic principle of the stabiliser formalism is that instead of working with specific quantum states, we can work with the operators that stabilise them instead. We can now define stabilisers more concretely. Let $S \leq \mathcal{P}_n^b$ and define V_S to be the set of n qubit states which are fixed by every element of S. V_S is the *vector space stabilised by* S, and S is said to be the *stabiliser* of the space V_S, since every element of V_S is stable under the action of elements in S. Furthermore, we actually only need to consider the vector stabilised by the generators of S as every element in S can be written as a product of the generators. Hence, if a vector is stabilised by the generators, then it is stabilised by any element of S.

For S to stabilise a non-trivial subspace, we need S to be abelian and $-I \notin S$. Before the proof that this is sufficient we need to develop a little more theory.

To begin with, let's assume we have $S = \langle g_1, \ldots, g_k\rangle$. We define the *check matrix* as a $k \times 2n$ matrix to indicate the X and Z components of each generator. To determine the entries on the ith row we consider the generator g_i. If g_i contains an I on the jth qubit, then the jth and $n+j$th column elements are 0; if it contains an X on the jth qubit then, the jth column element is 1 and if it contains a Z on the jth qubit then, the $n+j$th column element is 1. The check matrix does not contain information about the multiplicative factors so we are always working with *unnormalised* states in this context. It was also demonstrated in [19] that for some stabiliser group

$S = \langle g_1, \ldots, g_k \rangle$, where $-I \notin S$ the generators g_1 through g_k are independent if and only if the rows of the corresponding check matrix are linearly independent.

Now we can verify that V_S is non-trivial provided S is generated by independent commuting generators and $-I \notin S$. The proof is taken from [19] and included for completeness.

Proof:
The idea of the proof is that if there are $k = n - l$ generators, then V_S is 2^l-dimensional, based on the intuitive argument that each additional generator for the stabiliser cuts the dimension of V_S by a factor of 2, as we might naively expect because the $+1$ and -1 eigenspaces for a tensor product of Pauli matrices divide the total Hilbert space into two subspaces of equal dimension.

To formalise this, let $S = \langle g_1, \ldots, g_{n-l} \rangle$ be generated by $n - l = k$ independent and commuting elements from $\mathcal{P}_n^b$, and such that $-I \notin S$. Then V_S is a 2^l-dimensional vector space.

Let $x = (x_1, \ldots, x_{n-l})$ be a vector of $n - l$ elements of $\mathbb{Z}_2$. Define

$$P_S^x \equiv \frac{\prod_{j=1}^{n-l} (I + (-1)^{x_j} g_j)}{2^{n-l}}. \tag{1.5}$$

Because $\frac{(I+g_j)}{2}$ is the projector onto the $+1$ eigenspace of g_j, it is easy to see that $P_S^{(0,\ldots,0)}$ must be the projector onto V_S. For each x there exists $g_x \in \mathcal{P}_n^b$ such that $g_x P_S^{(0,\ldots,0)} (g_x)^\dagger = P_S^x$, and therefore the dimension of P_S^x is the same as the dimension of V_S. Furthermore, for distinct x the P_S^x are easily seen to be orthogonal.

The proof is concluded with the algebraic observation that

$$I = \sum_x P_S^x. \tag{1.6}$$

The left hand side is a projector onto a 2^n-dimensional space, while the right hand side is a sum over 2^{n-l} orthogonal projectors of the same dimension as V_S, and thus the dimension of V_S must be 2^l. □

There is one final comment to make in relation to stabilisers. Let V_s be the vector space stabilised by S, $|\psi\rangle \in V_S$ and $g \in S$. Then, for a unitary operator U,

$$U|\psi\rangle = Ug|\psi\rangle = UgU^\dagger U|\psi\rangle \tag{1.7}$$

thus, the state $U|\psi\rangle$ is stabilised by $UgU^\dagger$. It follows that the vector space UV_S is stabilised by the group $USU^\dagger \coloneqq \{UgU^\dagger \,|\, g \in S\}$. Very importantly, if $g_1, \ldots, g_k$ generate S, then $Ug_1U^\dagger, \ldots, Ug_kU^\dagger$ generate $USU^\dagger$, so to compute the change in the stabiliser we need only compute how it affects the generators of the stabiliser.

1.5 Qudits

Given that qubits belong to a Hilbert space of dimension two, $\mathcal{H}_2 \cong \mathbb{C}^2$, it is natural to consider whether this can be extended to arbitrary dimensions. This is where *qudits* come in.

Qudits are the d-dimensional version of qubits, they can be described by vectors in $\mathcal{H}_d \cong \mathbb{C}^d$ [20]. The space is spanned by a set of orthonormal basis vectors $\{|0\rangle, |1\rangle, ..., |d-1\rangle\}$ - this is the canonical basis of $\mathbb{R}^d$. A general qudit has the form

$$|\alpha\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + \alpha_2|2\rangle + ... + \alpha_{d-1}|d-1\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{d-1} \end{bmatrix} \in \mathbb{C}^d, \quad \text{where } \sum_{i=0}^{d-1} |\alpha_i|^2 = 1. \tag{1.8}$$

For the rest of the thesis, if a Hilbert Space $\mathcal{H}$ is referred to in the context of n qudits of dimension d, you can assume $\mathcal{H} \cong \mathbb{C}^{d^n}$.

1.5.1 Information

It is obvious that a single qudit can describe more states than a single qubit because it has d possible basis states instead of 2. But, we can generalise this to analyse the potential information difference.

n qubits are described by a 2^n-dimensional Hilbert space and n qudits are described by a d^n-dimensional Hilbert space. Comparing the number of distinguishable states between n qudits and qubits we have

$$\text{Information Ratio} = \frac{d^n}{2^n} = \left(\frac{d}{2}\right)^n. \tag{1.9}$$

For $d > 2$, qudits describe exponentially more information than qubits. For example, if $d = 3$ n qutrits can represent 1.5^n times more states than n qubits and if $d = 10$ qudits represent 5^n times more states than n qubits. Thus, higher-dimensional qudits can significantly increase the amount of information that can be described and processed compared to qubits. From this, we can determine that for each logical qudit we would need $\lceil \log_2(d) \rceil$ logical qubits to describe it.

1.5.2 Qudit Pauli Operators

Analogous to the qubit case, there are natural extensions to the Pauli X and Z operators. We define Pauli X in the qudit case as $X|j\rangle = |j+1\rangle$ (note, it is commonly defined as $X|j+1\rangle = |j\rangle$ too, there are varying conventions). Addition in the ket is performed modulo d. Similarly, we define Pauli Z in the qudit case as $Z|j\rangle = \omega^j|j\rangle$, where $\omega = e^{2\pi i/d}$. For example, if $d = 5$

$$X = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \omega & 0 & 0 & 0 \\ 0 & 0 & \omega^2 & 0 & 0 \\ 0 & 0 & 0 & \omega^3 & 0 \\ 0 & 0 & 0 & 0 & \omega^4 \end{pmatrix}. \tag{1.10}$$

1.5.3 Qudit Pauli Group

Similar to the qubit case, we can define a Pauli Group for qudits. Let's introduce some notation first. All qudit operators (of dimension d) take the form $X_1^{x_1} Z_1^{z_1} \otimes ... \otimes X_n^{x_n} Z_n^{z_n}$. Let $x, z \in \mathbb{Z}_d^n$ and $\eta \coloneqq \begin{bmatrix} x \\ z \end{bmatrix} \in \mathbb{Z}_d^{2n}$ where

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad z = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}, \quad \eta = \begin{pmatrix} x_1 \\ \vdots \\ x_n \\ z_1 \\ \vdots \\ z_n \end{pmatrix}. \tag{1.11}$$

Then we can parametrise qudit operators as $X^x Z^z = XZ(\eta) \coloneqq X_1^{x_1} Z_1^{z_1} \otimes ... \otimes X_n^{x_n} Z_n^{z_n}$. It follows that for some qudit $\sigma \in \mathbb{Z}_d^n$ $XZ(\eta)|\sigma\rangle = \omega^{z^T \sigma} |\sigma + x\rangle$.

Definition 1.5.1 *The **qudit Pauli group** $\mathcal{P}_n^d$ (the superscript denotes either qu**d**it or qu**b**it) is defined as*

$$\mathcal{P}_n^d = \{\gamma^\delta XZ(\eta) : \gamma = \omega^{1/2}, \delta \in \mathbb{Z}_{2d}, \eta \in \mathbb{Z}_d^{2n}\}. \tag{1.12}$$

The *multiplication* of elements of the Pauli qudit group is defined as

$$\gamma^\delta XZ(\alpha)\gamma^\varepsilon XZ(\beta) \coloneqq \gamma^{\delta+\varepsilon+2\alpha^T M\beta} XZ(\alpha+\beta) \tag{1.13}$$

where $\alpha + \beta$ is modulo d, $\delta + \varepsilon + 2\alpha^T M\beta$ is modulo $2d$ and $M \coloneqq \begin{pmatrix} 0_n & 0_n \\ I_n & 0_n \end{pmatrix}$. The *commutativity* rule for elements of the Pauli group is $XZ(\alpha)XZ(\beta) = \omega^{\alpha^T P\beta} XZ(\beta)XZ(\alpha)$ where $P \coloneqq M - M^T$ mod d [21].

1.5.4 Clifford Group

Definition 1.5.2 *The **Clifford Group**, $\mathcal{C}_n^d$ is defined as the set of unitary matrices Q such that $\forall p \in \mathcal{P}_n^d$, $QpQ^\dagger \in \mathcal{P}_n^d$.*

1.5.5 Stabilisers

Analagous to the qubit case let's define stabilisers for qudits. Suppose $S \leq \mathcal{P}_n^d$ and define V_S to be the set of n qudit states which are fixed by every element of S. V_S is the *vector space stabilised by* S, and S is said to be the *stabiliser* of the space V_S, since every element of V_S is stable under the action of elements in S.

The *stabiliser matrix* is an equivalent to the qubit check matrix. Let S be an Abelian subgroup of $\mathcal{P}_n^d$ which has trivial intersection with the center of $\mathcal{P}_n^d$ so, $-I \notin \mathcal{P}_n^d$. Furthermore, let $\{g_1, g_2, \ldots, g_k\}$ be a minimal set of generators for S where $g_i = \gamma_i X^{\alpha_i} Z^{\beta_i}$, γ_i is an arbitrary scalar and $(\alpha_i, \beta_i) \in \mathbb{F}_d^n \times \mathbb{F}_d^n$. Then $(\alpha_i, \beta_i) \in \mathbb{F}_d^n \times \mathbb{F}_d^n$ parametrise a minimal generating set of S and the stabiliser matrix of the corresponding stabilizer code takes the form

$$\left(\begin{array}{c|c} \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \\ \vdots & \vdots \\ \alpha_k & \beta_k \end{array}\right) \in \mathbb{F}_d^{k \times 2n}. \tag{1.14}$$

Chapter 2

Inspiration

Several papers have been published involving the tapering of qubits. Both [14] and [15] taper qubits from fermionic Hamiltonians using the symmetries of $\mathbb{Z}_2$. These Hamiltonians commonly describe molecules with geometric symmetries such as rotations or reflections. The aforementioned papers both demonstrate how to taper qubits from the hydrogen molecule H_2. Let's look at a representation of the molecule [22] and visualise the $\mathbb{Z}_2$ symmetries. The molecule has two atomic sites, each with a proton and two electrons in a bound state. Looking at the figure below we see without the impact of any external interactions there are reflection and rotation symmetries; the reflection in Π_z is isomorphic to the $\mathbb{Z}_2$ symmetry discussed.

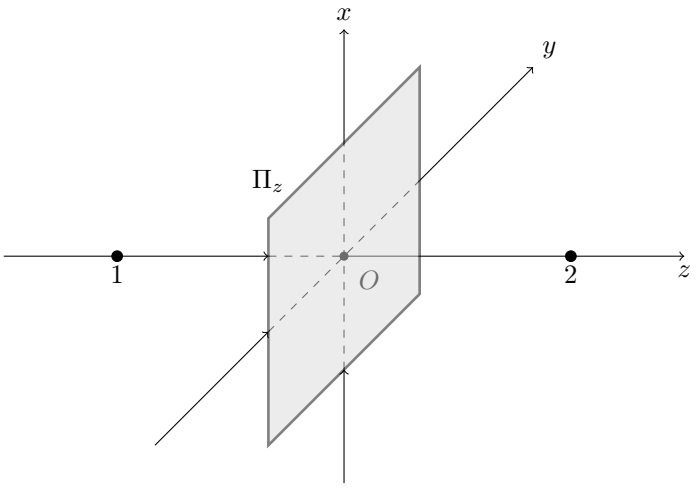

Figure 2.1: Symmetries of H_2

Although the applications to molecules are clear we can taper qubits from any Hamiltonian that can be written in a Pauli basis. Let's review the historical approach and take inspiration to generalise this from $\mathbb{Z}_2$ to $\mathbb{Z}_d$.

2.1 Historical Qubit Approach

The aforementioned papers follow near identical logic.

They establish the Hamiltonian as

$$H_{sim} = \sum_{j=1}^{r} h_j \sigma_j \tag{2.1}$$

where h_j are some real scalars and σ_j are n-qubit Pauli operators i.e. elements of the Pauli Group. Recall, the Pauli Group is defined as all operators taking the form $\mathcal{P}_n^b \coloneqq \{\langle \pm I, \pm iI, X, Z\rangle^{\otimes n}, *\}$. In addition, let's recall the Clifford Group, $\mathcal{C}_n^b$ is defined as the set of unitary operators U such that $U\sigma U^\dagger \in \mathcal{P}_n^b$ for all $\sigma \in \mathcal{P}_n^b$.

They construct an abelian subgroup of $\mathcal{P}_n^b$ namely, $\mathcal{S}$. This subgroup is known as the symmetry group of H_{sim}; it is defined such that any element of $\mathcal{S}$ commutes each Pauli term in H_{sim} and $-I \notin \mathcal{S}$. Daniel Gottesman demonstrated [13] that such a symmetry group has a set of independent generators, $\mathcal{S} = \langle \tau_1, ..., \tau_k \rangle$ and they claim there exists some unitary operator $U \in \mathcal{C}_n^b$ where

$$\tau_i = UX_iU^\dagger. \tag{2.2}$$

The Hamiltonian is then transformed by this U

$$H'_{sim} = U^\dagger H_{sim} U = \sum_{j=1}^{r} h_j \eta_j \tag{2.3}$$

where $\eta_j \in \mathcal{P}_n^b$. They claim conjugation by a unitary U does not change the spectrum of a matrix, we will prove it now for completeness.

Proof:

Let $Av = \lambda v$ for some matrix $A \in \mathbb{C}^{n\times n}$, vector $v \in \mathbb{C}^n$ and scalar λ. Now, consider the matrix $B = UAU^\dagger$. We want to show that B has the same eigenvalue λ.

Suppose $w = Uv$. Since U is unitary, w is a non-zero vector. We will show that w is an eigenvector of B with eigenvalue λ. We have

$$Bw = (UAU^\dagger)(Uv).$$

Using the fact that $U^\dagger U = I$, this simplifies to

$$Bw = U(AU^\dagger U)v = UAIv = UAv$$

Substituting $Av = \lambda v$,

$$Bw = U(\lambda v) = \lambda(Uv) = \lambda w.$$

This shows that w is indeed an eigenvector of B with eigenvalue λ and hence unitary conjugation does not change the spectrum of a matrix $\square$.

Using this result, we know H'_{sim} and H_{sim} have the same eigenvalues. Furthermore, by definition of the symmetry group $\mathcal{S}$, $[\tau_i, \sigma_j] = 0 \ \forall i, j$. Now, conjugating $[\tau_i, \sigma_j]$ by U,

$$\begin{aligned} U^\dagger[\tau_i, \sigma_j]U = 0 \\ \implies U^\dagger(\tau_i\sigma_j - \sigma_j\tau_i)U = 0 \\ \implies U^\dagger\tau_i\sigma_j U - U^\dagger\sigma_j\tau_i U = 0 \\ \implies U^\dagger\tau_i U U^\dagger\sigma_j U - U^\dagger\sigma_j U U^\dagger\tau_i U = 0 \\ \implies [U^\dagger\tau_i U, U^\dagger\sigma_j U] = 0 \\ \implies [X_i, \eta_j] = 0 \quad \forall i, j \end{aligned} \tag{2.4}$$

it follows that $[X_i, \eta_j] = 0$ for all i, j. Converting symbols to words, we have that all Pauli terms in H'_{sim} commute with the X-type Pauli operators on the first k qubits. This only occurs if each η_j acts on the first k qubits by I or X. As such, we can replace the first k X-type Pauli operators with their eigenvalues ± 1 and taper the first k qubits from the simulator system. The validation of this approach involves the von Neumann Bicommutant Theorem which is quite involved. As a result, the full discussion of this is saved for the qudit approach but the underlying principles remain the same.

It's essential to use the minimal generating set of the symmetry group as detailed in the preamble. To find this, both papers apply a symplectic version of the Gram-Schmidt orthogonolisation [23] to the basis vectors of the kernel of the parity check matrix of the Pauli operators in H_{sim}. This returns a set of linearly independent vectors which span the kernel of the parity matrix and parameterise the generators τ_i.

