Towards Fault-Tolerant Quantum Computation with Higher-Dimensional Systems

By

Hussain Anwar

A thesis submitted to

University College London

for the degree of

Doctor of Philosophy

Department of Physics and Astronomy
University College London
February 23, 2014

I, Hussain Anwar confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

List of Publications and Preprints

The majority of the work presented in this thesis contains materials from the following publications:

H. Anwar, E.T. Campbell, and D.E. Browne, *Qutrit Magic State Distillation*, New J. Phys. **14**, 063006 (2012).

E.T. Campbell, H. Anwar, and D.E. Browne, *Magic-State Distillation in All Prime Dimensions Using Quantum Reed-Muller Codes*, Phys. Rev. X **2**, 4, 041021 (2012).

H. Anwar, B. Brown, E.T. Campbell, and D.E. Browne, *Efficient Decoders for the Qudit Toric Code*, Preprint arXiv:1311.4895.

Other publications:

M.J. Hoban, J.J. Wallman, H. Anwar, N. Usher, R. Raussendorf, and D.E. Browne, *On the hardness of sampling and measurement-based classical computation*, Preprint arXiv:1304.2667 (To appear in Physical Review Letters).

To my parents

Acknowledgements

First and foremost, I need to thank my supervisor Dan Browne for his guidance, patience and continuous support throughout my PhD. For the past four years, listening to his eloquent explanations and ideas was a truly enlightening experience.

I would like to thank my examiners Dr. Simon Benjamin and Dr. Jiannis Pachos for all their helpful comments and suggestions, and without whom this document could not have been in its current form.

I wish to thank all the staff members of the quantum information group at UCL for creating such a friendly and stimulating research environment. In particular, I must thank Sugato Bose (my secondary supervisor) and Alessio Serafani (Doctoral Training Program organiser) for their useful comments and help with my early research. I especially thank Earl Campbell, who is a collaborator on the work presented in this thesis, for all his help with my early research and for all the time he patiently spent to teach me the basics of quantum information theory. I also wish to thank Janet Anders for all the friendly conversations and for organising various events (such as the regular group meetings and the 3-day quantum technologies workshop) which has greatly contributed to having a very joyful experience in this group.

It is a great pleasure to thank my fellow PhD students for all the interesting chats and for creating such lively environment. I thank Matty Hoban for all his help during my first two years at UCL. I wish to thank Ben Brown (an Imperial based student, and a collaborator on some of the work presented in this thesis) for all the stimulating discussions we had on topological error correction. I thank Naïri Usher for the countless physics-related discussions (and arguments)—they were truly fun! I thank Tanapat Deesuwan (Ω) for all the board discussions and for allowing me to use his books and computer whenever I needed. I thank Fern Watson for exchanging interesting research ideas and for keeping up with my incoherent explanations. I also wish to thank many other PhD students and Post-Docs that I frequently had the pleasure to interact with; Chris Perry, Mercedes Gimeno-Segovia, Seto Balian, James Millen, Bobby Antonio, Hamed Mohammady, Lia Han, Ilhan Candan, Sai-Yun Ye, Edward O'Reilly, Shirwan Abdullah, George Pender, and Hoda Nejad.

Away from UCL, there are many friends that I wish to thank for being just wonderful and for making my life less stressful. I especially would like to thank my best friends; Asad Sarfarazi, Dip Patel and Rajeev Ananth.

Last but not least, my deepest thanks goes to my entire family. My parents, Ahmad and Sawsan, have struggled in their life and have travelled to many places to ensure that I get the best education possible. I especially thank my father for his everlasting encouragement and for supporting my decision of studying physics, and my mother for her unbounded kindness, support and prayers. I thank my four sisters and my extended family members for all their love and encouragement through out my life.

Towards Fault-Tolerant Quantum Computation with Higher-Dimensional Systems

By

Hussain Anwar

Doctor of Philosophy of Physics

University College London

Dr. Dan E. Browne, Supervisor

Abstract

The main focus of this thesis is to explore the advantages of using higher-dimensional quantum systems (qudits) as building blocks for fault-tolerant quantum computation. In particular, we investigate the two main essential ingredients of many state-of-the-art fault-tolerant schemes [133], which are magic state distillation and topological error correction. The theory for both of these components is well established for the qubit case, but little has been known for the generalised qudit case.

For magic state distillation, we first present a general numerical approach that can be used to investigate the distillation properties of any stabilizer code. We use this approach to study small three-dimensional (qutrit) codes and classify, for the first time, new types of qutrit magic states. We then provide an analytic study of a family of distillation protocols based on the quantum Reed-Muller codes. We discover a particular five-dimensional code that, by many measures, outperforms all known qubit codes.

For the topological error correction, we study the qudit toric code serving as a quantum memory. For this purpose we examine an efficient renormalization group decoder to estimate the error correction threshold. We find that when the qudit toric code is subject to a generalised bit-flip noise, and for a sufficiently high dimension, a threshold of 30% can be obtained under perfect decoding.

Contents

List of Tables

List of Figures

Chapter 1

Basic Elements of Quantum Fault-Tolerance Theory

The aim of this chapter is to give a brief review of the theory of quantum fault-tolerance. We will restrict the discussions to the relevant concepts that are needed for later chapters. Throughout this chapter we will assume that the reader is familiar with the postulates of quantum mechanics and has a good understanding of quantum computation. For more information regarding the theory of quantum computation and fault-tolerance, the following resources are recommended [120, 139, 166, 60].

1.1 Quantum Computation with Qudits

Any quantum computational model consists of three main elements. First, the quantum states where the information is stored or transmitted. Second, some form of physical operations that manipulate and process the information (such as unitary gates). Third, a measurement set that determines the output of the computations. In this section, we review these elements for finite-dimensional quantum systems (*qudits*), and we then describe how a discrete set of operations (gates) can perform universal quantum computation.

1.1.1 Quantum States

The most elementary unit of quantum information is the quantum-bit, or a *qubit* [141], which is a two-level quantum mechanical system. In this thesis we are interested in studying the higher dimensional generalisation of a qubit, where we have a $d-$level quantum system called a quantum dit, or a *qudit*, such that d is *prime* number. Associated with each quantum state is a complex vector space, the Hilbert space $\mathcal{H}_d$. The Hilbert space is a complex vector space with an inner product $\langle \cdot | \cdot \rangle$ and a norm $|| \cdot ||$. The Hilbert space is spanned by the computational basis states $|0\rangle , |1\rangle , \ldots , |d-1\rangle$. In this basis, a single pure quantum state $|\psi\rangle$ is represented as

$$|\psi\rangle = \sum_{j=0}^{d-1} \alpha_j |j\rangle , \tag{1.1}$$

satisfying $||\psi|| = 1$ (or $\sum_j |\alpha_j|^2 = 1$). Since we are only considering prime dimensions, the above sum is taken over all the elements of a finite field[1] $\mathbb{F}_d$ (or $\mathbb{Z}_d$). We will often make use of the cyclic property of $\mathbb{Z}_d$ in modulo d arithmetic[2], which will allow us to express many equations in a succinct form. The coefficients $\alpha_j \in \mathbb{C}$ are called the probability amplitudes, and they represent the information stored by the quantum state. The square of the amplitudes $|\alpha_j|^2$ gives the probability that a measurement (in the basis $|j\rangle$) would result in the output state $|j\rangle$. A pure state contains the maximal information of a quantum system—meaning that it is an eigenstate of a well defined observable.

A composite system of n qudits is constructed by taking the *tensor product* of the individual Hilbert spaces $\mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_n$. In the computational basis, the resultant pure quantum state is represented as

$$|\psi_1 \psi_2 \ldots \psi_n\rangle = \sum_{\mathbf{j} \in \mathbb{Z}_d^n} \alpha_{j_1 j_2 \ldots j_n} |j_1 j_2 \ldots j_n\rangle ,$$

where $\mathbf{j}$ is a vector of n dits. As we can see, to represent such pure state completely would require specifying all the d^n coefficients, which is exponential in the number of qudits. In contrast, the state of a classical system can only be a single n dit string at a time.

[1] For our discussion, a field $\mathbb{F}_d$ is the set $\mathbb{Z}_d = \{0, 1, \ldots , d-1\}$ such that all the arithmetic operations are carried out modulo d (see [103] for further details).

[2] Cyclic means that the element $(d - a) \in \mathbb{Z}_d$ is equivalent to the element $-a \in \mathbb{Z}_d$.

A quantum system whose state is not known completely is described by the language of *density operators*. In this situation, the information of a quantum system is only known partially, such that it is described by an *ensemble* of pure states $\{p_j, |\psi_j\rangle\}$, where $|\psi_j\rangle$ is a possible state of the system with respective probability p_j. The density operator ρ is given by

$$\rho = \sum_{j \in \mathbb{Z}_d} p_j \left|\psi_j\right\rangle \left\langle\psi_j\right| . \tag{1.2}$$

A density operator is a $d \times d$ matrix that satisfies three main requirements. First, it must be Hermitian, or self-adjoint, such that $\rho = \rho^\dagger$. Secondly, it must be a semi-definite matrix (i.e. a positive operator with non-negative eigenvalues $\lambda_j \geq 0$). Thirdly, it must have a unit trace $\mathrm{tr}(\rho) = 1$, which is the condition required to preserve the probabilities. A density operator represents a pure state only if $\mathrm{tr}(\rho^2) = 1$, and a *mixed state* if $\mathrm{tr}(\rho^2) < 1$.

In the language of linear algebra, the density operator can be decomposed as a linear sum of a matrix basis set. For our purpose, we adopt the higher dimensional generalisation of the Pauli operators as basis to the density operators[3]. The $d-$dimensional single-qudit Pauli operators are defined as

$$X_d = \sum_{j=0}^{d-1} \left|j \oplus 1\right\rangle \left\langle j\right| , \quad Z_d = \sum_{j=0}^{d-1} \omega^j \left|j\right\rangle \left\langle j\right| , \tag{1.3}$$

where '$\oplus$' is taken to be addition modulo d throughout, and

$$\omega = e^{2\pi i/d}, \tag{1.4}$$

is the dth root of unity [65]. From this definition we see that X_d and Z_d are traceless non-Hermitian unitaries (except for the qubit case, where the conventional Pauli operators are Hermitian). For clarity, we will drop the subscript d when the dimension is apparent from the discussion. These operators generate the Pauli group (see Sec. 1.1.2). We define a general Pauli operator $P_{(j,k)}$ as follows:

$$P_{(j,k)} = X^j Z^k \quad (d > 2), \tag{1.5}$$

$$P_{(j,k)} = i^{jk} X^j Z^k \quad (d = 2). \tag{1.6}$$

[3] This basis is also known in the literature as the Heisenberg-Weyl basis.

In this basis a density operators is expressed as

$$\rho = \sum_{(j,k)\in\mathbb{Z}_d^2} \alpha_{j,k} P_{(j,k)}. \tag{1.7}$$

In analogy to the qubit case, we will refer to the coefficients $\alpha_{j,k}$ as the Bloch components, which can be evaluated as $\alpha_{j,k} = \mathrm{tr}(\rho P^{\dagger}_{(j,k)})$. We know that in the qubit case, a general density operator can be completely specified by three real parameters $(x, y, z) \in \mathbb{R}^3$ defining the Bloch sphere. However, there is no such geometric picture beyond the qubit case. We will discuss geometric properties of the state space in higher dimensions in Sec. 2.3.1.

1.1.2 Quantum Operations

To perform a computation on the quantum states, we need to be able to process the information stored. This is accomplished by applying a quantum operation that can either dynamically manipulate the states (via unitary gates) or destructively read out the stored information to determine the output of the computation (via a measurement). More formally, a quantum operation is a physical process, denoted by $\mathcal{E}$, that transforms a quantum state ρ to another quantum state ρ', such that $\rho' \propto \mathcal{E}(\rho)$. In our work here we are interested only in simple quantum operations such as unitary dynamics where $\mathcal{E}(\rho) = U\rho U^{\dagger}$ and measurements operators where $\mathcal{E}_m(\rho) \propto M_m \rho M_m^{\dagger}$ (with outcome m).

Unitary Dynamics

The evolution of a closed quantum system is described by a unitary operator U, satisfying the relation $UU^{\dagger} = U^{\dagger}U = \mathbb{1}$. Unitary operators are a type of trace preserving operations, which is a property that is easily verified as $\mathrm{tr}(\rho') = \mathrm{tr}(U\rho U^{\dagger}) = \mathrm{tr}(\rho U^{\dagger}U) = \mathrm{tr}(\rho)$. This means that unitary dynamics never leak any information to the environment. In the Bloch representation, a unitary operator does not change the length of the Bloch vector. Here, we are interested in two sets of unitary operators, which are the Pauli group $\mathcal{P}_d$ operators and the Clifford group $\mathcal{C}_d$ operators.

The Pauli group is generated by the Pauli unitaries X_d and Z_d defined in Eq. 1.3. These operators obey the commutation relation $XZ = \omega^{-1}ZX$. In the general case when the Pauli operators are raised to arbitrary integer powers the commutation relation becomes $X^a Z^b = \omega^{-ab} Z^b X^a$, where $a, b \in \mathbb{Z}_d$. In the case of a composite system of n qudits, the Pauli group $\mathcal{P}_d^n$ is generated by the $n-$fold tensor product of the single-qudit Pauli operators. To describe such $n-$particle Pauli operators we use the so-called symplectic notation, where

$$P_{(\mathbf{j},\mathbf{k})} = (X^{j_1} \otimes X^{j_2} \otimes \cdots \otimes X^{j_n})(Z^{k_1} \otimes Z^{k_2} \otimes \cdots \otimes Z^{k_n}). \tag{1.8}$$

Two such Pauli operators commute if and only if the *symplectic inner product* is zero. In other words

$$P_{(\mathbf{j},\mathbf{k})} P_{(\mathbf{j}',\mathbf{k}')} = P_{(\mathbf{j}',\mathbf{k}')} P_{(\mathbf{j},\mathbf{k})} \iff \mathbf{j}.\mathbf{k}' - \mathbf{j}'.\mathbf{k} = 0 \mod d. \tag{1.9}$$

Using the above notation, we define the $n-$particle Pauli group $\mathcal{P}_d^n$ as the following set of unitaries

$$\mathcal{P}_d^n = \{\omega^l P_{(\mathbf{j},\mathbf{k})} \mid \mathbf{j}, \mathbf{k} \in \mathbb{Z}_d^n,\ l \in \mathbb{Z}_d\}. \tag{1.10}$$

As we will see below, the Pauli group is used in the construction of stabilizer codes and fault-tolerant schemes. Another set of gates that play an important role in quantum fault-tolerance theory are the so-called Clifford operators. The Clifford group C_d^n is the normalizer of the Pauli group—consisting of unitary operators that leave the Pauli group invariant under conjugations. More formally

$$C_d^n = \{U \mid \forall P \in \mathcal{P}_d^n,\ \ UPU^\dagger \in \mathcal{P}_d^n\}. \tag{1.11}$$

The Clifford group is generated by three gates[4] only:

$$C_d^n = \langle H_d, P_d, \Lambda(X)\rangle, \tag{1.12}$$

where the gate H_d is the $d-$dimensional single-qudit Hadamard gate

$$H_d = \frac{1}{\sqrt{d}} \sum_{(j,k)\in\mathbb{Z}_d^2} \omega^{jk} |j\rangle \langle k|, \tag{1.13}$$

and P_d is the $d-$dimensional (diagonal) single-qudit phase gate

$$P_d = \sum_{j\in\mathbb{Z}_d} \omega^{\frac{j}{2}(j-1)} |j\rangle \langle j|, \tag{1.14}$$

[4] The proof can be found in [65, 43, 75].

and the last gate $\Lambda(X)$ is the two-qudit controlled-X_d gate (often called the SUM gate), defined as

$$\Lambda(X) = \sum_{(j,k)\in\mathbb{Z}_d^2} |j\rangle |j \oplus k\rangle \langle j| \langle k| . \tag{1.15}$$

In Ref. [66], Gottesman and Chuang generalised the construction of the Clifford group operations by introducing an infinite hierarchy of qubit gates, where each level of the Clifford hierarchy maps the Pauli operators, under conjugation, to the level that precedes it. The hierarchy generalise naturally to qudits. We denote the gates sets at level k of the Clifford hierarchy by $\mathcal{C}_d^n(k)$, and define the hierarchy recursively as follows.

Definition 1. *The $k\mathrm{th}$ level of the Clifford hierarchy for n qudits is the set*

$$\mathcal{C}_d^n(k) = \{U \in \mathrm{U}(d^n) | \forall P \in \mathcal{P}_d^n, UPU^\dagger \in \mathcal{C}_d^n(k-1)\}, \tag{1.16}$$

where the first level is obviously the Pauli group, i.e. $\mathcal{C}_d^n(1) \equiv \mathcal{P}_d^n$. Similarly, the second level is the Clifford group $\mathcal{C}_d^n(2) \equiv \mathcal{C}_d^n$. Higher levels consists of sets of gates without a group structure. The set of gates of the lower levels are subsets of the higher levels of the hierarchy, such that $\mathcal{C}_d^n(1) \subset \mathcal{C}_d^n(2) \subset \cdots \subset \mathcal{C}_d^n(k) \subset \mathcal{C}_d^n(k+1) \subset \ldots$.

An important example of a qubit gate—from which other gates in higher levels of the hierarchy can be derived—is the single-qubit Pauli Z gate. It is not hard to see that by taking the roots of this gate, one can derive a gate in any level of the hierarchy:

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix} \in \mathcal{C}_2^1(k) \backslash \mathcal{C}_2^1(k-1). \tag{1.17}$$

As expected, for the first level we have $Z_2 \in \mathcal{C}_2^1(1) = P_2^1$, and for the second level we have the Clifford gate $P_2 \in \mathcal{C}_2^1(2) = \mathcal{C}_2^1$. Interestingly, for the third level we get the qubit T–gate (or $\pi/8$–phase gate), where we have $T \in \mathcal{C}_2^1(3)$. This gate plays a crucial role for many fault-tolerant schemes. In particular, as we will see, it is an important element of magic state distillation protocols, and it is sufficient to promote the Clifford group to universality. One of our main contributions in this thesis is providing a generalisation of the T–gate to higher dimensions (see Chap. 3).

Measurements

In general, observing a quantum system by making measurement would collapse the quantum state of the system and destroy the superposition. The most general description of a quantum measurement is given by the positive-operator valued measure (POVM) formalism. A POVM is defined as any set of Hermitian semi-definite operators $\{E_m\}$ (with measurement outcomes m) satisfying the completeness relation $\sum_m E_m = \mathbb{1}$. The completeness relation manifest the fact that probabilities of the measurement outcomes add up to unity. In this formalism, any quantum measurement process consisting of measurement operators $\{M_m\}$ can be expressed in term of POVM elements by defining $M_m \equiv \sqrt{E_m}$. This definition guarantees that the completeness relation is satisfied by observing that $\sum_m M_m^\dagger M_m = \sum_m E_m = \mathbb{1}$. If the state of the quantum system before the measurement is ρ, the probability that outcome m occurs is given by

$$\mathrm{Prob}(m) = \mathrm{tr}(M_m^\dagger M_m \rho), \tag{1.18}$$

and the outcome state of the system ρ^{out} immediately after the measurement is

$$\rho^{\mathrm{out}} = \frac{M_m \rho M_m^\dagger}{\mathrm{tr}(M_m^\dagger M_m \rho)}. \tag{1.19}$$

For our purposes we are only interested in the measurement of the Pauli operators $P_{(\mathbf{j},\mathbf{k})} \in \mathcal{P}_d^n$. Notice, however, that beyond the qubit case, these are non-Hermitian unitary operators, and hence cannot be measured directly because they are not valid physical observables. Nevertheless, a general Pauli unitary $P_{(\mathbf{j},\mathbf{k})}$ can always be expressed as $U = \sum_{a_m \in \mathbb{Z}_d} \omega^{a_m} M_m$ where $\{M_m\}$ are projectors onto eigenspaces that can be taken as elements of a POVM.

1.1.3 Universal Gate Sets

A universal set of gates can approximate any unitary gate to an arbitrary accuracy. Such a set provides a practical means to perform a computation, where only a finite number of gates have to be experimentally realised. The existence of such a gate set is not related to how efficiently the approximation is achieved in terms of the cost of gates required. In fact, there exist unitaries that require an exponential number of universal gates to approximate [120] (and it is the job of the quantum algorithm to ensure that an efficient description for its unitaries exists).

The earliest account of a universal gate set is due to Shor [144], where he showed that the gate set $\{H_2, P_2, \Lambda_2(X)\}$ is universal[5]. In addition, Shor showed how this gate set can perform fault-tolerant quantum computation. Subsequent work followed that established various other universal gate sets that don't necessarily have a fault-tolerant implementation [14, 15, 2, 142], and other sets which can be implemented fault-tolerantly [114, 89, 24].

Of more relevance to our work are fault-tolerant schemes that are based primarily on the Clifford group operations. Unfortunately, the Clifford unitaries do not form a universal set of gates. To achieve universality, the Clifford group must be augmented with any single non-Clifford gate [63]. In the qubit case, this additional gate performs a non-trivial rotation by an irrational multiple of π, which allows the Clifford group to forms a dense set in $\mathrm{SU}(2)$ [120, 24]. This fact also applies to the qudit case, which is the subject of the following theorem.

Theorem 1. *The set of gates $\{\mathcal{C}_d^n, K\}$, where K is a single-qudit non-Clifford gate, is a universal set.*

Proof. Our proof make use of two known theorems due to Nebe, Rains and Sloane in order to show that the set of unitaries $\{\mathcal{C}_d^n, K\}$ is dense in $\mathrm{SU}(d^n)$.

In Ref. [118], theorem 7.3 implies that any *finite* group that contains the Clifford group must be generated by the Clifford group and a gate proportional to the identity. Thus a group J generated by the Clifford group $\mathcal{C}_d^n$ and a non-Clifford unitary K that is not proportional to the identity cannot be finite, and must therefore be of infinite order.

In corollary 6.8.2 of Ref. [119], it is shown that any closed sub-group, J, satisfying $\mathcal{C}_d^n \subset J \subset \mathrm{U}(d^n)$, must either have finite order (ignoring global phase factors) or be $\mathrm{SU}(d^n)$. Combining this corollary with the previous theorem we conclude that the closure of the group generated by the Clifford group and any non-Clifford unitary is $\mathrm{SU}(d^n)$.

□

[5] The gate $\Lambda_2(X)$, is the qubit Toffoli gate, i.e. controlled-controlled-X gate. In fact, Shor never presented the proof in his work.

In later chapters we will make use of this theorem to prove that the qudit generalisation of the T–gate, which is non-Clifford, is sufficient to promote the Clifford group to quantum universality.

1.2 The Stabilizer Formalism

The stabilizer formalism presents a concise description of additive quantum error correction codes in terms of the set of Pauli operators [63, 120, 60]. In Sec. 1.2.1 we outline the general structure of a qudit stabilizer code, and define the stabilizer operations which can be used to construct any stabilizer code. Then, in Sec. 1.2.2, we explain the properties of a special class of stabilizer codes, the so-called CSS codes. These codes will be used for magic state distillation in Chap. 3.

1.2.1 Stabilizer Codes

A quantum error correction code protects k logical qudits by encoding them in n physical qudits such that $n > k$. Consider an Abelian subgroup $\mathcal{S}$ of the Pauli group $\mathcal{P}_d^n$ such that it is generated by $(n-k)$ independent and mutually commuting generators $\mathcal{S} = \langle g_1, g_2, \ldots, g_{n-k} \rangle$, and it contains the identity operator $\mathbb{1}$ but not any other phase multiples of the identity, i.e. $\omega^j \mathbb{1} \notin \mathcal{S}$ for all nonzero j. Then, the common " + 1" eigenspace, denoted as Π, of $\mathcal{S}$ forms a protected subspace of $\mathcal{H}_d^n$ called the stabilizer code, and the elements of $\mathcal{S}$ are called the *stabilizers* of the code.

For any stabilizer code-space of dimension k, there exist k pairs of logical Pauli operators $\bar{X}_j$ and $\bar{Z}_j$, where $j = 1, \ldots, k$. These logical operators conjugate as $\bar{X}_j \bar{Z}_j = \omega^{-1} \bar{Z}_j \bar{X}_j$, but commute with the whole stabilizer group $\mathcal{S}$. Furthermore, each pair generates a group of logical operators $\mathcal{A}_j = \langle \bar{X}_j, \bar{Z}_j \rangle$, and each element of this group has the form $\omega^a \bar{X}_j^b \bar{Z}_j^c$ for $a, b, c \in \mathbb{Z}_d$. There are many suitable choices of logical operators, since for any Pauli P and any element of the stabilizer $S \in \mathcal{S}$, we find that PS has the same effect as P on the code-space Finally, the *distance* of the code quantifies the number of errors that can be detected and corrected. A stabilizer code of distance δ can detect up to $(\delta - 1)$ errors and correct up to $\lfloor (\delta - 1)/2 \rfloor$ errors. The distance is equal to the minimum weight (number of non-trivial Pauli operators) of all operators that commute with the stabilizer but have a non-trivial logical effect, that is all $P = AS$ such that $S \in \mathcal{S}$, $A \in \mathcal{A}$ but $A \neq \mathbb{1}$. We denote such a code as $[\![n, k, \delta]\!]_d$.

Most of the early qubit quantum codes are types of stabilizer codes. Examples include, the famous 9-qubit code due to Shor $[\![9,1,3]\!]_2$ [143], Steane's 7-qubit code $[\![7,1,3]\!]_2$ [150] and the 5-qubit code $[\![5,1,3]\!]_2$ [100]. All of these codes were unified by the above stabilizer code construction [62, 32].

Error correction is performed by first measuring the stabilizer generators non-destructively to extract the *syndromes*. Such measurements may consist of applying unitary operators to entangle an ancilla qudit with the relevant physical qudits of the code, followed by measuring the ancilla destructively. Any errors that have occurred on the physical qudits will propagate to the ancilla, and hence measuring the ancilla will only reveal the information about the errors and will not alter the logical states of the code. The syndromes are then used to identify and locate the errors that have occurred in the code-space via a classical decoding algorithm.

All the encoding and decoding circuits of a stabilizer code can be implemented by using what we call the *stabilizer operations* [64]. These operation contain the Clifford unitaries along with other simple operations.

Definition 2. *The stabilizer operations are composed from the following:*

1. *Clifford unitaries;*
2. *Measurements and projections on stabilizer subspaces;*
3. *Preparation of stabilizer states;*
4. *Adaptive decision making based on measurement outcomes.*

For most stabilizer-based fault-tolerant schemes these operations are readily available (they have a fault-tolerant implementation). Although these operation are capable of generating long-range entanglement, they are not sufficient to perform universal quantum computation. In fact, there exists a famous theorem, called the Gottesman-Knill theorem, which asserts that a computation consisting of only the stabilizer operations can be efficiently classically simulated [63]. Further work by Aaronson and Gottesman [1] showed that the class of computations that can be performed by the stabilizer operations lie in the complexity class $\oplus\mathbf{L}$, a class that is believed to be weaker than $\mathbf{P}$.

1.2.2 CSS Codes

Calderbank, Shor and Steane identified a special class of quantum codes, which in their honour are now known as CSS codes [33, 149]. These codes have stabilizers generated by two subgroups, $\mathcal{S}_Z$ and $\mathcal{S}_X$, which contain only Z and X terms, respectively. All CSS codes, can also be described by a pair of classical vector spaces, which correspond to $\mathcal{S}_Z$ and $\mathcal{S}_X$. If we have a vector $\mathbf{u} \in \mathbb{F}_d^n$ and a single-qudit operator, U, then we define the n-qudit operator

$$U[\mathbf{u}] = \bigotimes_{k=1}^{n} U^{u_k}. \tag{1.20}$$

The $k\text{th}$ element of the vector, $\mathbf{u}$, tells us what multiple of U acts on the $k\text{th}$ qudit. It follows that for every $s \in \mathcal{S}_Z$ we can find a $\mathbf{u}$ such that $s = Z[\mathbf{u}]$. In fact, $\mathcal{S}_Z = \{Z[\mathbf{u}]; \mathbf{u} \in \mathcal{L}_Z\}$ where $\mathcal{L}_Z$ is a linear vector space. The closure of the stabilizer group under multiplication is easily seen to directly correspond to closure of $\mathcal{L}_Z$ under addition modulo d. Similarly we can find a linear code, $\mathcal{L}_X$, for $\mathcal{S}_X$. The whole stabilizer must be Abelian and so for all $\mathbf{u} \in \mathcal{L}_X$ and $\mathbf{v} \in \mathcal{L}_Z$ we require $\langle \mathbf{u}, \mathbf{v} \rangle = \oplus_j u_j v_j = 0$. Furthermore, for any code, $\mathcal{L}$, we define the dual code $\mathcal{L}^\perp = \{\mathbf{u}; \langle \mathbf{u}, \mathbf{v} \rangle = 0, \forall \mathbf{v} \in \mathcal{L}\}$. In terms of duality, commutation inside the stabilizer equates to $\mathcal{L}_X \subset \mathcal{L}_Z^\perp$ and $\mathcal{L}_Z \subset \mathcal{L}_X^\perp$. The dimensionalities of the duals are related by $\mathrm{Dim}(\mathcal{L}^\perp) = n - \mathrm{Dim}(\mathcal{L})$, where n is the dimension of the vector field they inhabit, namely $\mathbb{F}_d^n$. For a CSS code $k = n - \mathrm{Dim}(\mathcal{L}_Z) - \mathrm{Dim}(\mathcal{L}_X)$ gives the number of logical qudits supported by the code.