All that remains is to define the unitary operator U which transforms the Hamiltonian. Note, they both restrict the symmetry generators $\tau_1, ..., \tau_k$ to Z-type and choose a subset of the qubits $q(1), ..., q(k)$ (q is a permutation operator) such that

$$X_{q(i)}\tau_j = (-1)^{\delta_{i,j}}\tau_j X_{q(i)}. \tag{2.5}$$

A unitary Clifford operator U_i is defined as

$$U_i = \frac{1}{\sqrt{2}}(X_{q(i)} + \tau_i) \tag{2.6}$$

such that $U_i^2 = I$, $U_i X_{q(i)} U_i^\dagger = \tau_i$ and $U_j \sigma_{q(i)} U_j^\dagger = X_i$ for all $i \neq j$. Letting, $U = U_1 U_2 ... U_k W$ where W is the permutation matrix we have defined a unitary U such that $\forall \tau_i \in \mathcal{S}$

$$\tau_i = U X_i U^\dagger$$

and thus, the qubit approach is complete.

2.2 Overview of Qubit Methods

Let's break down the approach into some concise steps,

(I) Convert the Hamiltonian with $\mathbb{Z}_2$ symmetry to a Pauli basis namely, H_{sim}.

(II) Define the symmetry group $\mathcal{S}$, its generators $\tau_1, \ldots, \tau_k$ and the unitary U.

(III) Construct the generators of the symmetry group (as minimally and efficiently as possible).

(IV) Construct the unitary U.

(V) Transform the elements of the Hamiltonian by conjugating with U.

(VI) Taper the first k qudits in the Hamiltonian by projecting terms into a block diagonal form.

Looking at these steps we see it is relatively straightforward to taper qubits from appropriate Hamiltonians. We will translate this 2-dimensional approach into d dimensions; enabling the reduction of qudits from Hamiltonians involving more complex systems and interactions.

Chapter 3

von Neumann Bicommutant Theorem

We have established the preliminary material in algebra and quantum mechanics. We have also defined qubits, qudits and their respective operators belonging to the Pauli and Clifford group. Finally, we deconstructed the qubit tapering approach into its intrinsic components. We possess nearly all the building blocks to construct a method to taper qudits from Hamiltonians with higher dimensional symmetries. This chapter discusses the final required component for the approach. We will demonstrate, using abstract mathematics, why we can taper qudits and preserve the Hamiltonian. The approach, more formally, is an application of the von Neumann Bicommutant Theorem. Before we can understand this we need to build some theory. The following utilises subsections 3.1 and 3.2 from [24] and many sections from [25].

3.1 Topologies on $\mathcal{B}(\mathcal{H})$

Let's begin with the definition of $\mathcal{B}(\mathcal{H})$. A linear map (operator) $a : \mathcal{H} \to \mathcal{K}$ is said to be bounded if $\exists K \in \mathbb{R}$ with $\|a\xi\| \leq K\|\xi\|$ $\forall \xi \in \mathcal{H}$. The infimum of all such K is called the *norm* of a, written $\|a\|$. The set of all bounded operators from $\mathcal{H}$ to $\mathcal{K}$ is written $\mathcal{B}(\mathcal{H}, \mathcal{K})$ and if $\mathcal{H} = \mathcal{K}$ we use $\mathcal{B}(\mathcal{H})$. Boundedness of an operator is equivalent to continuity.

We need to define several new objects before we can proceed to the von Neumann Bicommutant Theorem.

Definition 3.1.1 *The **power set**, of a set $\mathcal{S}$ is defined as $\mathbb{P}(\mathcal{S}) \coloneqq \{T : T \subseteq \mathcal{S}\}$*

Definition 3.1.2 *Let $\mathcal{S}$ be a set. A **topology** on $\mathcal{S}$ is a subset $\tau \subseteq \mathbb{P}(\mathcal{S})$ of the power set of $\mathcal{S}$ that satisfies the **open set axioms**,*

***O1**: The union of an arbitrary subset of τ is an element of τ.*

***O2**: The intersection of any two elements of τ is an element of τ.*

***O3**: $\mathcal{S}$ is an element of τ.*

*If τ is a topology on $\mathcal{S}$, then $(\mathcal{S}, \tau)$ is called a **topological space**. The elements of τ are called the **open sets** of $(\mathcal{S}, \tau)$.*

Definition 3.1.3 *A **seminorm** p on a vector space V is a function $p : V \to \mathbb{R}$ that satisfies the following properties for all vectors $x, y \in V$ and scalars $\alpha \in \mathbb{R}$*

***S1:** (non-negativity) $p(x) \geq 0$*

***S2:** (absolute homogeneity) $p(\alpha x) = |\alpha| p(x)$*

***S3:** (triangle inequality $p(x + y) \leq p(x) + p(y)$*

Unlike a norm, a seminorm can assign a zero value to non-zero vectors.

Definition 3.1.4 *A set $C \subset V$ is **convex** if, for any two points $x, y \in C$ and any $t \in [0, 1]$, the point $tx + (1 - t)y$ is also in C. This means that any line segment between two points in C lies entirely within C.*

Definition 3.1.5 *Let $\mathbb{F} \in \{\mathbb{R}, \mathbb{C}\}$. Let X be a vector space over $\mathbb{F}$ and let P be a set of seminorms on X. Then we say the pair (X, P) is a locally convex space over $\mathbb{F}$.*

The following definitions of topologies on $\mathcal{B}(\mathcal{H})$ are pivotal to the proof of the von Neumann Bicommutant Theorem.

Definition 3.1.6 *The norm or uniform topology on $\mathcal{B}(\mathcal{H})$ is given by the norm $||a||$*

Definition 3.1.7 *The **strong operator topology** on $\mathcal{B}(\mathcal{H})$ is the locally convex topology defined by the family of semi norms $P_x(A) = \|Ax\|$ where $x \in \mathcal{H}$. A basis of neighbourhoods of $a \in \mathcal{B}(\mathcal{H})$ is formed by the basic neighbourhoods*

$$N(a, \zeta_1, \zeta_2, \ldots, \zeta_n, \epsilon) = \{b : \|(b - a)\zeta_i\| < \epsilon \quad \forall i = 1, \ldots, n\} \tag{3.1}$$

Definition 3.1.8 *The **weak operator topology** on $\mathcal{B}(\mathcal{H})$ is the locally convex topology defined by the family of semi norms $P_{x,y}(A) = |\langle x, Ay\rangle|$ where $x, y \in \mathcal{H}$. The weak operator topology is formed by the basic neighbourhoods*

$$N(a, \zeta_1, \zeta_2, \ldots, \zeta_n, \eta_1, \eta_2, \ldots, \eta_n, \epsilon) = \{b : |\langle (b - a)\zeta_i, \eta_i\rangle| < \epsilon \quad \forall i = 1, \ldots, n\} \tag{3.2}$$

We state the following proposition as fact, the proof is excluded but can be found in [24].

Proposition 3.1.1 The weak topology is weaker than the strong topology which is itself weaker than the uniform topology. Once $\mathcal{H}$ is infinite dimensional then these comparisons are strict but for our situation this will never occur.

We have now defined the strong operator topology and weak operator topology and can proceed to the definition of commutant.

3.2 Commutant

Similar to classical commutation we define the commutant as follows,

Definition 3.2.1 *Let $\mathcal{M} \subseteq \mathcal{B}(\mathcal{H})$. We define the **commutant** of $\mathcal{M}$ by*

$$\mathcal{M}' = \{b \in \mathcal{B}(\mathcal{H}) \mid bm = mb, \forall m \in \mathcal{M}\}. \tag{3.3}$$

We also have that $(\mathcal{M}')' = \mathcal{M}''$

3.3 von Neumann Bicommutant Theorem

The proofs in this section have been replicated in many ways, the ones shown here take strong inspiration from [24] and [25]. First we need to define closure as well as its weak/strong counterparts.

Definition 3.3.1 *Let $T = (S, \tau)$. A point $x \in S$ is a **limit point** of A if and only if every open set $U \in \tau$, such that $x \in U$, contains some point of A distinct from x.*

Definition 3.3.2 *Let $T = (S, \tau)$ be a topological space and $H \subseteq S$. The **closure** of H in T is defined as $\overline{H} \coloneqq H \cup \partial H$ where ∂H is the set of all limit points in H.*

Definition 3.3.3 *Let $T = (S, \tau)$ be a topological space and $H \subseteq S$. H is considered **closed** if and only if it equals $\overline{H}$.*

Definition 3.3.4 *A sequence of operators $\{A_n\} \subset \mathcal{B}(\mathcal{H})$ **converges** to $A \in \mathcal{B}(\mathcal{H})$*

- *in norm if: $\|A_n - A\| \to 0$*
- *in the strong operator topology if: $\|A_n x - Ax\| \to 0 \ \forall \ x \in \mathcal{H}$*
- *in the weak operator topology if: $\langle A_n x, y\rangle \to \langle Ax, y\rangle \ \forall \ x, y \in \mathcal{H}$*

Definition 3.3.5 *A set $A \subseteq \mathcal{B}(\mathcal{H})$ is **weakly closed** if it is closed in the weak topology. This means that if a sequence $\{A_n\}$ of operators in A converges to an operator $\tilde{A}$ in the weak topology, then $\tilde{A}$ must also be in A.*

Definition 3.3.6 *A set $A \subseteq \mathcal{B}(\mathcal{H})$ is **strongly closed** if it is closed in the strong topology. This means that if a sequence $\{A_n\}$ of operators in A converges to an operator $\tilde{A}$ in the strong topology, then $\tilde{A}$ must also be in A.*

Now, we have defined all the prerequisites we can move onto the main theorems.

Theorem 3.3.1. Let $\mathcal{H}$ be a finite dimensional Hilbert space and also, let $\mathcal{M}$ be a self-adjoint subalgebra of $\mathcal{B}(\mathcal{H})$ containing I. We claim that in this case $\mathcal{M} = \mathcal{M}''$.

Proof.
Trivially, $\mathcal{M} \subseteq \mathcal{M}''$, so it suffices to prove $m \in \mathcal{M}'' \implies m \in \mathcal{M}$.

Instead of considering the action of $\mathcal{M}$ on $\mathcal{H}$ we consider an extended action on the larger space $\mathcal{H} \otimes \mathcal{H}$ defined by $x(\varphi \otimes \psi) = x\varphi \otimes \psi$ where $x \in \mathcal{M}$ and $\varphi, \psi \in \mathcal{H}$. Let $\{v_1, v_2, \ldots, v_n\}$ be an orthonormal basis of $\mathcal{H}$. The space $\mathcal{H} \otimes \mathcal{H}$ can be considered as a direct sum of n copies of $\mathcal{H}$ as each copy of $\mathcal{H}$ in the direct sum $\bigoplus_{i=1}^{n} \mathcal{H}$ can be associated with one of the basis vectors in the original space $\mathcal{H}$. For each fixed i, the vector space $\mathcal{H}_i$ in the direct sum corresponds to $\mathcal{H} \otimes v_i$. Thus,

$$\mathcal{H} \otimes \mathcal{H} \cong \bigoplus_{i=1}^{n} \mathcal{H}.$$

This essentially partitions the n^2-dimensional space into n blocks, each block being n-dimensional.

On the extended space $\mathcal{H} \otimes \mathcal{H}$, the algebra $\mathcal{M}$ is embedded as $n \times n$ matrices that are constant on the diagonal, where each diagonal entry is an element of $\mathcal{M}$ itself. Each block on the diagonal corresponds to an operator from $\mathcal{M}$, and all off-diagonal entries are zero.

The commutant $\mathcal{M}'$ of $\mathcal{M}$ on $\mathcal{H} \otimes \mathcal{H}$ consists of all $n \times n$ matrices where each entry is from the commutant of $\mathcal{M}$ in $\mathcal{B}(\mathcal{H})$. The matrix contains arbitrary entries from $\mathcal{M}'$, not just along the diagonal but throughout the entire matrix.

The embedding of $\mathcal{M}''$ on $\mathcal{H} \otimes \mathcal{H}$, is all operators which commute with terms in $\mathcal{M}'$. On $\mathcal{H} \otimes \mathcal{H}$, it follows by their definitions, that $\mathcal{M}''$ must consist of matrices that are again constant on the diagonal, but with elements from the original algebra $\mathcal{M}$.

To formalise this we need one new result. For a closed subspace $\mathcal{K}$ of $\mathcal{H}$, let $P_{\mathcal{K}} : \mathcal{H} \to \mathcal{K}$ be the map which sends any point in $\mathcal{H}$ to the nearest point in $\mathcal{K}$. We use without proof (see [25]) the fact that $P_{\mathcal{K}}$ is linear, a projection and that if $a = a^*$ for some $a \in \mathbb{B}(\mathcal{H})$ then $a\mathcal{K} \subseteq \mathcal{K} \iff [a, P_{\mathcal{K}}] = 0$.

Now, we can proceed with the formalism, let $v = \bigoplus_{i=1}^{n} v_i \in \bigoplus_{i=1}^{n} \mathcal{H}$ and let $V = \mathcal{M}v \subseteq \mathcal{H} \otimes \mathcal{H}$ then, $\mathcal{M}V \subseteq V$. $\mathcal{M} = \mathcal{M}^\dagger \implies P_V \in \mathcal{M}'$ on $\mathcal{H} \otimes \mathcal{H}$. If $m \in \mathcal{M}''$ on $\mathcal{H} \otimes \mathcal{H}$, then m commutes with P_V and $m\mathcal{M}V \subseteq \mathcal{M}v$. Importantly, this implies that if $m(Iv) = xv$ for some $x \in \mathcal{M}$ we have $mv_i = xv_i \ \forall i$ and so, $m = x \in \mathcal{M}$.

Since we already have $\mathcal{M} \subseteq \mathcal{M}''$ and demonstrated that $\mathcal{M}'' \subseteq \mathcal{M}$, we have $\mathcal{M} = \mathcal{M}''$ completing the proof. □

Theorem 3.3.2. (von Neumann) Let $\mathcal{M}$ be a self-adjoint subalgebra of $\mathcal{B}(\mathcal{H})$ with $I \in \mathcal{M}$ and let $\overline{\mathcal{M}}$ be the closure of $\mathcal{M}$ under strong operator topology. Then $\mathcal{M}'' = \overline{\mathcal{M}}$

Proof.
Commutants are always closed so $\overline{\mathcal{M}} \subseteq \mathcal{M}''$. Now we need to verify $\mathcal{M}'' \subseteq \overline{\mathcal{M}}$.

To prove the inclusion, $\mathcal{M}'' \subseteq \overline{\mathcal{M}}$, consider any operator $a \in \mathcal{M}''$. We need to show that a can be approximated by elements in $\mathcal{M}$ with respect to the strong operator topology. Specifically, we need to demonstrate that for any strong neighborhood $N(a, \zeta_1, \zeta_2, \ldots, \zeta_n, \epsilon)$ of a, there exists an operator $x \in \mathcal{M}$ within this neighborhood this implies a belongs to the closure $\overline{\mathcal{M}}$.

Let $v = \bigoplus_{i=1}^{n} \zeta_i \in \bigoplus_{i=1}^{n} \mathcal{H}$. Recall, from the previous proof, that we can extend the action of operators from $\mathcal{B}(\mathcal{H})$ diagonally on this direct sum space. Specifically, each operator $x \in \mathcal{B}(\mathcal{H})$ acts independently on each component of the direct sum $\bigoplus_{i=1}^{n} \mathcal{H}$.

Now, consider the projection $P_{\mathcal{M}v}$ onto the subspace $\mathcal{M}v \subseteq \mathcal{H} \otimes \mathcal{H}$. Since $\mathcal{M}$ is self-adjoint and contains the identity, $\mathcal{M}$ commutes with $P_{\mathcal{M}v}$. Moreover, $P_{\mathcal{M}v}$ also commutes with a because $a \in \mathcal{M}''$. Thus, $a\mathcal{M}v \subseteq \mathcal{M}v$.

Since a commutes with the projection $P_{\mathcal{M}v}$, and $v = \bigoplus_{i=1}^{n} \zeta_i$,

$$av = \bigoplus_{i=1}^{n} a\zeta_i.$$

Given that $a \in \mathcal{M}''$ and $\mathcal{M}$ contains the identity, the vector av must lie in the closure of $\mathcal{M}v$. Therefore, there exists an operator $x \in \mathcal{M}$ such that for each $\epsilon > 0$, we have

$$\|(x - a)\eta_i\| < \epsilon \quad \text{for all } i = 1, \ldots, n.$$

Hence a can be approximated by elements of $\mathcal{M}$ in the strong operator topology, meaning $a \in \overline{\mathcal{M}}$. Therefore, we conclude that $\mathcal{M}'' \subseteq \overline{\mathcal{M}}$. Since we already have $\overline{\mathcal{M}} \subseteq \mathcal{M}''$, it follows that $\mathcal{M}'' = \overline{\mathcal{M}}$, completing the proof. □

Corollary 3.3.1
If $\mathcal{M} = \mathcal{M}^\dagger$ is a subalgebra of $\mathcal{B}(\mathcal{H})$ with $I \in \mathcal{M}$, then the following are equivalent:

1. $\mathcal{M} = \mathcal{M}''$
2. $\mathcal{M}$ is strongly closed.
3. $\mathcal{M}$ is weakly closed.

Definition 3.3.7 *A subalgebra of $\mathcal{B}(\mathcal{H})$ satisfying the conditions of Corollary 3.3.1 is called a **von Neumann algebra**.*

We have defined and proved the essential theorems and corollary to validate the tapering approach however the full use of them may still be unclear. Let's clarify the implications in an unambiguous way.

3.4 Simultaneous Diagonalisation

Block diagonalisation refers to the process of finding a basis in which an operator (or a set of operators) is represented as a block diagonal matrix. This is a powerful technique because it simplifies the operator into smaller, more manageable pieces.

If A is a commutative, unital, self-adjoint subalgebra of $\mathcal{B}(\mathcal{H})$, it reflects the symmetries of the system. If a Hamiltonian H commutes with every element of A, then H belongs to the commutant A'. Since A is strongly closed and forms a von Neumann algebra, a Hamiltonian $H \in A'$ can be block-diagonalised in the basis that diagonalises A, preserving the symmetry structure captured by A. Each block corresponds to an eigenspace of the commuting operators in A, known as sectors, with H acting independently on each block.

How does this help us? If we can construct a commutative, unital, self-adjoint subalgebra of $\mathcal{B}(\mathcal{H})$, namely $\mathcal{Z}$ and suppose $\mathcal{Z}$ commutes with the first k qudits in a Hamiltonian, H_{sim}, or a version of it under unitary conjugation. Then, we know $\mathcal{Z}$ forms a von Neumann algebra and by the von Neumann Bicommutant Theorem we can block diagonalise, *id est taper*, the first k qudits of the Hamiltonian while preserving its symmetries.

3.5 Application to Qudit Tapering

We will later define a method to transform Hamiltonians which can be written in a Pauli basis such that its eigenspaces are preserved and the first k qudits commute with either I or Pauli Z type operators. As $\{Z_1, \ldots, Z_k\}$ generate a commutative, unital, self-adjoint subalgebra of $\mathcal{B}(\mathcal{H})$ we know

that we can taper the first k qudits while preserving the symmetries of the original Hamiltonian.

Let's explicitly demonstrate how to implement the block diagonalisation assuming the correct set up. Block diagonalisation is generally more complex than this case but, due to the diagonal nature of the Pauli Z operator, we can simplify the process.

Consider a general Hamiltonian of the form:

$$H = \sum_i h_i \bigotimes_{j=1}^{n} \sigma_i^{(j)} = \sum_i h_i \sigma_i^{(1)} \sigma_i^{(2)} \dots \sigma_i^{(n)} \tag{3.4}$$

where h_i are real coefficients and $\sigma_i^{(j)}$ are elements of the d-dimensional qudit Pauli group acting on qudit j.

We represent the eigenstates of the first k qudits by $|0\rangle, |1\rangle, \dots, |d^k - 1\rangle$, which denote the basis states for the combined system of the first k qudits.

Each Hamiltonian term $h_i \sigma_i^{(1)} \dots \sigma_i^{(n)}$ is projected onto the $|b\rangle$ states of the first k qudits. This gives

$$h_i \langle b|\sigma_i^{(1)} \dots \sigma_i^{(k)}|b\rangle \sigma_i^{(k+1)} \dots \sigma_i^{(n)} \tag{3.5}$$

where $\langle b|\sigma_i^{(1)} \dots \sigma_i^{(k)}|b\rangle$ results in a scalar value depending on the action of $\sigma_i^{(1)}, \dots, \sigma_i^{(k)}$ on the basis state $|b\rangle$ and $\sigma_i^{(k+1)} \dots \sigma_i^{(n)}$ acts on the remaining qudits, from $k+1$ to n.

In each sector $|b\rangle$, the Hamiltonian becomes:

$$H_b = \sum_i h_i \langle b|\sigma_i^{(1)} \dots \sigma_i^{(k)}|b\rangle \sigma_i^{(k+1)} \dots \sigma_i^{(n)} \tag{3.6}$$

here, the h_i coefficients are scaled by $\langle b|\sigma_i^{(1)} \dots \sigma_i^{(k)}|b\rangle$ and the remaining operators $\sigma_i^{(k+1)} \dots \sigma_i^{(n)}$ act on the unaffected qudits.

The full block-diagonalised Hamiltonian, projected onto the different eigenstates $|b\rangle$ of the first k qudits, is

$$H = \bigoplus_b H_b = \bigoplus_b \sum_i h_i \langle b|\sigma_i^{(1)} \dots \sigma_i^{(k)}|b\rangle \sigma_i^{(k+1)} \dots \sigma_i^{(n)}. \tag{3.7}$$

Or in matrix form,

$$H = \begin{pmatrix} H_0 & 0 & 0 & \cdots & 0 \\ 0 & H_1 & 0 & \cdots & 0 \\ 0 & 0 & H_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & H_{d^k-1} \end{pmatrix}. \tag{3.8}$$

Chapter 4

Qudit Tapering

We have now constructed enough theory (and validated it) to begin the qudit tapering method. The approach, in a simplified form, involves finding a unitary $U \in \mathcal{C}_n^d$ which - under conjugation - maps the generators of the Hamiltonian to the Pauli Z operator. As determined by the previous section, this implies we can taper the qudits and block diagonalise the Hamiltonian while preserving its symmetries.

4.1 Commutation

Let H_{sim} be a Hermitian qudit Hamiltonian which can be written as a linear combination of n qudit Pauli operators. Such examples exist, like the Quantum Clock Model [16] (which will be examined in more detail later)

$$H_{sim} = \sum_{j=1}^{r} h_j \sigma_j \quad h_j \in \mathbb{R}, \sigma_j \in \mathcal{P}_n^d. \tag{4.1}$$

Let $\mathcal{S} \leq \mathcal{P}_n^d$. We say $\mathcal{S}$ is a stabiliser/symmetry group (terms can be used interchangeably) if $\mathcal{S}$ is abelian and any element of $\mathcal{S}$ commutes with each Pauli term σ_j, in H_{sim} and $-I \notin \mathcal{S}$. It is a well known result from the theory of stabiliser codes [13] that any symmetry group $\mathcal{S}$ has a set of independent generators $\mathcal{S} = \langle g_1, ..., g_k \rangle$ for some $k \in \mathbb{N}$ and for now, we assume $\exists U \in \mathcal{C}_n^d$ such that

$$g_i = U Z_i^a U^\dagger \tag{4.2}$$

$\forall i \in \{1, ..., k\}$ and $a \in \mathbb{Z}_d$. We let $a = 1$ without loss of generality. The construction of this unitary, U, will appear later.

We can transform this Hamiltonian by U as follows

$$H'_{sim} = U^\dagger H_{sim} U = \sum_j h_j \tau_j \quad \tau_j = U^\dagger \sigma_j U \in \mathcal{P}_n^d. \tag{4.3}$$

We proved in Chapter 2 that unitary conjugation does not alter the spectrum of the Hamiltonian hence, H_{sim} and H'_{sim} have the same eigenvalues. Moreover, by the definition of $\mathcal{S}$, we know that all elements of $\mathcal{S}$ commute with the Pauli terms in the Hamiltonian it stabilises - this can be formally written as $[g_i, \sigma_j] = 0 \ \forall i \in \{1, 2, \ldots, k\}$ and $\forall j$. Conjugating this commutation relation we observe

$$\begin{aligned} U^\dagger [g_i, \sigma_j] U = 0 \\ \implies U^\dagger (g_i \sigma_j - \sigma_j g_i) U = 0 \\ \implies U^\dagger g_i \sigma_j U - U^\dagger \sigma_j g_i U = 0 \\ \implies U^\dagger g_i U U^\dagger \sigma_j U - U^\dagger \sigma_j U U^\dagger g_i U = 0 \\ \implies [U^\dagger g_i U, U^\dagger \sigma_j U] = 0 \\ \implies [Z_i, \tau_j] = 0 \end{aligned} \tag{4.4}$$

All Pauli terms in H'_{sim} commute with Z-type Pauli operators on the first k qudits. This is only possible if each τ_j with $j \in \{1, \ldots k\}$, acts with either I or Z^b for some $b \in \mathbb{Z}_d$. Recalling the theory from Chapter 3 we know it is possible to taper qudits from the Hamiltonian by projecting onto the eigenspaces of the Pauli operators $Z_1, ..., Z_k$. Given the definition of qudit Pauli Z the eigenvalues are given by the solutions to $\gamma^d = 1$, these are the d^{th} roots of unity: $1, e^{\frac{2\pi i}{d}}, e^{\frac{4\pi i}{d}}, ..., e^{\frac{2(d-1)\pi i}{d}}$.

4.2 Construction of a Minimal Generating Set

We want to work with minimal generating sets as the setup requires independent generators and so will the algorithm to compute the unitary U. Given this, we need a way to construct a minimal generating set from a set of dependent generators. There are many methods to construct the minimal generating set; the one included mirrors the approach first outlined in [21].

We can construct a minimal generating set $\mathcal{S} = \langle XZ(G_1), \ldots, XZ(G_k) \rangle$ for the Hamiltonian's stabiliser given arbitrary stabiliser $\mathcal{S}'$ using the Smith Normal Form of its stabiliser matrix. Note, the scalar factors can be ignored as we are working with unnormalised states.

The G'_l are assembled in the stabiliser matrix G'. We compute the Smith Normal Form [26], this produces matrices satisfying $F = KG'L$ where $K \in \mathbb{Z}_d^{2n \times 2n}$ and $L \in \mathbb{Z}_d^{k' \times k'}$ and both are invertible. Note, $G'L$ is also a generator matrix of the stabiliser and by the definition of Smith Normal Form $G'L$ has the minimal number of non-zero columns - the rightmost $k - k'$ columns of $G'L$ are zero. Denote the new generator $\mathcal{S} = G'L$. No linear combination of the columns of $\mathcal{S}$ are zero unless coefficients are a multiple of the order of the columns: if $\sum_j r_j \mathcal{S} = 0 \ mod \ d$ then $r_j \mathcal{S} = 0 \ mod \ d$, so the generating set is minimal.

4.3 Constructing U in Prime Power Dimensions

To practically implement the qudit tapering method we need to construct a unitary U such that for each of the stabiliser generators g_i, $U^\dagger g_i U = Z_i$. The idea is to take the stabiliser matrix of the original stabiliser generators and reduce this matrix until we are left with zeroes aside from a diagonal of 1s in the Z half. It mirrors the way Gaussian Elimination is performed and this is a convenient way to think about it. For future reference, this manipulation of the stabiliser matrix will be called *matrix reduction*. We will construct several unitary qudit operators and consider how they conjugate X and Z Pauli operators. Then, we will find an equivalent representation of the operators which act on one qudit in $SL(2, \mathbb{F}_d)$ and for the operator which acts on two qudits in $SL(4, \mathbb{F}_d)$. These representations will be used to perform the matrix reduction on the stabiliser matrix while we keep track of their corresponding operators. When complete, we will have all the necessary components to construct the desired unitary. Let's begin with some definitions.

4.3.1 Definitions

Let $\omega = \exp(2\pi i/d)$ where $d = p^m$ where p is prime and m is an integer.

Let us also denote $\mathrm{tr}(\alpha)$ as the trace of an element $\alpha \in \mathbb{F}_d = \mathbb{F}_{p^m}$. This is defined as $\mathrm{tr}(\alpha) \coloneqq \sum_{i=0}^{m-1} \alpha^{p^i} \in \mathbb{F}_p$.

Lemma: This function acts as a map from $\mathbb{F}_d$ to $\mathbb{F}_p$ and is linear.

Proof.

Let $\mathbb{F}_d = \mathbb{F}_{p^m}$ be a finite field extension of the finite field $\mathbb{F}_p$, where p is a prime number. The trace function from $\mathbb{F}_d$ to $\mathbb{F}_p$ is defined as

$$\mathrm{tr}(\alpha) := \sum_{i=0}^{m-1} \alpha^{p^i} \quad \text{for } \alpha \in \mathbb{F}_d.$$

We want to prove that the trace function is linear, meaning that for any $\alpha, \beta \in \mathbb{F}_d$ and any $a, b \in \mathbb{F}_p$, the following holds

$$\mathrm{tr}(a\alpha + b\beta) = a\mathrm{tr}(\alpha) + b\mathrm{tr}(\beta).$$

First we prove *additivity*, consider the trace of the sum $\alpha + \beta$

$$\mathrm{tr}(\alpha + \beta) = \sum_{i=0}^{m-1} (\alpha + \beta)^{p^i}.$$

Using the binomial theorem in characteristic p, where $(x+y)^p = x^p + y^p$ for any $x, y \in \mathbb{F}_d$, we have

$$(\alpha + \beta)^{p^i} = \alpha^{p^i} + \beta^{p^i}.$$

Therefore,

$$\mathrm{tr}(\alpha+\beta) = \sum_{i=0}^{m-1}(\alpha^{p^i}+\beta^{p^i}) = \sum_{i=0}^{m-1}\alpha^{p^i} + \sum_{i=0}^{m-1}\beta^{p^i}$$

which simplifies to

$$\mathrm{tr}(\alpha+\beta) = \mathrm{tr}(\alpha) + \mathrm{tr}(\beta).$$

This proves the *additivity* of the trace function. Now onto *homogeneity*, consider the trace of a scalar multiple $a\alpha$, where $a \in \mathbb{F}_p$

$$\mathrm{tr}(a\alpha) = \sum_{i=0}^{m-1}(a\alpha)^{p^i}.$$

Since $a \in \mathbb{F}_p$, and $\mathbb{F}_p$ is the base field, we know that $a^{p^i} = a$ for all i. Therefore,

$$\mathrm{tr}(a\alpha) = \sum_{i=0}^{m-1} a\alpha^{p^i} = a\sum_{i=0}^{m-1}\alpha^{p^i} = a\mathrm{tr}(\alpha).$$

This proves the *homogeneity* of the trace function.

Since the trace function $\mathrm{tr}(\alpha)$ satisfies both additivity and homogeneity, it is *linear*:

$$\mathrm{tr}(a\alpha+b\beta) = a\mathrm{tr}(\alpha) + b\mathrm{tr}(\beta) \qquad \square$$

We define several operators [20]

1. $X^\alpha \coloneqq \sum_{x\in\mathbb{F}_d} |x+\alpha\rangle\langle x|$ for $\alpha \in \mathbb{F}_d$
2. $Z^\beta \coloneqq \sum_{z\in\mathbb{F}_d} \omega^{\beta z}|z\rangle\langle z|$ for $\beta \in \mathbb{F}_d$
3. $M_\gamma \coloneqq \sum_{y\in\mathbb{F}_d} |\gamma y\rangle\langle y|$ for $\gamma \in \mathbb{F}_d\backslash\{0\}$
4. $F \coloneqq \frac{1}{\sqrt{d}}\sum_{x,z\in\mathbb{F}_d} \omega^{xz}|z\rangle\langle x|$
5. $CX^{(i,j)} \coloneqq \sum_{x,y\in\mathbb{F}_d} |x\rangle_i|x+y\rangle_j\langle y|_j\langle x|_i = \sum_{\alpha\in\mathbb{F}_d}|\alpha\rangle\langle\alpha|\otimes X_\alpha$

with the significant commutation relation $X^\alpha Z^\beta X^{\alpha\dagger} = \omega^{-\alpha\beta}Z^\beta$ [27]. The superscripts in the $CX^{(i,j)}$ operator refer to the i^{th} and j^{th} qudit. Observe that X^α and Z^β are defined identically to the previous section but, we include α and β to represent their powers more generally.

Lemma: We claim that each of the operators are unitary.

Proof.

Let's verify the unitary condition step by step for each of the five operators above.

1. The operator X^α is defined as: $X^\alpha \coloneqq \sum_{x\in\mathbb{F}_d}|x+\alpha\rangle\langle x|$, the conjugate transpose $X^{\alpha\dagger} \coloneqq \sum_{x\in\mathbb{F}_d}|x\rangle\langle x+\alpha|$.

We need to show that $X^{\alpha\dagger}X^{\alpha} = I$.

$$\begin{aligned} X^{\alpha\dagger}X^{\alpha} &= \left(\sum_{y\in\mathbb{F}_d} |y\rangle\langle y+\alpha|\right)\left(\sum_{x\in\mathbb{F}_d} |x+\alpha\rangle\langle x|\right) \\ &= \sum_{y\in\mathbb{F}_d}\sum_{x\in\mathbb{F}_d} |y\rangle\langle y+\alpha|x+\alpha\rangle\langle x|. \end{aligned}$$

Using the orthogonality condition $\langle y+\alpha|x+\alpha\rangle = \delta_{y,x}$

$$X^{\alpha\dagger}X^{\alpha} = \sum_{y\in\mathbb{F}_d} |y\rangle\langle y| = I.$$

It follows with identical logic that $X^{\alpha}X^{\alpha\dagger} = I$. Hence X^{α} is a unitary operator.

2. Z^{β} is defined as: $Z^{\beta} \coloneqq \sum_{x\in\mathbb{F}_d} \omega^{\beta x}|x\rangle\langle x|$. The conjugate transpose is $Z^{\beta\dagger} \coloneqq \sum_{x\in\mathbb{F}_d} \omega^{-\beta x}|x\rangle\langle x|$.

We need to show that $Z^{\beta\dagger}Z^{\beta} = I$.

$$\begin{aligned} Z^{\beta\dagger}Z^{\beta} &= \left(\sum_{x\in\mathbb{F}_d} \omega^{-\beta x}|x\rangle\langle x|\right)\left(\sum_{y\in\mathbb{F}_d} \omega^{\beta y}|y\rangle\langle y|\right) \\ &= \sum_{x\in\mathbb{F}_d}\sum_{y\in\mathbb{F}_d} \omega^{-\beta x}\omega^{\beta y}|x\rangle\langle x|y\rangle\langle y|. \end{aligned}$$

Using the orthogonality condition $\langle x|y\rangle = \delta_{x,y}$ and noting that $\omega^{-\beta x}\omega^{\beta x} = 1$ we have

$$\begin{aligned} Z^{\beta\dagger}Z^{\beta} &= \sum_{x\in\mathbb{F}_d} \omega^{-\beta x}\omega^{\beta x}|x\rangle\langle x| \\ \implies Z^{\beta\dagger}Z^{\beta} &= \sum_{x\in\mathbb{F}_d} |x\rangle\langle x| = I. \end{aligned}$$

It follows with identical logic that $Z^{\beta}Z^{\beta\dagger} = I$. Hence Z^{β} is a unitary operator.