Our main interest here is in stabilizer codes of only d dimensions, i.e. encoding a single logical qudit. In this case, the basis spanning the code are the single logical Pauli operators $\bar{Z}$ and $\bar{X}$. It follows that there exists a corresponding orthonormal logical basis, $\{|j\rangle_L\}$, of stabilizer states that obey $\bar{Z}|j\rangle_L = \omega^j |j\rangle_L$, $\bar{X}|j\rangle_L = |j \oplus 1\rangle_L$. In this basis, the code projector can be expressed as $\Pi = \sum_j |j\rangle_L \langle j|_L$. We also make use of the X-basis that we denote $|+_j\rangle$ for single qudits stabilized by $\omega^{-j}X$ and $|+_j\rangle_L$ for logical encoded states stabilized by $\omega^{-j}\bar{X}$. Typically, such logical operators can also be expressed in terms of vectors, such as $\bar{X} = X[\mathbf{u}]$ where commutation of $\bar{X}$ with $\mathcal{S}_Z$ entails $\mathbf{u} \subset \mathcal{L}_Z^\perp$ and $\mathcal{L}_Z \subset \mathbf{u}^\perp$.

Given this vector description, a useful fact is that $\mathcal{L}_Z = (\mathrm{span}(\mathcal{L}_X, \mathbf{u}))^\perp$ where the $\mathrm{span}(..., ...)$ is the vector space generated by its arguments. Let us prove this by first observing that since $\mathcal{L}_Z \subset \mathbf{u}^\perp$ and

$\mathcal{L}_Z \subset \mathcal{L}_X^\perp$ we have that $\mathcal{L}_Z \subset (\mathrm{span}(\mathcal{L}_X, \mathbf{u}))^\perp$. That $\mathcal{L}_Z$ can be no smaller than this set follows from dimension counting, that is

$$\begin{aligned}\mathrm{Dim}[(\mathrm{span}(\mathcal{L}_X, \mathbf{u}))^\perp] &= n - \mathrm{Dim}[\mathrm{span}(\mathcal{L}_X, \mathbf{u})], \qquad (1.21)\\ &= n - \mathrm{Dim}(\mathcal{L}_X) - 1.\end{aligned}$$

Since we have a single logical qudit, we know also that $\mathrm{Dim}(\mathcal{L}_Z) = n - \mathrm{Dim}(\mathcal{L}_X) - 1$. Since the dimensionality match, the assertion is proven. Taking also $\bar{Z} = Z[\mathbf{v}]$ and noting $(\mathcal{L}^\perp)^\perp = \mathcal{L}$, many such results for single qudit codes can be deduced by similar reasoning,

$$\mathcal{L}_Z = [\mathrm{span}(\mathcal{L}_X, \mathbf{u})]^\perp, \qquad (1.22)$$

$$\mathcal{L}_Z^\perp = \mathrm{span}(\mathcal{L}_X, \mathbf{u}), \qquad (1.23)$$

$$\mathcal{L}_X = [\mathrm{span}(\mathcal{L}_Z, \mathbf{v})]^\perp, \qquad (1.24)$$

$$\mathcal{L}_X^\perp = \mathrm{span}(\mathcal{L}_Z, \mathbf{v}). \qquad (1.25)$$

For CSS codes it suffices to consider phase- and bit-flip noise separately. For an operator $U[\mathbf{u}]$ its "size" is measured by the Hamming weight, $|\mathbf{u}|_H = \{\# x_j ; x_j \neq 0\}$, so the number qudits upon which the operator acts non-trivially. The robustness to phase noise is measured by the distance, $\delta_Z = \min\{|\mathbf{v}|_H ; Z[\mathbf{v}]\Pi = \bar{Z}\Pi\}$, and for bit-flip noise $\delta_X = \min\{|\mathbf{v}|_H ; X[\mathbf{v}]\Pi = \bar{X}\Pi\}$. The overall distance of the code is $\delta = \min\{\delta_X, \delta_Z\}$. Finally we remark that for any stabilizer code there always exists a Clifford unitary that decodes the encoded qudits back to the physical qudits, such that $U\bar{Z}U^\dagger = Z$ and $U\bar{X}U^\dagger = X$.

1.3 Fault-Tolerant Schemes

We have shown how error correction codes can be used to protect information in principle. Such codes could be useful for transmitting data over a noisy channel or storing quantum systems with active error correction. However, the existence of an error correction code is not sufficient to ensure that the information remains protected. The reason being that the operations involved in the encoding and decoding steps are themselves subject to errors, and hence errors can easily spread and multiply throughout each code block beyond the capability of the code to correct them. This suggests that each operation of an error correction code has to be designed such that the errors are carefully controlled. Loosely speaking, if such a design exists for an operation within the code block, then that operation is said to be fault-tolerant.

For the purpose of fault-tolerant computation, Shor [144] proposed that the computation is to be carried at the logical level without any decoding required. This avoids the decoding step which would expose the stored information to the environment. To accomplish such a computation, we need to introduce the notion of an *encoded operation* $\bar{U}$ which act on the logical encoded states just like a single operation U acts on the single physical state. We have already come across such an operation in our discussion of the logical gates. A *fault-tolerant scheme* consists of a quantum error correction code to protect the information and a set of universal encoded operations.

For a single-qudit encoded operation to be fault-tolerant we require that it does not entangle multiple qudits within a single code block. In other words, the encoded operation has to be an $n-$fold product operator of the form $\bar{U} = U^{\otimes n}$. This requirement ensures that the errors do not spread within the code block by an error-prone physical operator U. Such an operation is called *transversal*. Similarly, for a multi-qudit encoded operation to be fault-tolerant we require that it does not entangle a qudit in one code block with multiple qudits in a second code block. It is natural then to ask: does there exist a quantum error correction code that permits the existence of a universal transversal set of gates?

Unfortunately, for the qubit stabilizer codes this was proven in the negative by [170], which was later generalised to the qudit case [42]. Moreover, it turns out that such a set cannot exist for much larger class of codes, as was shown by a powerful no-go theorem proven by Eastin and Knill[6] [52]. For completeness we restate their theorem here:

[6] It is worth pointing out that there exist schemes that cleverly circumvent this theorem. For example, very recently in [122] a scheme was proposed that exploits transversality properties of the tri-orthogonal codes, that were introduced by Bravyi and Haah [26]. It was shown that these codes have a transversal controlled-controlled-Z gate (CCZ). In addition, a simple application of the product of Hadamard gates $\bar{H}_2$ acts non-trivially (not fault-tolerantly) on only some of the encoded qubits (gauge qubits) which can be discarded (or corrected fault-tolerantly). The set $\langle CCZ, H_2\rangle$ is known to be a universal set.

Theorem 2. *For any non-trivial local-error-detecting quantum code, the set of logical unitary product operators is not universal.*

In Tab. (1.1) we have listed few examples of well known stabilizer codes and the set of gates that have a transversal implementation.

Code $[\![n, k, \delta]\!]_d$	Transversal Gates	Non-transversal Gates
$[\![5, 1, 3]\!]_2$	PH	H, P, CNOT, T
$[\![7, 1, 3]\!]_2$	H, P, CNOT	T
$[\![9, 1, 3]\!]_2$	CNOT	H, P, T
$[\![15, 1, 3]\!]_2$	CNOT, T	H
$[\![d^m - 1, 1, 2]\!]_d$	CNOT, $\mathcal{M}_d(m)$	$\mathrm{H_d}$

Table 1.1: The set of transversal gates for various well known codes. The last row is our qudit generalisation of the quantum Reed-Muller codes (see Chap. 3). The structure of the table is motivated by that in [170].

An encoded operation that cannot be implemented transversally would require other techniques. Notably, such operations are often implemented via a teleportation circuit that consumes some uniquely prepared ancillary system. Such circuit are called state-injection circuits[7]. For example the state-injection circuit that implements the non-Clifford T–gate is given in Fig. (1.1), which is sufficient to promote the Clifford group to universality. There exist stabilizer codes, such as Steane's 7-qubit code that admit a transversal implementation for the entire group of Clifford unitaries. Hence, to perform a fault-tolerant universal computation all we need to ensure is a fault-tolerant implementation of the state-injection circuit. It is straightforward to see that all the Clifford operations of the state-injection circuit in Fig. (1.1) can be implemented by replacing them with their transversal version at the logical

[7] Note that there is no classical analogue for the state-injection circuit. In fact, in the classical case, there exists a set of transversal universal gates. For example, the Toffoli gate—a universal gate for classical computation—has a transversal implementation in the simple repetition code!

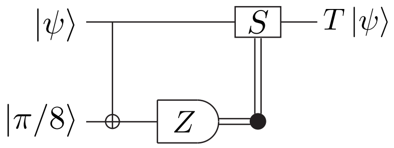

Figure 1.1: The state-injection circuit of the equatorial state $|\pi/8\rangle = (|0\rangle + e^{i\pi/4})/\sqrt{2}$.

level. What remains is to come up with a fault-tolerant preparation procedure for the ancilla state $|\pi/8\rangle$, and for this purpose various techniques have been developed (see p. 304 of [139]).

Finally, given the existence of a set of universal fault-tolerant operations, the accuracy threshold theorem of quantum fault-tolerance theory [144, 125] asserts that arbitrary accurate and long computations can be performed efficiently provided that all the physical operations have a failure probability below a certain constant threshold [94, 169, 89, 3, 95]. The exact numerical value of the threshold is very sensitive to the computational model and the assumptions concerning the noise in the fault-tolerant scheme, with thresholds ranging between $10^{-5} - 10^{-2}$.

The scheme of most interest to us is the one proposed by Raussendorf *et al.* in a series of papers [132, 133, 131]. This scheme shows how to perform fault-tolerant quantum computation for the measurement-based cluster state model [129, 128] by employing techniques from topological error correcting codes for robust error tolerance combined with magic state distillation [27] for universality. A detailed review of this scheme can be found in [56, 57]. This scheme require only *local* interactions and achieves the highest fault-tolerant threshold known to date of about 1%. In particular, the original scheme in [132, 131] achieves a threshold of 0.75%. This threshold was improved further to about 0.9% [58]. Achieving thresholds close to 1% represents one of the milestones for the theory of fault-tolerance.

Our aim in this thesis is to generalise all the necessary components of this scheme to qudit case. It is known that the formalism of measurement-based quantum computing generalises to higher dimensions naturally [172]. In the remaining of this thesis we provide the first qudit generalisation of magic state distillation and construct an efficient qudit decoder for qudit topological codes, hence providing all ingredients for a fault-tolerant qudit computation.

Chapter 2

Magic State Distillation: Introduction and Numerical Investigation

In this chapter we will describe the motivation behind Magic State Distillation (MSD), and why it is a crucial element to all stabilizer-based fault-tolerant schemes. We will often restrict the discussions and examples in the first few sections to the well-established qubit case as it is easier to grasp. However, all the points raised can also be applied to higher dimensions too, and we reserve the explicit generalisation to the sections that follow. We begin in Sec. 2.1 by introducing the magic state model which describes the different set of operations allowed to perform the distillation. This is followed in Sec. 2.2 by some comments on the computational power of magic states and what elements of the state-injection circuits makes the magic state model inefficient to simulate classically. In Sec. 2.3 we give a general description of the distillation map for any qudit stabilizer code. We use this map in Sec. 2.4 to explicitly study the distillation properties of the five-qutrit code, and we identify all the magic states that are distillable by this code and how they can be used to promote the Clifford group to quantum universality.

2.1 Magic States as a Quantum Resource

In the previous chapter we have seen that the operations that have a readily fault-tolerant implementation are often the stabilizer operations. We saw that there exist stabilizer codes, such as Steane's $7-$qubit code $[\![7,1,3]\!]_2$, where the entire group of Clifford operators $\mathcal{C}_2(2)$ can be performed transversally, and hence can be implemented fault-tolerantly. Unfortunately, however, the Clifford gates alone do not form a quantum universal set of gates, and to achieve universality the stabilizer operations must be augmented with certain *non-stabilizer* states (such as the $|\pi/8\rangle$ state). These additional states are consumed by the state-injection circuits to implement the non-Clifford gates needed in the computation (such as the $T-$gate, see Fig. (1.1)).

Since these states have to be prepared separately without being protected by an error correction code, they are thus expected to be highly mixed. Any direct injection of such noisy states will jeopardise the accuracy of the whole computation. Despite this fact, Bravyi and Kitaev [27] showed how mixed non-stabilizer states can allow universal quantum computation. They proposed a distillation process that distils pure non-stabilizer states, the so-called *magic states*, from a set of more mixed states using stabilizer operations alone. This process is called magic state distillation (MSD) and it is essentially a preparation procedure for the magic states. The term "magic" was used by Bravyi and Kitaev to reflect the powerful role of the magic states with respect to the stabilizer operations. In particular, the magic states are distilled by using only stabilizer operations and yet they are then used to promote the stabilizer operations to universality. As stated previously, the state $|\pi/8\rangle$ is in fact an example of a qubit magic state. Based on this basic motivation, we define a magic state, denoted by $|M\rangle$, in the broadest sense as follows:

Definition 3. *A magic state $|M\rangle$ is any non-stabilizer pure state that can be distilled from multiple copies of mixed states by stabilizer operations alone.*

Associated with every MSD protocol is a stabilizer code $[\![n,k,\delta]\!]_d$ which is used to perform the distillation. Recall that every stabilizer code can be constructed by using Clifford gates alone, hence only stabilizer operations will be involved in the distillation. We refer to the non-stabilizer mixed states from which the magic states are distilled as the *resource states* for MSD, denoted by ρ_{res}, defined as follows:

Definition 4. *The resource states ρ_{res} are the set of mixed states from which a magic state can be distilled with arbitrarily high purity by a magic state distillation protocol. The resource states are determined by the choice of the distillation protocol.*

The resource states form a convex region in the state space around the magic states, and one of the important problems in the theory of MSD is to identify the plausible region of resource states. We will expand on this point in more depth shortly.

In MSD the stabilizer operations are assumed to be perfect, and the resource states are the only noisy elements in the distillation protocol. The justification behind these assumptions is that the stabilizer operations have a direct fault-tolerant implementation in many computational models. For instance, in topological quantum computation, the stabilizer operations have an intrinsic protection due to the nature of the physical systems, which further justify these assumptions[1]. Computational models that share such a property and allow for imperfect state preparation became known as the magic state computational models:

Definition 5. *A magic state model consists of perfect (noise-free) stabilizer operations and the ability to prepare any number of resource states ρ_{res}.*

All known MSD protocols have an iterative structure [27, 36], with each iteration having three steps:

1. *Initialization:* Prepare n copies of the qudit resource state ρ_{res}.

The resource states are supposed to be noisy in nature due to the experimental imperfection in the preparation device. Also, the resource states are assumed to be generated independently, resulting in a product state $\rho_{\mathrm{res}}^{\otimes n}$ after each initialization step.

[1] There are studies of magic state distillation where the these assumptions have been relaxed by allowing noisy stabilizer operations [81].

2. *Projection:* Measure the Pauli generators of a stabilizer code $[\![n,k,\delta]\!]_d$ and post-select on the '+1' outcome[2].

This step will project the resource states onto the code-space. The measurement outcomes of the generators $\{g_j\}$ of a stabilizer code forms the syndrome vector $\lambda = \{\lambda_1,\ldots,\lambda_{n-k}\}$, where $\lambda_j \in \{1,\omega,\ldots,\omega^{d-1}\}$. Post-selecting on the trivial +1 syndromes corresponds to the resource states being acted upon by the following projector:

$$\Pi = \frac{1}{d^{n-k}}\prod_{j=1}^{n-k}(I+g_j). \tag{2.1}$$

Note that each stabilizer measurement is probabilistic, and therefore there is an associated success probability p_{succ} for measuring the trivial '+1' syndrome.

3. *Decoding:* Perform a Clifford unitary that maps the d–dimensional code-space onto a single physical output qudit ρ^{out}.

When successful, the output state ρ^{out} is used as one of the input copies on the next level of iteration. After every successful iteration the fidelity with respect to the pure magic states, given by $\langle M|\,\rho^{\text{out}}\,|M\rangle$, is increased. Magic states are distilled until a high enough fidelity is achieved as required for the target accuracy of the fault-tolerant computation.

Interestingly, there is a strong analogy between MSD and entanglement distillation [17, 40, 39, 74]. In entanglement distillation, n copies of arbitrary entangled states are transformed into fewer copies of higher-fidelity Bell states using only local operations and classical communication (LOCC). Under the restricted set of LOCC, entanglement is considered as a *resource* for quantum communication (e.g. quantum teleporation [18]) [124, 74]. Similarly, in the case of MSD, under the restricted set of stabilizer operations, the magic states can be considered as the resource for universal quantum computation. This analogy was formulated (rigorously) only very recently in [160], where various monotones for $magicness$ where introduced.

[2] It is worth noting that post-selection is not always necessary. As we will see in the next chapter, the general outcome of some types of stabilizer measurements can always be corrected to the '+1' eigenspace by the application of an adaptive Clifford operator.

For every MSD protocol there are various questions that need to be addressed. First, what are the types of magic states that can be distilled? For example, the qubit $|\pi/8\rangle$ state is an equatorial state in the Bloch sphere; is there a family of such equatorial magic states? Second, what are the regions of resource states in the state space? Can *all* the non-stabilizer mixed states be distilled by a certain MSD protocol? Finally, what is the *yield* of the distillation protocol? The yield captures the overall performance of the MSD protocol—for a give target error probability in the distilled magic state, the yield quantifies the *fraction* of the initial copies of resource states that will be distilled.

Although the rigorous structure of MSD was not introduced until [27], there exist earlier accounts of distillation protocols that serve the same purpose as MSD. The earliest work that we are aware of is due to Dennis [46], where he showed how, for a non-generic noise model, a certain three-qubit ancilla state can be distilled to implement the Toffoli gate required in Shor's fault-tolerant scheme [144]. In addition, and in parallel work to [27], Knill has introduced state preparations protocols for his fault-tolerant scheme of post-selected quantum computer [92, 91, 93], which although seems distinct from MSD, they were later shown to be equivalent to MSD by Reichardt [136].

In the qubit case, MSD has an elegant geometrical visualization in terms of the Bloch sphere picture. In this representation, the stabilizer states form an octahedron, whose vertices are the Pauli eigenstates. Moreover, the single Clifford group unitaries coincide with the rotational symmetries of this octahedron. There are two types of magic state distilled by the distillation protocols in [27], which correspond to pure states invariant under two types of rotational symmetries. The first type is the set of magic states that are invariant under $180°$ rotations around the centres of the edges of the octahedron, which contains the Hadamard gates, and we refer to these state as Hadamard-type or H–type magic states. The second type is the set of states invariant under $120°$ rotations around the centres of the faces of the octahedron, known as T–type magic states. Fig. (2.1) shows all these magic states on the Bloch sphere. The H– and T–type magic states were distilled in [27] using the $5-$qubit code $[\![5,1,3]\!]_2$ and the $15-$qubit Reed-Muller code $[\![15,1,3]\!]_2$, respectively.

Identifying the set of mixed states that provide suitable resource states ρ_{res} for MSD has interesting consequences for the theory of quantum information. If we treat the magic state model as a resource

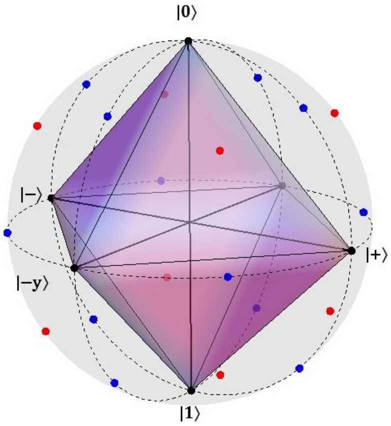

Figure 2.1: The Bloch sphere with the qubit stabilizer octahedron and the magic states. There are 8 (red) T–type magic states and 12 (blue) H–type magic states. These states were characterised and shown to be distillable by Bravyi and Kitaev [27] using the $5-$qubit code $[\![5,1,3]\!]_2$ and the $15-$qubit Reed-Muller code $[\![15,1,3]\!]_2$.

theory for universal quantum computation, then the resource states are essentially what promote the stabilizer operations to universality. We can identify straight away the states within the stabilizer octahedron as not useful as resource states for MSD, due to the Gottesman-Knill theorem (see Sec. 1.2.1). In other words, there cannot exist a MSD protocol that distils non-stabilizer magic states from the set of states within the stabilizer octahedron because that would promote the stabilizer operations to universality, which would contradict the Gottesman-Knill theorem. It is natural then to ask whether *all* the non-stabilizer mixed states are useful resources for MSD. A positive indication comes from Reichardt [136] who showes explicitly how all the states above the edges of the octahedron can be distilled by Steane's code $[\![7,1,3]\!]_2$.

However, we now know that there are limits on the suitable resource states ρ_{res}. First, Campbell and Browne [37, 36] showed that for any iterative protocol there will always exist undistillable qubit states (bound states) above the faces of the stabilizer octahedron. This result applies to the iterative structure of MSD protocols, and non-iterative qubit protocols could be an interesting possibility to circumvent this result; for example, compare with the hashing protocol [39, 47] and quantum polar codes [138] used in an analogous context of entanglement distillation. In addition, Campbell [34] introduced an activation protocol that can activate qubit states from above the octahedron face to the known regions of resource states. Second, Veitch *et al.* [159] showed that for *odd-dimensional* systems there exist other types of bound states—states with positive Wigner distribution—that cannot contribute any enhancement to the computation power of the stabilizer operations [159, 108]. We will show explicit examples of how these states are derived below. Finally, as we will see later, there could be other family of mixed states that are ruled out as a resource for MSD by some other no-go theorems that are yet to be discovered.

Another very interesting investigation is due to van Dam and Howard [157, 158] who studied noise thresholds using qudit systems. They identified a set of *robust* qudit states that are the most resilient to depolarizing noise and found that the degree of noise needed to map such states to the set of stabilizer states increases with the dimension of the qudits—scaling with $d/(d+1)$. Thus, the higher-dimensional states have the potential to offer higher MSD threshold due to the potentially larger region of non-stabilizer resource states.

In addition to the above work, there has been a considerable number of investigations to improve the original qubit schemes in [27]. For example, modifications to the magic state model with noisy stabilizer operations were studied in [81]. Also, distillation protocols for other "equatorial" type magic states were proposed in [59, 101] that could directly implement a family of non-Clifford diagonal phase gates—this could in turn improve the overhead in the gate synthesis of some quantum algorithms.

All the above protocols involve a quantum code that encodes one qubit, i.e. $[\![n, 1, \delta]\!]_2$. However, recently Meier *et al.* [110] proposed a distillation protocol that can distil multi-qubit magic states and achieve significant reduction in the overhead of the resource states needed for the distillation. This has led to more investigations building on the idea of multi-level MSD for various codes [26, 85, 84, 83, 82, 55]. Finally, MSD protocols are used to refine the upper-bound error tolerance threshold for many stabilizer-based fault-tolerant schemes [123, 127].

2.2 Computational Power of Magic States

In the last section, we stated that the magic states are used to promote the stabilizer operations to quantum universality. Since the stabilizer operations are classically efficiently simulated, this suggests that the magic states are what give quantum computers their computational speed-up in comparison to classical computers. In this section we are interested in identify the aspects of the state-injection circuit that give magic states their non-classical power. For this purpose we analyse a generic structure of the known state-injection circuits, given in Fig. (2.2i). Note that this circuit is the generalisation of the injection-circuit for the equatorial $|\pi/8\rangle$ (H–type) magic state given in Fig. (1.1).

The Clifford gates in $\mathcal{C}_d(2)$ and the Pauli measurement $\mathcal{C}_d(1)$ are stabilizer operations, and hence can be efficiently classically simulated. Also, the magic state $|M\rangle$ is by definition a non-stabilizer state, therefore has to be an element that cannot be efficiently classically simulated. This indeed was shown to be the case in the early work of Aaronson and Gottesman [1], who considered (under assumptions of quantum speed-up) the classical simulation of stabilizer circuits with arbitrary non-stabilizer initial states, and showed that such a simulation would have an exponential time complexity in the number of non-stabilizer initial states involved.

The last remaining element to check is the classically-controlled unitary gate $\mathcal{C}_d(2)/\mathcal{C}_d(1)$, denoted here as CC-$U$. There is a general misconception that this is a Clifford gate because $\mathcal{C}_d(2)/\mathcal{C}_d(1)$ is a non-Pauli Clifford gate. However, it turns out that the gate CC-U is in fact a non-Clifford gate. The easiest way to see the validity of this statement is by considering the circuit identity in Fig. (2.2ii) [173], which allows us to get rid of the classical adaptivity of the Pauli measurement. It is not hard to be convinced with a simple calculation that all the non-Pauli controlled-Clifford gates would map, for example, the operators $X \otimes X$ to a non-Pauli operator under conjugation, implying that, indeed, the CC-U gate is a non-Clifford gate.

Whether magic state-injection without such a classically controlled gate is possible is still unknown. If such a circuit exists, it would help us to pin down the resource that is responsible for the quantum speed-up to be exactly the magic states. In essence, loosely speaking, the magic states will be what

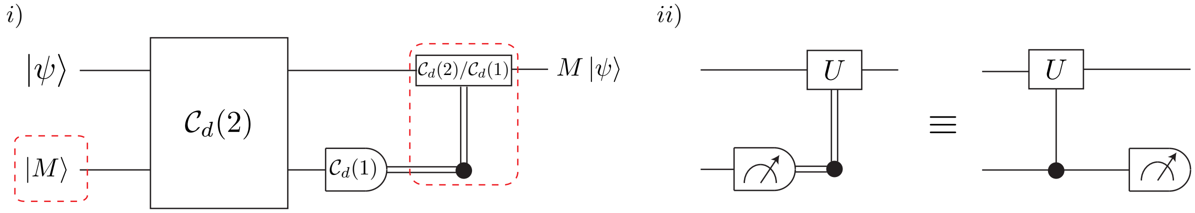

Figure 2.2: i) The general structure of a state-injection circuit. The elements in the red dashed boxes are non-stabilizer operations. ii) A circuit identity that shows how an adaptive classical-controlled unitary gate can be replaced by a quantum-controlled gate followed by the measurement.

would give the apparent distinction between the computational complexity classes **BQP** and **BPP**[3].

Another related question is whether the magic states can be reused using only Clifford gates in the injection-circuit. In other words, does there exist a state-injection circuit such that the magic state is not consumed? Such a circuit would allows us to inject a single magic state as many times as non-Clifford gates are required without any dependence on the input size of the quantum computation. As a result,the overhead for MSD will be reduced to a constant $O(1)$. This would have drastic consequences, as was shown by Anderson [4], it will imply that the class **BQP** is equivalent to class of the stabilizer circuits $\oplus$**L**. Such a collapse of the class **BQP** is extremely unlikely, and in turn indicates that such a circuit for reusable magic states should not exist. Nevertheless, this suggests that the magic state model could be a suitable model to probe open questions on computational complexity questions.

2.3 General Distillation Map

In this section, we will analyse the three steps of MSD in more detail for qudit systems (and not qubits). In particular, we derive a general formula for the distillation map which relates the Bloch components of the distilled state ρ^{out} after one round of MSD, to terms of the Bloch components of the initial input

[3] These computational classes contain the decision problems solvable by a quantum and classical computers, respectively, in polynomial time [120].

state ρ_{res}.

2.3.1 Useful Basis Set

We start by defining a new basis set that will prove to be very convenient for studying the distillation map. For completeness, we will state again the $d-$dimensional single-qudit X and Z Pauli operators defined previously:

$$X = \sum_{j=0}^{d-1} |(j+1) \mod d\rangle \langle j|, \;\; Z = \sum_{j=0}^{d-1} \omega^j |j\rangle \langle j|, \tag{2.2}$$

where $\omega = e^{2\pi i/d}$ is the dth root of unity. We define a slightly different form of the single-qudit Pauli operators as follows:

$$\{\sigma_{j,k} = \omega^{cjk} X^j Z^k;\; (j,k) \in \mathbb{Z}_d^2\}, \tag{2.3}$$

where $c = (1-d)/2$. The main reason for choosing this definition and the significance of the extra phase ω^{cjk} will become apparent shortly. But we shall first outline some of the properties of the Pauli operators based on this definition. For a single qudit, the composition of two Pauli operators can easily be verified to be

$$\sigma_{j,k}\sigma_{j',k'} = \omega^{j'k - c(jk' + j'k)} \sigma_{(j+j'),(k+k')}. \tag{2.4}$$

For the case of a composite system of n qudits we use the symplectic notation to represent the $n-$fold tensor products of Pauli operators

$$\begin{aligned} \sigma_{j_1,k_1} \otimes \sigma_{j_2,k_2} \otimes \cdots \otimes \sigma_{j_n,k_n} &\equiv \sigma_{j_1 j_2 \ldots j_n, k_1 k_2 \ldots k_n} \\ &\equiv \sigma_{\boldsymbol{j},\boldsymbol{k}}, \end{aligned} \tag{2.5}$$

where $\boldsymbol{j}$ and $\boldsymbol{k}$ are vectors in $\mathbb{Z}_d^n$. The Pauli operators satisfy a generalised commutation relation

$$\sigma_{\boldsymbol{j},\boldsymbol{k}}\sigma_{\boldsymbol{j}',\boldsymbol{k}'} = \omega^{\boldsymbol{k}.\boldsymbol{j}' - \boldsymbol{j}.\boldsymbol{k}'} \sigma_{\boldsymbol{j}',\boldsymbol{k}'}\sigma_{\boldsymbol{j},\boldsymbol{k}}, \tag{2.6}$$

where $\boldsymbol{k}.\boldsymbol{j}' - \boldsymbol{j}.\boldsymbol{k}'$ is the symplectic inner product.

We will use $\sigma_{j,k}$ as the basis set to represent a qudit state ρ. However, notice that $\sigma_{j,k}$ is a *non-Hermitian* unitary operator. To guarantee the Hermiticity of ρ we must impose the Hermiticity condition

$\rho = \rho^\dagger$. We begin by expressing ρ as

$$\rho(\boldsymbol{\alpha}) = \frac{1}{d} \sum_{(j,k)} \alpha_{j,k} \sigma_{j,k}, \tag{2.7}$$

where the summation is over all pair elements of $\mathbb{Z}_d^2$ (and we assume that the identity Bloch component $\alpha_{0,0} = 1$). We will use $\sigma_{0,0}$ and the conventional $\mathbb{1}$ interchangeably. We refer to $\alpha_{j,k}$ as the Bloch components, generalising the qubit convention, and $\boldsymbol{\alpha}$ as the Bloch vector, which has the Bloch components as its elements. Observe that for $d = 2$, the set of $\sigma_{j,k}$ is Hermitian and the Bloch components will be real, but in the general qudit case the Bloch components are complex and need to be constrained by a Hermiticity relation in order for $\rho = \rho^\dagger$ to hold. To work out the Bloch components' relation we start by explicitly writing $\rho = \rho^\dagger$:

$$\sum_{(j,k)} \alpha_{j,k} \sigma_{j,k} = \sum_{(j,k)} \alpha^*_{j,k} \sigma^\dagger_{j,k}. \tag{2.8}$$

The importance of the extra phase factor of ω^{cjk} in our definition in Eq. (2.3) is to ensure that $\sigma^\dagger_{j,k} = \sigma_{-j,-k}$, as shown by the following simple calculation:

$$\sigma^\dagger_{j,k} = \omega^{-cjk} (X^j Z^k)^\dagger = \omega^{-cjk} Z^{-k} X^{-j}, \tag{2.9}$$

$$= \omega^{-cjk} \omega^{jk} X^{-j} Z^{-k} = \omega^{-cjk} \omega^{jk} \omega^{-cjk} \sigma_{-j,-k}, \tag{2.10}$$

$$= \omega^{(1-2c)jk} \sigma_{-j,-k} = \sigma_{-j,-k}, \tag{2.11}$$

where in the last line the aforementioned constant $c = (1-d)/2$ was substituted. Using this relation, and after relabelling, Eq. (2.8) reduces to

$$\alpha^*_{j,k} = \alpha_{-j,-k}. \tag{2.12}$$

Without including the extra phase factor, the above relationship will have some phase dependence, which could complicate many of the expressions that will be calculated in the next section. So in effect, these phases have been absorbed in the definition of the Pauli operators $\sigma_{j,k}$. In addition, a direct implication of Eq. (2.12) is that only half the Bloch components, or $(d^2 - 1)/2$, are *independent*, as the other half are simply the complex conjugates. Hence, only half the Bloch components are needed to define the density operator. Of course, the independent Bloch components are complex numbers and we still have

$(d^2 - 1)$ *real* parameters defining the density operator. A Bloch component $\alpha_{j,k}$ can be evaluated using the following relation:

$$\alpha_{j,k} = \mathrm{tr}(\rho\sigma^{\dagger}_{j,k}) = \mathrm{tr}(\rho\sigma_{-j,-k}). \tag{2.13}$$

Beyond the qubit case, it is not possible to visualise the entire state space with a geometrical picture similar to the Bloch sphere. However, there have been some attempts to study the geometry of the state space in higher dimensions [23, 16], and to define a Bloch-type representation for qutrits [99, 67]. It is interesting to have a general intuition of the geometric features of qudit state space (and the qutrit space in particular). Having fixed normalization and Hermiticity, a *mixed* qutrit state is described by a complex vector $\boldsymbol{\alpha} \in \mathbb{C}^{(d^2-1)/2}$. Associated with this complex vector space are few geometrical measures, which are the inner product $\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle = \sum_j \alpha_j^* \beta_j$, and a norm $||\boldsymbol{\alpha}|| = \sqrt{\langle \boldsymbol{\alpha}, \boldsymbol{\alpha} \rangle}$. The relation between these concepts and the density matrix representation is

$$\mathrm{tr}(\rho^{\dagger}(\boldsymbol{\alpha})\rho(\boldsymbol{\beta})) = (1 + 2\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle)/d. \tag{2.14}$$

Recall that all states must obey $\mathrm{tr}(\rho^2) \leq 1$, and using the above equation, this implies that

$$|\boldsymbol{\alpha}|^2 \leq \frac{d-1}{2}. \tag{2.15}$$

This entails that all physical states are within a Bloch-like *ball* of radius $\sqrt{(d-1)/2}$ about the origin, with the origin being the maximally-mixed state and the pure states on the surface of the Bloch ball. However, the above condition does not guarantee that all the states on the surface of the Bloch ball are positive physical states. The qubit state space is of course a special case in which all the points on the surface of the Bloch sphere correspond to positive physical states. The additional condition required to ensure the positivity of all states, as shown in [86], is $\mathrm{tr}(\rho^2) = \mathrm{tr}(\rho^3) = 1$.

In general, identifying the vectors $\boldsymbol{\alpha}$ corresponding to physical states (including mixed states) $\rho(\boldsymbol{\alpha})$ is not a trivial task. Nevertheless, within certain higher-dimensional planes, called hyperplanes, the structure of the state space is very simple. Consider hyperplanes defined by a set of d positive *orthonormal* density operators, $\{\rho(\boldsymbol{\alpha}_j)\}$. Such states must satisfy the orthogonality condition $\mathrm{tr}(\rho(\boldsymbol{\alpha}_j)\rho(\boldsymbol{\alpha}_k)) = \delta_{j,k}$. Geometrically, this is equivalent to

$$\langle \boldsymbol{\alpha}_j, \boldsymbol{\alpha}_k \rangle = \frac{1}{2}(d\delta_{j,k} - 1). \tag{2.16}$$

We now consider the hyperplane spanned by the orthogonal vectors $\{\boldsymbol{\alpha}_j\}$, such that $\boldsymbol{\gamma} = \sum_j a_j \boldsymbol{\alpha}_j$. It follows that an operator $\rho(\boldsymbol{\gamma})$ is positive if and only if $\sum_j a_j \leq 1$ and $b_j = |b_j|$ for all j. Hence, the physical Bloch components $\boldsymbol{\alpha}$ lie within the convex polytope with γ_j as vertices. We have d vertices all equally separated from each other and residing within a real $d-1$ dimensional hyperplane. For $d = 3$, and a corresponding 2-dimensional plane, the physical states reside within an equilateral triangle. We will see explicit examples below of how qutrit orthogonal states form equilateral triangles.

Finally, it proves useful to discuss the *orbits* of the single qudit Pauli group $\mathcal{P}_d$ when acting on a general qudit state $\rho(\boldsymbol{\alpha})$ with *conjugation* being the group action. The singular orbit of a general state $\rho(\boldsymbol{\alpha})$, denoted by $\text{Orb}(\rho(\boldsymbol{\alpha}))$, is defined as

$$\text{Orb}(\rho(\boldsymbol{\alpha})) = \{\rho(\boldsymbol{\alpha}') = \sigma_{\boldsymbol{j}',\boldsymbol{k}'}\ \rho(\boldsymbol{\alpha})\ \sigma^{\dagger}_{\boldsymbol{j}',\boldsymbol{k}'}\ \ \forall\, \sigma_{\boldsymbol{j}',\boldsymbol{k}'} \in \mathcal{P}_n\}. \tag{2.17}$$

In our Bloch representation we are using Pauli group elements as a basis set for the states, thus conjugation by Pauli operators will not transform the basis elements themselves, but will add a phase of the form ω^j for some $j \in \mathbb{Z}_d$. The overall effect of this conjugation is to add certain phases to the Bloch components. The exact form of the phases is exactly given by:

$$\rho(\boldsymbol{\alpha}') = \sum_{(\boldsymbol{j},\boldsymbol{k})} \alpha_{\boldsymbol{j},\boldsymbol{k}} \sigma_{\boldsymbol{j}',\boldsymbol{k}'} \sigma_{\boldsymbol{j},\boldsymbol{k}} \sigma_{-\boldsymbol{j}',-\boldsymbol{k}'}, \tag{2.18}$$

$$= \sum_{(\boldsymbol{j},\boldsymbol{k})} \omega^{\boldsymbol{j}\boldsymbol{k}'-\boldsymbol{j}'\boldsymbol{k}} \alpha_{\boldsymbol{j},\boldsymbol{k}} \sigma_{\boldsymbol{j},\boldsymbol{k}}. \tag{2.19}$$

where the $\sigma^{\dagger}_{\boldsymbol{j}',\boldsymbol{k}'} = \sigma_{-\boldsymbol{j}',-\boldsymbol{k}'}$, the commutation Eq. (2.6) and composition Eq. (2.4) relations were used in the last step.

Qutrits

So far the discussion has been for all prime dimensions, but in the remainder of this chapter we will often discuss the qutrit case only. Therefore, we shall outline some of above results explicitly for the $d = 3$ case. Our definition of the qutrit Pauli basis set in Eq. (2.3) is $\sigma_{j,k} = \omega^{-jk} X^j Z^k$, where $\omega = e^{2\pi i/3}$

$\sigma_{\boldsymbol{j'},\boldsymbol{k'}}$	$\sigma_{\boldsymbol{j'},\boldsymbol{k'}}\rho(\alpha_{1,0},\alpha_{0,1},\alpha_{1,1},\alpha_{1,2})\sigma^{\dagger}_{\boldsymbol{j'},\boldsymbol{k'}}$
$\sigma_{0,0}$	$\rho(\alpha_{1,0},\alpha_{0,1},\alpha_{1,1},\alpha_{1,2})$
$\sigma_{\pm 1,0}$	$\rho(\alpha_{1,0},\omega^{\mp 1}\alpha_{0,1},\omega^{\mp 1}\alpha_{1,1},\omega^{\pm 1}\alpha_{1,2})$
$\sigma_{0,\pm 1}$	$\rho(\omega^{\pm 1}\alpha_{1,0},\alpha_{0,1},\omega^{\pm 1}\alpha_{1,1},\omega^{\pm 1}\alpha_{1,2})$
$\sigma_{\pm 1,\pm 1}$	$\rho(\omega^{\pm 1}\alpha_{1,0},\omega^{\mp 1}\alpha_{0,1},\alpha_{1,1},\omega^{\mp 1}\alpha_{1,2})$
$\sigma_{\pm 1,\mp 1}$	$\rho(\omega^{\mp 1}\alpha_{1,0},\omega^{\mp 1}\alpha_{0,1},\alpha_{1,1}\omega^{\pm 1},\alpha_{1,2})$

Table 2.1: The qutrit orbital Bloch phases.

and $c = -1$. The explicit qutrit $\rho(\boldsymbol{\alpha})$ state is:

$$\begin{aligned}\rho(\boldsymbol{\alpha}) &\equiv \rho(\alpha_{1,0},\alpha_{0,1},\alpha_{1,1},\alpha_{1,2}),\\ &= \frac{1}{3}\Big(\mathbb{1} + \alpha_{1,0}\sigma_{1,0} + \alpha^{*}_{1,0}\sigma_{2,0} + \alpha_{0,1}\sigma_{0,1} + \alpha^{*}_{0,1}\sigma_{0,2} + \\ &\qquad \alpha_{1,1}\sigma_{1,1} + \alpha^{*}_{1,1}\sigma_{2,2} + \alpha_{1,2}\sigma_{1,2} + \alpha^{*}_{1,2}\sigma_{2,1}\Big).\end{aligned} \tag{2.20}$$

As we can see, completely specifying a qutrit state would only require 4 complex independent parameters $(\alpha_{1,0},\alpha_{0,1},\alpha_{1,1},\alpha_{1,2})$. In terms of the Bloch components, the purity condition $\mathrm{tr}(\rho^2) = 1$ for a general qutrit state can be shown to be $||\boldsymbol{\alpha}|| \leq 1$.

The Pauli group orbits for a single qutrit state can be evaluated using Eq. (2.19). We are interested in knowing how the phases of the four independent Bloch components change when an element from the nine qutrit $\sigma_{j,k}$ operators is conjugated with the general qutrit state. The result is summarised in Table 2.1. We refer to these phases as the orbital Bloch phases. These represent the phases which generate the set of states Pauli-equivalent to any state. The magic states that we will find are unique up to a Bloch orbital phase. In other words, inserting one of the phases from the set in Table 2.1 into the Bloch components of the magic states would also give a valid magic state with the same distillation properties.

2.3.2 Protocol Structure

Using the definitions and notations we have developed in the previous section, we will show how the three steps of a MSD protocol described in Sec. 2.1 can be formulated to study the distillation properties of any stabilizer code of any prime dimension.

Resource state preparation: Recall that the computational model considered when studying MSD consists of perfect stabilizer operations and the ability to prepare n identical copies of a noisy resource state ρ_{res}. By repeating the preparation procedure n times, the state $\rho_{\text{res}}^{\otimes n}$ will be prepared. As an input to the MSD protocol we consider a general state

$$\rho(\boldsymbol{\alpha})^{\otimes n} = \frac{1}{d^n} \sum_{(\boldsymbol{j},\boldsymbol{k}) \in \mathbb{Z}_d^n} \alpha_{j_1 \dots j_n, k_1 \dots k_n} \sigma_{j_1 \dots j_n, k_1 \dots k_n}. \tag{2.21}$$

By performing the remaining steps of the iteration on the above general form, we will determine the map on the Bloch components of the initial general state $\rho(\boldsymbol{\alpha})$. Then, by searching the state space for different initial states (different $\boldsymbol{\alpha}$ vectors), we can identify the resource states as those that when used as an input to the protocol the output state has a higher fidelity with respect to a pure non-stabilizer state (the magic state), and ultimately distilling this non-stabilizer pure state. If the search is done systematically, one can in principle identify the entire region of resource states ρ_{res}.

Stabilizer measurement and Decoding: The $(n-k)$ stabilizer generators of a stabilizer code $[\![n,k,\delta]\!]_d$ are measured successively post-selecting on the $+1$ outcome of each measurement. That is, if one of the outcomes is ω^k (for some non-zero $k \in \mathbb{Z}_d$) then the protocol is aborted[4], and the procedure is repeated with a fresh state $\rho^{\otimes n}$. Also, it is important to notice that the error correction code is not being used for the usual purpose of correcting errors since the syndrome measurements are performed on the product state $\rho^{\otimes n}$. If successful, the measurement of the stabilizers simply project the state to the code-space. The projector operator describing this measurement procedure can be put into the following convenient form:

$$\Pi = \frac{1}{d^{n-k}} \prod_{j=1}^{n-k} (I + g_j) = \frac{1}{d^{n-k}} \sum_{\boldsymbol{j} \in \mathbb{Z}_d^{n-k}} g_1^{j_1} g_2^{j_2} \dots g_{n-k}^{j_{n-k}}. \tag{2.22}$$

[4] As pointed out earlier, post-selection can sometime be avoided if there exists a correction Clifford that maps the state to the '$+1$' eigenspace. We will not come across such an example until the next chapter.

After the successful measurements, the following map will be performed:

$$\rho^{\otimes n} \mapsto \frac{\Pi\rho^{\otimes n}\Pi^{\dagger}}{\mathrm{tr}\left(\rho^{\otimes n}\Pi\right)}. \tag{2.23}$$

The resultant state is decoded via a Clifford operator [36]. In a Heisenberg picture, the decoding operation maps logical operators $\bar{\sigma}_{j,k}$ on the code-space to unencoded operators acting on a single qudit. Hence, the output Bloch-components after decoding, denoted by $\alpha^{\mathrm{out}}_{j,k}$, correspond to the components of the logical operators prior to decoding $\bar{\sigma}_{j,k}$. After one round of the distillation, these can be evaluated as follows:

$$\alpha^{\mathrm{out}}_{j,k} = \frac{\mathrm{tr}(\Pi\rho^{\otimes n}\Pi^{\dagger}\bar{\sigma}^{\dagger}_{j,k})}{\mathrm{tr}(\rho^{\otimes n}\Pi)}. \tag{2.24}$$

The resultant expressions for the output Bloch components will be multi-variable complex polynomials of order n. For the qutrit codes we consider next, we have not found analytic solutions for the fixed points of the map. However, the problem is tractable by using numerical methods to study the distillation behaviours and find the fixed-points to a high accuracy.

2.4 Qutrit Distillation

Using the generalised formulation of MSD in the previous section, we study in this section the distillation properties of the five-qutrit code (with the stabilizer generators shown in Tab. (2.2)). This code was chosen because of its small size and also for being the simplest generalisation of the five-qubit code.

2.4.1 $[\![5,1,3]\!]_3$ Distillation

The stabilizer generators of the general five-qudit code $[\![5,1,3]\!]_d$ takes the same form in all dimensions [100, 41]. It is usually presented in terms of the conventional generalised Pauli operators of Eq. (2.2), as shown in Tab. (2.2). It is easy to see that these generators commute and $\bar{X}$ and $\bar{Z}$ form logical operators.

Based on these stabilizers we will study the distillation map of Eq. (2.24) for the four Bloch components of a general input qutrit state. Notice that the decoding of a stabilizer code is not unique, but one of an equivalence class of unitaries—a coset of the Clifford group—which are all equally valid choices.

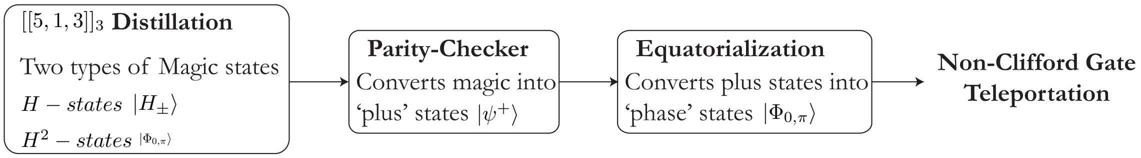
$[[5,1,3]]_3$ Distillation
Two types of Magic states
$H-states\ |H_{\pm}\rangle$
$H^2-states\ |\Phi_{0,\pi}\rangle$
Parity-Checker
Converts magic into 'plus' states $|\psi^+\rangle$
Equatorialization
Converts plus states into 'phase' states $|\Phi_{0,\pi}\rangle$
Non-Clifford Gate Teleportation

Figure 2.3: An outline of the different qutrit protocols in Secs. 2.4 and 2.5 and how they are related. In Sec. 2.4 we discover two types of magic states distillable by the $[\![5,1,3]\!]_3$, the so-called $H-$states $|H_{\pm}\rangle$ and H^2-states $|\phi\rangle$. We then consider two sub-protocols in Sec. 2.5, the Parity-Checker and Equatorialization, that produce a suitable magic states (the phase states $|\Phi_{0,\pi}\rangle$) which are then used to implement a qutrit non-Clifford gate.

$g_1 =$	X	Z	Z^{-1}	X^{-1}	I
$g_2 =$	I	X	Z	Z^{-1}	X^{-1}
$g_3 =$	X^{-1}	I	X	Z	Z^{-1}
$g_4 =$	Z^{-1}	X^{-1}	I	X	Z
$\bar{X} =$	Z	Z	Z	Z	Z
$\bar{Z} =$	X	X	X	X	X

Table 2.2: The stabilizer generators of the five-qudit code $[\![5,1,3]\!]_d$.

The choice of decoding will affect the *iterative* distillation behaviour. The decoding specified by the logical operators in Tab. (2.2) is the canonical one, though we found the behaviour was simplified by following each iterate with the following additional correction Clifford unitary U_c:

$$U_c = \begin{pmatrix} 1 & \omega & \omega \\ \omega^2 & \omega & \omega^2 \\ \omega^2 & \omega^2 & \omega \end{pmatrix}, \tag{2.25}$$

where this maps the qutrit density operator $\rho(\boldsymbol{\alpha}) \equiv \rho(\alpha_{1,0}, \alpha_{0,1}, \alpha_{1,1}, \alpha_{1,2})$ such that:

$$\{\alpha_{1,0}, \alpha_{0,1}, \alpha_{1,1}, \alpha_{1,2}\}_{U_c} \mapsto \{\alpha^*_{1,2}, \alpha_{1,1}, \alpha^*_{0,1}, \alpha_{1,0}\}. \tag{2.26}$$

$g_1 =$	$\sigma_{1,0}$	$\sigma_{0,1}$	$\sigma_{0,-1}$	$\sigma_{-1,0}$	$\sigma_{0,0}$
$g_2 =$	$\sigma_{0,0}$	$\sigma_{1,0}$	$\sigma_{0,1}$	$\sigma_{0,-1}$	$\sigma_{-1,0}$
$g_3 =$	$\sigma_{-1,0}$	$\sigma_{0,0}$	$\sigma_{1,0}$	$\sigma_{0,1}$	$\sigma_{0,-1}$
$g_4 =$	$\sigma_{0,-1}$	$\sigma_{-1,0}$	$\sigma_{0,0}$	$\sigma_{1,0}$	$\sigma_{0,1}$
$\sigma^L_{1,0} =$	$\sigma_{0,1}$	$\sigma_{0,1}$	$\sigma_{0,1}$	$\sigma_{0,1}$	$\sigma_{0,1}$
$\sigma^L_{0,1} =$	$\sigma_{1,0}$	$\sigma_{1,0}$	$\sigma_{1,0}$	$\sigma_{1,0}$	$\sigma_{1,0}$

Table 2.3: The stabilizer generators of the five-qudit code and the qutrit logical operators expressed in the $\sigma_{j,k}$ notation.

Without this corrective Clifford one observes a cycling behaviour throughout the distillation process, we will discuss this behaviour at the end of Sec. (2.4.3).

We will now compute the exact distillation map on the Bloch components after a single round of distillation. We start by expressing the stabilizer generators and the logical operators of the $[\![5,1,3]\!]_3$ in terms of the $\sigma_{j,k}$ operators as shown in Tab. (2.3). The input to the distillation protocol is five identical copies of a qutrit state $\rho(\boldsymbol{\alpha})$. Using Eq. (2.21), the input state is expressed as:

$$\rho(\boldsymbol{\alpha})^{\otimes 5} = \frac{1}{3^5} \sum_{(\boldsymbol{j},\boldsymbol{k})\in\mathbb{Z}_3^5} \alpha_{j_1\dots j_5,k_1\dots k_5}\sigma_{j_1\dots j_5,k_1\dots k_5}, \tag{2.27}$$

Using Eq. (2.22), a successful measurement of the four stabilizer generators with outcome '+1' corresponds to the following projector:

$$\Pi = \frac{1}{3^4} \sum_{\boldsymbol{q}\in\mathbb{Z}_3^4} g_1^{q_1} g_2^{q_2} g_3^{q_3} g_4^{q_4}. \tag{2.28}$$

Substituting the stabilizer generators in Tab. (2.3) into the above expression and using the composition law in Eq. (2.4), the projector becomes

$$\begin{aligned}\Pi =& \frac{1}{81} \sum_{\boldsymbol{q}\in\mathbb{Z}_3^4} \Big(\sigma_{(q_1-q_3),(-q_4)} \otimes \sigma_{(q_2-q_4),(q_1)} \otimes \sigma_{(q_3),(-q_1+q_2)} \\ & \otimes \sigma_{(-q_1+q_4),(-q_2+q_3)} \otimes \sigma_{(-q_2),(-q_3+q_4)}\Big).\end{aligned} \tag{2.29}$$

We can simplify the notation of the above expression by using Eq. (2.5), which removes the tensor product sign:

$$\Pi = \frac{1}{81} \sum_{\boldsymbol{q} \in \mathbb{Z}_3^4} \sigma_{(q_1-q_3)_1(q_2-q_4)_2(q_3)_3(-q_1+q_4)_4(-q_2)_5,(-q_4)_1(q_1)_2(-q_1+q_2)_3(-q_2+q_3)_4(-q_3+q_4)_5}. \tag{2.30}$$

The distillation map in Eq. (2.24) can be put into a simpler form:

$$\alpha_{j,k}^{\text{out}} = \frac{\text{tr}(\Pi \rho^{\otimes n} \Pi^\dagger \bar{\sigma}_{j,k}^\dagger)}{\text{tr}(\rho^{\otimes n}\Pi)} = \frac{\text{tr}(\rho^{\otimes n}\Pi \bar{\sigma}_{-j,-k})}{\text{tr}(\rho^{\otimes n}\Pi)}. \tag{2.31}$$

where in the last step Eq. (2.13), $\Pi = \Pi^\dagger$, $[\bar{\sigma}_{j,k}, \Pi] = 0$ and the cyclic property of the trace were used. The remaining task is to substitute Eqs. (2.27) and (2.30) into Eq. (2.31) to calculate the distillation map on the Bloch components.

Let us start by evaluating $\text{tr}(\rho^{\otimes 5}\Pi)$. Recall that all the $\sigma_{j,k}$ operators are traceless except for the identity operator $\sigma_{0,0}$. Therefore, the only terms that will survive in $\text{tr}(\rho^{\otimes 5}\Pi)$ are the coefficients of the identity operator. We get the identity operator in $\rho^{\otimes 5}\Pi$ when the $\sigma_{j,k}$ operators in $\rho^{\otimes 5}$ and the $\sigma_{j',k'}$ in Π have the opposite subscripts (i.e. $\sigma_{j,k}\sigma_{j',k'} = \sigma_{0,0}$ if and only if $j = -j'$ and $k = -k'$). As a result, $\text{tr}(\rho^{\otimes 5}\Pi)$ will be the sum of all the Bloch components that are the coefficient of the $\sigma_{j,k}$ operators such that the subscripts (j,k) are the negative of the subscripts in Eq. (2.29). In fact, since the summation is over all the elements of the ring $\mathbb{Z}_3^4$ it is possible to multiply all the subscripts by (-1) without changing the actual value of the summation. Hence, $\text{tr}(\rho^{\otimes 5}\Pi)$ can be compactly expressed as follows

$$\text{tr}(\rho^{\otimes 5}\Pi) = \frac{1}{81} \sum_{\boldsymbol{q} \in \mathbb{Z}_3^4} \alpha_{(q_1-q_3)_1(q_2-q_4)_2(q_3)_3(-q_1+q_4)_4(-q_2)_5,(-q_4)_1(q_1)_2(-q_1+q_2)_3(-q_2+q_3)_4(-q_3+q_4)_5}. \tag{2.32}$$

In a similar way, we can express $\text{tr}(\rho\Pi\bar{\sigma}_{-j,-k})$ for all four logical operators. For example, in the case of evaluating the output Bloch component $\alpha_{1,0}^{\text{out}}$, Eq. (2.31) becomes

$$\alpha_{1,0}^{\text{out}} = \frac{\text{tr}(\rho^{\otimes 5}\Pi\bar{\sigma}_{-1,0})}{\text{tr}(\rho^{\otimes 5}\Pi)}, \tag{2.33}$$

with $\text{tr}(\rho^{\otimes 5}\Pi\bar{\sigma}_{-1,0})$ given by

$$\text{tr}(\rho^{\otimes 5}\Pi\bar{\sigma}_{-1,0}) = \frac{1}{81} \sum_{\boldsymbol{q} \in \mathbb{Z}_3^4} \alpha_{(q_1-q_3)_1(q_2-q_4)_2(q_3)_3(-q_1+q_4)_4(-q_2)_5,(-q_4+1)_1(q_1+1)_2(-q_1+q_2+1)_3(-q_2+q_3+1)_4(-q_3+q_4+1)_5}. \tag{2.34}$$

We have evaluated the expressions for the four output Bloch components. However, writing them out in terms of $\alpha_{j,k}$ notation is cumbersome. Therefore, for clarity, we will relabel the four qutrit Bloch components as follows $(\alpha_{1,0}, \alpha_{0,1}, \alpha_{1,1}, \alpha_{1,2}) \equiv (A, B, C, D)$. For example, tr$(\rho^{\otimes 5}\Pi)$ is given in Eq. (2.35), where the subscript r represent the number of the distillation rounds with $r = 0$ corresponding to the initial input state.

$$\begin{aligned}\mathrm{tr}\left(\rho^{\otimes 5}\Pi\right) =& \frac{1}{81}\Big(1 + 10\left(|A_r|^2 + |D_r|^2\right)\left(|B_r|^2 + |C_r|^2\right) + 5\big(B_r^2 A_r^* C_r^{*2} + D_r^2 A_r^{*2} B_r^* + \\ & D_r\left(A_r^2 D_r C_r^* + B_r^2 C_r^2\right) + B_r^{*2}\left(A_r C_r^2 + C_r^{*2} D_r^*\right) + D_r^{*2}\left(A_r^2 B_r + C_r A_r^{*2}\right)\big)\Big). \end{aligned} \tag{2.35}$$

Furthermore, it can be shown that the resultant expressions for the four output Bloch components can compactly be expressed in terms of the following single function:

$$\begin{aligned}\mathcal{F}(A, B, C, D) =& \frac{1}{81}\Big(B_r^5 + 10B_r\left(D_r A_r^* + B_r^*\right)\left(A_r C_r^* + C_r D_r^*\right) + 5\big(A_r C_r^2 |A_r|^2 + D_r^2\left(A_r B_r^{*2} + C_r\right) + \\ & A_r^{*2}\left(B_r^{*2} D_r^* + C_r^*\right) + D_r^{*2}\left(A_r^2 + D_r C_r^{*2}\right) + |C_r|^4 B_r^*\big)\Big)/\mathrm{tr}\left(\rho^{\otimes 5}\Pi\right). \end{aligned} \tag{2.36}$$

Based on this function the distillation map can be expressed as

$$A_{r+1} = \mathcal{F}(A, B, C, D), \tag{2.37}$$

$$B_{r+1} = \mathcal{F}(B^*, A, D, C^*), \tag{2.38}$$

$$C_{r+1} = \mathcal{F}(A^*, C, B, D), \tag{2.39}$$

$$D_{r+1} = \mathcal{F}(B^*, D^*, A^*, C). \tag{2.40}$$

These four expressions represents the complete distillation map, as the remaining four components are simply the complex conjugates of these expressions. However, the above expressions do not incorporate the additional corrective Clifford U_c. We need to ensure that the map in Eq. (2.26) is applied after every iteration. This can easily be achieved in our formalism by the appropriate relabelling as follows:

$$A_{r+1} = \mathcal{F}(D^*, C, B^*, A), \tag{2.41}$$

$$B_{r+1} = \mathcal{F}(C^*, D^*, A, B), \tag{2.42}$$

$$C_{r+1} = \mathcal{F}(D, B^*, C, A), \tag{2.43}$$

$$D_{r+1} = \mathcal{F}(C^*, A^*, D, B^*), \tag{2.44}$$

which is the corrected distillation map.