3. M_{γ} is defined as $M_{\gamma} \coloneqq \sum_{y\in\mathbb{F}_d} |\gamma y\rangle\langle y|$. The conjugate transpose is: $M_{\gamma}^{\dagger} \coloneqq \sum_{y\in\mathbb{F}_d} |y\rangle\langle\gamma y|$.

We need to show that $M_{\gamma}^{\dagger}M_{\gamma} = I$.

$$\begin{aligned} M_{\gamma}^{\dagger}M_{\gamma} &= \left(\sum_{y\in\mathbb{F}_d} |y\rangle\langle\gamma y|\right)\left(\sum_{z\in\mathbb{F}_d} |\gamma z\rangle\langle z|\right) \\ &= \sum_{y\in\mathbb{F}_d}\sum_{z\in\mathbb{F}_d} |y\rangle\langle\gamma y|\gamma z\rangle\langle z|. \end{aligned}$$

Using the orthogonality condition $\langle\gamma y|\gamma z\rangle = \delta_{y,z}$

$$M_{\gamma}^{\dagger}M_{\gamma} = \sum_{y\in\mathbb{F}_d} |y\rangle\langle y| = I.$$

It follows with identical logic that $M_\gamma M_\gamma^\dagger = I$. Hence M_γ is a unitary operator.

4. The operator F is defined as $F \coloneqq \frac{1}{\sqrt{d}} \sum_{x,z \in \mathbb{F}_d} \omega^{xz} |z\rangle\langle x|$. The conjugate transpose is $F^\dagger \coloneqq \frac{1}{\sqrt{d}} \sum_{x,z \in \mathbb{F}_d} \omega^{-xz} |x\rangle\langle z|$.

   We need to show that $F^\dagger F = I$.

$$\begin{aligned} F^\dagger F &= \left(\frac{1}{\sqrt{d}} \sum_{x',z' \in \mathbb{F}_d} \omega^{-x'z'} |x'\rangle\langle z'| \right) \left(\frac{1}{\sqrt{d}} \sum_{x,z \in \mathbb{F}_d} \omega^{xz} |z\rangle\langle x| \right) \\ &= \frac{1}{d} \sum_{x,z,x',z' \in \mathbb{F}_d} \omega^{-x'z'} \omega^{xz} |x'\rangle\langle z'|z\rangle\langle x|. \end{aligned}$$

   Using the orthogonality condition $\langle z|z'\rangle = \delta_{z,z'}$

$$F^\dagger F = \frac{1}{d} \sum_{x,x',z \in \mathbb{F}_d} \omega^{-x'z'} \omega^{xz} |x'\rangle\langle x|,$$

   simplifying the exponent,

$$F^\dagger F = \frac{1}{d} \sum_{x,x' \in \mathbb{F}_d} \left(\sum_{z \in \mathbb{F}_d} \omega^{z(x-x')} \right) |x'\rangle\langle x|.$$

   We know that

$$\sum_{z \in \mathbb{F}_d} \omega^{z(x-x')} = \begin{cases} d & \text{if } x = x' \\ 0 & \text{if } x \neq x' \end{cases}$$

   therefore

$$F^\dagger F = \frac{1}{d} \sum_{x,x' \in \mathbb{F}_d} d\delta_{x,x'} |x'\rangle\langle x| = \sum_{x \in \mathbb{F}_d} |x\rangle\langle x| = I.$$

   It follows with identical logic that $FF^\dagger = I$. Hence F is a unitary operator.

5. The operator $CX^{(i,j)}$ is defined as $CX^{(i,j)} \coloneqq \sum_{x,y \in \mathbb{F}_d} |x\rangle_i\langle x|_i \otimes |x+y\rangle_j\langle y|_j$. The conjugate transpose is $CX^{(i,j)\dagger} \coloneqq \sum_{x,y \in \mathbb{F}_d} |x\rangle_i\langle x|_i \otimes |y\rangle_j\langle x+y|_j$.

   We need to show that $CX^{(i,j)\dagger} CX^{(i,j)} = I$.

$$CX^{(i,j)\dagger} CX^{(i,j)} = \left(\sum_{x,y \in \mathbb{F}_d} |x\rangle_i\langle x|_i \otimes |y\rangle_j\langle x+y|_j \right) \left(\sum_{x',y' \in \mathbb{F}_d} |x'\rangle_i\langle x'|_i \otimes |x'+y'\rangle_j\langle y'|_j \right)$$

   ,

   expanding the product,

$$CX^{(i,j)\dagger} CX^{(i,j)} = \sum_{x,y,x',y' \in \mathbb{F}_d} \left(|x\rangle_i\langle x|_i \otimes |y\rangle_j\langle x+y|_j \right) \left(|x'\rangle_i\langle x'|_i \otimes |x'+y'\rangle_j\langle y'|_j \right).$$

   Using the distributive property of the tensor product and the inner product, we get

$$CX^{(i,j)\dagger} CX^{(i,j)} = \sum_{x,y,x',y' \in \mathbb{F}_d} \left(|x\rangle_i\langle x|_i|x'\rangle_i\langle x'|_i \right) \otimes \left(|y\rangle_j\langle x+y|_j|x'+y'\rangle_j\langle y'|_j \right).$$

Using the orthogonality condition $\langle x|x'\rangle = \delta_{x,x'}$ and $\langle y|y'\rangle = \delta_{y,y'}$

$$CX^{(i,j)\dagger}CX^{(i,j)} = \sum_{x,y\in\mathbb{F}_d} |x\rangle_i\langle x|_i \otimes |y\rangle_j\langle y|_j = I.$$

It follows with identical logic that $CX^{(i,j)}CX^{(i,j)\dagger} = I$. Hence $CX^{(i,j)}$ is a unitary operator.

We have shown the operators are unitary as $UU^\dagger = U^\dagger U = I$ holds in each case. □

There is one more operator we need to define but it is different depending on whether d is odd or even. If d is odd

$$P_\gamma \coloneqq \sum_{y\in\mathbb{F}_d} \omega^{-\frac{1}{2}\gamma y^2}|y\rangle\langle y|$$

Lemma: P_γ is unitary when d is odd.

Proof:

P_γ is defined as $P_\gamma \coloneqq \sum_{y\in\mathbb{F}_d} \omega^{-\frac{1}{2}\gamma y^2}|y\rangle\langle y|$. The conjugate transpose, $P_\gamma^\dagger \coloneqq \sum_{y\in\mathbb{F}_d} \omega^{\frac{1}{2}\gamma y^2}|y\rangle\langle y|$.

We need to show that $P_\gamma^\dagger P_\gamma = I$.

$$P_\gamma^\dagger P_\gamma = \left(\sum_{y\in\mathbb{F}_d} \omega^{\frac{1}{2}\gamma y^2}|y\rangle\langle y|\right)\left(\sum_{z\in\mathbb{F}_d} \omega^{-\frac{1}{2}\gamma z^2}|z\rangle\langle z|\right),$$

expanding the product

$$P_\gamma^\dagger P_\gamma = \sum_{y,z\in\mathbb{F}_d} \omega^{\frac{1}{2}\gamma y^2}\omega^{-\frac{1}{2}\gamma z^2}|y\rangle\langle y|z\rangle\langle z|.$$

Using the orthogonality $\langle y|z\rangle = \delta_{y,z}$

$$P_\gamma^\dagger P_\gamma = \sum_{y\in\mathbb{F}_d} \omega^{\frac{1}{2}\gamma y^2}\omega^{-\frac{1}{2}\gamma y^2}|y\rangle\langle y|.$$

Since $\omega^{\frac{1}{2}\gamma y^2}\omega^{-\frac{1}{2}\gamma y^2} = 1$

$$P_\gamma^\dagger P_\gamma = \sum_{y\in\mathbb{F}_d} |y\rangle\langle y| = I.$$

It follows with identical logic that $P_\gamma P_\gamma^\dagger = I$. Hence P_γ is unitary. □

The arithmetic in $\mathbb{F}_d$ is determined by operations modulo p^m. This field is an extension of the base field $\mathbb{F}_p$, which contains p elements. If p is odd, then p^m is also odd, and division by 2 is possible because 2 has a multiplicative inverse in $\mathbb{F}_p$ and therefore in $\mathbb{F}_{p^m}$. This means that the term $\frac{1}{2}\gamma y^2$ is well-defined. If $p = 2$, then $d = 2^m$, in this case the field $\mathbb{F}_{2^m}$ is a binary field, and division by 2 is not possible because there is no element $x \in \mathbb{F}_{2^m}$ such that $2x = 1$. In this field, $2x = 0$ for any x, making $\frac{1}{2}$ undefined. This means that $\frac{1}{2}\gamma y^2$ is not valid, hence, we need a new definition in the event that d is even [28].

Firstly, let's define a new function. Let $d = 2^m$ and let $B = \{b_1, \ldots, b_m\}$ be an arbitrary self-dual basis of $\mathbb{F}_d$ over $\mathbb{F}_2$. Furthermore, we define an integer-valued function on $\mathbb{F}_d$ as

$$\text{wgt} : \mathbb{F}_d \to \mathbb{Z},\ \alpha \mapsto |\{j : j \in \{1, 2, \ldots, m\} \mid \text{tr}(\alpha b_j) \neq 0\}|.$$

Then the following holds for all $\alpha, y \in \mathbb{F}_d$

$$i^{\text{wgt}(y+\alpha)} = i^{\text{wgt}(\alpha)} i^{\text{wgt}(y)} (-1)^{\text{tr}(\alpha y)}.$$

The function wgt assigns an integer value to each element α in the finite field $\mathbb{F}_d$, where $d = 2^m$. The value of $\text{wgt}(\alpha)$ is the number of basis elements b_j (from the self-dual basis $B = \{b_1, b_2, \ldots, b_m\}$) for which the trace $\text{tr}(\alpha b_j)$ is non-zero. The function, wgt, takes an element α from the finite field $\mathbb{F}_d$ and outputs an integer, which represents the *weight* of α.

Now we define the first component of P_γ

$$P_1 := \sum_{y \in \mathbb{F}_d} (-i)^{\text{wgt}(y)} |y\rangle\langle y| = \sum_{y \in \mathbb{F}_d} \prod_{j=1}^{m} (-i)^{\text{tr}(y b_j)} |y\rangle\langle y|.$$

Then, for any $\gamma \in \mathbb{F}_d \backslash \{0\}$, we define P_γ as

$$P_\gamma := M_{\gamma_0}^{-1} P_1 M_{\gamma_0} \quad \text{where } \gamma_0^2 = \gamma.$$

(Note that γ_0 is uniquely defined as $x \mapsto x^2$ is an automorphism of $\mathbb{F}_d = \mathbb{F}_{2^m}$.)

Lemma: P_γ is unitary when d is even.

Proof:

To demonstrate that P_γ is unitary all we need to do is very that P_1 is unitary as we have already shown M_γ to be.

To show that the operator P_1 is unitary, we need $P_1^\dagger P_1 = P_1 P_1^\dagger = I$.

The operator P_1 is defined as

$$P_1 := \sum_{y \in \mathbb{F}_{2^m}} (-i)^{\text{wgt}(y)} |y\rangle\langle y| = \sum_{y \in \mathbb{F}_{2^m}} \prod_{j=1}^{m} (-i)^{\text{tr}(y b_j)} |y\rangle\langle y|,$$

and the adjoint is

$$P_1^\dagger = \sum_{y \in \mathbb{F}_{2^m}} (i)^{\text{wgt}(y)} |y\rangle\langle y|.$$

Next, we calculate the product $P_1^\dagger P_1$:

$$P_1^\dagger P_1 = \left(\sum_{y \in \mathbb{F}_{2^m}} (i)^{\text{wgt}(y)} |y\rangle\langle y| \right) \left(\sum_{z \in \mathbb{F}_{2^m}} (-i)^{\text{wgt}(z)} |z\rangle\langle z| \right).$$

Since the states $|y\rangle$ are orthonormal (i.e., $\langle y|z\rangle = \delta_{y,z}$), this simplifies to

$$P_1^\dagger P_1 = \sum_{y\in\mathbb{F}_{2^m}} (i)^{\mathrm{wgt}(y)}(-i)^{\mathrm{wgt}(y)}|y\rangle\langle y|.$$

Now, we simplify the product $(i)^{\mathrm{wgt}(y)} \cdot (-i)^{\mathrm{wgt}(y)}$:

$$(i)^{\mathrm{wgt}(y)} \cdot (-i)^{\mathrm{wgt}(y)} = (i \cdot -i)^{\mathrm{wgt}(y)} = (-i^2)^{\mathrm{wgt}(y)} = (-(-1))^{\mathrm{wgt}(y)} = 1^{\mathrm{wgt}(y)} = 1.$$

For any $y \in \mathbb{F}_{2^m}$, the product of the phase factors equals 1, regardless of the value of $\mathrm{wgt}(y)$. Since each term $(i)^{\mathrm{wgt}(y)}(-i)^{\mathrm{wgt}(y)} = 1$, the sum simplifies to:

$$P_1^\dagger P_1 = \sum_{y\in\mathbb{F}_{2^m}} |y\rangle\langle y| = I,$$

Since $P_1^\dagger P_1 = I$ and $P_1 P_1^\dagger = I$ by identical logic, the operator P_1 is unitary and hence, so is P_γ when d is even. □

4.3.2 Some useful results

We have defined several unitary qudit operators for the construction of U. Now, we need to understand how they affect the X and Z type Pauli operators via conjugation. Firstly, let's consider the conjugation of the qudit Fourier transform operator F.

$$\begin{aligned} F^\dagger Z^\beta F &= \frac{1}{\sqrt{d}} \sum_{i,j\in\mathbb{F}_d} \omega^{-ij}|i\rangle\langle j| \sum_{z\in\mathbb{F}_d} \omega^{\beta z}|z\rangle\langle z| \frac{1}{\sqrt{d}} \sum_{k,l\in\mathbb{F}_d} \omega^{kl}|k\rangle\langle l| \\ &= \frac{1}{d} \sum_{i,l\in\mathbb{F}_d} \sum_{j\in\mathbb{F}_d} \omega^{-ij}\omega^{\beta j}\omega^{jl}|i\rangle\langle l| \\ &= \frac{1}{d} \sum_{i,l\in\mathbb{F}_d} \sum_{l\in\mathbb{F}_d} \omega^{j(l-i+\beta)}|i\rangle\langle l| \\ &= \sum_{l\in\mathbb{F}_d} |l+\beta\rangle\langle l| = X^\beta. \end{aligned} \tag{4.5}$$

$$\begin{aligned} F^\dagger X^\alpha F &= \frac{1}{\sqrt{d}} \sum_{i,j\in\mathbb{F}_d} \omega^{-ij}|i\rangle\langle j| \sum_{x\in\mathbb{F}_d} |x+\alpha\rangle\langle x| \frac{1}{\sqrt{d}} \sum_{k,l\in\mathbb{F}_d} \omega^{kl}|k\rangle\langle l| \\ &= \frac{1}{d} \sum_{i,l\in\mathbb{F}_d} \sum_{x\in\mathbb{F}_d} \omega^{-i(x+\alpha)}\omega^{xl}|i\rangle\langle l| \\ &= \frac{1}{d} \sum_{i,l\in\mathbb{F}_d} \omega^{-i\alpha} \sum_{x\in\mathbb{F}_d} \omega^{x(l-i)}|i\rangle\langle l| \\ &= \sum_{i\in\mathbb{F}_d} \omega^{-i\alpha}|i\rangle\langle i| = Z^{-\alpha}. \end{aligned} \tag{4.6}$$

Same for M_γ,

$$\begin{aligned} M_\gamma^\dagger X^\alpha Z^\beta M_\gamma &= \sum_{y\in\mathbb{F}_d} |y\rangle\langle\gamma y| \sum_{x\in\mathbb{F}_d} |x+\alpha\rangle\langle x| \sum_{z\in\mathbb{F}_d} \omega^{\beta z}|z\rangle\langle z| \sum_{v\in\mathbb{F}_d} |\gamma v\rangle\langle v| \\ &= \sum_{y\in\mathbb{F}_d} |y\rangle\langle\gamma y| \sum_{x\in\mathbb{F}_d} |x+\alpha\rangle\langle x| \sum_{v\in\mathbb{F}_d} \omega^{\beta\gamma v}|\gamma y\rangle\langle v| \\ &= \sum_{y\in\mathbb{F}_d} |\gamma^{-1}y\rangle\langle y| \sum_{v\in\mathbb{F}_d} \omega^{\beta\gamma v}|\gamma v+\alpha\rangle\langle v| \\ &= \sum_{v\in\mathbb{F}_d} \omega^{\beta\gamma v}|v+\gamma^{-1}\alpha\rangle\langle v| \\ &= \sum_{x\in\mathbb{F}_d} |x+\gamma^{-1}\alpha\rangle\langle x| \sum_{z\in\mathbb{F}_d} \omega^{\beta\gamma z}|z\rangle\langle z| \\ &= X^{\gamma^{-1}\alpha} Z^{\gamma\beta}. \end{aligned} \tag{4.7}$$

For odd d and P_γ, we note that it commutes with Z^β. Furthermore, X^α acts on $P_\gamma^\dagger$ as follows:

$$\begin{aligned} X^{\alpha\dagger} P_\gamma^\dagger X^\alpha &= \sum_{x\in\mathbb{F}_d} |x\rangle\langle x+\alpha| \sum_{y\in\mathbb{F}_d} \omega^{\frac{1}{2}\gamma y^2}|y\rangle\langle y| \sum_{z\in\mathbb{F}_d} |z+\alpha\rangle\langle z| \\ &= \sum_{y\in\mathbb{F}_d} \omega^{\frac{1}{2}\gamma(y+\alpha)^2}|y\rangle\langle y| \\ &= \omega^{\frac{1}{2}\gamma\alpha^2} \sum_{z\in\mathbb{F}_d} \omega^{\gamma\alpha z}|z\rangle\langle z| \sum_{y\in\mathbb{F}_d} \omega^{\frac{1}{2}\gamma y^2}|y\rangle\langle y| \\ &= \omega^{\frac{1}{2}\gamma\alpha^2} Z^{\gamma\alpha} P_\gamma^\dagger. \end{aligned} \tag{4.8}$$

Similarly, we have $P_\gamma^\dagger X^\alpha P_\gamma = \omega^{\frac{1}{2}\gamma\alpha^2} X^\alpha Z^{\gamma\alpha}$.