In order to calculate the fixed points of this map analytically, one would have to solve the above simultaneous complex multi-variable polynomials of order 5. It is known from the famous Abel-Ruffini theorem that there is no algebraic solution for a general polynomial of order five or above. Therefore, the best way to discover the fixed points of the distillation is through numerical means. We started with initial states $\rho(A, B, C, D)$ for certain Bloch components and computed the above expressions for a number of iterations, and observed whether there is a convergence toward a fixed point. If a fixed point corresponds to a non-stabilizer pure state, then it is a magic state. We identify two qualitatively different families of magic states. Firstly, those that are in the Hadamard plane, which satisfy $H\rho H^\dagger = \rho$. Secondly, we investigate the distillation of an interesting set of states outside the Hadamard plane. Each of these studies has its own merits. All quantum states can be mapped onto the Hadamard plane and so this is the study of most generic value.

2.4.2 Hadamard-like Distillation

In the qubit case, the eigenstates of the Hadamard gate are known to be magic states, distillable by the five-qubit code $[\![5, 1, 3]\!]_2$. This result was, however, not presented in the literature, so we include it in a footnote[5]. Since the exact generalised form of the Hadamard gate is defined in Eq. (1.13), a good starting point would be to investigate whether the qutrit Hadamard eigenstates can be distilled by $[\![5, 1, 3]\!]_3$. We begin by outlining some of the structural properties of the qutrit Hadamard eigenspace.

[5] We have repeated the calculations in Ref. [27] of the 5-qubit code distillation but for distilling the qubit Hadamard states instead of the T states. The output error probability ϵ^{out} as a function of the initial error probability ϵ can be shown to be:

$$\epsilon^{\text{out}}(\epsilon) = \frac{\epsilon(5 + 4\epsilon(5 - 4\epsilon(5 + (\epsilon - 5)\epsilon)))}{9 + 40\epsilon(\epsilon - 1)(2\epsilon(\epsilon - 1) + 1)}.$$

Solving the above equation for $\epsilon^{\text{out}}(\epsilon) = \epsilon$ gives an error threshold value of $\frac{1}{6}(3 - \sqrt{6})$. Also, for sufficiently small ϵ, $\epsilon^{\text{out}}(\epsilon) \approx 5\epsilon/9$. This suggests that there is a slow linear error suppression in contrast to the quadratic error suppression for the T–state distillation.

In the matrix representation the qutrit Hadamard is given by

$$H = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix}, \tag{2.45}$$

where $\omega = e^{2\pi i/3}$ and H has the eigenvalues $(+1, -1, i)$. We label the corresponding three eigenstates as $(|H_+\rangle, |H_-\rangle, |H_i\rangle)$. The density operators of the eigenstates have the form:

$$|H_{+1}\rangle\langle H_{+1}| \equiv \rho(a, a, b, b), \tag{2.46}$$

$$|H_{-1}\rangle\langle H_{-1}| \equiv \rho(b, b, a, a), \tag{2.47}$$

$$|H_i\rangle\langle H_i| \equiv \rho(c, c, c, c), \tag{2.48}$$

where $a = \frac{1}{4}\left(1 + \sqrt{3}\right)$, $b = \frac{1}{4}\left(1 - \sqrt{3}\right)$ and $c = -\frac{1}{2}$, are real parameters. This basis of pure states all lie on the hyperplane of operators of the form $\rho(x, x, y, y)$. Probabilistic mixtures of these states form an equilateral triangle, and as reviewed earlier, all points outside this triangle correspond to non-physical operators (with negative eigenvalues). Furthermore, any qutrit state can be projected onto the Hadamard plane by applying the following twirling operation to each copy of the input state ρ:

$$\rho \mapsto \sum_{j=1}^{4} \frac{1}{4} H^j \rho H^{j^\dagger}. \tag{2.49}$$

This twirling operation maps the general Bloch components as follows:

$$\alpha_{1,0} \text{ and } \alpha_{0,1} \mapsto \frac{\mathrm{Re}(\alpha_{1,0} + \alpha_{0,1})}{2}, \tag{2.50}$$

$$\alpha_{1,1} \text{ and } \alpha_{1,2} \mapsto \frac{\mathrm{Re}(\alpha_{1,1} + \alpha_{1,2})}{2}. \tag{2.51}$$

As such, we are interested in studying the distillable regions within the Hadamard plane, i.e. the states corresponding to the points inside the triangle.

In studying the distillable resource region in the Hadamard plane, it is also informative to rule out regions for which distillation is impossible by any protocol. We have already identified two such regions in Sec. 2.1. First, clearly, all stabilizer states are undistillable, and this is shown by the red region in

Fig. (2.4i). Second, results of Veitch *et al.* [159] prove undistillability of all qutrit states with a positive Wigner function. The numerically calculated positive-region is shown in Fig. (2.4i) as the yellow area[6].

Running the $[\![5,1,3]\!]_3$ distillation for all the remaining points as inputs states we have discovered that both the $|H_+\rangle$ and $|H_-\rangle$ states are distillable[7], but that $|H_i\rangle$ is not an attractor. The distillable regions are enclosed by the blue dashed triangles. The states $|H_+\rangle$ and $|H_-\rangle$ are equally valuable as magic states because $U_c\,|H_\pm\rangle \propto |H_\mp\rangle$, which perhaps also explains the symmetry in their distillation regions.

The path of the distillation takes the form shown in Fig. (2.4ii) where we have chosen the $|H_+\rangle$ blue triangle as an example. The small black points are few examples of input states to the distillation, and the black lines represent the distillation paths toward the $|H_+\rangle$ state. Notice how the distillation does not follow a straight line, but rather a curved path. In analogy to the "magic axes" in the qubit case, we call this plane the Hadamard magic plane. The curved distillation path can be understood by studying how the noise of a resource state in the Hadamard plane is suppressed in different directions by the distillation. We start by considering a general state $\rho_\triangle$ inside the triangle of the form

$$\rho_\triangle = (1-\epsilon_1-\epsilon_2)\,|H_+\rangle\,\langle H_+| + \epsilon_1\,|H_-\rangle\,\langle H_-| + \epsilon_2\,|H_i\rangle\,\langle H_i|\,, \tag{2.52}$$

with $\epsilon_1+\epsilon_2 \leq 1$. For clarity, we write the Bloch components of the above state as $\rho_\triangle(A_\triangle, B_\triangle, C_\triangle, D_\triangle)$. They can be calculated explicitly using Eq. (2.13) as

$$\begin{aligned} A_\triangle = B_\triangle &= \frac{1}{4}\left(1+\sqrt{3}-2\sqrt{3}\epsilon_1-\left(3+\sqrt{3}\right)\epsilon_2\right), \\ C_\triangle = D_\triangle &= \frac{1}{4}\left(1-\sqrt{3}+2\sqrt{3}\epsilon_1-\left(3-\sqrt{3}\right)\epsilon_2\right). \end{aligned} \tag{2.53}$$

[6] We have calculated these states by using the formalism in [159]. Here, the generalised Pauli operators are define as $T_{\boldsymbol{u}} \equiv T_{(j,k)} = \omega^{-jk/2}Z^jX^k$, where $\boldsymbol{u} = (j,k) \in \mathbb{Z}_\mathrm{d}^2$. Associated with each state ρ, when expressed in the $T_{\boldsymbol{u}}$ basis, is a discrete representation—called the Wigner representation, denoted by $W_\rho(\boldsymbol{u})$—which is uniquely specified by the *phase space point operators* $A_{\boldsymbol{u}}$. The phase point operators are defined as $A_{\mathbf{0}} = \sum_{\boldsymbol{u}} T_{\boldsymbol{u}}$ and $A_{\boldsymbol{u}} = T_{\boldsymbol{u}}A_{\mathbf{0}}T_{\boldsymbol{u}}^\dagger$. The results of [159] show that, for all odd prime dimensions, if a state ρ has a positive Wigner representation, such that $W_\rho(\boldsymbol{u}) \geq 0 \forall \boldsymbol{u} \in \mathbb{Z}_d^2$, then it is a bound state for MSD.

[7] The $|H_\pm\rangle$ states are unique up to a Bloch orbital phase. In other words, inserting one of the phases from the set in Table 2.1 would also give a H–type magic state with the same distillation properties. In general, this is not sufficient to specify the complete set of the H–type magic states.

Since we know the general distillation map for any set of Bloch components, we can simply substitute the above expressions into Eqs. (2.41-2.44) to evaluate the output Bloch components $\boldsymbol{\alpha}^{\text{out}} = (A^{\text{out}}_{\triangle}, B^{\text{out}}_{\triangle}, C^{\text{out}}_{\triangle}, D^{\text{out}}_{\triangle})$. The output state is then $\rho^{\text{out}}_{\triangle} = \rho(\boldsymbol{\alpha}^{\text{out}})$. We have numerically calculated the output ϵ^{out}_1 and ϵ^{out}_2 to the first-order terms and obtained

$$\epsilon^{\text{out}}_1(\epsilon_1, \epsilon_2) = \langle H_-| \, \rho^{\text{out}t}_{\triangle} \, |H_-\rangle \approx (0.38 + 0.09\epsilon_2)\, \epsilon_1, \tag{2.54}$$

$$\epsilon^{\text{out}}_2(\epsilon_1, \epsilon_2) = \langle H_i| \, \rho^{\text{out}}_{\triangle} \, |H_i\rangle \approx (0.77 + 3.55\epsilon_1)\, \epsilon_2. \tag{2.55}$$

The above expressions show an asymmetric error suppression in the ϵ_1 (along the $|H_+\rangle$—$|H_-\rangle$ line) and ϵ_2 (along the $|H+\rangle$—$|H_i\rangle$ line) directions. The particular distillation paths of Fig. (2.4ii) can be explained by observing the difference in the coefficients of $\epsilon^{\text{out}}(\epsilon_1, 0)$ and $\epsilon^{\text{out}}(0, \epsilon_2)$, where we see that in the distillation region of the $|H_+\rangle$ state there is a stronger attraction toward the $|H_i\rangle$ state compared to the $|H_-\rangle$ state.

The above analysis shows that the performance of the $[\![5, 1, 3]\!]_3$ code in distilling the qutrit Hadamard states is not as good as the qubit case where the 15 qubit code by [27] has an output error probability of $\epsilon^{\text{out}} \approx 35\epsilon^3$. This is to be expected given the similar performance of the five-qubit code in distilling the H–type qubit magic state[5].

The state $|H_i\rangle$ is not distillable by $[\![5, 1, 3]\!]_3$. In fact, this state belongs to the family of states with maximally non-positive Wigner function [158]. As we can see this state is the furthest away from the stabilizer region in the Hadamard plane and to bring it to the stabilizer region would require a depolarizing noise with an error threshold of 75% (i.e. $d/(d+1)$ for $d = 3$). Whether such a state is distillable by some stabilizer code is still an open question.

To improve the size of the distillation region we have investigated a qutrit version of the seven-qubit code $[\![7, 1, 3]\!]_2$ proposed in Ref. [136]. We started with the stabilizer generators of $[\![7, 1, 3]\!]_2$ code and by adding the (-1) power to the appropriate X and Z Pauli operators, we constructed a set of generalised 7–qudit commuting stabilizer generators as shown in Tab. (2.4). We repeated the distillation procedure for this set of generators for the case $d = 3$ (exact calculations are omitted here) and we investigated its distillation capability in the Hadamard plane. We found that this code attracts towards the *non-stabilizer* segments of the line joining the $|H_+\rangle$ and $|H_-\rangle$ states with the distillation region enclosed by the green

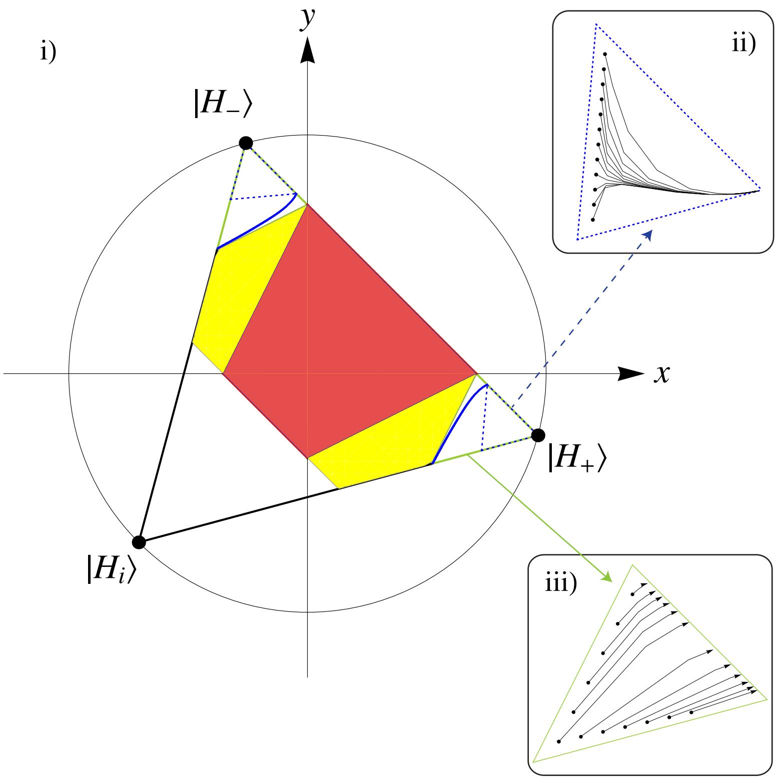

Figure 2.4: A representation of the Hadamard plane. The Hadamard eigenstates are the vertices of the equilateral triangle which lies on a circle of radius $1/\sqrt{2}$. i) The red region contains the stabilizer states. The yellow and red regions combined form the states with positive Wigner function. The dashed blue and the green triangles contain the states that are distillable by the $[\![5,1,3]\!]_3$ and $[\![7,1,3]\!]_3$ codes, respectively. ii) and iii) shows the distillation paths for the $|H_+\rangle$ state and the mixed states for the $[\![5,1,3]\!]_3$ and $[\![7,1,3]\!]_3$ codes, respectively.

$g_1 =$	I	I	I	X^{-1}	X	X	X^{-1}
$g_2 =$	X	I	X^{-1}	I	X^{-1}	I	X
$g_3 =$	I	X	X^{-1}	I	I	X^{-1}	X
$g_4 =$	I	I	I	Z	Z	Z	Z
$g_5 =$	Z	I	Z	I	Z	I	Z
$g_6 =$	I	Z	Z	I	I	Z	Z
$\bar{X} =$	X	X^{-1}	X	X	X^{-1}	X	X^{-1}
$\bar{Z} =$	Z	Z	Z	Z	Z	Z	Z

Table 2.4: The stabilizer generators of the seven-qudit code.

triangle in Fig. (2.4). In other words, the 7−qutrit code distils not pure, but mixed states. Regardless, the protocol may be useful for bringing states into the region distillable by the 5-qutrit code. The distillation path for the $|H_+\rangle$ state is shown in Fig. (2.4iii). This code increase the distillation region as shown by the solid blue curve in Fig. (2.4). For example, a state between the solid blue line and the dashed blue triangle is first distilled by the seven-qutrit code to a state within the dashed blue triangle, after which the $[\![5,1,3]\!]_3$ code is used to distil the $|H_\pm\rangle$ states.

2.4.3 Hadamard-Squared Subspace

In this section, we introduce a second class of magic states distilled by the $[\![5,1,3]\!]_3$ code. This magic state is an eigenstate of the H^2 operator, but lies within a degenerate eigenspace for this operator, and so is not uniquely defined by it. The magic state considered here has the form:

$$|\varphi\rangle = a\,|0\rangle + b\,|1\rangle + b\,|2\rangle\,, \tag{2.56}$$

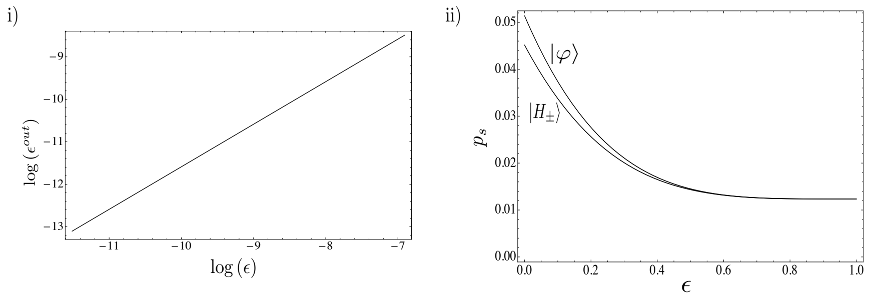

Figure 2.5: i) The log-log plot of the output error probability ϵ^{out} for the input state ρ_{dep} and very small depolarizing noise ϵ. ii) The success probability for the trivial syndrome measurements for the case where the magic states $|\varphi\rangle$ and $|H_\pm\rangle$ are undergoing depolarizing noise.

where up to 4 decimal places

$$a = -0.1203 - 0.0272i, \tag{2.57}$$

$$b = 0.7017. \tag{2.58}$$

The equality of the $|1\rangle$ and $|2\rangle$ components follows from the H^2 symmetry. In the Bloch representation, the $\boldsymbol{\alpha}$ vector is,

$$\boldsymbol{\alpha} = \{0.3236, -0.4772, 0.5438, 0.6098\}, \tag{2.59}$$

with all components are real and being related again by the H^2 symmetry. There are four such distillable magic states which are related by Clifford unitaries (see aside note below). Our numerical analysis shows that these states are clearly attractor fixed points of the distillation protocol, but we do not have closed form analytic expressions for them. As a consequence, we will not be able to determine analytically how the error is suppressed as we did in the previous section for the Hadamard magic states. Nevertheless, we can still gain a numerical indication of how the error of $|\varphi\rangle$ states is suppressed. Lets us start with a $|\varphi\rangle$ state undergoing depolarizing noise:

$$\rho_{\mathrm{dep}} = (1-\epsilon)\,|\varphi\rangle\,\langle\varphi| + \epsilon I/3. \tag{2.60}$$

For a sufficiently small ϵ the distilled state $\rho_{\rm dep}^{\rm out}$ will also be of the above form (i.e on the depolarizing axis). We can then calculate the output error probability as follows:

$$\epsilon^{\rm out} = 1 - \langle\varphi|\,\rho_{\rm dep}^{\rm out}\,|\varphi\rangle\,. \tag{2.61}$$

In general, we expect that $\epsilon^{\rm out} \approx \epsilon^{k}$ for very small ϵ. Therefore, the power k can be evaluated as the gradient of a $\log - \log$ plot of $\epsilon^{\rm out}$ versus ϵ. From Fig. (2.5i), we see that $k \approx 1$. This indicates that the error suppression is linear, which is worse than what is observed in [27].

For completeness, we include the success probability $p_{\rm succ}$ of the syndrome measurements. Successful syndrome measurements, where all outputs of the stabilizer measurements are +1, are described by the projector Π given in Eq. (2.30). Hence, the probability of this measurement is simply:

$$p_{\rm succ} = {\rm tr}(\Pi\rho^{\otimes 5}\Pi) = {\rm tr}(\rho^{\otimes 5}\Pi), \tag{2.62}$$

which is given in Eq. (2.32) for all sets of Bloch components. We have computed $p_{\rm succ}$ for both $|\varphi\rangle$ and $|H_{\pm}\rangle$ undergoing depolarizing noise as an input states to the distillation. A plot of $p_{\rm succ}$ is given in Fig. (2.5ii).

Aside: Clifford equivalences and cycling behaviour

We have chosen a particular decoding to avoid a certain cycling behaviour. If instead, the canonical decoding was used, without the addition of correction U_c permutation, then purification would still occur, but between each iterate the output would cycle between different states. In the Hadamard plane, we would observe an oscillation between $|H_{\pm}\rangle$, which is trivial to see since $U_c\,|H_{\pm}\rangle = |H_{\mp}\rangle$. Whereas, for the $|\varphi\rangle$ state there is a more complex 4-cycle behaviour, illustrated in Fig. (2.6), such that for the distillation map for one iterate, denoted by $\mathcal{D}$, performs $\mathcal{D}(|\varphi_j\rangle) = |\varphi_{j+1}\rangle$ and $|\varphi\rangle = |\varphi_1\rangle = |\varphi_5\rangle$. The four cycling states are related to $|\phi_1\rangle$ by

$$\begin{aligned} |\varphi_2\rangle &= U_c^{\dagger}\,|\varphi_1\rangle\,, \\ |\varphi_3\rangle &= H\,|\varphi_1\rangle\,, \\ |\varphi_4\rangle &= U_c^{\dagger}U_c^{\dagger}\,|\varphi_1\rangle\,. \end{aligned} \tag{2.63}$$

However, by considering $\mathcal{D}'(\rho) = U_c\mathcal{D}(\rho)U_c^\dagger$ after each iterate, this cycling behaviour vanishes. Note also, that this cycling behaviour is not only seen for the pure states but for depolarized states, and so all of these states are distilled by the 5-qutrit code.

2.5 Promoting the Clifford group

In this section, we show how to use the two families of magic states $|H_+\rangle$ and $|\varphi\rangle$ to simulate a non-Clifford gate, thus achieving a universal set of gates. In the current form, these states cannot be injected directly, so we introduce additional sub-protocols that convert these magic states into another form of magic state we call *phase* states, that are useful for state-injection. The phase states hold only phase information with respect to the computational basis, having the form

$$|\Phi_{\theta,\phi}\rangle = \frac{1}{\sqrt{3}}\left(|0\rangle + e^{i\theta}|1\rangle + e^{i\phi}|2\rangle\right). \tag{2.64}$$

In the next chapter, we will derive a more general form of these states and show how they can be distilled directly by the quantum Reed-Muller codes.

We describe, in Sec 2.5.1, the parity-checker protocol, which is used to convert both $|H_+\rangle$ and $|\varphi\rangle$ into the *plus*-state $|\Psi^+\rangle \propto |0\rangle + |1\rangle$. These plus states are then input into the equatorialization procedure, in Sec. 2.5.2, to output the desired phase state. Finally, in Sec. 2.5.3, we show how the phase states are used to implement a non-Clifford gate.

2.5.1 The parity-checker protocol

The parity-checker protocol is a simple distillation protocol (see [137]) which is very efficient against a specific type of noise, but vulnerable against another type of noise. However, both $|H_+\rangle$ and $|\varphi\rangle$ have zero overlap with the “bad” noise term and so the protocol can be efficiently used to convert these states into a *plus* state. Before beginning the iterative protocol some manipulation of the input states $|H_+\rangle$ and $|\varphi\rangle$ is required:

1. *Preparation 1*, uniformly randomly choose from the set of unitaries $\{1, H^2\}$ and apply;

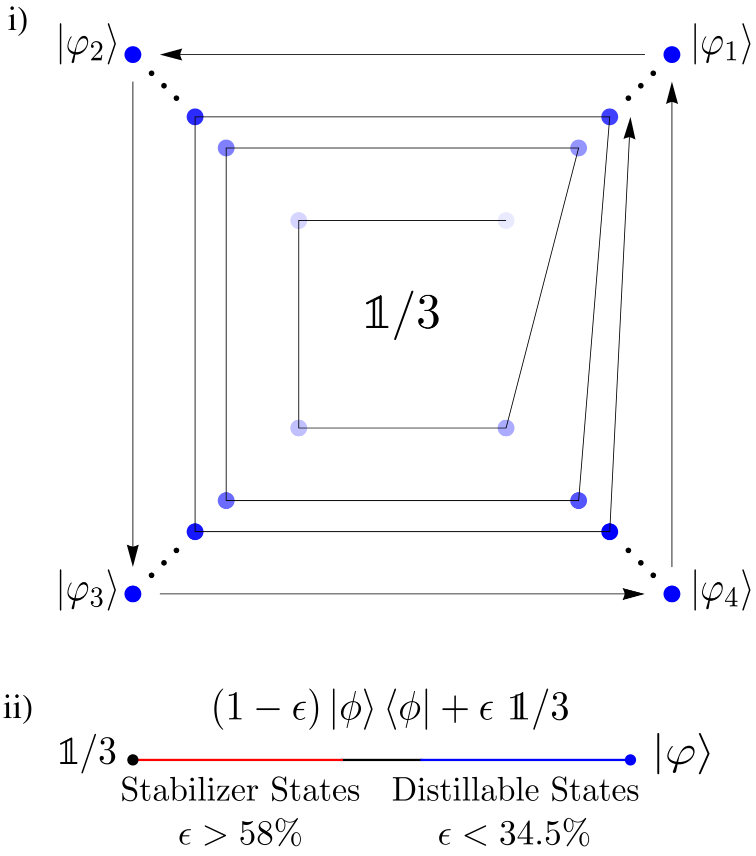

Figure 2.6: An illustrative picture of the cycling behaviour of $|\varphi\rangle$. i) Starting with a mixed state (light blue point) the protocol will increase the purity of the states while cycling between them and ultimately reaching the fix pure points (dark blue points). ii) The convex line between one of the cycling states $|\varphi\rangle$ and the completely mixed state $\mathbb{1}/3$ with an accurate ratio of the noise threshold.

2. *Preparation 2*, apply $X^\dagger$;

3. *Preparation 3*, uniformly randomly choose from the set of unitaries $\{1, S, S^2\}$and apply, where $S = |0\rangle\langle 0| + |1\rangle\langle 1| + \omega |2\rangle\langle 2|$.

This preparation procedure maps all quantum states to

$$\rho(\delta_0, \eta) = (1 - \eta_0 - \delta_0)|\Psi^+\rangle\langle\Psi^+| + \delta_0 |\Psi^-\rangle\langle\Psi^-| + \eta_0 |2\rangle\langle 2|, \tag{2.65}$$

where $|\Psi^\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$. Note that $|\Psi^-\rangle$ is Clifford equivalent to the Hadamard eigenstate $|H_i\rangle$. For imperfect $|H_+\rangle$ and $|\varphi\rangle$ states, with depolarizing noise ϵ, we have

$$\eta_0 = c(|\psi\rangle) + \delta_0, \tag{2.66}$$

$$\delta_0 = \epsilon/3, \tag{2.67}$$

where for the two magic states of interest $c(|H_+\rangle) = 0.2113$ and $c(|\varphi\rangle) = 0.0152$. The parity-checker protocol will exponentially suppress the value of η_0, whereas δ_0 will linearly increase. However, this is not problematic as δ_0 can be made arbitrarily small via distillation by the 5-qutrit protocol.

The iterative parity-checker is now fairly simple. On the $(n+1)$th round we have

1. Take two copies of $\rho(\delta_n, \eta_n)$;

2. Measure the observable $Z_1 Z_2^\dagger$ and post-select on +1;

3. Decode the state such that $|j, j\rangle \to |j\rangle$;

4. Use the output state $\rho(\delta_{n+1}, \eta_{n+1})$ as an input in the next iterate.

It is straightforward to verify the iterative relations are

$$\eta_{n+1} = \eta_n^2/p_n, \tag{2.68}$$

$$\delta_{n+1} = \delta_n(1 - \eta_n - \delta_n)/p_n, \tag{2.69}$$

where p_n is the success probability

$$p_n = (1 + \eta_n(3\eta_n - 2))/2. \tag{2.70}$$

When the small noise component is zero, so $\delta_0 = 0$, and the large noise is not too large, $\eta_0 < 1/3$, then η_n vanishes exponentially quickly such that $\eta_n \sim (2\eta_0)^n$. Allowing for non-zero δ_0, the protocol can be iterated for approximately $n \sim \log(\delta_0)$ rounds before the δ noise becomes problematic.

Let us consider a concrete example. If we have a $|\varphi\rangle$ magic state with depolarization noise $\epsilon = 10^{-8}$, this is first prepared into a noisy plus state with $\eta_0 \sim 0.0152$ and $\delta_0 = 10^{-8}/3$. The total noise, $\eta_n + \delta_n$, will decease for the first 3 rounds of parity checking. After the fourth round we have a plus state with a total error of only 2.707×10^{-8}. This illustrates that high-fidelity plus states can be prepared from high fidelity $|\varphi\rangle$ states in a small number of rounds.

2.5.2 Equatorialization

In this section, we describe a magic state protocol that converts the plus states to the desired phase states. The phase states lie on a generalisation of the qubit Bloch sphere equator, hence the term *Equatorialization*. This protocol is probabilistic but not iterative. We take two highly purified copies of a plus state, $|\Psi^+\rangle$. We measure a 2-qutrit stabilizer operator and post-select such that we project onto the subspace spanned by:

$$\begin{aligned} |0_L\rangle &\propto |0,0\rangle + \omega |1,2\rangle + \omega^2 |2,1\rangle, \\ |1_L\rangle = X_1 X_2 |0_L\rangle &\propto |1,1\rangle + \omega |2,0\rangle + \omega^2 |0,2\rangle, \\ |2_L\rangle = X_1^2 X_2^2 |0_L\rangle &\propto |2,2\rangle + \omega |0,1\rangle + \omega^2 |1,0\rangle, \end{aligned}$$

and then decode onto a single qutrit. When successful this produces the following transformation:

$$|\Psi^+\rangle^{\otimes 2} \to (|0\rangle + |1\rangle + (\omega + \omega^2)|2\rangle)/\sqrt{3}. \tag{2.71}$$

Noticing that $\omega + \omega^2 = -1$, we find that the output is a phase state required

$$|\Phi_{0,\pi}\rangle = (|0\rangle + |1\rangle - |2\rangle)/\sqrt{3}. \tag{2.72}$$

2.5.3 Non-Clifford Gate

Let us begin by considering a general phase state $|\Phi_{\theta,\phi}\rangle$. Given such a magic state, and a second qutrit state in any state $|\psi\rangle$ we can perform a gate teleportation by measuring $Z_1 Z_2^\dagger$. Given measurement outcome ω^k, we perform the decoding $|x, y\rangle \to |y + k\rangle$. The ω^k outcome applies a unitary $U_{k,\theta,\phi}$ to $|\psi\rangle$, which is diagonal in the computational basis with eigenvalues

$$\begin{aligned} U_{0,\theta,\phi} &= (1, e^{i\theta}, e^{i\phi}), \\ U_{1,\theta,\phi} &= (e^{i\theta}, 1, e^{i\theta}), \\ U_{2,\theta,\phi} &= (e^{i\phi}, e^{i\theta}, 1). \end{aligned} \tag{2.73}$$

Each unitary occurs with equal probability because the phase state contains no variation in amplitudes. For our $|\Phi_{0,\pi}\rangle$ state, the corresponding unitaries are non-Clifford and take a simple form $U_{k,0,\pi}$, for $k \in \{0, 1, 2\}$. Any such non-Clifford unitary augmented with the Clifford operations forms a universal set of gates as shown in Thm. (1).

2.6 Summary and Open Problems

In this chapter we have presented an overview of MSD in the qubit case, and we described a numerical approach that can be used to study the distillation properties of any stabilizer code by deriving the general distillation map. Our approach, however, is not efficient in the size n of the code. The reason being that in deriving the distillation map we sum over over all the stabilizer elements of the code, which is an exponential number of elements d^{n-k}. For this reason, we have only studied the small 5-qutrit code $[\![5, 1, 3]\!]_3$ as a first attempt to explore magic states distillation beyond the qubit case.

The most interesting open question in this work is whether the state $|H_i\rangle$ is distillable. This state, as we saw, is member of the family of most robust states and can tolerate the highest depolarising noise. Distilling this state can give a clear indication that qudit systems do definitely tolerate higher noise levels. But, the fact that the bound states (positive Wigner states) and the stabilizer states coincide in the direction of the $|H_i\rangle$ could perhaps suggest that there are other (unknown) no-go theorems that could imply that this mixed-state region is not distillable.

Chapter 3

Distillation with Quantum Reed-Muller Codes

In this chapter we investigate the distillation properties of the qudit quantum Reed-Muller codes. More specifically, we provide a generalisation of the $[\![15,1,3]\!]_2$ *distillation protocol by Bravyi and Kitaev [27]. This code has the interesting property of having a transversal non-Clifford gate, namely the* $\pi/8-$*phase gate. We start in Sec. 3.1 by exploiting this remarkable property in relation to magic state distillation. We introduce a generalisation of the phase gates, referred to as the magic gates* $M \in \mathcal{M}_d^m$*, and define their properties. We then state our main theorem of this chapter for the existence of a MSD protocol based on the Reed-Muller codes, which will be proven in the subsequent sections. In Sec. 3.2 we describe our generalised distillation protocol for the family of CSS codes. Then in Sec. 3.3, using the tools developed earlier, we study the distillation properties of quantum Reed-Muller codes (which are a sub-class of the CSS codes) in all prime dimensions. In Sec. 3.4 we develop different measures to study the performance of the distillation protocols and we provide concrete examples of distillation using the two small* 3*- and* 5*-dimensional Reed-Muller codes. Finally, in 3.5 we show how the magic states distilled by the Reed-Muller codes can be injected to implement a magic gate.*

3.1 Exploiting Transversality

The Reed-Muller codes are classical linear error correction codes [134, 117] that contain many families of known important codes[1] with diverse properties [103, 10, 88, 165, 45, 72]. To our knowledge, these codes were first generalised to the quantum case by Knill *et al.* [94], and was further developed by Steane [151]. Furthermore, the qudit (non-binary) generalisation was later investigated by Sarvepalli and Klappenecker in [140]. Our aim in this chapter is to generalise the MSD protocol by Bravyi and Kitaev [27] for distilling the $H-$type qubit magic states which uses the $15-$qubit Reed-Muller code $[\![15,1,3]\!]_2$. But before we introduce our generalisation, we start by exploring a very special property of this code.

As we will see below, the code $[\![15,1,3]\!]_2$ is a CSS stabilizer code. It has a remarkable property of having a transversal non-Clifford operator, which is the T–gate (or $\pi/8-$phase gate). Following the definition of a transversal operations introduced in Sec. 1.3, this means that the product operator $T^{\otimes 15} = \bar{T}$ acts on the logical basis of this code as a non-Clifford operator[2]. This property plays a central role in MSD for several reasons. Most importantly, there is a direct relationship between the $T-$gate and the $H-$type magic state $|\pi/8\rangle$, where $|\pi/8\rangle = T\,|+\rangle$. This property will be exploited by the distillation protocol (see Eq. (3.19)), and it will ultimately allow us to employ powerful techniques of classical coding theory to obtain the distillation properties of the quantum Reed-Muller codes for all prime dimensions. Moreover, recall that the T–gate is a non-Clifford unitary that belongs to the third level of the Clifford hierarchy $C_d^1(3)$, and hence it is sufficient to promote the Clifford group to quantum universality.

For our qudit generalisation we start first by generalising the T–gate to higher dimensions, and we call such gates the M–gates (a shorthand for a *magic* gate). We then show that our M–gates are indeed transversal for qudit Reed-Muller codes.

[1] Interestingly, the Reed-Muller codes contain the special family of Reed-Solomon codes [135], which has been used in the Voyager space program to transmit images and in other commercial products such as compact disks.

[2] We will prove this fact explicitly for the qudit Reed-Muller codes in Sec. 3.3.3.

3.1.1 Qudit Magic Gates

Using the above properties of the T–gate as a guideline, we demand that those properties hold for the qudit generalisation. For this purpose we define a family of qudit magic gates $M \in \mathcal{M}_d^m$ and state the general conditions that such gates must satisfy. Note that the parameter m is directly related to formal definition of quantum Reed-Muller codes, which will be explained thoroughly in due time.

Definition 6. *The set of gates $\mathcal{M}_d^m$ contains all M such that:*

1. *M is diagonal in the computational basis $\{|0\rangle, \ldots, |d-1\rangle\}$;*
2. *$M \in SU(d)$;*
3. *$M^{d^m} = \mathbb{1}$;*
4. *$M \in \mathcal{C}_d^1(3)/\mathcal{C}_d^1(2)$;*

Conditions $1-2$ require that the qudit M–gate to be a diagonal *phase-type* unitary gate similar to the qubit T–gate. In addition, condition 3 is directly related to the transversality of the M–gate for our quantum Reed-Muller codes. Furthermore, if we express the eigenvalues of M as $\exp(i2\lambda_j\pi/d^m)$ then condition 3 entails that λ_j are integers and condition 2 is satisfied when $\sum_j \lambda_j = 0$. Finally, condition 4 requires that while M is a member of the third level of the Clifford hierarchy, it is not a member of the Clifford group itself. From this we conclude that the operator

$$C_M = MX_dM^\dagger, \tag{3.1}$$

must be a non-Pauli Clifford operator. The eigenstates of C_M will be the attractor of our distillation protocols[3], which is why it is essential that C_M is a non-Pauli operator.

For every set $\mathcal{M}_d^m$ we will design protocols that distil eigenstates of C_M. However, we need to know that such gates exist. In the qubit setting, the T–gate provides such a unitary for $m = 4$. However, for $m < 4$ it is easy to check that all qubit gates with the form required by conditions (1-3) of the above

[3] This is in direct analogy to the qubit case where the equatorial H–type magic state $|\pi/8\rangle = (|0\rangle + e^{\pi i/4}|1\rangle)/\sqrt{2}$ is an eigenstate of $TXT^\dagger$.

definition are Clifford unitaries and so fail condition 4. Remarkably, for all odd prime dimensions $d \geq 3$ we can find such gates for $m = 2$, and when $d \geq 5$ these gates exist for $m = 1$. Using tall brackets to denote binomial coefficients we have the following theorem.

Theorem 3. *For all odd primes d, there exists a gate M such that*

1. *for $d = 3$ we have $M \in \mathcal{M}_d^m$ for all $m \geq 2$;*
2. *for $d \geq 5$ we have $M \in \mathcal{M}_d^m$ for all $m \geq 1$.*

One such gate is the following

$$M = \sum_j \exp(i2\lambda_j \pi / d^m) \left|j\right\rangle \left\langle j\right|, \tag{3.2}$$

with

$$\lambda_j = d^{m-2} \left[d\binom{j}{3} - j\binom{d}{3} + \binom{d+1}{4} \right]. \tag{3.3}$$

We refer to this M as the canonical $\mathcal{M}_d$ gate.

In particular, the canonical $\mathcal{M}_d$ gate is associated with the non-Pauli Clifford unitary

$$C_M = MXM^\dagger \propto XP, \tag{3.4}$$

where P is the Clifford gate introduced earlier in Eq. (1.14). Clearly, a different M exists for every dimension d. Solving for M requires only basic algebra, as shown next.

Proof. Here we verify the assertions of Thm. 3 and show that the canonical M is a member of $\mathcal{M}_d^m$ for the asserted values of d and m. We begin by showing that

$$C_M = MXM^\dagger \propto XP. \tag{3.5}$$

Left multiplying by $X^\dagger$ gives $X^\dagger MXM^\dagger \propto P$. The left hand side is then

$$X^\dagger MXM^\dagger = \sum_j \exp(i2\pi(\lambda_{j\oplus 1} - \lambda_j)/d^m) \left|j\right\rangle \left\langle j\right|. \tag{3.6}$$

This equals P, up to a global phase, if for all $0 \leq j \leq d-1$,

$$\lambda_{j\oplus 1} - \lambda_j = d^{m-1}\binom{j}{2} + c, \tag{3.7}$$

for some c. We first solve for the cases where $j \oplus 1 = j + 1$, that is $j \neq d - 1$. For this set of equations, we may use standard arithmetic and recurrence equation methods, and the general solution is

$$\lambda_j = d^{m-1}\binom{j}{3} + jc + \lambda_0, \tag{3.8}$$

for all j, where c and λ_0 are integers to be determined. These integer variables will be fixed by demanding that Eq. (3.7) with $j = d - 1$ holds, and also that $\sum_j \lambda_j = 0$. First, let us impose the former condition and substitute Eq. (3.8) into Eq. (3.7) for $j = d - 1$, to yield

$$\begin{aligned} \lambda_0 - \lambda_{d-1} &= \lambda_0 - \left[d^{m-1}\binom{d-1}{3} + j(d-1)c + \lambda_0 \right], \\ &= d^{m-1}\binom{j-1}{2} + c. \end{aligned}$$

Solving this equation for c yields

$$c = -d^{m-2}\binom{d}{3}. \tag{3.9}$$

For $m \geq 2$, inspection reveals that c is integer-valued for all d. For $m = 1$, c is integer-valued for all prime $d \geq 5$. This follows from the fact that when $m = 1$, $c = -(d-1)(d-2)/6$. We use the fact that $6 = 3 \times 2$. Since $d \geq 5$ is a prime number not equal to three, then either $(d-1)$ or $(d-2)$ must be divisible by three. Since $d \geq 5$ is a prime number not equal to two then $(d-1)$ must be divisible by 2. Hence the product $(d-1)(d-2)$ is divisible by 6 for all primes $d \geq 5$, and c is an integer for $m = 1$ and $d \geq 5$.

It remains to fix λ_0 by imposing that $\sum_j \lambda_j = 0$. Performing the summation and simplifying, we find that

$$\lambda_0 = d^{m-2}\binom{d+1}{4}. \tag{3.10}$$

Again, for $m \geq 2$ this is (by inspection) integer-valued for all d. For $m = 1$, this is integer-valued for all prime $d \geq 5$, and the proof for this latter case is similar to above. When $m = 1$, $\lambda_0 = (d+1)(d-1)(d-2)/24$. We observe that $24 = 3 \times 2 \times 4$. Since $d \geq 5$ is a prime number not equal to three, then either $(d-1)$ or $(d-2)$ must be divisible by three. Since d is an odd prime number both $(d+1)$ and $(d-1)$ must be divisible by two, and one of this pair must be divisible by 4. Hence $(d+1)(d-1)(d-2)$ must be divisible by 24 and consequently λ_0 is an integer for $m = 1$ and $d \geq 5$.

Thus, the gate M as defined in theorem 1, satisfies all the requirements to be a member of $\mathcal{M}_d^m$. For $m = 1$ and $d = 3$, λ_j, is not integer-valued for all values of j and so the above argument does not provide a member of $\mathcal{M}_3^1$. Indeed, for $d = 3$ it is easy to numerically search the sets of gates with integer λ_j and verify that none are non-Clifford and so $\mathcal{M}_3^1$ is empty.

□

Finally, in parallel work to us, a comprehensive classification of families of gates that generalise the T–gates to higher dimension has been derived by Howard and Vala [76], using tools from symplectic geometry. We also remark that these gates, for $d = 3, 5$, are Clifford equivalent to those found in Ref. [158] to be the most robust to depolarizing noise before becoming stabilizer operations.

3.1.2 Existence of a MSD Protocol

The eigenstates of C_M are non-stabilizer states, which we label $|M_k\rangle$. In direct analogy to the qubit case, we note that $|M_k\rangle = M\,|+_k\rangle$, where $|+_k\rangle$ is an eigenstate of X with eigenvalue ω^k. Here, we aim to use magic state distillation to purify $|M_0\rangle$ states from n copies of input noisy states[4] ρ_{res}. For generality, we will drop the label 'res', and consider the input state to the protocol in the most general form denoted as ρ. To quantify the general noise (as opposed to fidelity) of the state ρ with respect to the magic state $|M_0\rangle$, we will use the following form of the error probability

$$\epsilon = 1 - \langle M_0|\,\rho\,|M_0\rangle\,. \tag{3.11}$$

After distilling a magic state with a sufficient output probability ϵ^{out}, they will be used to for fault-tolerant state-injection of the magic unitary M. Our aim here will be to show that the magic states $|M_0\rangle$ can be distilled by using a higher-dimensional quantum Reed-Muller code, which will brings us the main theorem of this chapter:

Theorem 4. *Consider any $M \in \mathcal{M}_d^m$ for any odd prime* d *and any integer $m \geq 2$, or any odd prime $d \geq 5$ and $m \geq 1$. There exists a quantum Reed-Muller code $\mathcal{QRM}_d(m)$, such that $[\![n = d^m - 1, 1, 2]\!]_d$ that*

[4] Recall from the last chapter that the resource state ρ_{res} cannot be from the region of mixed states with positive Wigner representation (which also includes the stabilizer region) due to a theorem in [159].

iteratively distils the magic state $|M_0\rangle$. The code takes n copies of a qudit state ρ with error probability

$$\epsilon = 1 - \langle M_0| \rho |M_0\rangle . \tag{3.12}$$

and with non-zero probability the protocol outputs a state ρ^{out} such that

$$\epsilon^{\text{out}} = 1 - \langle M_0^\dagger| \rho^{\text{out}} |M_0^\dagger\rangle . \tag{3.13}$$

Moreover, there exists a constant $K > 0$ such that for all ϵ we have $\epsilon^{\text{out}} \leq K\epsilon^2$. Consequently, there exists a threshold $\epsilon^* > 0$ such that if $0 < \epsilon < \epsilon^*$ then $\epsilon^{\text{out}} < \epsilon$.

All the assertions of the above theorem will be proved in the sections that follows. However, there are two important observations to make. First, notice that after a single iteration, using as input noisy $|M_0\rangle$ states, the protocol will output a noisy $|M_0^\dagger\rangle$ state. This is due to the cycling phenomenon that we saw in the last chapter, which as we will see, it can be prevented by some Clifford unitary correction. Second, the rate of error suppression is always quadratic, and so these results give the first better than linear error reductions in higher-dimensional systems that we saw in the last chapter.

Moreover, it is important to point out that the Clifford unitary C_M plays two central roles in our distillation protocol. First, it is used as part of the *Clifford correction*, which significantly increases the success probability of the protocol. Secondly, it is used for twirling the general input state canonical form into the $|M_k\rangle$ basis. This is similar to the twirling process in the $H-$plane we saw in the chapter. Here, we will refer to the twirling process as *C_M-twirling*. It consists of choosing a random integer $k \in 1, \dots, d$ and applying the gate C_M^k to the input states ρ, which will convert it into the canonical form that depends only on $d-1$ independent parameters, such that

$$\frac{1}{d} \sum_{k \in \mathbb{Z}_d} C_M^k \rho (C_M^k)^\dagger = \sum_k f_k |M_k\rangle \langle M_k| . \tag{3.14}$$

For the distillation of the $|M_0\rangle$ state the protocol will have to increase the value of f_0.

3.2 Distillation using CSS Codes

As has been stated previously, the quantum Reed-Muller codes $\mathcal{QRM}_d(m)$ that we will construct are a subclass of the CSS codes. Here, we will use the definitions and notations developed in Sec. (1.2.2) to define our distillation protocol for the general CSS construction. Recall that a CSS code is a stabilizer code such that $\Pi = \Pi_{\mathcal{S}_X}\Pi_{\mathcal{S}_Z}$. Our only main demand so far is the existence of an $M \in \mathcal{M}_d^m$ gate that has to obey the transversality property. For this purpose, we define an $\mathcal{M}_d^m$–distillation code as follows.

Definition 7. *An n-qudit stabilizer code, Π, is an $\mathcal{M}_d^m$–distillation code if all of the following hold*

1. *all $M \in \mathcal{M}_d^m$ are transversal such that $M^{\otimes n}\Pi(M^{\otimes n})^\dagger = \bar{M}^\dagger \Pi \bar{M}$;*
2. *it has distance, $D \geq 2$;*
3. *it has logical Pauli operators $\bar{X} = X[\mathbf{1}]$ and $\bar{Z} = Z[-\mathbf{1}]$.*

We have introduced the vector shorthand $\mathbf{1} = (1, 1, \ldots, 1)$. Notice that we require a special kind of transversality, such that the logical operator, $\bar{M}^\dagger$, is implemented by applying $M^{\otimes n}$. The need for complex transposition will be explained later, and will be seen to result in a cycling phenomenon in the distillation protocol.

Here we show that all $\mathcal{M}_d^m$–distillation codes can be used to perform distillation for magic states of the form $|M_0\rangle = M|+_0\rangle$ for all $M \in \mathcal{M}_d^m$. Due to cycling, after a single iteration using as input noisy $|M_0\rangle$ states, the protocol will output a noisy $|M_0^\dagger\rangle$ state. Later we show the existence of the required codes with $D = 2$, which will then entail Thm. 4. For now we show how to proceed given such a code.

3.2.1 The Distillation Protocol

Now we state our distillation protocol given that an n-qudit $\mathcal{M}_d^m$–distillation code exits. For any $M \in \mathcal{M}_d^m$, we have the following iterative protocol:

1. Take n copies of the state ρ and C_M–twirl;
2. Measure generators of the phase stabilizer $\mathcal{S}_Z$;
3. Accept all outcomes, but perform a Clifford correction operator $C_M[\mathbf{w}]$ for every syndrome vector $\boldsymbol{w}$.;
4. Measure generators of the bit-flip stabilizer $\mathcal{S}_X$;

5. Post-select on all '+1' measurement outcomes;

6. Decode the encoded qudit to a single qudit output state ρ^{out};

7. Use ρ^{out} as input in the next iteration.

Step 1 is simply the initialization step of the distillation protocol which prepares the product state $\rho^{\otimes n}$ in the canonical form. Steps 2 – 5 correspond to the stabilizer measurement of the stabilizer generators of the code. We have not yet defined the Clifford correction $C_M[\mathbf{w}]$, this step simply increases the success probability of the $\mathcal{S}_Z$ measurement, as will be shown in Sec. 3.2.3. For now, we will assume that step 2 generates all '+1' measurement outcomes, for which $C_M[\mathbf{w}] = \mathbb{1}$, i.e. the $\Pi_{\mathcal{S}_Z}$ projection is deterministic. The remaining step 6 is a simple decoding map that returns a single purified qudit. Next, we describe all these steps in more detail.

After C_M–twirling the n copies have the form

$$\rho^{\otimes n} = \sum_{\mathbf{v}\in\mathbb{Z}_d^n} \alpha_{\mathbf{v}} \left|M_{\mathbf{v}}\right\rangle \left\langle M_{\mathbf{v}}\right|, \tag{3.15}$$

where

$$\left|M_{\mathbf{v}}\right\rangle = \left|M_{v_1}\right\rangle \left|M_{v_2}\right\rangle \dots \left|M_{v_n}\right\rangle, \tag{3.16}$$

and

$$\alpha_{\mathbf{v}} = \prod_{k\in\mathbb{Z}_d} f_k^{\text{wt}_k(\mathbf{v})}, \tag{3.17}$$

where $\text{wt}_k(\mathbf{v})$ is the k-weight, the number of elements in $\mathbf{v}$ equal to k, and $f_k = \langle M_k|\rho|M_k\rangle$. We now exploit the transversality property, note that

$$\rho^{\otimes n} = \bar{M}^{\dagger}\left(\sum_{\mathbf{v}\in\mathbb{Z}_d^n} \alpha_{\mathbf{v}} \left|+_{\mathbf{v}}\right\rangle \left\langle +_{\mathbf{v}}\right|\right)\bar{M}, \tag{3.18}$$

where $\bar{M}^{\dagger} = M^{\otimes n}$. Upon a successful projection onto the code subspace, we have

$$\Pi\rho^{\otimes n}\Pi = \bar{M}^{\dagger}\left(\sum_{\mathbf{v}\in\mathbb{Z}_d^n} \alpha_{\mathbf{v}}\Pi \left|+_{\mathbf{v}}\right\rangle \left\langle +_{\mathbf{v}}\right|\Pi\right)\bar{M}, \tag{3.19}$$

as the projector commutes with $\bar{M}$. All is left to do is to determine the effect of each term $\Pi\left|+_{\mathbf{v}}\right\rangle$, which we will find to be

$$\begin{aligned} \Pi\left|+_{\mathbf{v}}\right\rangle &= 0; \forall \mathbf{v} \notin \mathcal{L}_X^{\perp}; & (3.20)\\ \Pi\left|+_{\mathbf{v}}\right\rangle &= \sqrt{c}\left|+_j\right\rangle_L; \forall \mathbf{v} \oplus j\mathbf{1} = \mathbf{w}, \text{s.t.} \mathbf{w} \in \mathcal{L}_Z. & (3.21)\end{aligned}$$

It is not hard to see that the first equation covers all $\mathbf{v} \notin \mathcal{L}_X^{\perp}$ and the second equation covers all $\mathbf{v} \in \operatorname{span}(\mathcal{L}_Z, \mathbf{1})$. Using the known relation that $\mathcal{L}_X^{\perp} = \operatorname{span}(\mathcal{L}_Z, \mathbf{1})$, we conclude that these equations account for all possible $\mathbf{v}$. Note that the constant c gives the probability of this projection when the initial state is pure, i.e.

$$c = \operatorname{tr}(\Pi\left|+_0\right\rangle\left\langle+_0\right|^{\otimes n}). \tag{3.22}$$

Furthermore, $\left|+\right\rangle^{\otimes n}$ is an eigenstate of $\Pi_{\mathcal{S}_X}$ and so this randomness can be completely attributed to the Z stabilizer measurements, which can be made deterministic by Clifford correction. Now we present the reasoning that leads to these two equations, namely Eqs. (3.20,3.21). We divide the action of the projector into three distinct cases of errors: a detected error, no error and an undetected error.

When $\mathbf{v} \notin \mathcal{L}_X^{\perp}$, an error is present that is detected by the code and so the state vanishes, $\Pi\left|+_{\mathbf{v}}\right\rangle = 0$. To see this we recall that $X\left|+_k\right\rangle = \omega^k\left|+_k\right\rangle$ and so more generally $X[\mathbf{u}]\left|+_{\mathbf{v}}\right\rangle = \omega^{\langle\mathbf{v},\mathbf{u}\rangle}\left|+_{\mathbf{v}}\right\rangle$. Projecting onto the '+1' eigenspace of all $X[\mathbf{u}] \in \mathcal{S}_X$ entails that the state will vanish unless $\langle\mathbf{v},\mathbf{u}\rangle = 0$ for all $\mathbf{u} \in \mathcal{L}_X$. This is simply the requirement that $\mathbf{v}$ is in the dual of $\mathcal{L}_X$, which proves Eq. (3.20).

For the "no error" instances, $\mathbf{v} \in \mathcal{L}_Z$, the state does not vanish under projection. Furthermore, since $\left|+_{\mathbf{v}}\right\rangle = Z[\mathbf{v}]\left|+\right\rangle^{\otimes n}$ and $\Pi Z[\mathbf{v}] = \Pi$ we have $\Pi\left|+_{\mathbf{v}}\right\rangle = \Pi\left|+\right\rangle^{\otimes n}$ and so all such states must be projected onto the same logical state. Finally, we observe that $\left|+\right\rangle^{\otimes n}$ is stabilized by $X_L = X^{\otimes n}$ and so $\Pi\left|+\right\rangle^{\otimes n} = \sqrt{c}\left|+_0\right\rangle_L$.

All other possibilities correspond to undetected errors, resulting in a projection onto other logical states. In such cases, $\mathbf{v} \in \mathcal{L}_X^{\perp}$ and so there must exist a $j \in \mathbb{Z}_d$ such that $\mathbf{w} = \mathbf{v} \oplus j\mathbf{1} \in \mathcal{L}_Z$. In terms of Pauli operators, we have $Z[\mathbf{w}] = Z[\mathbf{v}]Z[j\mathbf{1}]$ and so $Z[\mathbf{v}] = Z[\mathbf{w}]Z[-\mathbf{1}]$. Since the logical operator is $Z_L = Z[-\mathbf{1}]$ it follows that $Z[\mathbf{v}] = Z[\mathbf{w}]Z_L^j$. In terms of the quantum state, we have $\left|+_{\mathbf{v}}\right\rangle = Z[\mathbf{w}]Z_L^j\left|+\right\rangle^{\otimes n}$ and so after projection $\Pi\left|+_{\mathbf{v}}\right\rangle = \sqrt{c}Z_L^j\left|+_0\right\rangle_L = \sqrt{c}\left|+_j\right\rangle_L$.