For even d, the matrix M_γ is a permutation matrix, hence P_γ is diagonal and commutes with Z_β. The action of P_γ on X_α is given by

$$\begin{aligned} P_\gamma^\dagger X_\alpha P_\gamma &= M_{\gamma_0}^\dagger P_1^\dagger M_{\gamma_0} X_\alpha M_{\gamma_0}^\dagger P_1 M_{\gamma_0} \\ &= M_{\gamma_0}^\dagger P_1^\dagger X_\alpha \gamma_0 P_1 M_{\gamma_0} \\ &= M_{\gamma_0}^\dagger i^{\mathrm{wgt}(\alpha\gamma_0)} X_\alpha \gamma_0 Z_{\alpha\gamma_0} M_{\gamma_0} \\ &= i^{\mathrm{wgt}(\alpha\gamma_0)} X_\alpha Z_{\alpha\gamma_0^2} \\ &= i^{\mathrm{wgt}(\alpha\gamma_0)} X_\alpha Z_{\alpha\gamma}. \end{aligned} \tag{4.9}$$

For $CX^{(i,j)}$, we see it trivially commutes with qudit pairs of the form $Z^\alpha \otimes X^\beta$ and for qudit pairs

of the form $X^\alpha \otimes Z^\beta$, we get

$$\begin{aligned} CX^{(i,j)\dagger}(X^\alpha \otimes Z^\beta)CX^{(i,j)} &= CX^{(i,j)\dagger} \sum_{v,w\in\mathbb{F}_d} \omega^{\beta w}|v+\alpha\rangle_i\langle w|_j\langle w|_j\langle v|_i \sum_{x,y\in\mathbb{F}_d} |x\rangle_i\langle x+y|_j\langle y|_j\langle x|_i \\ &= CX^{(i,j)\dagger} \sum_{x,y\in\mathbb{F}_d} \omega^{\beta(x+y)}|x+\alpha\rangle_i\langle x+y|_j\langle y|_j\langle x|_i \\ &= \sum_{v,w\in\mathbb{F}_d} |v\rangle_i\langle w|_j\langle w+w|_j\langle v|_i \sum_{x,y\in\mathbb{F}_d} \omega^{\beta(x+y)}|x+\alpha\rangle_i\langle x+y|_j\langle y|_j\langle x|_i \\ &= \sum_{x,y\in\mathbb{F}_d} \omega^{\beta x}\omega^{\beta y}|x+\alpha\rangle_i\langle y-\alpha\rangle_j\langle y|_j\langle x|_i \\ &= (X^\alpha Z^\beta) \otimes (X^{-\alpha} Z^\beta). \end{aligned} \tag{4.10}$$

4.3.3 Action of the Operators

Let's define $\mathcal{J}_{\mathbb{F}_d} := \langle F, P_\gamma, M_\gamma : \gamma \in \mathbb{F}_d^* \rangle$. The action of the group $\mathcal{J}_{\mathbb{F}_d}$ on $\overline{\mathcal{P}_1^d} := \frac{\mathcal{P}_1^d}{Z(\mathcal{P}_1^d)}$ is $SL(2,\mathbb{F}_d)$. This is not immediately obvious but, if we consider a qudit $XZ(\alpha|\beta)$ where $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\beta = (\beta_1, \ldots, \beta_n)$ where (α_i, β_i) are the powers of X_i and Z_i respectively. Then, we can define a set of operators $\{\overline{F}, \overline{P_\gamma}, \overline{M_\gamma}\}$, namely

$$\overline{F} := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \quad \overline{P_\gamma} := \begin{pmatrix} 1 & \gamma \\ 0 & 1 \end{pmatrix}, \quad \overline{M_\gamma} := \begin{pmatrix} \gamma^{-1} & 0 \\ 0 & \gamma \end{pmatrix} \tag{4.11}$$

such that for $\overline{J} \in \{\overline{F}, \overline{P_\gamma}, \overline{M_\gamma}\}$ we have that $(\alpha, \beta)\overline{J}$ is equivalent to $J^\dagger(XZ(\alpha|\beta))J$.

The proof then follows naturally as the action of $\overline{F}$, $\overline{P_\gamma}$ and $\overline{M_\gamma}$ on $\overline{\mathcal{P}_1^d}$ generates $SL(2,\mathbb{F}_d)$. We also know that $\mathcal{J}_{\mathbb{F}_d}$ acts transitively on the non-trivial elements of $\overline{\mathcal{P}_1^d}$ [29] i.e. the group action of $\mathcal{J}_{\mathbb{F}_d}$ on $\overline{\mathcal{P}_1^d}$ only has one orbit. In simpler terms, this means that the group $SL(2,\mathbb{F}_d)$ can transform any non-zero vector in the vector space $\mathbb{F}_d^2$ to any other non-zero vector in $\mathbb{F}_d^2$ through its group actions. We have roughly shown that with one qudit, we can transform any qudit into another; this is the first required component for our *matrix reduction*. We now need to define the matrix representation of CX which is similar but, will act on pairs of qudits, $(\alpha_i, \beta_i, \alpha_j, \beta_j)$. This is defined as

$$\overline{CX}^{(i,j)} := \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}. \tag{4.12}$$

Now, we have the capability to transform multiple qudits and can proceed to the algorithm for computing U.

4.3.4 Algorithm

For an abelian stabiliser group $\mathcal{S}$ we show how to construct $U \in \mathcal{C}_n^d$ such that up to some qudit permutation π,

$$U^\dagger \mathcal{S} U =_\pi \langle Z_1, ..., Z_k \rangle \tag{4.13}$$

The algorithm [28] to perform the matrix reduction and thus, compute U is shown below.

Algorithm 1 Stabiliser Matrix Reduction

Require: a stabiliser matrix $(X|Z) \in \mathbb{F}_d^{k \times 2n}$

$L \leftarrow \emptyset$
for each row $i = 1$ to k **do**
  for each column $j = 1$ to n **do**
    if $(X_{ij}, Z_{ij}) \neq (0, 0)$ **then**
      find a transformation $T_{ij} \in \mathcal{J}_{\mathbb{F}_d}$ such that $(X_{ij}, Z_{ij}) \cdot \overline{T_{ij}} = (1, 0)$
      for each row $l = 1$ to k **do**
        $(X_{lj}, Z_{lj}) \leftarrow (X_{lj}, Z_{lj}) \cdot \overline{T_{ij}}$
      end for
    else
      $T_{ij} \leftarrow id$
    end if
  end for
  $T_i \leftarrow T_{i,1}^{(1)} \otimes T_{i,2}^{(2)} \otimes \ldots \otimes T_{i,n}^{(n)}$
  find the first column $l \notin L$ where $X_{il} \neq 0$
  include l into L
  $A_i \leftarrow id$
  for each column $j = 1$ to n **do**
    if $l \notin L$ and $X_{ij} = 1$ **then**
      $A_i \leftarrow A_i \cdot CX^{(l,j)}$
      for each row $\mu = i$ to k **do**
        $(X_{\mu j}, Z_{\mu j}) \leftarrow (X_{\mu j} - X_{lj}, Z_{\mu j} + Z_{lj})$
      end for
    end if
  end for
end for
$F \leftarrow \prod_{l \in L} F_l$
return $U = T_1 A_1 T_2 A_2 \ldots T_k A_k F$

4.3.5 Elucidation

It might not be obvious how to follow this algorithm however, let's break it down into more simple components for clairfication and then compute an example of the stabiliser matrix reduction. It is important to observe that the algorithm works by reducing the stabiliser matrix to be all zeroes aside from 1s along the diagonal in the X block then, we apply the Fourier transforms to shift the diagonal ones to the Z block. A natural question to ask is why we bother applying the Fourier transform as the previous steps can be modified very easily to work with X-type operators. The answer lies in the physical considerations for encoding qudits (if we only care about theory it is more efficient to use X's as we need fewer gates). Z-type operators are typically easier to implement in physical systems as

(I) Z operators are diagonal in the computational basis, making them simpler to represent mathematically [30].

(II) In physical implementations like superconducting circuits, Z rotations can often be implemented virtually using frame changes, without requiring physical pulses [31].

(III) For multi-level systems like transmons, the energy levels naturally correspond to the computational basis states, making Z operations more straightforward [31].

Proceeding with the simplification, given a qudit parametrisation of the form $(\alpha|\beta)$ our operators have the following mapping

1. $F : (\alpha|\beta) \to (\beta| - \alpha)$
2. $M_\gamma : (\alpha|\beta) \to (\alpha\gamma^{-1}|\beta\gamma)$
3. $P_\gamma : (\alpha|\beta) \to (\alpha|\alpha\gamma + \beta)$
4. $CX^{(i,j)} : ((\alpha_i, \beta_i), (\alpha_j, \beta_j)) \to ((\alpha_i, \beta_i + \beta_j), (\alpha_j - \alpha_i, \beta_j))$

If we consider a component of a stabiliser matrix given by

$$\left(\begin{array}{c|c} 0 & 2 \\ 1 & 1 \\ 2 & 2 \\ 1 & 2 \end{array}\right). \tag{4.14}$$

We know the first step is to convert the 0 to a 1 in the X block and convert the 2 to a 0 in the Z block. To do this we apply M_2H which has the action of mapping $(\alpha|\beta) \to (2^{-1}\beta|-\alpha) = (2\beta|-2\alpha)$,

$$M_2H : \left(\begin{array}{c|c} 0 & 2 \\ 1 & 1 \\ 2 & 2 \\ 1 & 2 \end{array}\right) \to \left(\begin{array}{c|c} 1 & 0 \\ 2 & 1 \\ 1 & 2 \\ 1 & 1 \end{array}\right). \tag{4.15}$$

4.3.6 Example

Let's work through a larger example to demonstrate how the algorithm works now we have a better understanding of each individual component. We consider an example over qutrits (qudits when $d = 3$). A stabiliser matrix of the code $\mathcal{C} = [9, 5, 3]_3$ is given by

$$(X|Z) = \left(\begin{array}{ccccccccc|ccccccccc} 1 & 0 & 0 & 2 & 1 & 2 & 2 & 0 & 1 & 0 & 0 & 2 & 1 & 2 & 2 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 & 0 & 2 & 2 & 1 & 0 & 0 & 0 & 1 & 2 & 1 & 1 & 0 & 2 & 2 \\ 0 & 0 & 2 & 1 & 2 & 2 & 0 & 1 & 1 & 1 & 0 & 2 & 0 & 0 & 1 & 2 & 1 & 2 \\ 0 & 0 & 1 & 2 & 1 & 1 & 0 & 2 & 2 & 0 & 1 & 2 & 1 & 1 & 0 & 2 & 0 & 2 \end{array}\right). \tag{4.16}$$

Transform each pair (α_i, β_i) to $(1, 0)$ using

$$T_1 = I \otimes I \otimes M_2F \otimes P_1M_2 \otimes P_1 \otimes P_2M_2 \otimes M_2F \otimes P_2. \tag{4.17}$$

The resulting stabiliser matrix is

$$(X|Z) = \left(\begin{array}{ccccccccc|ccccccccc} 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 2 & 1 & 1 & 0 & 2 & 2 \\ 0 & 0 & 1 & 2 & 2 & 1 & 0 & 1 & 1 & 1 & 0 & 2 & 2 & 2 & 1 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 & 1 & 2 & 0 & 0 & 2 & 0 & 1 & 1 & 0 & 2 & 1 & 1 & 1 & 0 \end{array}\right). \tag{4.18}$$

Remove the 1s in the first row of the X block using

$$A_1 := CX^{(1,3)}CX^{(1,4)}CX^{(1,5)}CX^{(1,6)}CX^{(1,7)}CX^{(1,8)}CX^{(1,9)} \tag{4.19}$$

we obtain

$$(X|Z) = \left(\begin{array}{ccccccccc|ccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 2 & 1 & 1 & 0 & 2 & 2 \\ 0 & 0 & 1 & 2 & 2 & 1 & 0 & 1 & 1 & 0 & 0 & 2 & 2 & 2 & 1 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 & 1 & 2 & 0 & 0 & 2 & 0 & 1 & 1 & 0 & 2 & 1 & 1 & 1 & 0 \end{array}\right). \tag{4.20}$$

We now proceed in a similar fashion for the remaining rows until they are all zero aside from the diagonal 1s. We begin with the second row using the transformations

$$T_2 := I \otimes I \otimes P_1M_2 \otimes P_1 \otimes F \otimes P_2 \otimes I \otimes P_3M_2 \otimes M_2F \tag{4.21}$$

and

$$A_2 := CX^{(2,3)}CX^{(2,4)}CX^{(2,5)}CX^{(2,6)}CX^{(2,7)}CX^{(2,8)}CX^{(2,9)} \tag{4.22}$$

to obtain

$$(X|Z) = \left(\begin{array}{ccccccccc|ccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 2 & 1 & 0 & 2 & 2 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 2 & 1 \\ 0 & 0 & 2 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 2 & 1 & 2 & 2 \end{array}\right). \tag{4.23}$$

Now we use the transformations

$$T_3 := I \otimes I \otimes M_2 \otimes P_1 M_2 \otimes P_1 M_2 \otimes I \otimes F \otimes P_3 M_2 \otimes P_1 M_2 \tag{4.24}$$

and

$$A_3 := CX^{(3,4)} CX^{(3,5)} CX^{(3,6)} CX^{(3,7)} CX^{(3,8)} CX^{(3,9)} \tag{4.25}$$

to obtain

$$(X|Z) = \left(\begin{array}{ccccccccc|ccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 2 & 2 & 0 & 0 & 0 & 1 & 2 & 2 & 0 & 1 & 1 \end{array}\right). \tag{4.26}$$

For the last row, we use the transformations

$$T_4 := I \otimes I \otimes P_2 \otimes M_2 F \otimes P_1 \otimes I \otimes P_1 M_2 \otimes P_1 M_2 \tag{4.27}$$

and

$$A_4 := CX^{(4,5)} CX^{(4,6)} CX^{(4,8)} CX^{(4,9)} \tag{4.28}$$

to obtain

$$(X|Z) = \left(\begin{array}{ccccccccc|ccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right). \tag{4.29}$$

We can subtract the third row from the fourth to obtain to remove the superfluous 1 as row addition corresponds to generator multiplication. A Fourier transformation on the first four indices transforms the stabiliser group into the form $\{Z_1, Z_2, Z_3, Z_4\}$ exactly as we desire.

4.4 Overview

We proved, via von Neumann's Bicommutant Theorem that if we can find a commutative, unital self-adjoint subalgebra of $\mathcal{B}(\mathcal{H})$ which commutes with the first k qudits in a Hamiltonian then, we can *taper* those qudits, namely we can block diagonalise the Hamiltonian while preserving the original symmetries. We also concluded that that the subalgebra $\mathcal{Z}$ which is generated by $\{Z_1, \dots Z_k\}$ satisfies these conditions. Then we determined an algorithm to find a unitary $U \in \mathcal{C}_n^d$ which maps a given Hamiltonian to a new basis preserving its eigenspaces in such a way that the first k qudits commute with $\mathcal{Z}$. Ergo, enabling the block diagonalisation, or *tapering*, of the first k components of the Hamiltonian. It is also worth recalling that the original qubit approach restricted the generators of the symmetry group to Z type only. Hence, we have actually extended their method in the qubit case too (when $d = 2$), as there are no restrictions on the type of Pauli operators the generators can contain in this tapering method.

Chapter 5

Quantum Clock Model Example

Theory only takes you so far; to see the effectiveness of the tapering method let's apply it to a real example, in particular the Quantum Clock Model [16] which is a generalisation of the transverse-field Ising model .

5.1 Hamiltonian and Stabiliser Group

The Hamiltonian is defined as

$$H = -J\left(\sum_{\langle i,j\rangle}^{n}(Z_i^\dagger Z_j + Z_i Z_j^\dagger) + g\sum_{i=1}^{n}(X_i + X_i^\dagger)\right) \tag{5.1}$$

where X_i and Z_i are the d-dimensional qudit Pauli operators and J and g are scalars. We can equivalently write this Hamiltonian as a single sum of scaled Pauli operators $\sigma_i \in \mathcal{P}_n^d$

$$H = \sum_{i=1}^{m} h_i \sigma_i. \tag{5.2}$$

We want to compute the stabiliser group $\mathcal{S}$, of this Hamiltonian; if we recall we need $\mathcal{S}$ to be an abelian subgroup of $\mathcal{P}_n^d$, where any element of $\mathcal{S}$ will commute with each Pauli term in H and $-I \notin \mathcal{S}$. We claim $\mathcal{S} = \langle P\rangle \cong \mathbb{Z}_d$ where $P = \prod_{i=1}^{n} X_i$.

Proof.

Trivially $\langle P\rangle$ is abelian subgroup of $\mathcal{P}_n^d$ as all elements are X-type Pauli and P commutes with $\sum_i (X_i + X_i^\dagger)$ by the same logic. It is also easy to see that $-I \notin \langle P\rangle$.

All that is left to do is show $\langle P\rangle$ commutes with $\sum_{\langle i,j\rangle}(Z_i^\dagger Z_j + Z_i Z_j^\dagger)$. Noting that,

$$X_i Z_i^a X_i^\dagger = \omega^a Z_i^a \implies P Z_i^a P^\dagger = \omega^a Z_i^a \tag{5.3}$$

where $\omega = \exp(2\pi i/d)$ and $a \in \mathbb{Z}_d$. We also have,

$$\begin{aligned} PZ_i^\dagger Z_j P^\dagger &= PZ_i^\dagger P^\dagger PZ_j P^\dagger = \omega^{-1+1} Z_i^\dagger Z_j = Z_i^\dagger Z_j \\ PZ_i Z_j^\dagger P^\dagger &= PZ_i P^\dagger PZ_j^\dagger P^\dagger = \omega^{1-1} Z_i Z_j^\dagger = Z_i Z_j^\dagger \end{aligned} \tag{5.4}$$

So, P commutes with $\sum_{\langle i,j \rangle}(Z_i^\dagger Z_j + Z_i Z_j^\dagger)$ which implies $\langle P \rangle$ commutes with H. Hence, $\mathcal{S} = \langle P \rangle$ is the stabiliser group of the Quantum Clock Model Hamiltonian.

5.2 Minimal Generating Set

Before we proceed to the matrix reduction algorithm to construct U we need to ensure the generating set is minimal. Given that the generating set has size 1 it is already minimal so we can skip this step.

5.3 Constructing U

Now we need to find a unitary $U \in \mathcal{C}_n^d$ such that $U^\dagger P U = Z_1$. Given our generating set has size 1 the stabiliser matrix will only have 1 row. It follows that no matter the value of d the stabiliser matrix will take the form

$$\Big(\underbrace{1 \quad 1 \quad \dots \quad 1}_{n} \Big| \underbrace{0 \quad 0 \quad \dots \quad 0}_{n} \Big). \tag{5.5}$$

We want to convert all the columns to 0 aside from the first one. To do this we apply

$$A = \mathrm{CX}^{(1,2)} \mathrm{CX}^{(1,3)} \dots \mathrm{CX}^{(1,n)}. \tag{5.6}$$

Now the stabiliser matrix takes the form

$$\Big(\underbrace{1 \quad 0 \quad \dots \quad 0}_{n} \Big| \underbrace{0 \quad 0 \quad \dots \quad 0}_{n} \Big). \tag{5.7}$$

All that's left to do is apply F_1. Giving $U = AF_1$ where, $U^\dagger P U = Z_1$.

5.4 Transformed Hamiltonian when $n = 2$

To visualise a concrete example, let's consider the case when $n = 2$ note, the value of d has no impact on this result. To taper qudits we need to transform the Hamiltonian as follows $H \to U^\dagger H U$.

Here, the Hamiltonian is given by

$$H = \sum_i h_i \sigma_i = h_1 Z_1^\dagger Z_2 + h_2 Z_1 Z_2^\dagger + h_3 X_1 + h_4 X_1^\dagger + h_5 X_2 + h_6 X_2^\dagger \tag{5.8}$$

and stabilised by $\langle X_1X_2\rangle$. Writing $CX^{(1,2)}$ as CX our unitary is $U = CXF_1$. Let's first verify that $U^\dagger(X_1X_2)U = Z_1$

$$F_1^\dagger CX^\dagger(X_1X_2)CXF_1 = F_1^\dagger X_1F_1 = Z_1. \tag{5.9}$$

This is as we want. Now, we need to transform each of the terms in the Hamiltonian - we will ignore the scalars as they are preserved exactly by the conjugation.

1. For $\sigma_1 = Z_1^\dagger Z_2$ let's first compute $CX^\dagger(Z_1^\dagger Z_2)CX$, we can do this by considering the action of $CX^\dagger(Z_1^\dagger Z_2)CX$ on a basis state $|x\rangle|y\rangle$

$$\begin{aligned} CX^\dagger(Z_1^\dagger Z_2)CX|x\rangle|y\rangle &= CX^\dagger(Z_1^\dagger Z_2)|x\rangle|x+y\rangle \\ &= \omega^{-x}\omega^{x+y}CX^\dagger|x\rangle|x+y\rangle \\ &= \omega^y|x\rangle|(x+y)-x\rangle \\ &= Z_2|x\rangle|y\rangle. \end{aligned}$$

   It follows naturally that,

$$F_1^\dagger CX^\dagger(Z_1^\dagger Z_2)CXF_1 = Z_2. \tag{5.10}$$

2. For $\sigma_2 = Z_1Z_2^\dagger$ let's first compute $CX^\dagger(Z_1Z_2^\dagger)CX$, we can do this by considering the action of $CX^\dagger(Z_1Z_2^\dagger)CX$ on a basis state $|x\rangle|y\rangle$

$$\begin{aligned} CX^\dagger(Z_1Z_2^\dagger)CX|x\rangle|y\rangle &= CX^\dagger(Z_1Z_2^\dagger)|x\rangle|x+y\rangle \\ &= \omega^x\omega^{-(x+y)}CX^\dagger|x\rangle|x+y\rangle \\ &= \omega^{-y}|x\rangle|(x+y)-x\rangle \\ &= Z_2^\dagger|x\rangle|y\rangle. \end{aligned}$$

   It follows naturally that,

$$F_1^\dagger CX^\dagger(Z_1Z_2^\dagger)CXF_1 = Z_2^\dagger. \tag{5.11}$$

3. For $\sigma_3 = X_1$ let's first compute $CX^\dagger(X_1)CX$, we can do this by considering the action of $CX^\dagger(X_1)CX$ on a basis state $|x\rangle|y\rangle$

$$\begin{aligned} CX^\dagger(X_1)CX|x\rangle|y\rangle &= CX^\dagger(X_1)|x\rangle|x+y\rangle \\ &= CX^\dagger|x+1\rangle|x+y\rangle \\ &= |x+1\rangle|(x+y)-(x+1)\rangle \\ &= |x+1\rangle|y-1\rangle \\ &= X_1X_2^\dagger|x\rangle|y\rangle. \end{aligned}$$

   It follows naturally that,

$$F_1^\dagger CX^\dagger(X_1)CXF_1 = Z_1X_2^\dagger. \tag{5.12}$$

4. For $\sigma_4 = X_1^\dagger$ let's first compute $CX^\dagger(X_1^\dagger)CX$, we can do this by considering the action of $CX^\dagger(X_1^\dagger)CX$ on a basis state $|x\rangle|y\rangle$

$$\begin{aligned} CX^\dagger(X_1^\dagger)CX|x\rangle|y\rangle &= CX^\dagger(X_1^\dagger)|x\rangle|x+y\rangle \\ &= CX^\dagger|x-1\rangle|x+y\rangle \\ &= |x-1\rangle|(x+y)-(x-1)\rangle \\ &= |x-1\rangle|y+1\rangle \\ &= X_1^\dagger X_2|x\rangle|y\rangle. \end{aligned}$$

It follows naturally that,

$$F_1^\dagger CX^\dagger(X_1^\dagger)CXF_1 = Z_1^\dagger X_2. \tag{5.13}$$

5. For $\sigma_5 = X_2$ let's first compute $CX^\dagger(X_2)CX$, we can do this by considering the action of $CX^\dagger(X_2)CX$ on a basis state $|x\rangle|y\rangle$

$$\begin{aligned} CX^\dagger(X_2)CX|x\rangle|y\rangle &= CX^\dagger(X_2)|x\rangle|x+y\rangle \\ &= CX^\dagger|x\rangle|x+y+1\rangle \\ &= |x\rangle|(x+y+1)-x\rangle \\ &= |x\rangle|y+1\rangle \\ &= X_2|x\rangle|y\rangle. \end{aligned}$$

It follows naturally that,

$$F_1^\dagger CX^\dagger(X_2)CXF_1 = X_2. \tag{5.14}$$

6. For $\sigma_6 = X_2^\dagger$ let's first compute $CX^\dagger(X_2^\dagger)CX$, we can do this by considering the action of $CX^\dagger(X_2^\dagger)CX$ on a basis state $|x\rangle|y\rangle$

$$\begin{aligned} CX^\dagger(X_2^\dagger)CX|x\rangle|y\rangle &= CX^\dagger(X_2^\dagger)|x\rangle|x+y\rangle \\ &= CX^\dagger|x\rangle|x+y-1\rangle \\ &= |x\rangle|(x+y-1)-x\rangle \\ &= |x\rangle|y-1\rangle \\ &= X_2^\dagger|x\rangle|y\rangle. \end{aligned}$$

It follows naturally that,

$$F_1^\dagger CX^\dagger(X_2^\dagger)CXF_1 = X_2^\dagger. \tag{5.15}$$

In summary,

$$\begin{aligned} F_1^\dagger CX^\dagger (Z_1^\dagger Z_2) CXF_1 &= Z_2 \\ F_1^\dagger CX^\dagger (Z_1 Z_2^\dagger) CXF_1 &= Z_2^\dagger \\ F_1^\dagger CX^\dagger (X_1) CXF_1 &= Z_1 X_2^\dagger \\ F_1^\dagger CX^\dagger (X_1^\dagger) CXF_1 &= Z_1^\dagger X_2 \\ F_1^\dagger CX^\dagger (X_2) CXF_1 &= X_2 \\ F_1^\dagger CX^\dagger (X_2) CXF_1 &= X_2^\dagger. \end{aligned} \tag{5.16}$$

Now, considering the terms in the transformed Hamiltonian we see that operators acting on the first qudit are either identity or Z-type, that is to say they commute with Z_1. This is exactly what we aimed for in the original construction. Hence, we can taper the first qudit from the Hamiltonian in a simulation of the quantum clock model.

Let's actually compute the block corresponding to projection onto $|0\rangle$, in other words let's compute the 0 sector. Reintroducing the scalars we have the following Hamiltonian terms

$$\sigma_1 = -JI_1Z_2,\ \ \sigma_2 = -JI_1Z_2^\dagger,\ \ \sigma_3 = -JgZ_1X_2^\dagger,\ \ \sigma_4 = -JgZ_1^\dagger X_2,\ \ \sigma_5 = -JgI_1X_2,\ \ \sigma_6 = -JgI_1X_2^\dagger. \tag{5.17}$$

Now let's compute the projections in each case, implementing the procedure outlined in Section 3.5

1. $-J\langle 0|I_1|0\rangle Z_2 = -JZ_2$
2. $-J\langle 0|I_1|0\rangle Z_2^\dagger = -JZ_2^\dagger$
3. $-Jg\langle 0|Z_1|0\rangle X_2^\dagger = -JgX_2^\dagger$
4. $-Jg\langle 0|Z_1^\dagger|0\rangle X_2 = -JgX_2$
5. $-Jg\langle 0|I_1|0\rangle X_2 = -JgX_2$
6. $-Jg\langle 0|I_1|0\rangle X_2^\dagger = -JgX_2^\dagger$

Now, we combine all the projected terms to form zero sector block, namely

$$H_0 = -JZ_2 - JZ_2^\dagger - JgX_2^\dagger - JgHX_2 - JgX_2 - JgX_2^\dagger = -J(Z_2 + Z_2^\dagger + 2g(X_2 + X_2^\dagger)). \tag{5.18}$$

To complete the full block diagonal version of the Hamiltonian we would repeat the process for $|1\rangle$ and $|2\rangle$ giving H_1 and H_2 respectively then these blocks would form the diagonal of the Hamiltonian,

$$H = \begin{pmatrix} H_0 & 0 & 0 \\ 0 & H_1 & 0 \\ 0 & 0 & H_2 \end{pmatrix}. \tag{5.19}$$

5.5 General n

Given the previous result, it is natural to consider what happens in a more general case for arbitrary n.

We already showed for arbitrary n the general form of U is $U = AF_1$ where $A = CX^{(1,2)}...CX^{(1,n)}$, as the stabiliser of the Hamiltonian is generated by only one element. Referring back to the general form of the Hamiltonian, $H = -J\left(\sum_{\langle i,j\rangle}^{n}(Z_i^\dagger Z_j + Z_i Z_j^\dagger) + \sum_{i=1}^{n} g(X_i + X_i^\dagger)\right)$ we are interested in the result of $U^\dagger Z_i^\dagger Z_j U$, $U^\dagger Z_i^\dagger Z_j^\dagger U$, $U^\dagger X_i U$ and $U^\dagger X_i^\dagger U$ for $i \neq j$. We will compute the result for each Pauli term in both the $i = 1$ and $i \neq 1$ cases.

1. Let's begin with $\sigma = Z_i^\dagger Z_j$ and $i = 1$. Similar to the previous example it's more straightforward to begin with the action of $A^\dagger(Z_1^\dagger Z_j)A$ on some basis state $|x_1\rangle|x_2\rangle\ldots|x_n\rangle$

$$\begin{aligned} A^\dagger(Z_1^\dagger Z_j)A|x_1\rangle\ldots|x_n\rangle &= A^\dagger(Z_1^\dagger Z_j)|x_1\rangle|x_2+x_1\rangle\ldots|x_n+x_1\rangle \\ &= \omega^{-x_1}\omega^{x_1+x_j}A^\dagger|x_1\rangle|x_2+x_1\rangle\ldots|x_n+x_1\rangle \\ &= \omega^{x_j}|x_1\rangle|(x_2+x_1)-x_1\rangle\ldots|(x_n+x_1)-x_1\rangle \\ &= Z_j|x_1\rangle|x_2\rangle\ldots|x_n\rangle. \end{aligned}$$

   It follows naturally that,

$$F_1^\dagger A^\dagger(Z_1^\dagger Z_j)AF_1 = F_1^\dagger(Z_j)F_1 = Z_j \quad \text{for } i = 1.$$

   For $i \neq 1$ we proceed in exactly the same way

$$\begin{aligned} A^\dagger(Z_i^\dagger Z_j)A|x_1\rangle\ldots|x_n\rangle &= A^\dagger(Z_i^\dagger Z_j)|x_1\rangle|x_2+x_1\rangle\ldots|x_i+x_1\rangle\ldots|x_n+x_1\rangle \\ &= \omega^{-(x_i+x_1)}\omega^{x_1+x_j}A^\dagger|x_1\rangle|x_2+x_1\rangle\ldots|x_i+x_1\rangle\ldots|x_n+x_1\rangle \\ &= \omega^{x_j-x_i}|x_1\rangle|(x_2+x_1)-x_1\rangle\ldots|(x_n+x_1)-x_1\rangle \\ &= Z_i^\dagger Z_j|x_1\rangle|x_2\rangle\ldots|x_n\rangle. \end{aligned}$$

   It follows that,

$$F_1^\dagger A^\dagger(Z_i^\dagger Z_j)AF_1 = F_1^\dagger(Z_i^\dagger Z_j)F_1 = Z_i^\dagger Z_j \quad \text{for } i \neq 1. \tag{5.20}$$

2. Now for $\sigma = Z_i Z_j^\dagger$ and $i = 1$. Similar to the previous example it's more straightforward to begin with the action of $A^\dagger(Z_1 Z_j^\dagger)A$ on some basis state $|x_1\rangle|x_2\rangle\ldots|x_n\rangle$

$$\begin{aligned} A^\dagger(Z_1 Z_j^\dagger)A|x_1\rangle\ldots|x_n\rangle &= A^\dagger(Z_1 Z_j^\dagger)|x_1\rangle|x_2+x_1\rangle\ldots|x_n+x_1\rangle \\ &= \omega^{x_1}\omega^{-(x_1+x_j)}A^\dagger|x_1\rangle|x_2+x_1\rangle\ldots|x_n+x_1\rangle \\ &= \omega^{-x_j}|x_1\rangle|(x_2+x_1)-x_1\rangle\ldots|(x_n+x_1)-x_1\rangle \\ &= Z_j^\dagger|x_1\rangle|x_2\rangle\ldots|x_n\rangle. \end{aligned}$$

It follows that,

$$F_1^\dagger A^\dagger (Z_1 Z_j^\dagger) A F_1 = F_1^\dagger (Z_j^\dagger) F_1 = Z_j^\dagger \quad \text{for } i = 1. \tag{5.21}$$

For $i \neq 1$ we proceed in exactly the same way

$$\begin{aligned} A^\dagger (Z_i Z_j^\dagger) A |x_1\rangle \dots |x_n\rangle &= A^\dagger (Z_i Z_j^\dagger) |x_1\rangle |x_2 + x_1\rangle \dots |x_i + x_1\rangle \dots |x_n + x_1\rangle \\ &= \omega^{x_i + x_1} \omega^{-(x_1 + x_j)} A^\dagger |x_1\rangle |x_2 + x_1\rangle \dots |x_i + x_1\rangle \dots |x_n + x_1\rangle \\ &= \omega^{x_i - x_j} |x_1\rangle |(x_2 + x_1) - x_1\rangle \dots |(x_n + x_1) - x_1\rangle \\ &= Z_i Z_j^\dagger |x_1\rangle |x_2\rangle \dots |x_n\rangle. \end{aligned}$$

It follows that,

$$F_1^\dagger A^\dagger (Z_1 Z_j^\dagger) A F_1 = F_1^\dagger (Z_i Z_j^\dagger) F_1 = Z_i Z_j^\dagger \quad \text{for } i \neq 1. \tag{5.22}$$