Summary and Distillation Map

In summary, the transversality of M in Eq. (3.19) allowed us to consider the distillation of magic states $|M_0\rangle$ as equivalent to the simpler problem of distillation in the X basis. Substituting Eqs. (3.20,3.21), we get

$$\Pi\rho^{\otimes n}\Pi = c\bar{M}^\dagger \left(\sum_{j\in\mathbb{Z}_d} \sum_{\mathbf{v}\oplus j\mathbf{1}\in\mathcal{L}_Z} \alpha_{\mathbf{v}} |+_j\rangle_L \langle +_j|_L \right) \bar{M}. \tag{3.23}$$

Again, due to the cycling behaviour, that output state is diagonal in the basis $\bar{M}^\dagger |+_j\rangle_L$ rather than the desired $\bar{M}|+_j\rangle_L$. Decoding onto a single qudit we have

$$\rho^{\text{out}} \propto c \sum_{j\in\mathbb{Z}_d} \sum_{\mathbf{v}\oplus j\mathbf{1}\in\mathcal{L}_Z} \alpha_{\mathbf{v}} |M_j^\dagger\rangle \langle M_j^\dagger| . \tag{3.24}$$

By expanding out $\alpha_{\mathbf{v}}$, we get an iterative formula for $f_k^{\text{out}} = \langle M_k| \rho^{\text{out}} |M_k\rangle$, such that

$$f_j^{\text{out}} = \frac{\sum_{\mathbf{v}\oplus j\mathbf{1}\in\mathcal{L}_Z} \prod_{k\in\mathbb{Z}_d} f_k^{\text{wt}_k(\mathbf{v})}}{P_{\text{succ}}}, \tag{3.25}$$

which has been renormalized by dividing through by the success probability P_{succ}. This probability equals the sum of the numerators, which is

$$P_{\text{succ}} = \sum_{j\in\mathbb{Z}_d} \sum_{\mathbf{v}\oplus j\mathbf{1}\in\mathcal{L}_Z} \prod_{k\in\mathbb{Z}_d} f_k^{\text{wt}_k(\mathbf{v})}. \tag{3.26}$$

The summation over all j, such that $\mathbf{v}\oplus j\mathbf{1} \in \mathcal{L}_Z$, is equivalent to a sum over all $\mathbf{v} \in \text{span}(\mathcal{L}_Z, -\mathbf{1})$. Using the properties of the CSS codes we know $\text{span}(\mathcal{L}_Z, -\mathbf{1}) = \mathcal{L}_X^\perp$ and so

$$P_{\text{succ}} = \sum_{\mathbf{v}\in\mathcal{L}_X^\perp} \prod_{k\in\mathbb{Z}_d} f_k^{\text{wt}_k(\mathbf{v})}. \tag{3.27}$$

Notice that we have dropped a factor of c from the success probability, which will be justified later by Clifford correction. Finally, both numerator and denominator of f_j^{out} can be calculated from the classical codes $\mathcal{L}_X$ and $\mathcal{L}_Z$ only, and will result in polynomials of degree n.

3.2.2 Analyzing the Iterative Formulae

Here we will study the behaviour of the above iterative formulae under a simple depolarizing noise model and give a Taylor series approximation. This will prove to be useful when we later study the performance of the distillation protocols. When the noise is depolarizing, it means that $f_{j\neq 0} = \epsilon/(d-1)$ and $f_0 = 1-\epsilon$. The formula for the fidelity simplifies to

$$f_0^{\text{out}} = \frac{\sum_{\mathbf{v}\in\mathcal{L}_Z} f_0^{n-|\mathbf{v}|_H} f_{j\neq 0}^{|\mathbf{v}|_H}}{\sum_{\mathbf{v}\in\mathcal{L}_X^\perp} f_0^{n-|\mathbf{v}|_H} f_{j\neq 0}^{|\mathbf{v}|_H}}, \tag{3.28}$$

where $|\dots|_H$ is the Hamming weight. The factors f_0^n appear on both numerator and denominator and so cancel. Making use of the shorthand

$$\mu = \frac{f_{j\neq 0}}{f_0} = \frac{\epsilon}{(d-1)(1-\epsilon)}, \tag{3.29}$$

we can further simplify the fidelity formula to

$$f_0^{\text{out}} = \frac{\sum_{\mathbf{v}\in\mathcal{L}_Z} \mu^{|\mathbf{v}|_H}}{\sum_{\mathbf{v}\in\mathcal{L}_X^\perp} \mu^{|\mathbf{v}|_H}}. \tag{3.30}$$

The above equation depends on only a single parameter and the simple Hamming weights. We will show later, in Sec. 3.3.4, that this simple form can be further simplified by employing some powerful techniques from classical coding theory.

Now, we make the following observation in regard to the quadratic error suppression asserted by Thm. 4. Taylor-expanding the numerator and denominator to second order we have

$$f_0^{\text{out}} \sim \frac{1 + a\mu^D + O(\mu^{D+1})}{1 + b\mu^D + O(\mu^{D+1})}, \tag{3.31}$$

where a (b) is the number of weight d elements of $\mathcal{L}_Z$ ($\mathcal{L}_X^\perp$). Notice that both $\mathcal{L}_Z$ and $\mathcal{L}_X^\perp$ must contain a single weight zero element, $\mathbf{v} = \mathbf{0} = (0, 0, \dots, 0)$. Also, by definition, both classical codes contain no other elements with weights smaller than D. By further approximating the denominator and using $f_0^{\text{out}} = 1 - \epsilon^{\text{out}}$ gives

$$\epsilon^{\text{out}} \sim (b-a)\mu^D + O(\mu^{D+1}). \tag{3.32}$$

So the suppression of errors is degree D as $\mu \sim \epsilon$. In particular, since we know that $D \geq 2$ (if fact, $D = 2$ for our protocols) this implies that the error suppression is at least quadratic, as claimed.

Finally, we remark that the above approximation can be generalised further (omitted here) to demonstrate error suppression and existence of a threshold for all possible noise models.

3.2.3 Clifford Correction

So far we have assumed that the Z stabilizer measurements all yield the desired '+1' outcome. Next we consider the process of Clifford correction, as outlined by step 3 of our protocol. This additional strategy significantly increases the success probability of each round, so much so that success is guaranteed in the limit of pure initial states. The general idea is that for any measurement outcomes, with resulting projector $\Pi'_{\mathcal{S}_Z}$, there exists a Clifford $C_M[\mathbf{w}]$ such that $C_M[\mathbf{w}]\Pi'_{\mathcal{S}_Z} = \Pi_{\mathcal{S}_Z} C_M[\mathbf{w}]$. The key fact exploited is that for a single qudit $C_M Z = \omega^{-1} Z C_M$, and so for many qudits $C_M[\mathbf{w}]Z[\mathbf{v}] = \omega^{-\langle \mathbf{w},\mathbf{v}\rangle} Z[\mathbf{v}]C_M[\mathbf{w}]$. To proceed we must specify the projector $\Pi'_{\mathcal{S}_Z}$. We begin by expressing the linear code as $\mathcal{L}_Z = \{G\mathbf{u} : \mathbf{u} \in \mathbb{Z}_d^m\}$ where $m = \mathrm{Dim}(\mathcal{L}_Z)$ and G is an m by n matrix called the generator matrix of $\mathcal{L}_Z$. Each column of G gives an individual generator of $\mathcal{L}_Z$ and hence $\mathcal{S}_Z$. When the measurement corresponding to the $j\mathrm{th}$ generator gives outcome ω^{k_j}, the resulting projection is

$$\Pi'_{\mathcal{S}_Z} = \frac{1}{2^m} \sum_{\mathbf{u} \in \mathbb{Z}_d^m} \omega^{\langle \mathbf{k},\mathbf{u}\rangle} Z[G\mathbf{u}]. \tag{3.33}$$

Conjugating with a Clifford correction $C_M[\mathbf{w}]$ yields

$$C_M[\mathbf{w}]\Pi'_{\mathcal{S}_Z} = \frac{1}{2^m} \sum_{\mathbf{u} \in \mathbb{Z}_d^m} \omega^{\langle \mathbf{k},\mathbf{u}\rangle - \langle \mathbf{w}, G\mathbf{u}\rangle} Z[G\mathbf{u}]C[\mathbf{w}], \tag{3.34}$$

and so the correction works when for all $\mathbf{u}$ we have $\langle \mathbf{k},\mathbf{u}\rangle = \langle \mathbf{w}, G\mathbf{u}\rangle$ mod d. We can always choose a canonical form for the generator matrix, such that $G = (\mathbb{1}_m|G')$, where the identity acts on the first m rows of G and G' labels the remainder of the matrix. For such a canonical generator matrix we choose $\mathbf{w}$ to equal $\mathbf{w} = (k_1, k_2, \ldots, k_m, 0, 0, \ldots, 0)$ so it matches the measurement outcomes on the first m entries. This yields $\langle \mathbf{w}, G\mathbf{u}\rangle = \langle \mathbf{k},\mathbf{u}\rangle$ and so Clifford correction achieves its goal.

3.3 Reed-Muller Codes

Discussing quantum Reed-Muller codes is a challenging task since there many different definitions of these codes in the literature [94, 151, 171, 101]. One reason for this is because the classical Reed-Muller codes—from which the quantum Reed-Muller codes are constructed—contain a lot of symmetry, and different investigations tend to exploit such symmetries in different ways, which results in different definitions. In our work here we start by introducing the classical Reed-Muller codes in their general form, and then following Knill *et. al.* [94] we modify these codes to obtain the *shortened* classical Reed-Muller codes. Next, we show how these modified codes can be used to construct the quantum Reed-Muller codes $\mathcal{QRM}_d(m)$ with the promised properties suitable for the above distillation protocol.

3.3.1 Classical Reed-Muller Codes

We start by reviewing the non-binary, or d-ary, generalisations of Reed-Muller codes and define the necessary properties needed for later sections. A d-ary classical Reed-Muller code, denoted by $\mathcal{RM}_d(u, m)$, is conventionally defined by two parameters u and m [88, 165, 45, 72]. As before, the dimension d tells us the relevant field $\mathbb{Z}_d$ the code is defined over. The parameter u determines the *order* of the code, and m determines the *size* of the code (i.e. number of codewords). Associated with all Reed-Muller codes are polynomials with degree of order u, which *uniquely* define the code[5]. In this work, we are interested in degree 1 polynomials, i.e. linear functions, and hence $u = 1$ is assumed throughout. The dual of a Reed-Muller is another Reed-Muller code, though it may have a different order. Although this means that Reed-Muller codes of higher order will be involved in our work, we will only need to define them in terms of duality. For our quantum Reed-Muller code construction we will not use the codes we define in this section but a shortened version we introduce in the next section. Nevertheless, it proves useful to start our construction by reviewing classical Reed-Muller codes in their more general form.

Reed-Muller codes are defined in terms of linear maps between finite fields. For every field $\mathbb{Z}_d^m$, there are d^m linear maps g from $\mathbb{Z}_d^m$ onto $\mathbb{Z}_d$. All such maps can be labelled by vectors $\mathbf{u}$ themselves,

[5] For readers that wish to understand the relationship between the polynomial's order and the structure of a finite field, we recommend [103].

such that $g_{\mathbf{u}} : \mathbb{Z}_d^m \to \mathbb{Z}_d$ where $\mathbf{u} \in \mathbb{Z}_d^m$, and then the function will evaluate to $g_{\mathbf{u}}(\mathbf{a}) = \langle \mathbf{u}, \mathbf{a} \rangle = \oplus_j u_j a_j$. Next, we consider another mapping, $U_d^m : \mathbb{Z}_d^m \to \mathbb{Z}_d^n$, where $n = d^m$, such that

$$U_d^m(\mathbf{u}) = (\langle \mathbf{u}, \mathbf{a}_0 \rangle, \langle \mathbf{u}, \mathbf{a}_1 \rangle, \dots, \langle \mathbf{u}, \mathbf{a}_{n-1} \rangle), \tag{3.35}$$

where $\mathbf{a}_j$ is the base-d representation of the natural number j. For example, with $d = 3$ and $m = 2$ we would have the *ordered* set

$$\{\mathbf{a}_j\} = \{(0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1), (2,2)\}. \tag{3.36}$$

Hence for $\mathbf{u} = (0,1)$ we have

$$U_3^2(\mathbf{u}) = U_3^2(0,1) = (0,1,2,0,1,2,0,1,2). \tag{3.37}$$

For any positive integers d and m, the set $\mathcal{L} = \{\bar{\mathbf{u}} = U_d^m(\mathbf{u}); \mathbf{u} \in \mathbb{Z}_d^m\}$ is a linear vector space[6]. The codes of interest are constructed by considering all affine functions, which are linear maps plus an additional constant c such that they map $\mathbf{u}$ to $U_d^m(\mathbf{u}) \oplus c\mathbf{1}$, this brings us to our formal definition of the classical $\mathcal{RM}_d(1,m)$ before being shortened:

Definition 8. *Unshortened Reed-Muller codes, $\mathcal{RM}_d(1,m)$, are classical linear codes on $\mathbb{Z}_d^n$, where $n = d^m$, of dimension $m+1$. They are the set of codewords $\mathcal{RM}_d(1,m) = \{U_d^m(\mathbf{u}) \oplus c\mathbf{1} : \mathbf{u} \in \mathbb{Z}_d^m, c \in \mathbb{Z}_d\}$ defined in terms of affine functions.*

Before we explore the properties of these codes further we need to introduce one more definition:

Definition 9. *We say a function $\Lambda : \mathbb{Z}_d^n \to \mathbb{Z}$ is a λ-function if there exists a set of d integers $\{\lambda_0, \dots, \lambda_{d-1}\}$ such that $\sum_{j \in \mathbb{Z}_d} \lambda_j = 0$ and*

$$\Lambda(\mathbf{v}) = \sum_{j=1}^{n} \lambda_{v_j}. \tag{3.38}$$

This definition is closely related to the non-Clifford gates introduced in Def. (6), and the relationship between these two definitions will become clear soon. Our main observation here is the following.

[6] Recall that a vector space has to satisfy the closure property of under addition. In our definition, this follows directly from the closure under addition of homogeneous (same order) linear maps.

Lemma 1. *Given a λ-function Λ and an unshortened code $\mathcal{RM}_d(1,m)$ all $\mathbf{v} \in \mathcal{RM}_d(1,m)$ satisfy $\Lambda(\mathbf{v}) = 0$ mod d^m.*

Proof. To prove the lemma we first consider codewords where $\bar{\mathbf{u}} = \mathbf{0}$, and so $\mathbf{v} = (c, c, c, \ldots, c)$, then

$$\Lambda(\mathbf{v}) = d^m \lambda_c, \tag{3.39}$$

which vanishes modulo d^m. Let us now consider the codeword for the unit vector, $\bar{\mathbf{u}} = (1, 0, 0, \ldots, 0)$, and $c = 0$. The corresponding codeword has a repetitive structure as in Eq. (3.37), where each element of $\mathbb{Z}_d$ appears d^{m-1} times. Hence,

$$\Lambda(\mathbf{v}) \quad = \quad d^{m-1} \sum_{j=0}^{d-1} \lambda_j = 0, \tag{3.40}$$

since we required in our definition of a λ-function that $\sum_{j=0}^{d-1} \lambda_j = 0$. The above argument looks tailored to codewords for a unit vector $\bar{\mathbf{u}}$, but a similar argument holds for all codewords with non-trivial $\bar{\mathbf{u}}$. That is, for any non-trivial vector $\bar{\mathbf{u}}$, there are d^{m-1} *different* linear maps that evaluate to each possible output. To see this, consider that the family of linear maps is invariant under change of variables that preserve linearity. Hence, the family of functions can always be expressed in a basis such that $\bar{\mathbf{u}}$ *is* a unit vector. Furthermore, these codewords have uniform multiplicity of every value $\mathbb{Z}_d$, and so adding $c\mathbf{1}$ will only reorder the elements and not the multiplicity with which they appear. This proves our lemma. □

In summary, unshortened Reed-Muller codes have a huge amount of symmetry that they inherit from the families of affine and linear maps. However, they actually have too much symmetry for our purposes. We break just enough of that symmetry by shortening the code.

3.3.2 Shortened Classical Reed-Muller Codes

Given a code $\mathcal{L}$ over $\mathbb{Z}_d^n$, the corresponding shortened code, denoted $\mathcal{L}^*$, is over $\mathbb{Z}_d^{n-1}$. In terms of the codewords, a $\mathcal{L}^*$ code differs from $\mathcal{L}$ in two aspects. First, it contains only the codewords of $\mathcal{L}$ that has 0 in the first position. This will remove, for example, the $\mathbf{1}$ codeword. Secondly, it deletes the first 0 position from all the codewords. On its own, the second property is known as puncturing, where the first position is removed but *all* codewords are kept. We shall give a self-contained definition of a shortened Reed-Muller code as follows.

Definition 10. *Shortened Reed-Muller codes, $\mathcal{RM}_d^*(1,m)$, are classical linear codes on $\mathbb{Z}_d^n$, where $n = d^m - 1$, of dimension m. They are the set of codewords $\mathcal{RM}_d^*(1,m) = \{P_d^m(\mathbf{u}) : \mathbf{u} \in \mathbb{Z}_d\}$ defined in terms of linear maps.*

Here P_d^m is the same map as U_d^m, but omitting the first element. For example, the shortened version of Eq. (3.37) is

$$P_3^2(\mathbf{u}) = P_3^2(0,1) = (1,2,0,1,2,0,1,2), \tag{3.41}$$

Notice that the above definition makes use of only linear maps and not affine maps as in the $\mathcal{RM}_d(1,m)$. This is because, in the unshortened code we had a generator $\mathbf{1}$ that corresponded to the constant term in affine functions, but the $\mathbf{1}$ generator is dropped in the definition of the shortened Reed-Muller code. Consequently, the dimension of the code drops by one; $\mathrm{Dim}(\mathcal{RM}_d^*(1,m)) = \mathrm{Dim}(\mathcal{RM}_d(1,m)) - 1$. For completeness, lets now consider the shortened analogue of Lem. 1.

Lemma 2. *Given a λ-function Λ and a shortened code $\mathcal{RM}_d^*(1,m)$ all $\mathbf{v} \in \mathcal{RM}_d^*(1,m)$ satisfy $\Lambda(\mathbf{v} \oplus c\mathbf{1}) = -\lambda_c$ mod d^m.*

Proof. This follows quickly from Lem. 1. Given a $\mathbf{v} \in \mathcal{RM}_d^*(1,m)$, let us define

$$\mathbf{w} = (0, v_1, v_2, \ldots, v_n) \oplus c\mathbf{1} = (c, v_1 \oplus c, v_2 \oplus c, \ldots, v_n \oplus n), \tag{3.42}$$

where clearly $\mathbf{w}$ is a codeword of the unshortened code $\mathcal{RM}_d(1,m)$. Furthermore, $\Lambda(\mathbf{w}) = \Lambda(\mathbf{v}) + \lambda_c$ as it has an extra term appended. However, Lem. 1 tells us that $\Lambda(\mathbf{w}) = 0$, and so $\Lambda(\mathbf{v}) = -\lambda_c$. □

In the next section, we see that Lem. 2 is related to transversality of quantum gates for an associated quantum code.

3.3.3 Quantum Reed-Muller codes

Here we demonstrate how quantum Reed-Muller codes can be constructed from two classical $\mathcal{RM}_d^*(1,m)$ for general m and d.

Definition 11. *$\mathcal{QRM}_d(m)$ with $m \geq 1$ is a quantum CSS code over $n = d^m - 1$ qudits of prime dimension d. The code-space is defined by*

1. $\mathcal{L}_X = \mathcal{RM}_d^*(1,m)$*;*
2. $\mathcal{L}_Z = [\mathrm{span}(\mathcal{L}_X, \mathbf{1})]^{\perp}$*;*
3. $\bar{X} = X[\mathbf{1}]$*;*
4. $\bar{Z} = Z[-\mathbf{1}]$.

We could have equivalently specified $\mathcal{L}_Z$ as a higher order Reed-Muller code [27], though the above is simpler and more succinct. We now verify that $\mathcal{QRM}_d(m)$ codes are indeed quantum codes. By construction, the stabilizer is Abelian as $\mathcal{L}_Z \subset \mathcal{L}_X^{\perp}$. It is straightforward to check the logical operators are well defined: that $\bar{Z}$ commutes with the stabilizer; $\bar{X}$ commutes with the stabilizer; and $\bar{X}\bar{Z} = \omega^{-1}\bar{Z}\bar{X}$. Now our next main result can be concisely stated.

Theorem 5. *$\mathcal{QRM}_d(m)$ quantum codes are $\mathcal{M}_d^m$–distillation codes of distance 2.*

The main property we need to prove is transversality for all $M \in \mathcal{M}_d^m$. As with all CSS codes [120], we have that

$$|j\rangle_L = \frac{1}{\sqrt{|\mathcal{L}_X|}} \sum_{\mathbf{v} \in \mathcal{L}_X} |\mathbf{v} \oplus j\mathbf{1}\rangle \,. \tag{3.43}$$

Acting on this logical state with $M^{\otimes n}$ gives

$$M^{\otimes n} |j_L\rangle = \frac{1}{\sqrt{|\mathcal{L}_X|}} \sum_{\mathbf{v} \in \mathcal{L}_X} \exp\left(i \frac{2\pi}{d^m} \Lambda(\mathbf{v} \oplus j\mathbf{1}) \right) |\mathbf{v} \oplus j\mathbf{1}\rangle \,, \tag{3.44}$$

where Λ is a λ-function (Recall Def. 9) using the integers $\{\lambda_j\}$ associated with the eigenvalues of the unitary M. Now we use our key lemma 2 to conclude

$$\begin{aligned} M^{\otimes n} |j\rangle_L &= \frac{1}{\sqrt{|\mathcal{L}_X|}} \sum_{\mathbf{v}\in\mathcal{L}_X} \exp(-2i\pi\lambda_j/d^m) |\mathbf{v} \oplus j\mathbf{1}\rangle, \\ &= \exp(-2i\pi\lambda_j/d^m) |j\rangle_L = \bar{M}^\dagger |j\rangle_L, \end{aligned} \tag{3.45}$$

and so we can identify $M^{\otimes n}$ with $\bar{M}^\dagger$.

The second property we need to prove is that the distance of such quantum code is 2. This is a rather straightforward task as distance *2* is the smallest non-trivial distance. The relevant distance is D_z, the smallest $|\mathbf{v}|_H$ such that it produces a logical error $Z[\mathbf{v}]\Pi = Z_L^j\Pi$. For such an operator $\mathbf{v} \in \mathcal{L}_X^\perp$ but $\mathbf{v} \neq 0$, so the phase error commutes with the X stabilizer but is non-trivial. If such an operator existed with Hamming weight 1, it would entail that there existed a qudit upon which $\mathcal{L}_X$ acted trivially, which there is not. The fact that the distance is not greater than 2 is shown in the following section.

3.3.4 Weight Enumerators and MacWilliams Identity

In the last section, we introduced our version of higher-dimensional Reed-Muller codes (based on the classical shortened Reed-Muller codes), and showed that they have the required non-Clifford transversality. In this section we wish to evaluate the exact iterative formula, Eq. (3.25), of these codes. However, for a general noise model, this task is not tractable because the size and complexity of the sets $\mathcal{L}_Z$ and $\mathcal{L}_X^\perp$ grow exponentially[7] with d and m. However, by considering depolarizing noise, the problem becomes much simpler as shown by Eq. (3.30), which we restate here

$$f_0^{\text{out}} = \frac{\sum_{\mathbf{v}\in\mathcal{L}_Z} \mu^{|\mathbf{v}|_H}}{\sum_{\mathbf{v}\in\mathcal{L}_X^\perp} \mu^{|\mathbf{v}|_H}}. \tag{3.46}$$

To evaluate the above equation we use very useful tools from classical coding theory, namely weight

[7] Notice for small codes (e.g. $\mathcal{QRM}_3(2)$ and $\mathcal{QRM}_5(1)$) it is computationally feasible to sum over all the elements of $\mathcal{L}_Z$ and $\mathcal{L}_X^\perp$. In fact, in the last chapter we were also able to calculate the entire distillation map because the five qutrit code $[\![5,1,3]\!]_3$ is small enough.

enumerators and the MacWilliams identities [103]. The above equation can be expressed as

$$f_0^{\mathrm{out}} = \frac{W_{\mathcal{L}_Z}(\mu)}{W_{\mathcal{L}_X^{\perp}}(\mu)}, \tag{3.47}$$

where $W_{\mathcal{L}}(\mu)$ is the weight enumerator

$$W_{\mathcal{L}}(\mu) = \sum_{v \in \mathcal{L}} \mu^{|\mathbf{v}|_H}. \tag{3.48}$$

As can be seen form the above equation, a weight enumerator is simply a polynomial—the coefficient being the number of codewords with Hamming weight equal to the power of each term in the polynomial. More importantly, a weight enumerator for a code $\mathcal{L}$ can be related to the weight enumerator for the dual code $\mathcal{L}^{\perp}$ by the MacWilliams identity

$$W_{\mathcal{L}^{\perp}}(\mu) = \frac{1}{d^{\mathrm{Dim}}(\mathcal{L})}[1 + (d-1)\mu]^n W_{\mathcal{L}}\left(\frac{1-\mu}{1+(d-1)\mu}\right). \tag{3.49}$$

For clarity we will use the shorthand

$$\tilde{\mu} = \frac{1-\mu}{1+(d-1)\mu}. \tag{3.50}$$

Using $\mathcal{L}_Z = [\mathrm{span}(\mathcal{L}_X, \mathbf{1})]^{\perp} = (\mathcal{L}'_X)^{\perp}$ (see Eq. 1.22) and the MacWilliams identity we have

$$f_0^{\mathrm{out}} = \frac{W_{\mathcal{L}'_X}(\tilde{\mu})}{dW_{\mathcal{L}_X}(\tilde{\mu})}. \tag{3.51}$$

The above form is more convenient for us because, as we will see, the codes $\mathcal{L}'_X$ and $\mathcal{L}_X$ are simpler than their duals, and so the MacWilliams identity has proven extremely helpful. We now derive a closed form for the weight enumerators $W_{\mathcal{L}'_X}(\tilde{\mu})$ and $W_{\mathcal{L}_X}(\tilde{\mu})$, which is the subject of the next lemma.

Lemma 3. *For any quantum Reed-Muller code with $\mathcal{L}_X = \mathcal{RM}_d(1, m)$ we have*

$$W_{\mathcal{L}_X}(\tilde{\mu}) = 1 + (d^m - 1)\tilde{\mu}^{(d^m - d^{m-1})}, \tag{3.52}$$

and

$$W_{\mathcal{L}'_X}(\tilde{\mu}) = W_{\mathcal{L}_X}(\tilde{\mu}) + (d-1)[\tilde{\mu}^{(d^m-1)} + (d^m - 1)\tilde{\mu}^{(d^m-1-d^{m-1})}]. \tag{3.53}$$

Proof. We need to find the weight enumerators for $\mathcal{L}_X = \mathcal{RM}^*_d(1,m)$ and $\mathcal{L}'_X = \mathrm{span}(\mathcal{L}_X, \mathbf{1})$. Notice that $\mathcal{L}_X \subset \mathcal{L}'_X$, therefore it is natural to start our evaluation with $\mathcal{L}_X$ and then add the remaining terms needed for $\mathcal{L}'_X$.