3. Now for $\sigma = X_i$ and $i = 1$. Similar to the previous example it's more straightforward to begin with the action of $A^\dagger(X_1)A$ on some basis state $|x_1\rangle|x_2\rangle \dots |x_n\rangle$

$$\begin{aligned} A^\dagger (X_1) A |x_1\rangle \dots |x_n\rangle &= A^\dagger (X_1) |x_1\rangle |x_2 + x_1\rangle \dots |x_n + x_1\rangle \\ &= A^\dagger |x_1 + 1\rangle |x_2 + x_1\rangle \dots |x_n + x_1\rangle \\ &= |x_1 + 1\rangle |(x_2 + x_1) - (x_1 + 1)\rangle \dots |(x_n + x_1) - (x_1 + 1)\rangle \\ &= |x_1 + 1\rangle |x_2 - 1\rangle \dots |x_n - 1\rangle \\ &= X_1 X_2^\dagger X_3^\dagger \dots X_n^\dagger |x_1\rangle |x_2\rangle \dots |x_n\rangle \\ &= X_1 \prod_{j=2}^{n} X_j^\dagger |x_1\rangle \dots x_n\rangle. \end{aligned}$$

It follows that,

$$F_1^\dagger A^\dagger (X_1) A F_1 = F_1^\dagger (X_1 \prod_{j=2}^{n} X_j^\dagger) F_1 = Z_1 \prod_{j=2}^{n} X_j^\dagger \quad \text{for } i = 1. \tag{5.23}$$

For $i \neq 1$ we proceed in exactly the same way

$$\begin{aligned} A^\dagger (X_i) A |x_1\rangle \dots |x_n\rangle &= A^\dagger (X_i) |x_1\rangle |x_2 + x_1\rangle \dots |x_n + x_1\rangle \\ &= A^\dagger |x_1\rangle |x_2 + x_1\rangle \dots |x_i + x_1 + 1\rangle \dots |x_n + x_1\rangle \\ &= |x_1\rangle |(x_2 + x_1) - x_1\rangle \dots |x_i + x_1 + 1 - x_1\rangle \dots |(x_n + x_1) - x_1\rangle \\ &= |x_1\rangle |x_2\rangle \dots |x_i + 1\rangle \dots |x_n\rangle \\ &= X_i |x_1\rangle |x_2\rangle \dots |x_n\rangle. \end{aligned}$$

It follows that,

$$F_1^\dagger A^\dagger (X_i) A F_1 = F_1^\dagger (X_i) F_1 = X_i \quad \text{for } i \neq 1. \tag{5.24}$$

4. Now for $\sigma = X_i^\dagger$ and $i = 1$. Similar to the previous example it's more straightforward to

begin with the action of $A^\dagger(X_1^\dagger)A$ on some basis state $|x_1\rangle|x_2\rangle\dots|x_n\rangle$

$$\begin{aligned}
A^\dagger(X_1^\dagger)A|x_1\rangle\dots|x_n\rangle &= A^\dagger(X_1^\dagger)|x_1\rangle|x_2+x_1\rangle\dots|x_n+x_1\rangle \\
&= A^\dagger|x_1-1\rangle|x_2+x_1\rangle\dots|x_n+x_1\rangle \\
&= |x_1-1\rangle|(x_2+x_1)-(x_1-1)\rangle\dots|(x_n+x_1)-(x_1-1)\rangle \\
&= |x_1-1\rangle|x_2+1\rangle\dots|x_n+1\rangle \\
&= X_1^\dagger X_2 X_3\dots X_n|x_1\rangle|x_2\rangle\dots|x_n\rangle \\
&= X_1^\dagger\prod_{j=2}^{n}X_j|x_1\rangle\dots x_n\rangle.
\end{aligned}$$

It follows that,

$$F_1^\dagger A^\dagger(X_1^\dagger)AF_1 = F_1^\dagger(X_1^\dagger\prod_{j=2}^{n}X_j)F_1 = Z_1^\dagger\prod_{j=2}^{n}X_j \quad \text{for } i=1. \tag{5.25}$$

For $i\neq 1$ we proceed in exactly the same way

$$\begin{aligned}
A^\dagger(X_i^\dagger)A|x_1\rangle\dots|x_n\rangle &= A^\dagger(X_i^\dagger)|x_1\rangle|x_2+x_1\rangle\dots|x_n+x_1\rangle \\
&= A^\dagger|x_1\rangle|x_2+x_1\rangle\dots|x_i+x_1-1\rangle\dots|x_n+x_1\rangle \\
&= |x_1\rangle|(x_2+x_1)-x_1\rangle\dots|x_i+x_1-1-x_1\rangle\dots|(x_n+x_1)-x_1\rangle \\
&= |x_1\rangle|x_2\rangle\dots|x_i-1\rangle\dots|x_n\rangle \\
&= X_i^\dagger|x_1\rangle|x_2\rangle\dots|x_n\rangle
\end{aligned}$$

It follows that,

$$F_1^\dagger A^\dagger(X_i^\dagger)AF_1 = F_1^\dagger(X_i^\dagger)F_1 = X_i^\dagger \quad \text{for } i\neq 1. \tag{5.26}$$

To summarise,

$$\begin{aligned}
F_1^\dagger CX^\dagger(Z_i^\dagger Z_j)CXF_1 &= \begin{cases} F_1^\dagger(Z_j)F_1 = Z_j & \text{for } i=1 \\ F_1^\dagger(Z_i^\dagger Z_j)F_1 = Z_i^\dagger Z_j & \text{for } i\neq 1 \end{cases} \\
F_1^\dagger CX^\dagger(Z_iZ_j^\dagger)CXF_1 &= \begin{cases} F_1^\dagger(Z_j^\dagger)F_1 = Z_j^\dagger & \text{for } i=1 \\ F_1^\dagger(Z_iZ_j^\dagger)F_1 = Z_iZ_j^\dagger & \text{for } i\neq 1 \end{cases} \\
F_1^\dagger CX^\dagger(X_i)CXF_1 &= \begin{cases} F_1^\dagger(X_1\prod_{j=2}^{n}X_j^\dagger)F_1 = Z_1\prod_{j=2}^{n}X_j^\dagger & \text{for } i=1 \\ F_1^\dagger(X_i)F_1 = X_i & \text{for } i\neq 1 \end{cases} \\
F_1^\dagger CX^\dagger(X_i^\dagger)CXF_1 &= \begin{cases} F_1^\dagger(X_1^\dagger\prod_{j=2}^{n}X_j)F_1 = Z_1^\dagger\prod_{j=2}^{n}X_j & \text{for } i=1 \\ F_1^\dagger(X_i^\dagger)F_1 = X_i^\dagger & \text{for } i\neq 1 \end{cases}
\end{aligned} \tag{5.27}$$

Once again, considering the terms in the transformed Hamiltonian we see that operators acting on the first qudit are either identity or Z-type, namely they commute with Z_1. This is exactly what we aimed for in the original construction. Hence, we can taper the first qudit from the general Hamiltonian in a simulation of the Quantum Clock Model.

5.6 Simulation

To complete the example, we simulate [32] both the original form of the Hamiltonian and the block diagonal version in Python. We fix $d = 3, J = 1$ and vary the number of qudits from $2 - 7$ as well as the parameter g which is known as the *coupling strength*. In each case, we compute the condition number, $\kappa = \frac{\|\lambda_{\max}\|}{\|\lambda_{\min}\|}$. For greater values of d and n the computation time becomes extremely long but, we can already deduce a result from the original values. Note, we add a small number, $\varepsilon = 10^{-8}$ along the diagonal of both matrices to improve numerical stability; if the smallest eigenvalue is close to 0 it can cause the condition number to explode. The results are shown on pages $52 - 53$.

Interestingly we see near identical results for both simulations. Despite adding the ε term the condition number explodes to around 10^7 in some situations but remains relatively stable, between 10 and 1000, in the other cases. The actual values for this brief comparison are not that significant instead the most important conclusion we can draw is that condition number is near identical (up to some computational error) for both scenarios. Despite performing the block diagonalisation and reducing the number of qudits required to simulate the Hamiltonian we have managed to preserve the condition number which plays an important role in the time complexity of many classical and quantum algorithms [33]. Fundamentally, it experimentally shows that the tapering process and block diagonalisation preserves the eigenvalues of the system. We will analyse and justify this mathematically in the next chapter.

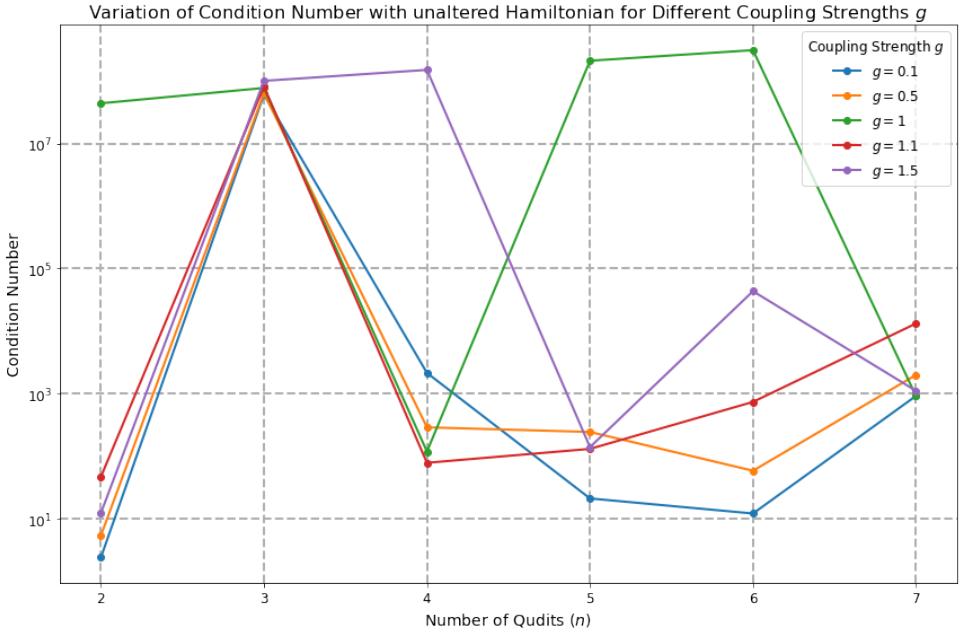

Figure 5.1: Original Hamiltonian Simulation Results

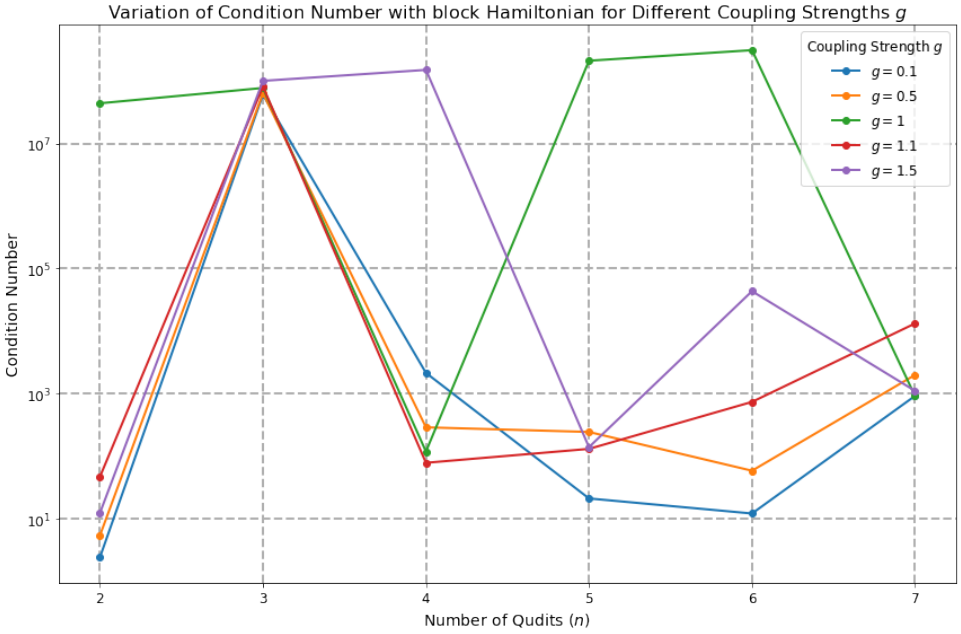

Figure 5.2: Block Hamiltonian Simulation Results

Chapter 6

Space and Time Complexity

A method to taper qudits is only as powerful as the time it takes produce the result. It is important for us to analyse the time complexity of the U construction; we know the only other computationally intensive portion is conjugating the Hamiltonian by U but, this is limited by the size of the system. In addition, we will also look at the impact of tapering on a selection of quantum solvers.

6.1 Complexity of the U construction

The time complexity to construct U was first determined in [28]. The justification is simple so, for completeness, it is included in full. For a stabiliser matrix, $(X|Z) \in \mathbb{F}_d^{k\times 2n}$ there are at most three nested *for* loops; one iterates over the n columns and the other two are over the k rows. Each of the k transformations A_j consists of at most j CX-gates. Therefore, the total number of CX gates is at most $nk - \binom{k+1}{2}$. Each of the k transformations T_j comprises $O(n)$ single qudit operations. Including the k local Fourier transformations, the total number of single qudit gates is $O(nk)$.

6.2 Impact on Quantum Algorithms

We want to consider the impact that tapering qudits has on the space and time complexity of quantum algorithms. To do this we make reference to the specific task outlined in [33] and the end-to-end complexities of a variety of quantum algorithms applied to this problem. Ideally, we want to find that tapering qudits does not detrimentally impact run time or gate complexity.

The aforementioned task is as follows, let A be an $N \times N$ Hermitian matrix which can be classically decomposed into the Pauli basis

$$A = \sum_{\ell=1}^{L} a_\ell P_\ell \tag{6.1}$$

where $a_\ell \in \mathbb{R}$ and P_ℓ is a multiple qubit operator. Recall, setting $d = 2$ in our approach is a more general version of the historical qubit tapering method so, in this section *only*, we use qubit and qudit interchangeably. Let $\vec{b}$ an N-dimensional vector, we aim to approximate the element $\frac{1}{c}.\left(A^{-1}\vec{b}\right)_i$ to additive error ε, for some choice of $i \in [N]$ and normalisation $c \in \mathbb{R}$. Let's analyse several solutions and the end-to-end complexity before and after tapering.

6.2.1 Quantum Linear Systems Solver (QLSS)

The quantum linear systems solver (QLSS) is an algorithm that can be used to provide a solution to our task. It does not exactly compute the result but instead returns the quantum state proportional to $A^{-1}\vec{b}$, where $\vec{b} := \sum_i b_i|i\rangle/\|\vec{b}\|$. Since the original QLSS algorithm [34], known colloquially as HHL, there have been many improvements utilising $\mathcal{O}(\log(N))$ algorithmic qubits.

We consider the current most optimised form of the QLSS [35]. A high-level description of the algorithm, as given in the paper, is as follows:

1. The matrix A is described by a block encoding and the vector b is described by a unitary operation to prepare $|b\rangle$.
2. From those descriptions, we construct a block encoding of a Hamiltonian $H(s)$, where an eigenstate of $H(0)$ is $|b\rangle$ and an eigenstate of $H(1)$ is the solution $|A^{-1}b\rangle$.
3. The unitary operator for the block encoding of $H(s)$, together with a reflection on the control ancilla, gives the (unitary) qubitised walk operator $W(s)$.
4. The discrete sequence of walk operators $W(s)$ for s varying from 0 to 1 is shown to give a good overlap with the solution $|A^{-1}b\rangle$ via the discrete adiabatic theorem.
5. At the end, we apply filtering via a linear combination of powers of $W(1)$ in order to obtain $|A^{-1}b\rangle$ with precision ε. If there is failure of the filtering, then the adiabatic evolution is repeated.

We observe the following results [33] for the untapered scenario.

Data Access Space	**Algorithmic Space**	**AE?**	**Gate Depth**	**Total Runtime**	**Norm**
$\mathcal{O}\left(N^2\right)$ qubits	$\log N + 6$ qubits	yes	$\widetilde{\mathcal{O}}\left(\kappa_F \log(N)/\epsilon\right)$	$\widetilde{\mathcal{O}}\left(\kappa_F \log(N)/\epsilon\right)$	$\|A^{-1}\vec{b}\|$
		no	$\widetilde{\mathcal{O}}\left(\kappa_F \log(N)\log(1/\epsilon)\right)$	$\widetilde{\mathcal{O}}\left(\kappa_F \log(N)/\epsilon^2\right)$	$\|A^{-1}\vec{b}\|$

Table 6.1: QLSS Performance Metrics

To be certain the tapering procedure does not have a detrimental effect we need to consider how it impacts the value of $\kappa_F = \|A^{-1}\|\|A\|_F$ where $\|*\|_F$ is the Frobenius norm defined as $\|M\|_F = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}|m_{ij}|} = \sqrt{\mathrm{Tr}(MM^\dagger)}$ and $\|*\|$ is the operator norm defined as $\|M\| = \sup_{\|x\|=1}\|Mx\|$.

As A is Hermitian, the block diagonalisation, which occurs as a result of tapering, is equivalent to conjugation by a unitary [36]. Thus, for the unitary U which performs this change of basis it is easy to see that

$$\|A_{\text{block}}\|_F = \|UA_{\text{block}}U^\dagger\|_F = \|A\|_F. \tag{6.2}$$

Similarly, it is a known result [37] that block diagonalisation does not change the eigenvalues of a matrix so the qudit tapering process will not change the operator norm either.

As κ_F remains unchanged by the tapering process the QLSS algorithm's gate depth and runtime is identical before and after tapering.

6.2.2 Randomised Quantum Solver

The work in [33] creates a novel technique to solve the task mentioned above. Recall, we want to approximate $\frac{1}{c}.\left(A^{-1}\vec{b}\right)_i$. We provide their summary of the algorithm not to understand but just to give an overview and provide some context for the complexity analysis. For the complete picture please refer to their paper.

Consider a Hermitian matrix A with known Pauli decomposition as in Eq. 6.1 with Pauli weight $\lambda \coloneqq \sum|a_\ell|$. Denote q as a freely chosen normalisation parameter. Finally, suppose we have the ability to prepare state $|\vec{b}\rangle$ in $\mathcal{O}(d_{\vec{b}})$ depth. Then, we find:

(a) Given the ability to implement $|\psi\rangle$ via unitary U_ψ in gate depth d_ψ, we have a randomised quantum algorithm to approximate

$$\frac{\langle\psi|A^{-1}|\vec{b}\rangle}{q}$$

up to additive error ϵ, with arbitrary constant success probability, utilising C^{ψ}_{sample} quantum circuits each consisting of C^{ψ}_{gate} layers of gates, where

$$C^{\psi}_{\text{sample}} = \widetilde{\mathcal{O}}\left(\frac{\|A^{-1}\|^2}{\epsilon^2 q^2}\right), \quad C^{\psi}_{\text{gate}} = \widetilde{\mathcal{O}}\left(\|A^{-1}\|^2\lambda^2 + d_\psi + d_{\vec{b}}\right).$$

(b) Given the ability to measure observable O; $\|O\| \leq 1$, we have a randomised quantum algorithm to approximate

$$\frac{\langle\vec{b}|A^{-1}OA^{-1}|\vec{b}\rangle}{q^2}$$

up to additive error ϵ, with arbitrary constant probability, utilising C^{O}_{sample} quantum circuits each consisting of C^{O}_{gate} layers of gates and one measurement of O, where

$$C^{O}_{\text{sample}} = \widetilde{\mathcal{O}}\left(\frac{\|A^{-1}\|^4}{\epsilon^2 q^4}\right), \quad C^{O}_{\text{gate}} = \widetilde{\mathcal{O}}\left(\|A^{-1}\|^2\lambda^2 + d_\psi + d_{\vec{b}}\right).$$

Similar to the previous case we have the tabulated complexity.

Data Access Space	**Algorithmic Space**	**AE?**	**Gate Depth**	**Total Runtime**	**Norm**
$\mathcal{O}\left(N^2\right)$ bits	$\log N + 2$ qubits	no	$\widetilde{\mathcal{O}}\left(\|A^{-1}\|^2\lambda^2\log^2(1/\varepsilon)\right)$	$\widetilde{\mathcal{O}}\left(\|A^{-1}\|^4\lambda^2/\varepsilon^2\right)$	$\|\vec{b}\|$

Table 6.2: Randomised Quantum Performance Metrics

In this scenario the gate depth and run-time are dependent on $\|A^{-1}\|$ and λ^2. We have already demonstrated that the operator norm is not affected by the tapering process so we just need to consider the impact it has on the Pauli coefficients. Recall, we transform the Hamiltonian (or matrix A in this notation) by unitary conjugation; the unitary is chosen in such a way that the generators of the symmetry group become Z-type Pauli matrices. However, this has no impact on the Pauli coefficients. For A as previously defined, some scalars h_j, unitary $U \in \mathcal{C}_N^b$ and $\sigma_j \in \mathcal{P}_N^b$ observe,

$$\begin{aligned} U^\dagger A U &= U^\dagger \left(\sum_{j=1}^{r} h_j \sigma_j \right) U \\ &= \sum_{j=1}^{r} h_j U^\dagger \sigma_j U \\ &= \sum_{j=1}^{r} h_j \tau_j \end{aligned} \tag{6.3}$$

where $\tau_j \in \mathcal{P}_N^b$ as $U \in \mathcal{C}_N^b$. Given that h_j has not been altered, the Pauli weight pre-unitary conjugation is equal to the Pauli weight post-unitary conjugation. Hence, the qudit tapering approach reduces the spatial complexity as overall k fewer qudits are required but, there is no compromise in terms of gate depth or total runtime. We can run quantum algorithms as efficiently but, with reduced number of qudits.

Chapter 7

Conclusion

7.1 Overview

This work built an approach to taper qudits from Hermitian Hamiltonians.

Several theoretical results were constructed towards that goal. We began by providing fundamental abstract mathematics background; then extended this, developing theory around qubits and qudits - notably stabilisers and operators which underpin the approach. Next, we studied the historical qubit method [14, 15]. It was demonstrated that by finding an appropriate unitary operator belonging to the qubit Clifford Group, it was possible to transform a Hamiltonian in such a way that its first k qubits commute with the qubit Pauli X operator. Using this knowledge they were able to taper the first k qubits from a Hamiltonian and demonstrated this in an example with the H_2 molecule. It provided the motivation for the qudit approach however; we needed to rigorously justify *why* we can taper without symmetry loss.

Chapter 3 developed more abstract mathematics to justify the eventual qudit tapering approach; we needed to ensure it was possible to block diagonalise (taper) a Hamiltonian without losing its original symmetries. This began by defining a topological space and topologies on $\mathcal{B}(\mathcal{H})$ to eventually prove the essential result: the von Neumann Bicommutant Theorem. As a corollary to the theorem we know if we can construct a commutative, unital, self-adjoint subalgebra of $\mathcal{B}(\mathcal{H})$, namely $\mathcal{Z}$ and suppose $\mathcal{Z}$ commutes with the first k qudits of a Hamiltonian, H_{sim}, or a version of it under unitary conjugation. Then, we know $\mathcal{Z}$ forms a von Neumann algebra and by the von Neumann Bicommutant Theorem we can taper (block diagonalise) the first k qudits of the Hamiltonian. This gave us the mathematical justification to proceed with the qudit tapering method. Recognising that $\{Z_1, \ldots, Z_k\}$ forms a commutative, unital, self-adjoint subalgebra of $\mathcal{B}(\mathcal{H})$ the approach became clear. For a minimal symmetry group, $\langle g_1, g_2, \ldots, g_k \rangle$, of a Hamiltonian we need to find a unitary operator $U \in \mathcal{C}_n^d$ which sends each g_i to Z_i under conjugation. Hence, we can

transform the original Hamiltonian by this unitary and then block diagonalise; tapering the first k qudits while preserving symmetries.

Given the mathematical tools and justification defined in earlier chapters we were able to begin the qudit tapering method. We demonstrated that a Hamiltonian, under conjugation by the appropriate unitary, commutes with Z-type qudit Pauli operators. Then, we highlighted a simple method to ensure we have a minimal generating set for the symmetry group. The bulk of this section revolved around constructing the unitary which is certainly non-trivial - in fact, we were only able to do so when d is a prime power. Much of the work relied on operations within a field and it is a well known fact that finite fields with non-prime power cardinality do not exist. We adjusted a mechanism originally defined for computing stabiliser codes [28] to build U, after developing a significant amount of qudit operator theory. At this stage we had established how to taper qudits and mathematically verified the block diagonalisation.

Next, we utilised the tapering approach on a practical example - the Quantum Clock Model. We found that the stabiliser matrix always takes the form

$$\Big(\underbrace{1 \quad 1 \quad \dots \quad 1}_{n} \Big| \underbrace{0 \quad 0 \quad \dots \quad 0}_{n} \Big)$$

no matter the value of d. After computing the value of U to be $U = AF_1$ where $A = CX^{(1,2)}...CX^{(1,n)}$ and then deriving the transformed Hamiltonian we considered some tangible examples. Including a worked example of the zero sector block for $n = 2$ and $d = 3$ as well as some basic simulations comparing the original Hamiltonian with the block diagonal version in this simple case. We found that the condition number remains constant across the scenarios despite the reduced dimensionality; this was discussed further in the space and time complexity section.

Finally, we evaluated the space and time complexity of implementing the approach. We found the total number of two qudit gates is at most $nk - \binom{k+1}{2}$ (for n qudits and a symmetry generator of cardinality k) and the number of single qudit gates to be of order nk. To complete the analysis we calculated the impact on gate-depth and total runtime after tapering in the optimal quantum linear systems solver and randomised quantum solver. We demonstrated that the operator norm, Frobenius norm and Pauli weight remain constant; hence, the tapering process reduces dimensionality with no additional detriment to the gate depth or total runtime in these quantum solvers. The complexity of many solvers and quantum problems is proportional to these constants so the tapering method is likely to be feasible in other scenarios too.

7.2 The Future

An immediate extension is apparent, we are restricted to the case when d is a prime power and it would be useful to extend this to all d. However, given the aforementioned restrictions to finite fields this remains an open question.

Furthermore, there is a possible extension to different symmetries. For example, consider the Quantum Heisenberg Model with Hamiltonian

$$H = \sum_{\langle i,j \rangle} J_{ij}(X_i X_j + Y_i Y_j + Z_i Z_j) \tag{7.1}$$

where J_{ij} is some scalar, X_k, Z_k and $Y_k = iX_kZ_k$ are the Pauli operators acting on the k^{th} qubit. The symmetries of this system are generated by the SU(2) algebra [38], namely

$$u_1 = \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix}, \quad u_2 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \quad u_3 = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}. \tag{7.2}$$

Despite knowing the symmetries of the system it is unknown how to implement the projection of the Hamiltonian on the trivial SU(2) representation on a quantum computer. The projection of Hamiltonians onto alternative symmetries remains an interesting and more feasible problem to approach in the future.

Chapter 8

Ethical Considerations

After careful review, there are no ethical considerations to address in relation to this project.

Bibliography

[1] Heggie DC. The classical gravitational n-body problem. In *Encyclopedia of Mathematical Physics*. Elsevier, 2006. doi: 10.1016/B0-12-512666-2/00003-1.

[2] Shor PW. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J Comput*, 26(5):1484–509, Oct 1997. ISSN 1095-7111. doi: 10.1137/s0097539795293172.

[3] Grover LK. A fast quantum mechanical algorithm for database search. In *Proc 28th Annu ACM Symp Theory Comput*, pages 212–9. ACM, 1996. doi: 10.1145/237814.237866.

[4] Abrams DS and Lloyd S. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Phys Rev Lett*, 83(24):5162–5, Dec 1999. ISSN 1079-7114. doi: 10.1103/physrevlett.83.5162.

[5] Farhi E, Goldstone J, Gutmann S, Lapan J, Lundgren A, and Preda D. A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. *Science*, 292: 472–5, May 2001. ISSN 1095-9203. doi: 10.1126/science.1057726.

[6] Maniscalco S and Borrelli EM et al. Quantum network medicine: rethinking medicine with network science and quantum algorithms, 2022. URL `https://arxiv.org/abs/2206.12405`.

[7] Kadian K, Garhwal S, and Kumar A. Quantum walk and its application domains: A systematic review. *Comput Sci Rev*, 41:100419, 2021. ISSN 1574-0137. doi: 10.1016/j.cosrev.2021.100419.

[8] Robert A, Barkoutsos PK, Woerner S, and Tavernelli I. Resource-efficient quantum algorithm for protein folding. *NPJ Quantum Inf*, 7(1), Feb 2021. ISSN 2056-6387. doi: 10.1038/s41534-021-00368-4.

[9] Batra K and Zorn K et al. Quantum machine learning algorithms for drug discovery applications. *Journal of Chemical Information and Modeling*, 61(6):2641–7, Jun 2021. doi: 10.1021/acs.jcim.1c00166.

[10] Roth TE, Ma R, and Chew WC. The transmon qubit for electromagnetics engineers: An introduction. *IEEE Antennas Propag Mag*, 65(2):8–20, Apr 2023. ISSN 1558-4143. doi: 10.1109/map.2022.3176593.

[11] Bruzewicz CD, Chiaverini J, McConnell R, and Sage JM. Trapped-ion quantum computing: Progress and challenges. *Appl Phys Rev*, 6(2), May 2019. ISSN 1931-9401. doi: 10.1063/1.5088164.

[12] Torma P and Stenholm S. Quantum logic using polarized photons. *Phys. Rev. A*, 54:4701–6, Dec 1996. doi: 10.1103/PhysRevA.54.4701.

[13] Gottesman D. An introduction to quantum error correction and fault-tolerant quantum computation, 2009. URL `https://arxiv.org/abs/0904.2557`.

[14] Bravyi S, Gambetta JM, Mezzacapo A, and Temme K. Tapering off qubits to simulate fermionic hamiltonians, 2017. URL `https://arxiv.org/abs/1701.08213`.

[15] Setia K and Chen R et al. Reducing qubit requirements for quantum simulations using molecular point group symmetries. *J Chem Theory Comput*, 16(10):6091–7, Aug 2020. ISSN 1549-9626. doi: 10.1021/acs.jctc.0c00113.

[16] Offei-Danso A, Surace FM, Iemini F, Russomanno A, and Fazio R. Quantum clock models with infinite-range interactions. *J Stat Mech Theory Exp*, 2020(7):073107, Jul 2020. ISSN 1742-5468. doi: 10.1088/1742-5468/aba0a1.

[17] Serge Lang. *Algebra*. Springer, New York, NY, 2002. ISBN 9781461300410 146130041X.

[18] Werner O. Amrein. *Hilbert Space Methods in Quantum Mechanics*. EPFL Press, Lausanne, Switzerland, 2009. ISBN 9781439827376.

[19] Nielsen MA and Chuang IL. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[20] Wang Y, Hu Z, Sanders BC, and Kais S. Qudits and high-dimensional quantum computing. *Front Phys*, 8, 2020. ISSN 2296-424X. doi: 10.3389/fphy.2020.589504.

[21] Hostens E, Dehaene J, and De Moor B. Stabilizer states and clifford operations for systems of arbitrary dimensions and modular arithmetic. *Phys Rev A*, 71(4), Apr 2005. ISSN 1094-1622. doi: 10.1103/physreva.71.042315.

[22] Brechet S, Reuse F, Maschke K, and Ansermet JP. Magnetoelectric effect in a hydrogen molecule. *Mod Phys Lett B*, 28:1450070, Apr 2014. doi: 10.1142/S0217984914500705.

[23] Klco N, Beck DH, and Savage MJ. Entanglement structures in quantum field theories: Negativity cores and bound entanglement in the vacuum. *Phys Rev A*, 107:012415, Jan 2023. doi: 10.1103/PhysRevA.107.012415.

[24] Attal S. *Elements of Operator Algebras and Modular Theory*, pages 69–105. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-33922-9. doi: 10.1007/3-540-33922-1_3.

[25] Takesaki M. *Theory of Operator Algebras II*. EMS Series on Operator Algebras. European Mathematical Society, 2002.

[26] Storjohann A. Near optimal algorithms for computing smith normal forms of integer matrices. *Proc ISSAC*, Jun 2000. doi: 10.1145/236869.237084.

[27] Zhong C, Wong Y, and Jiang L. Entanglement trimming in stabilizer formalism, 2021. URL `https://arxiv.org/abs/2103.09932`.

[28] Grassl M, Rottler M, and Beth T. Efficient quantum circuits for non-qubit quantum error-correcting codes. *Int J Found Comput Sci*, 14(5):757–75, Oct 2003. ISSN 1793-6373. doi: 10.1142/s0129054103002011.

[29] Huppert B. *Endliche Gruppen I*, volume 134 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag Berlin Heidelberg New York, 1967. Part of the book series: Grundlehren der mathematischen Wissenschaften (GL, volume 134).

[30] Das S and Caruso F. A hybrid-qudit representation of digital rgb images. *Sci Rep*, 13(1): 13671, 2023. doi: 10.1038/s41598-023-39906-9. Published 2023 Aug 22.

[31] Cao S and Bakr M et al. Emulating two qubits with a four-level transmon qudit for variational quantum algorithms. *Quantum Sci Technol*, 9(3):035003, Apr 2024. doi: 10.1088/2058-9565/ad37d4.

[32] George Hilton. Hamiltonian Sim Code v1.0.0, September 2024. URL `https://zenodo.org/doi/10.5281/zenodo.13730665`.

[33] Wang S, McArdle S, and Berta M. Qubit-efficient randomized quantum algorithms for linear algebra. *PRX Quantum*, 5(2), Apr 2024. ISSN 2691-3399. doi: 10.1103/prxquantum.5.020324.

[34] Harrow AW, Hassidim A, and Lloyd S. Quantum algorithm for linear systems of equations. *Phys Rev Lett*, 103:150502, Oct 2009. doi: 10.1103/PhysRevLett.103.150502.

[35] Costa PCS, An D, Sanders YR, Su Y, Babbush R, and Berry DW. Optimal scaling quantum linear-systems solver via discrete adiabatic theorem. *PRX Quantum*, 3:040303, Oct 2022. doi: 10.1103/PRXQuantum.3.040303.

[36] Shapiro H. Simultaneous block triangularization and block diagonalization of sets of matrices. *Linear Algebra Appl*, 25:129–37, 1979. ISSN 0024-3795. doi: 10.1016/0024-3795(79)90012-0.

[37] Eisenfeld J. Block diagonalization and eigenvalues. Technical Report 136, Mathematics Technical Papers, 1975. URL `https://mavmatrix.uta.edu/math_technicalpapers/136`.

[38] Beach KSD, Alet F, Mambrini M, and Capponi S. Quantum monte carlo study. *Phys Rev B*, 80(18), Nov 2009. ISSN 1550-235X. doi: 10.1103/physrevb.80.184401.