First, $\mathcal{L}_X$ contains a zero vector $(0,0,\dots,0)$ with zero Hamming weight. Second, all the remaining codewords, there are $d^m - 1$ such codewords, have $(d-1)$ zeros, i.e have Hamming weight $n-(d-1) = d^m - d$. Thus we have the weight enumerator

$$W_{\mathcal{L}_X}(\tilde{\mu}) = 1 + (d^m - 1)x^{(d^m - d)}. \tag{3.54}$$

The enumerator for $\mathcal{L}'_X$ can be broken up into d separate sums, since $\mathcal{L}'_X = \{\mathcal{L}_X, \mathcal{L}_X \oplus \mathbf{1}, \dots, \mathcal{L}_X \oplus (d-1)\mathbf{1}\}$, and so

$$\begin{aligned} W_{\mathcal{L}'_X}(\tilde{\mu}) &= \sum_{j=0}^{d-1} W_{\mathcal{L}_X \oplus j\mathbf{1}}(\tilde{\mu}), && (3.55)\\ &= W_{\mathcal{L}_X}(\tilde{\mu}) + \sum_{j=1}^{d-1} W_{\mathcal{L}_X \oplus j\mathbf{1}}(\tilde{\mu}). && (3.56)\end{aligned}$$

For the rest of this argument we focus on the $j \neq 0$ terms. First, each $j\mathbf{1}$ when added to the $(0,0,\dots,0)$ vector will generate a codeword of full Hamming weight ($n = d^m - 1$). Second, each $j\mathbf{1}$ when added to any other codeword of $\mathcal{L}_X$ (other than the $(0,0,\dots,0)$ vector) results in a codeword with d^{m-1} zero's and so Hamming weight $n - d^{m-1} = d^m - 1 - d^{m-1}$. For each $\mathcal{L}_X \oplus j\mathbf{1}$, there are $d^m - 1$ such codewords and so

$$W_{\mathcal{L}_X \oplus j\mathbf{1}}(\tilde{\mu}) = x^{(d^m - 1)} + (d^m - 1)x^{(d^m - 1 - d^{m-1})}. \tag{3.57}$$

For every $j \neq 0$ we get the same result and we have $d-1$ such sums, and so

$$W_{\mathcal{L}'_X}(\tilde{\mu}) = W_{\mathcal{L}_X}(x) + (d-1)W_{\mathcal{L}_X \oplus \mathbf{1}}(\tilde{\mu}), \tag{3.58}$$

which is the required expression. □

Using the result of the above lemma, we then obtain the exact analytic form of the iterative formula after one round of distillation and under depolarising noise as

$$f_0^{\mathrm{out}} = \frac{1 + (d^m - 1)\tilde{\mu}^{(d^m - d^{m-1})}}{d + d(d^m - 1)\tilde{\mu}^{(d^m - d^{m-1})} + d(d-1)\left(\tilde{\mu}^{(d^m - 1)} + (d^m - 1)\tilde{\mu}^{(d^m - 1 - d^{m-1})}\right)}, \tag{3.59}$$

Now we substitute back the original variable ϵ, such that

$$\tilde{\mu} = 1 + \frac{d}{(d-1)}\epsilon, \tag{3.60}$$

which would give us a closed analytic form. The exact expression is a bit long to reproduce here. Rather we present the Taylor expansion to second order in ϵ,

$$\epsilon^{\text{out}} = \frac{(d^m - 1)(d-2)}{2(d-1)}\epsilon^2 + O[\epsilon^3]. \tag{3.61}$$

The above formula holds many interesting facts. Notice that for all codes with odd prime d and all m, we see a quadratic error suppression. In contrast, the qubit Reed-Muller code used by Bravyi and Kitaev, namely $[\![15,1,3]\!]_2 \equiv \mathcal{QRM}_2(4)$, obtained a cubic reduction $\epsilon^{\text{out}} \sim 35\epsilon^3$. Our analysis also describes the Bravyi-Kitaev protocol, the only difference being that in the qubit case we need $m \geq 4$, and so the above formula also holds for qubits. It is intriguing to observe that the factor $(d-2)$ appears above and so the quadratic term would vanish only in the qubit case, and so in higher dimensions these Reed-Muller codes are only distance 2.

3.4 Distillation Performance

In this section we will outline two measures that can quantify the performance of our $\mathcal{QRM}_d(m)$ distillation protocols, namely the distillation *yield* and the distillation *thresholds* under the depolarizing noise model. Next, we will consider the performance of the smallest possible codes, which are the $8-$qutrit code $\mathcal{QRM}_3(2)$ and $5-$ququint code $\mathcal{QRM}_5(1)$, in more detail.

3.4.1 Distillation Yields

The yield captures the performance of the distillation protocol in terms of the cost of mixed states needed to distil a magic state. More precisely, given some target error probability ϵ_{tar}, the yield quantifies the expected fraction of the initial copies needed to achieve the desired final error probability. We will show that our protocols yield magic states at a rate that scales only polynomially with ϵ_{tar}.

For any protocol and any resource state ρ with error probability ϵ_{in}, there exists a number of distillation rounds $N(\rho, \epsilon_{\text{tar}})$ required to achieve ϵ_{tar}. Recall that the success probability P_{succ} for measuring the trivial syndromes depends on the error probability of the resource states. For the k^{th} round of distillation we denote the success probability by $P(k)$. Then the yield is simply

$$Y(\rho, \epsilon_{\text{tar}}) = \prod_{k=1,\ldots,N} \frac{P(k)}{n}, \tag{3.62}$$

where n is the number of copies used per iteration ($n = d^m - 1$ in our case). Our main interest here is to see how this expression scales as the target probability $\epsilon_{\text{tar}} \to 0$. In this limit, the success probability approaches 1, which means that $P(k)$ approaches 1 as k increases. Therefore, for all $p < 1$ these exists a c (number of iterations) such that for all $k > c$ we have $P(k) > P(c) = p$. This allows us to lower bound the yield such that

$$Y(\rho, \epsilon_{\text{tar}}) \geq C \left(\frac{P(c)}{n} \right)^{N-c}, \tag{3.63}$$

where C is a constant overhead, independent of ϵ_{tar}, which represents the yield for c iterations. Furthermore, after c iterations the error probability is now ϵ_c, and observe that for a single round we know that (for our protocols) $\epsilon^{\text{out}} \leq K\epsilon^2$ for some K—equivalently $K\epsilon^{\text{out}} \leq (K\epsilon)^2$. Therefore, the error probability after N iterations, ϵ_N, satisfies $K\epsilon_N \leq (K\epsilon_c)^{2^{N-c}}$. Taking $K\epsilon_c < 1$ allows us to bound the number of iterations needed such that

$$N - c < \log_2 \left(\frac{\log(\epsilon_{\text{tar}}^{-1}/K)}{\log(\epsilon_c^{-1}/K)} \right). \tag{3.64}$$

Now we make use of the following identity, for any positive a and b we have $a^{\log_2(b)} = b^{\log_2(a)}$. Using this relation and the above equation, we can express the yield as

$$Y(\rho, \epsilon_{\text{tar}}) \geq C \left(\frac{\log(\epsilon_{\text{tar}}^{-1}/K)}{\log(\epsilon_c^{-1}/K)} \right)^{\log_2(P(c)/n)}. \tag{3.65}$$

With the shorthand $\gamma = -\log_2(P_c/n)$, which is positive, we finally have

$$Y(\rho, \epsilon_{\text{tar}}) \geq C \frac{\log(\epsilon_c^{-1}/K)^{\gamma}}{\log(\epsilon_{\text{tar}}^{-1}/K)^{\gamma}}. \tag{3.66}$$

This decreases by a factor polynomial in $\epsilon_{\text{tar}}^{-1}$.

The expected resource cost of distillation is the inverse of the yield, and this increases only polynomially in $\epsilon_{\text{tar}}^{-1}$. As we can see the scaling is governed by the factor $\gamma = -\log_2(P(c)/n)$. For practical purposes we can assume that $P(c)$ is arbitrarily close to 1, i.e. we are working in the regime that the error probability is very small. Hence, the relevant scaling parameter is $\gamma^* = \log_2(n) = \log_2(d^m - 1)$, which we give in table 3.1. Therefore, the yield becomes

$$Y(\rho, \epsilon_{\text{tar}}) \sim O\left(\log(\epsilon_{\text{tar}}^{-1}/K)^{-\gamma^*}\right). \tag{3.67}$$

d	$m = 1$	$m = 2$	$m = 3$	$m = 4$
2	N/A	N/A	N/A	2.46497
3	N/A	3	4.70044	6.32193
5	2	4.58496	6.9542	9.2854
7	2.58496	5.58496	8.41785	11.2288
11	3.32193	6.90689	10.3772	13.8376
13	3.58496	7.39232	11.1007	14.8017
17	4	8.16993	12.2621	16.3498
19	4.16993	8.49185	12.7436	16.9917

Table 3.1: The scaling parameter, γ^*, for the $\mathcal{QRM}_d(m)$ distillation as governed by Eq. 3.67. The smaller the value of γ^*, the more resource efficient the protocol in the limit of many iterations. N/A indicates not applicable, as for those parameters no non-Clifford gates exist.

Notice that the code $\mathcal{QRM}_5(1)$ achieves the best yield scaling of all quantum Reed-Muller codes.

3.4.2 Depolarising Noise Thresholds

The second measure of performance of a distillation protocol is its capability in distilling the largest mixed state resource region. While some codes can distil large regions of non-stabilizer mixed states (i.e. have high distillation thresholds), others can have achieve better yields. For this purpose we have used the exact expression for ϵ^{out} to find the depolarizing noise threshold, denoted by ϵ_{dep}^*, below which the distillation occurs[8]. We have numerically evaluated $\epsilon^*_{\rm dep}$ for small values of d and m as shown in Tab. (3.2). Under a more general noise model, we denote the absolute threshold, which could be smaller, by ϵ^*. This threshold corresponds to the noise resulting in mixed states off the depolarizing 'axis', and we will provide a examples of how it is calculated in the next section.

As shown by Tab. (3.2), the threshold gets weaker for both increasing d and m. This is in correspondence with the approximate formula for $\epsilon^{\rm out}$ in Eq. (3.61). As m increases, the number of copies required per iteration increases and the depolarizing noise threshold decreases. This suggests to us that it is advantageous to use the smallest possible m. The only benefit of using codes with larger m is to distil a larger set of states by the protocol.

If we also compare our protocols with the threshold of the BK protocol for $d = 2$, the pattern of better threshold for smaller dimensions no longer holds. We see that the best threshold we observe is for $\mathcal{QRM}_5(1)$ with a fairly high threshold also observed for $\mathcal{QRM}_3(2)$. There are many subtle differences in the Clifford group between odd and even dimension, and here those differences work in our favour. In odd prime dimension we can construct smaller codes with transversal non-Clifford gates. Our code $\mathcal{QRM}_5(1)$ uses 4 ququints covering a Hilbert space of dimension 5^4, which to our knowledge is the smallest non-trivial stabilizer code with a transversal non-Clifford gate. Furthermore, research to date indicates that smaller codes lend themselves to better thresholds. A plausible explanation is that larger codes allow more undetected errors. Most of these undetected errors will have a large Hamming weight, and so while negligible for small ϵ, they will be damaging for the modest size ϵ relevant for threshold calculations.

[8] Note that we have used $\epsilon = 1 - \langle M_0 | \rho | M_0 \rangle$ to quantify the depolarizing noise. However, when a state is undergoing depolarising noise, it has the form

$$\rho = \epsilon_{\rm dep} \left| M_0 \right\rangle \left\langle M_0 \right| + (1 - \epsilon_{\rm dep}) \mathbb{1}/d, \tag{3.68}$$

as we have done in the previous chapter. Observe the dependence on the dimension appearing in the above equation. The two distinct noise measures are related by

$$\epsilon^*_{\rm dep} = (d-1)\epsilon_{\rm dep}/d. \tag{3.69}$$

This suggests that the above depolarising noise relation will give larger threshold values. For example the codes $\mathcal{QRM}_3(2)$ and $\mathcal{QRM}_5(1)$ having thresholds at $\epsilon_{\rm dep} = 0.317$ and $\epsilon_{\rm dep} = 0.453$, respectively.

d	m=1	m = 2	m = 3	m = 4
2	N/A	N/A	N/A	0.14148
3	N/A	0.211001	0.0657764	0.0214564
5	0.3631226	0.0614718	0.0119213	0.00236986
7	0.2322599	0.0291865	0.00409851	0.000584079
11	0.1341066	0.0111835	0.00100907	0.0000916717
13	0.1106148	0.00790156	0.000604487	0.0000464795
17	0.0818753	0.00454655	0.000266565	0.0000156773
19	0.072453	0.00362063	0.000190054	0.0000100014

Table 3.2: The distillation threshold ϵ^*_{dep} for depolarizing noise when distilled by $\mathcal{QRM}_d(m)$. Notice the threshold for the Bravyi-Kitaev $[\![15,1,3]\!]_2 \equiv \mathcal{QRM}_2(4)$ protocol. N/A indicates that Reed-Muller codes with those do not have a non-Clifford gate.

3.4.3 Examples: Qutrit $\mathcal{QRM}_3(2)$ and Ququint $\mathcal{QRM}_5(1)$ Codes

In this section we will outline the properties of the smallest two Reed-Muller codes, which are the three dimensional 8−qutrit $\mathcal{QRM}_3(2)$ code and the five dimensional 5−ququint $\mathcal{QRM}_5(1)$ code. Due to their small sizes, the iterative formula for these codes can be evaluated exactly for general noise models.

Performance of $\mathcal{QRM}_3(2)$

We begin by stating the definition of $\mathcal{QRM}_3(2)$ in terms of its stabilizer generators.

Definition 12. *$\mathcal{QRM}_3(2)$ is a CSS code over $n = 8$ qudits of dimension 3. The $\mathcal{L}_X$ code is generated by*

$$\begin{aligned}\mathbf{u}_1 &= (1,2,0,1,2,0,1,2), \qquad (3.70)\\ \mathbf{u}_2 &= (0,0,1,1,1,2,2,2).\end{aligned}$$

Similarly, $\mathcal{L}_Z$ is the code generated by

$$\begin{aligned} \mathbf{v}_1 &= (1,2,0,1,2,0,1,2), \\ \mathbf{v}_2 &= (0,0,1,1,1,2,2,2), \\ \mathbf{v}_3 &= (0,0,1,2,0,2,1,0), \\ \mathbf{v}_4 &= (1,1,0,1,1,0,1,1), \\ \mathbf{v}_5 &= (0,0,1,1,1,1,1,1). \end{aligned} \tag{3.71}$$

The logical operators are $\bar{Z} = Z[-\mathbf{1}] \equiv Z[2\mathbf{1}]$ and $\bar{X} = X[\mathbf{1}]$.

This code is transversal with respect to the canonical $\mathcal{M}_3$ non-Clifford gate:

$$M = \begin{pmatrix} \tau & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \tau^{-1} \end{pmatrix}, \tag{3.72}$$

where $\tau = \exp(i2\pi/9)$. In the distillation protocol, recall that we first apply the C_M–twirling on all the input qudits, which project them onto the plane spanned by the $|M_k\rangle$ orthogonal basis, such that $\rho = \sum_k f_k |M_k\rangle \langle M_k|$. Recall from the last chapter that for qutrit systems such a plane can be represented by two real parameters such that the eigenstates of the $\mathcal{M}_3$ operator form an equilateral triangle, we show such a representation in Fig. (3.1). Also, when we wish to distil $|M_0\rangle$, we can think of the weights f_1 and f_2 as representing different types of noise. For our purposes, a more convenient parameterization is $f_1 = \epsilon \cos^2(\theta)$ and $f_2 = \epsilon \sin^2(\theta)$ as we are mainly interested in how the *total* noise reduces. Our techniques allow us to find an analytic solution for ϵ' after a single iteration of magic state distillation with $\mathcal{QRM}_3(2)$. However, the expression is lengthy so here we truncate to 3rd order

$$\epsilon^{\text{out}} = \epsilon^2[3 + \cos(4\theta)] + \epsilon^3[9 - \cos(4\theta)] + O[\epsilon^4], \tag{3.73}$$

which is quadratically reduced. In Fig (3.2a), we show the exact output error probability for the whole range of different noise models (different θ) and depolarizing noise ($\theta = \pi/2$). We find that a threshold of $\epsilon^* = 0.20015$ for general noise and $\epsilon^*_{\text{dep}} = 0.211001$ for depolarizing noise. As such, for all θ, if $0 < \epsilon < \epsilon^*$ it follows that $\epsilon^{\text{out}} < \epsilon$.

The region of distillable states is actually slightly larger than the $\epsilon < \epsilon^*$ region, with a greater noise tolerance for some values of θ. To find the whole distillable region we resort to numerics and present the results as part of the plane in Fig. 3.1. We also identify the other regions, the stabilizer states and positive Wigner distributions (or bound magic states), which cannot be distilled by any MSD protocol. Notice the clear region in which neither our protocol works upon nor is ruled out from distillability by any known theorem. This is as a comparison to what we have observed in the last chapter.

Also important is the success probability of distillation with $\mathcal{QRM}_3(2)$, which for all states satisfy $P_{\text{succ}} \geq 1/9$ and for small ϵ is approximately

$$P_{\text{succ}} = 1 - 8\epsilon + [31 + \cos(4\theta)]\epsilon^2 + O(\epsilon^3). \tag{3.74}$$

Given these fairly high success probabilities and that we use only 8 copies per iteration, this protocol is competitive in comparison to the $\mathcal{QRM}_2(4)$ by Bravyi-Kitaev [27]. Their protocol has $P_{\text{succ}} \geq 1/16$ and for small ϵ it achieves $P = 1 - 15\epsilon + O(\epsilon^2)$. Our $\mathcal{QRM}_3(2)$ code requires fewer copies per iteration, but it would require more iterations to achieve the same error suppression as $\mathcal{QRM}_2(4)$, since $\mathcal{QRM}_2(4)$ has a cubic error suppression rather than just quadratic.

In Fig. (3.3.1a) we consider the exact yield of our protocol for depolarizing noise (i.e. $\theta = \pi/4$). For small error probability $\epsilon_{\text{in}} < 0.05$, the yield of our protocol $\mathcal{QRM}_3(2)$ is similar to $\mathcal{QRM}_2(4)$. Both protocols give yields of the same order of magnitude and which protocol is superior fluctuates depending on the required iterations. However, as the initial error probability ϵ_{in} increases, the yield of $\mathcal{QRM}_3(2)$ exceeds that of BK by many orders of magnitude.

Performance of $\mathcal{QRM}_5(1)$

The smallest $\mathcal{QRM}_d(1)$ with an $\mathcal{M}_d$ non-Clifford gate is a five dimensional $4-$ququint code, defined explicitly as follows.

Definition 13. *$\mathcal{QRM}_5(1)$ is a CSS code over $n = 4$ ququints of dimension 5. The $\mathcal{L}_X$ code is generated by*

$$\mathbf{u}_1 = (1, 2, 3, 4). \tag{3.75}$$

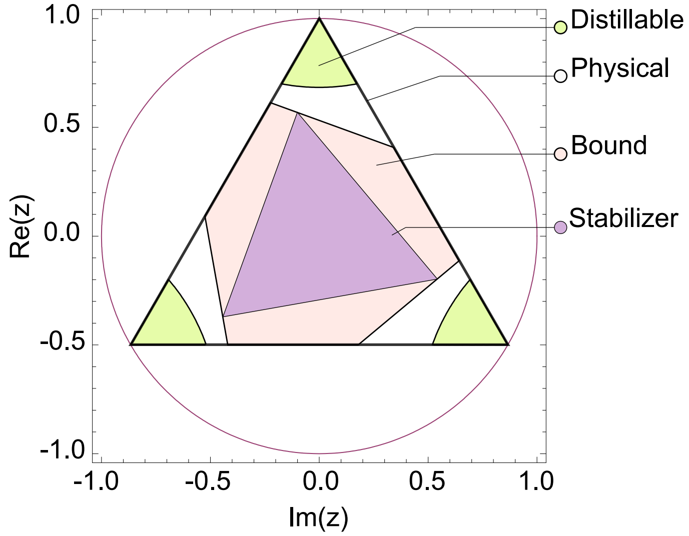

Figure 3.1: The canonical C_M–plane for a qutrit, $d = 3$, which any state can be projected onto by C_M–twirling. Every quantum state is a point in the complex plane for the complex number $z_\rho = \mathrm{tr}(C_M \rho)$. The three pure magic states, $|M_k\rangle$, take values $z = 1, \omega, \omega^2$, which have $|z|^2 = 1$ and so lie on a circle in the plane. All *physical* states have, $z = (1 - f_1 - f_2) + \omega f_1 + \omega^2 f_2$, and so lie in the convex hull of the pure magic states, forming a triangle of physical states. The *distillable* region of states can, by use of the $\mathcal{QRM}_3(2)$ protocol, be brought arbitrarily close to the nearest pure magic state. The *stabilizer* states are the convex hull over the set of points, z, taken for each of the pure stabilizer states. It is impossible to distil the stabilizer states and the states with positive Wigner distribution (the *bound* states). Note that the rotational symmetry is to be expected as the Pauli Z rotation performs a rotation in the C_M–plane.

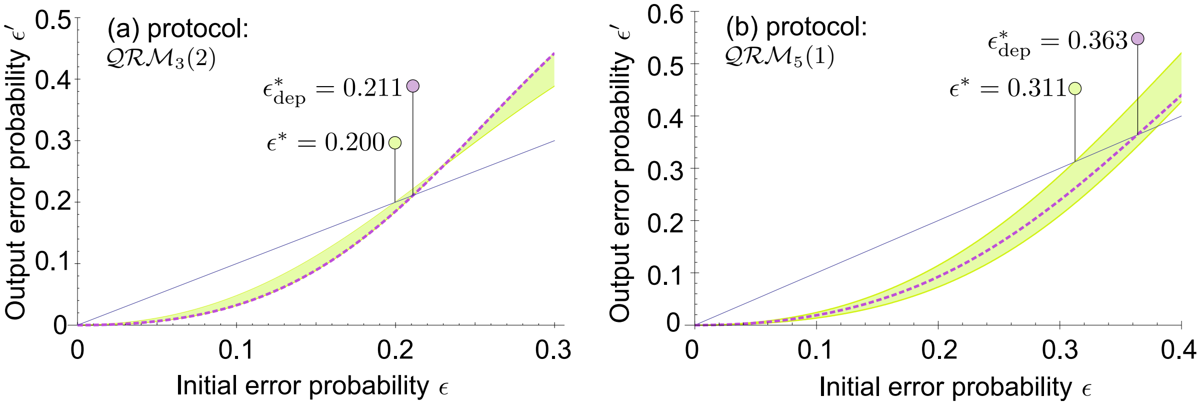

Figure 3.2: The output error, $\epsilon^{\rm out}$ against input error, ϵ for (a) $\mathcal{QRM}_3(2)$ and (b) $\mathcal{QRM}_5(1)$. For a fixed ϵ there are many different compatible states, and so there are many different possible output $\epsilon^{\rm out}$ and these are shown as a region rather than single curve. For the worst case noise we mark the threshold ϵ^*. The dashed line shows the specific instance of depolarizing noise, and the associated depolarizing threshold $\epsilon^*_{\rm dep}$ is also shown. The straight line is simply the "break even" line.

Similarly, $\mathcal{L}_Z$ is the code generated by

$$\begin{aligned} \mathbf{v}_1 &= (1,2,3,4), \\ \mathbf{v}_2 &= (1,4,4,1). \end{aligned} \tag{3.76}$$

The logical operators are $\bar{Z} = Z[-\mathbf{1}] \equiv Z[4]$ and $\bar{X} = X[\mathbf{1}]$.

For the above code is transversal with respect to the canonical $\mathcal{M}_5$ non-Clifford gate,

$$M = \begin{pmatrix} \omega^3 & 0 & 0 & 0 & 0 \\ 0 & \omega & 0 & 0 & 0 \\ 0 & 0 & \omega^{-1} & 0 & 0 \\ 0 & 0 & 0 & \omega^{-2} & 0 \\ 0 & 0 & 0 & 0 & \omega^{-1} \end{pmatrix}, \tag{3.77}$$

where here $\omega = \exp(i2\pi/5)$. The distillation protocol associated with this code have many distinguishing features already mentioned: it is the smallest known non-trivial code to have a transversal non-Clifford; it has the largest noise threshold against depolarizing noise ($\epsilon^*_{\text{dep}} = 0.363$); and it has the best known scaling in terms of expected yield (with $\gamma = 2$). All these features can be attributed to the fact that $d = 5$ is the smallest dimension where a diagonal non-Clifford gate exists with period d, allowing us to work with $m = 1$.

Again, the C_M–twirled states are parameterised by a fidelity, $f_0 = 1 - \epsilon$, and 4 independent noise parameters f_j for $j = 1, 2, 3, 4$. Unfortunately, this means that we cannot visualise the C_M–plane since it depends on 4 real parameters. Nevertheless, we can still obtain a numerical estimate for the distillation noise thresholds. In Fig. (3.2b) we show the range of different output error rates for general noise and for depolarising noise, which have thresholds of $\epsilon^* = 0.31195$ and $\epsilon^*_{\text{dep}} = 0.363122$.

For completeness we include the output error probability for the depolarizing noise case with $f_0 = 1 - \epsilon$ and $f_{j\neq 0} = \epsilon/4$. After a successful implementation of one round, a depolarized state is output with

$$\epsilon^{\text{out}} = \frac{\epsilon^2(96 - 160\epsilon + 75\epsilon^2)}{64 - 256\epsilon + 480\epsilon^2 - 400\epsilon^3 + 125\epsilon^4} \sim \frac{3\epsilon^2}{2} + \frac{7\epsilon^3}{2} + O[\epsilon^4], \tag{3.78}$$

and this occurs with probability

$$P_{\text{succ}} = \frac{(1 - 2\epsilon)^4(64 - 256\epsilon + 480\epsilon^2 - 400\epsilon^3 + 125\epsilon^4)}{64(-1 + \epsilon)^4} \sim 1 - 8\epsilon + \frac{51\epsilon^2}{2} + O[\epsilon^3]. \tag{3.79}$$

Finally, based on the above results we expect this protocol to have an excellent yield. We numerically studied the yield and again compared it against the qubit protocol $\mathcal{QRM}_2(4)$, see Fig (3.3.2b). The numerics confirm that across all parameter regimes $\mathcal{QRM}_5(1)$ offers a significant resource savings of potentially many orders of magnitude.

3.5 State-Injection and Quantum Universality

In the previous chapter, we showed how the qutrit magic states can be converted to a qutrit analogue of the Bloch sphere equator states, which were then used for state-injection of non-Clifford phase gate. Here we generalise these concepts further and define a qudit *equatorial* state as follows.

Definition 14. *A qudit quantum state $|\Theta\rangle$ is said to be equatorial, or a phase state, if $\Theta \in \mathbb{R}^d$ and*

$$|\Theta\rangle = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} e^{i\Theta_j} |j\rangle . \tag{3.80}$$

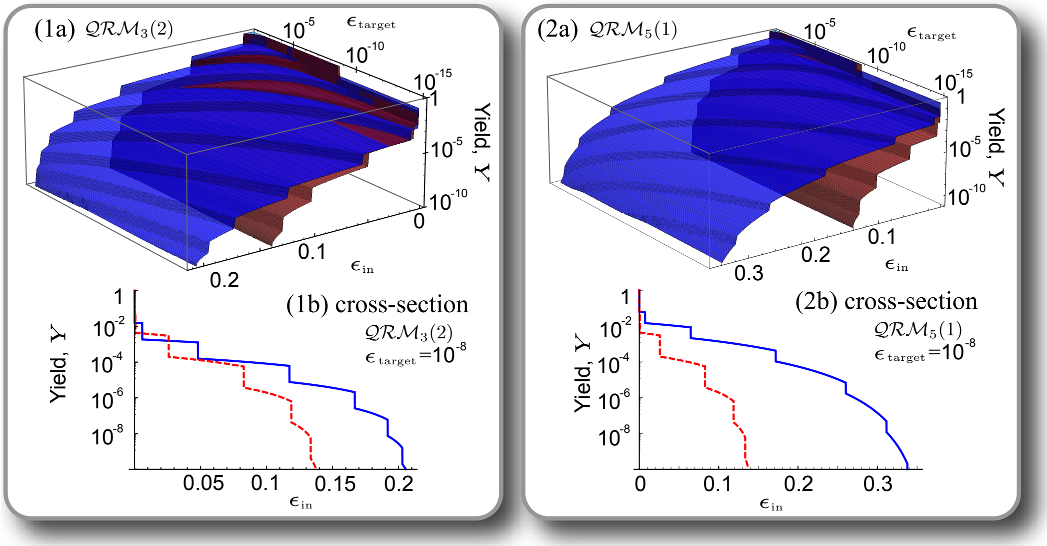

Figure 3.3: The yield on a log-scale of of our protocols, $\mathcal{QRM}_3(2)$ and $\mathcal{QRM}_5(1)$, (blue) compared with the Bravyi-Kitaev $\mathcal{QRM}_2(4)$ (red) protocol. Plots (1a) and (2a) are a function of initial error probabilities ϵ_{in} and target error probabilities ϵ_{tar}. For the qutrit and ququint states we consider depolarizing noise. Both (1a) and (2a) are further illustrated with cross-sections (1b) and (2b), respectively, with the target error probability held constant. The sudden changes in yield occurs because of discrete changes in the number of iterations required.

From the above definition we see immediately that the magic state $|M_0\rangle$ is in fact an equatorial state with $\Theta_j = 2\lambda_j\pi/d^m$. Just like with the qutrit magic phase state, the above equatorial states have the important property of being unbiased with respect to the computational bases. In other words, the outcomes of a Z measurement are completely random.

We can now use these equatorial states to simulate a non-Clifford unitary via a state-injection circuit. Let's consider the input to the circuit to be $|\Phi\rangle|\psi\rangle$, where the state $|\psi\rangle = \sum_j c_j |j\rangle$ is the general state of interest that we wish to act the non-Clifford gate on. We proceed by performing the measurement of the operator $ZZ^\dagger$. Each possible outcome k corresponds to the projector Π_k onto a subspace stabilized by $\omega^{-k}ZZ^\dagger$. The outcome state is then

$$\Pi_k|\psi\rangle|\Theta\rangle \propto \sum_j c_j e^{i\Theta_{j\oplus k}}|j\oplus k\rangle|j\rangle. \tag{3.81}$$

We then decode by performing a Clifford unitary that maps $|j\oplus k\rangle|j\rangle \rightarrow |k\rangle|j\rangle$, and tracing out the first system. This results, as expected, into a unitary transformation on the general states of the form $|\psi\rangle \rightarrow U_k(\Theta)|\psi\rangle$, where

$$U_k(\Theta) = \sum_j e^{i\Theta_{j\oplus k}}|j\rangle\langle j|. \tag{3.82}$$

This unitary can also be expressed as

$$U_k(\Theta) = (X^k)^\dagger U_0(\Theta) X^k, \tag{3.83}$$

such that

$$U_0(\Theta)|+_0\rangle = |\Theta\rangle. \tag{3.84}$$

This unitary transformation is randomly selected from d different possibilities (corresponding to the k different possible outcomes). The question is then, how can we apply (or teleport) a deterministic unitary?

For our purposes we are interested in the unitary gates produced by injecting the magic state $|M_0\rangle$, and wish to teleport the non-Clifford gate M deterministically. For this magic state the above calculations imply that unitary transformation is

$$U_k = (X^k)^\dagger M X^k, \tag{3.85}$$

where we only have the desired unitary when the outcome is $k = 0$. Now, to perform this unitary deterministically when $k \neq 0$ we make use of the relation $C_M = MXM^\dagger$. Observe that $C_M^k = MX^kM^\dagger$, hence the unitary can be expressed as

$$\begin{aligned} U_k &= (X^k)^\dagger MX^kM^\dagger M, && (3.86) \\ &= (X^k)^\dagger C_M^k M. && (3.87) \end{aligned}$$

This suggests that we can always recover the non-Clifford unitary M (for all outcomes k) by applying a correction Clifford unitary, which is the inverse of $(X^k)^\dagger C_M^k$. Therefore, we have shown how our magic states $|M_0\rangle$ can always teleport a non-Clifford gate M via a state-injection circuit. Using the Thm. (1) we conclude that the gate set $\langle \mathcal{C}, M\rangle$ is universal.

Finally, in the above discussion we have only considered the perfect magic state $|M_0\rangle$ in the injection circuit. However, we can consider a more realistic scenario where we have a slightly noisy distilled magic state σ, such that $\epsilon = 1 - \langle M_0|\,\sigma\,|M_0\rangle$, in the injection circuit. We report that such modification (omitted here) would only lead to similar error (linear in ϵ) in the output state $M\,|\psi\rangle$.

3.6 Summary and Open Problems

We have presented a complete framework for MSD based on the qudit Reed-Muller codes. Along the way, we have provided a generalisation of the qubit T–gate (the family of $\mathcal{M}_d(m)$ magic gates) and have characterised the "equatorial" magic states $|M\rangle$ for prime dimensions. We have observed how the qudit systems can allow for much smaller distillation protocols (such as the 5-ququint code $\mathcal{QRM}_5(1)$, which is the smallest known MSD protocol) that outperform many qubit protocols in terms of the yield and noise threshold that can be tolerated.

In our investigation of MSD we have only considered quantum codes that encode a single qudit, i.e. $[\![n, 1, \delta]\!]_d$. However, in recent developments there have been qubit MSD protocols proposed that distil multiple magic states per iteration. Notably, in Ref. [110], a novel protocol takes ten noisy magic states per iteration and outputs two magic states. They call this protocol the 10-to-2 distillation protocol, and it has the benefit of increasing the yield. Similar techniques could potentially also be used to design higher-dimensional protocols, improving the performance of our protocols even further.

Chapter 4

Qudit Topological Quantum Memory

In this chapter we investigate the properties of qudit topological codes serving as a quantum memory. In particular, we describe an efficient decoder for the qudit toric code based on a Renormalization Group (RG) algorithm. We start in Sec. 4.1 by introducing the general properties of topological error correction codes and state the results of this chapter. In Sec. 4.2, we describe the properties of the qudit toric code. This is followed by a discussion of the noise model used here and the numerical method we adopt to estimate the thresholds. In Sec. 4.3 we provide a formal description of our RG decoder and we present the thresholds we obtain. In the limit of high qudit dimension d*, the thresholds achieved by the decoder reach the saturating value of about ~* 18%*. We explain, in Sec. 4.4, this behaviour to be due to a percolation phenomenon and we provide numerical evidence to support our claim. More precisely, the saturating threshold we observe is tightly upper-bounded by what we call the syndrome percolation threshold. To beat this upper-bound, we introduce an enhanced version of this decoder and show that it can boost the threshold, in the limit of high* d*, to about ~* 30%*. Finally, in Sec. 4.5, we briefly compare our decoder to other known RG decoders and discuss the generalisation of our work for a complete fault-tolerant simulation.*

4.1 Motivation

Any fault-tolerant scheme is associated with a form of error correction code that protects the information when stored or during transmission. We have already seen that the theory of fault-tolerant quantum computation defines a probability threshold, for each operation in the scheme, below which accurate and long computation can be performed. A great amount of research has been devoted in exploring various quantum systems with potential fault-tolerant schemes that achieve high threshold values and have experimentally favourable features. Arguably, the most important feature required for many current technologies is to have the quantum systems on a 2D geometry where only the nearest neighbours can interact.

Conventional schemes that are based on concatenation of stabilizer quantum error correcting codes (QECC) suffer from involving long-range interactions by construction. Such schemes have non-local stabilizer generators[1]. Embedding these schemes on a 2D geometry requires the introduction of many additional error-prone swapping operations, which will reduce the threshold substantially [152]. For example, modelling Steane's 7−qubit code on a 2D architecture with nearest-neighbour interactions, a fault-tolerant threshold of 1.85×10^{-5} was obtained in [154]. Under this setting, currently, the highest threshold achieved is 2.02×10^{-5} [147] using the Bacon-Shor code [11]. On the other hand, schemes that allow for long-range interactions and post-selection can achieve a threshold as high as 3% [93].

A more practical approach is to include the locality constraint of the interactions within the design of the QECC. The first example of such construction was introduced by Kitaev [90] in his discovery of topological error correction codes or *surface codes*. The surface codes are a class of stabilizer codes where the qubits are placed on a 2D lattice and the stabilizer generators are local Pauli operators. While the locality of the stabilizer depends on the geometry of the lattice, the error correction properties and the encoded space depend on the *topology* of the surface [28, 112]. Moreover, Kitaev demonstrated how to perform universal quantum computation that is fault-tolerant by its physical nature using topologically

[1] By non-local stabilizers we mean stabilizers such that the non-trivial Pauli operators acts on geometrically distant qudits. In other words, we cannot structure the qudits on a 2D geometrical lattice such that the stabilizer generators involve measuring nearest-neighbours qudits.

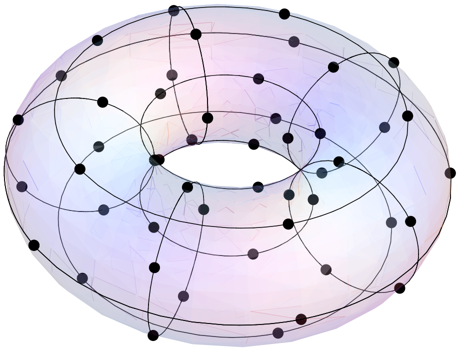

Figure 4.1: An example of the toric code where the qubits (black dots) are attached to the edges of a square lattice on the surface of a torus.

ordered many-body quantum systems supporting anyonic excitations. In our study here, we are not so much interested in the ‘topological’ features of the surface codes. We will simply consider these codes as stabilizer codes and study their capability in storing qudits.

More specifically, in this work we will study the *toric code*, which is a simplified version of the surface codes. Kitaev introduced the toric code as a toy model for topological error correction. The toric code captures all the essential features needed to store quantum information robustly in 2D topologically ordered systems. In the toric code the quantum systems are attached to the edges of a square lattice on a torus (see Fig. 4.1). The torus can be thought of as a 2D plane with two periodic boundaries. Converting the toric code to a surface code with open boundaries requires little effort and causes a negligible effect on the threshold. For this reason, we will only limit our study to the toric code.

For the toric code to serve as a quantum memory, the physical errors must be corrected at a sufficient rate to prevent the errors from accumulating and causing a logical error. This is achieved by performing active error correction and decoding the toric code periodically. Each decoding step consist of measuring all the stabilizer generators to obtain the syndromes. The syndrome values are then processed by a *classical* algorithm to guess the errors that could given rise to observed syndromes, which can then be corrected. In our study here we will assume that the stabilizer measurements to be noise-free, i.e. we are only considering *perfect decoding*.

Currently there are a variety of classical algorithms that efficiently decode the *qubit* toric code. The early extensive study by Dennis *et al.* [46] demonstrated how the toric code model can be mapped to a statistical mechanical model, the random-bond Ising model (RBIM), and showed how the optimal error threshold $p_{\mathrm{th}}^{\mathrm{opt}}$ of the toric code corresponds to a phase transition point in the RBIM known as the Nishimori point. In the case of independent, identically distributed (IID) bit-flip noise model and perfect decoding, the optimal threshold[2] is estimated (when mapped to the Nishimori point) to be $p_{\mathrm{th}}^{\mathrm{opt}} \approx 10.9\%$ [73, 111, 121, 44]. The decoding algorithm that achieves the closest threshold to $p_{\mathrm{th}}^{\mathrm{opt}}$ is the minimum-weight perfect matching algorithm (MWPMA) which has an efficient implementation based on Edmonds' blossom algorithm [53]. This algorithm has been extensively studied and the highest threshold it can achieve has been estimated to be about 10.3% [162, 163, 58].

Of more relevance to our work is the so-called Renormalization Group (RG) decoder proposed recently by Bravyi and Haah [25]. This decoder was introduced as an efficient decoder for general topological codes and its correction capabilities were studied for the 3D cubic code [70] and the qubit toric code. With perfect measurements, this decoder was shown to achieve a threshold of 6.7% for the qubit toric code. As we will see, one of the remarkable features of this decoder is that it has the construction independent of the qudit dimensions, which makes it very suitable for our investigation, and it will allow us to obtain a numerical estimate of the threshold for any qudit dimension.

This is not the only RG decoder that exists. An earlier RG-type decoder that was used for decoding the toric code is due to Duclos-Cianci and Poulin [48, 49]. This decoder achieves a threshold of 8.2% for the qubit toric code. Very recently, Duclos-Cianci and Poulin [51] have generalised their decoder to the qudit toric code. However, due to the fact that their decoder has a run-time complexity that depends on the dimension as $O(d^7)$, they were not able to obtain thresholds values beyond dimension 6.

Although these can both be considered as a type of RG-decoder, their constructions differ greatly. To distinguish between these decoders, we will follow the distinction given in [50] and we will refer to the first decoder from [25] as the *hard-decisions* RG decoder (HDRG), for reasons that will become clear

[2] Under similar statistical mechanical arguments, the optimal threshold for other noise models, such as the depolarizing noise, has been estimated to be around 10.9% [20].

in due time. Our main interest in this chapter is to investigate the performance of the HDRG decoder in decoding the qudit toric code [31]. A detailed comparison between these decoders can be found in [6].

One of the motivations behind exploring qudit systems is that the outcomes of the stabilizer measurements have the potential to give more information on where the errors have occurred (we will discuss a specific example in due time). If this extra information is exploited correctly, higher thresholds might be obtained than is possible in the qubit case. To our knowledge, such an improvement in the threshold was first observed in [5]. Other studies of the thermal stability of the toric code also indicate some advantages in using qudit systems [161].

Finally, it is also worth pointing out why we did not choose to generalise the MWPMA. In the qubit case, the MWPMA typically consists of constructing a complete weighted graph where the nodes are the non-trivial syndromes and the weight of the edges is the shortest Manhattan distance between the nodes. Then using Edmonds' algorithm [53, 96] the perfect matching of minimum weight can be efficiently determined. The algorithm then finds the maximum-likelihood error for the code. We can think of each node in the qubit code as having charge 1, and hence when two nodes are matched, they are annihilated (since $1 + 1 = 0 \mod 2$). However, in higher dimensions, the charge of the nodes are from the set $\{1, \ldots, d-1\}$. Hence we must consider all the possibilities of matching all neutral sets of *multiple* nodes in order to annihilate them. Thus to find the lowest weight error correction chains for the qudit code requires an algorithm, which must minimize weights on a hypergraph whose hyperedges consist of all charge neutral subsets of nodes. Minimum weight hypergraph matching is an NP-Hard problem in computational complexity theory [87]. We therefore do not expect good speed performance for such a decoder, and have not pursued it here.

4.2 The Qudit Toric Code

4.2.1 Properties

In the toric code, the *physical* qudits are attached to the edges of a square lattice embedded on the surface of a torus, as shown by the lattice in Fig. (4.2a). The toric code is a quantum stabilizer code with two

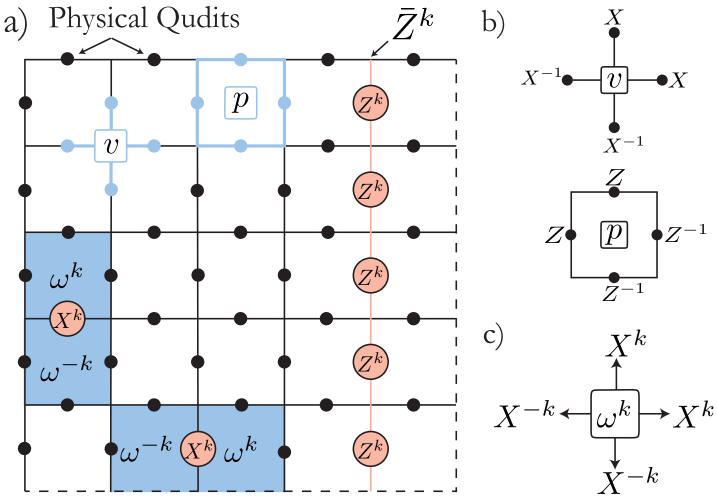

Figure 4.2: a) The toric code lattice with the periodic boundaries indicated by the dashed line. b) The qudit vertex and plaquette operators. c) Transporting a plaquette in the four directions requires applying the indicated X operators.

types of weight-four commuting stabilizer generators, the so-called *vertex* and *plaquette* stabilizers, see Fig. (4.2b). It is easy to see that all these stabilizer generators commute. This code is also known in the literature as the $\mathbb{Z}_d$–Kitaev's code. It is a member of the family of Abelian quantum double models [90].

Notice that a square lattice is self-dual—meaning that when taking the lattice dual the vertices and plaquettes are switched. We will often refer to the vertex and plaquette operators collectively as the *check* operators. For an $L \times L$ lattice there are $n = 2L^2$ physical qudits and $m = L^2$ possible plaquette or vertex operators. Due to the periodic boundary condition, however, *each* plaquette (vertex) can be expressed as the product of the remaining $L^2 - 1$ plaquette (vertex) operators; implying that there are only $n - k = 2(2m - 1)$ check operators that are independent, where k is the number of encoded qudits. Hence, $k = 2$ meaning that a toric code encodes only two qudits.

As can be seen from the definition of the plaquette and vertex operators, the toric code is a CSS code [33, 149] where the stabilizer generators consist either of X^j or Z^j operators. Hence, the vertex operators detect only Z^j errors, and similarly the plaquette operators detect X^j errors. Recall that the possible outcomes of measuring a stabilizer generator are phases from the set $\{1, \omega, \dots, \omega^{d-1}\}$. We

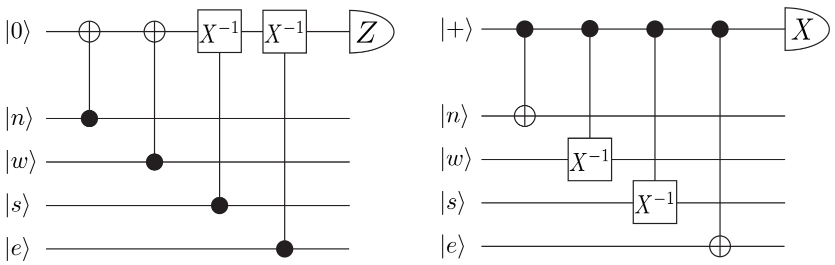

Figure 4.3: The stabilizer measurement circuits for the plaquette (left) and vertex (right) operators. The labels (*n,w,s,e*), refer to the (*north, west, south, east*) physical qudits.

will refer to any non-trivial syndrome as *charged*, with the charge being an element from the set of the non-trivial phases $\{\omega, \omega^2, \ldots, \omega^{d-1}\}$. For simplicity, we can equivalently work with integer powers of the non-trivial phases $\{1, \ldots, d-1\}$.

The stabilizer measurement of the plaquette and vertex operators is performed using the circuit in Fig. (4.3), where an ancillary qudit is measured destructively in the process. Geometrically, we can think of these ancillary qudits to be placed at the centre of the plaquettes and the vertices, with the generalised SUM gate $\Lambda(X^k)$ entangling the nearest-neighbouring qudits. This gate is defined as

$$\Lambda(X^k): \quad |i\rangle\,|j\rangle \mapsto |i\rangle\,|ki \oplus j\rangle\,, \tag{4.1}$$

and in the Heisenberg picture, it maps the Pauli operators as

$$X \otimes \mathbb{1} \quad \mapsto \quad X \otimes X^k, \tag{4.2}$$

$$\mathbb{1} \otimes X \quad \mapsto \quad \mathbb{1} \otimes X, \tag{4.3}$$

$$Z \otimes \mathbb{1} \quad \mapsto \quad Z \otimes \mathbb{1}, \tag{4.4}$$

$$\mathbb{1} \otimes Z \quad \mapsto \quad Z^{-k} \otimes Z. \tag{4.5}$$

All the error detecting and correcting elements of these circuits are assumed to be perfect with the only source of errors being the noise channel on the physical qudits. By considering single physical errors X^j (or Z^j), one can use the above map to track how the errors propagate through these circuits and determine the corresponding measurement outcome.

The topological nature of the code is evident within the structure of the stabilizer group. Every element in the stabilizer corresponds to a product of closed loops of X^j and Z^j operators. It is convenient to adopt some terminology from topology to characterise these operators. Closed loops which enclose a surface are called *homologically trivial.* The logical operators $\bar{X}^j$ and $\bar{Z}^j$ correspond to *non-trivial homological* loops of errors. An example of a $\bar{Z}^j$ is shown in Fig. 4.2(a). Given any chain on the lattice, deformation to any homologically equivalent chain can be achieved by a multiplication by an element of the stabilizer. Hence all homologically equivalent operators correspond to the same logical operator on the encoded qudits. Recall that the distance of a code is equal to the minimum weight of all possible logical operators, and this is easily seen as the error chain of the lattice length L. In the notation of the stabilizer codes, the toric code can then be denoted by $[\![2L^2, 2, L]\!]_d$.

It proves useful to introduce some terminology to describe the relationship between the qudit errors and the detected syndromes. For example, we say that a single X^j error *creates* a pair of plaquettes. Similarly, we say a chain of X^j errors consisting of different weights (i.e. different j values) will creates a *trail* of plaquettes. Notice that, in the qubit case a chain of X errors will still create a pair of plaquettes at the endpoints of the error chain—without providing any information about the actual path of the error chain. But as the qudit dimension increases, the trail reveals more information on where exactly the actual error chain has occurred, see the example provided in Fig. (4.4). It is this additional information that allows us to make better judgement when performing error correction and ultimately leads to higher thresholds than the qubit case.

To correct the errors, we apply correction chains, as instructed by the decoder, such that charged plaquettes (vertices) are *annihilated.* The annihilation can be thought of as a physical process that *transports* and *fuses* the syndromes until the charges cancel. As an example, the transportation rule in Fig. (4.2c) shows which operators need to be applied to transport a plaquette in a certain direction.

There are an exponential (in L) number of different error configurations $\mathbb{E}(E)$, with a total of $d^n = d^{2L^2}$ possible errors of each type. Note that the code is degenerate, meaning that many different error configurations correspond to each syndrome. In fact, given the symmetry of square lattice, one would expect that there are $d^m = d^{L^2}$ possible plaquettes (or vertex) outcomes. But because we are only

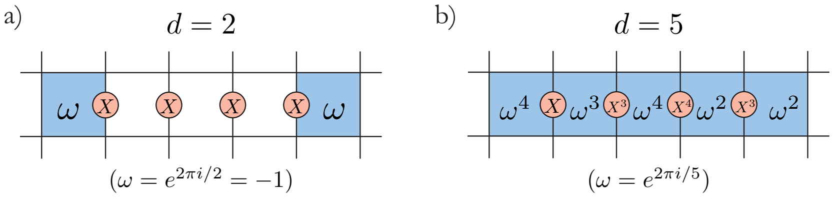

Figure 4.4: An example showing that in higher dimensions the syndrome measurements reveal more information about the path of the errors on the lattice of the toric code. We have chosen two error chain with two different dimension (d = 2 and d = 5) as a demonstration. Fig. a) shows a qubit error chain creating two non-trivial syndromes at its endpoints. Fig. b) shows a random error chains of local dimension five being detected by the syndromes measurement along the path of the chain, thus revealing more information about the actual path of the error in comparison to Fig. a).

considering perfect syndrome measurements, each error configuration must have a *total* charge zero. Hence the number of possible configurations is divided by the number of different charges, which is d. This leads to d^{L^2-1} possible syndrome configurations.

Based on the above simple counting argument, each syndrome can be caused by $d^n/d^{m-1} = d^{L^2+1}$ possible error configurations, which is exponential in the size of the lattice. An optimal decoder will have then to search through an exponential number of errors to find the most probably homological class. For the majority of stabilizer codes, the problem of optimal decoding is known to be an **NP**-hard problem [77, 98].

4.2.2 Noise Model and Monte Carlo Simulation

In this section we discuss the noise model the lattice is subjected to, and explain how we estimate the thresholds of the HDRG decoder.

As stated previously, we will assume that the measurement of the check operators is perfect, and the only source of errors is the noise acting on the physical qudits. Furthermore, the noise model we consider is the independent error model (also called the uncorrelated noise model) where X^j and Z^j physical errors occur independently with equal probability. Therefore, under such a symmetric noise model it suffices to study the error-correction properties for one type of errors[3]. For this reason, in the discussions that follow we will often only discuss X^j error chains and the plaquette operators. It should always be assumed that the same discussion can be applied to the case of Z^j errors chains and vertex operators by working on the dual lattice. More formally, under this noise model, each qudit e suffers an X^j error with the following probabilities:

$$\text{Prob}(X^0) \equiv \text{Prob}(\mathbb{1}) = 1 - p, \quad \text{Prob}(X^j) = p/(d-1), \tag{4.6}$$

for all nonzero $j \in \mathbb{Z}_d$.

We estimate the error correction threshold numerically via a Monte Carlo simulation. For a single Monte Carlo sample, we initiate a lattice in a pure state of the code-space, and generate random error configuration E of the edges of the lattice using the above noise model for physical error rate p. The syndromes of the error configuration are then measured and fed to the decoder. The decoding algorithm will return a correction configuration E' that will annihilate all the observed syndromes. After the correction step is completed, we compute $E'' = E \oplus E'$, and if E'' contains no logical chain we regard the decoder as successful, otherwise the decoder has failed. In the simulation we repeat this procedure N times for a given p, and we evaluate the success probability p_{succ} as the fraction of times the decoder succeeds. The standard deviation in the estimated success probability is $\sigma = \sqrt{p_{\text{succ}}(1 - p_{\text{succ}})/N}$.

To determine the threshold, we plot p versus p_{succ} for different lattice sizes as shown, for example, in Fig. (4.6). The threshold p_{th} is defined to be the *point* at which the success probability curves intersect in the limit $L \to \infty$. In other words, the threshold represents the point below which arbitrarily high p_{succ} can be achieved provided that the lattice is large enough. However, in the actual simulations, the data points can only be obtained for a relatively small lattice sizes L, and such lattices are subject to small system size effects, which can affect the evaluation of p_{th}. This is easily seen in the $L = 16$ curve of Fig. (4.6).

To account for the small system size effects, we estimate p_{th} by using the fitting proposed by Har-

[3] Keeping in mind that a qudit suffering a combination of X^j and Z^k errors simultaneously such as $X^j Z^k$ (i.e. a Y-type error) will be detected by both the adjacent plaquettes and vertices, and hence the decoder will return a independent correction of $X^{j'}$ and $Z^{k'}$, which when combined becomes $X^{j'} Z^{k'}$—thus such combinations of errors are also correctable.

rington *et al.* [162, 71]. In this fitting, *all* the data points (for all that lattice sizes L) are fitted to the curve

$$A + Bx + Cx^2 + DL^{-1/\mu}, \tag{4.7}$$

where $x = (p - p_{th})L^{1/\nu}$, as shown, for example, in the boxed plot in Fig. (4.6). In particular, the last term in the fitting, $DL^{-1/\mu}$, accounts for the small size effects. We can see that, in the limit of $L \to \infty$, this term tends to 0 (where μ is positive). We have used the **NonlinearModelFit** function in Mathematica to estimate the fitting parameters[4] $\{A, B, C, D, p_{\text{th}}, \nu, \mu\}$.

4.3 HDRG Decoder

The HDRG decoder was introduced in [25]. In this section we will present a refined version of this decoder and show how it can achieve higher thresholds for the toric code.

4.3.1 Decoder Description

The HDRG decoder has a simple and elegant intuition behind its construction, and before we introduce it formally we shall give a heuristic description of how it works. The HDRG consists of multiple levels of decoding that will eventually annihilate all the syndromes completely. Each level of decoding is associated with a geometric measure of distance on the square lattice, such that the distance gets bigger as the levels increase. At each level, the syndromes are divided into *disjoint clusters* such that the syndromes in each cluster are separated by (at most) the distance determined by the decoding level. If the charge of a cluster is zero (modulo d), then the syndromes of the clusters are annihilated *locally*. Otherwise, charged clusters are passed to the next higher level until ultimately they become part of a neutral cluster and gets annihilated. Next, we will define and explain all the aspects of this decoder more rigorously.

Let us just work with plaquettes with the analogous vertex formalism being obvious. We will need

[4] In particular we have used the options “BestFitParameters” to extract the parameter estimates, and “ParameterErrors” to estimate the standard deviation error in each parameter.

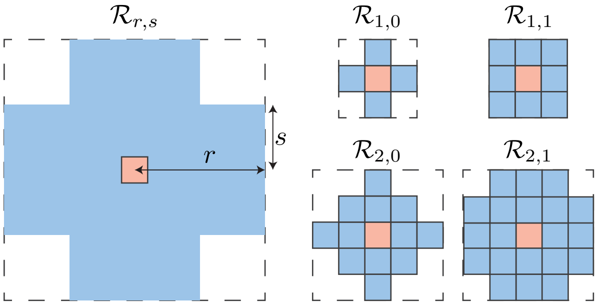

Figure 4.5: The refined regions $\mathcal{R}_{r,s}$ on a taxi-cab geometry (left hand side) with examples of the first four levels (right hand side).

a notion of distance between the plaquettes, and for this purpose start by associating a coordinate vector $\mathbf{x} = (x_1, x_2)$ for each plaquette on the 2D lattice. For any two plaquettes with coordinates $\mathbf{x}$ and $\mathbf{x}'$, we will use two distance measures. First, the taxi-cab distance is $D^{(1)}(\mathbf{x}, \mathbf{x}') = |x_1 - x_1'| + |x_2 - x_2'|$. Our second distance is the Max distance, and it is defined as $D^{(\infty)}(\mathbf{x}, \mathbf{x}') = \max\{|x_1 - x_1'|, |x_2 - x_2'|\}$. Both can be used to define balls like regions B of a certain radius r, centred on a plaquettes $q(\mathbf{x})$, such that

$$B_r^{(1)}(\mathbf{x}) = \{\mathbf{x}' | D^1(\mathbf{x}, \mathbf{x}) \le r\}, \tag{4.8}$$

$$B_r^{(\infty)}(\mathbf{x}) = \{\mathbf{x}' | D^\infty(\mathbf{x}, \mathbf{x}) \le r\}, \tag{4.9}$$

Although we call these balls, the Max distance generates squares and the taxi-cab distance picks out diamonds. Here, however, we will be interested in regions that combine these notions of distance. For any integer r and s, we define region $\mathcal{R}_{r,s}$ that when centred on a point $\mathbf{x}$ are

$$\mathcal{R}_{r,s}(\mathbf{x}) = B_{r+s}^{(1)}(\mathbf{x}) \cap B_r^{(\infty)}(\mathbf{x}), \tag{4.10}$$

and so is simply the intersection of two balls with different metrics. The first few instances are shown in Fig. (4.5). Clearly we only need to consider $s \le r$. Note that the regions are symmetric, so if $\mathbf{x} \in \mathcal{R}_{r,s}(\mathbf{y})$ then $\mathbf{y} \in \mathcal{R}_{r,s}(\mathbf{x})$. When this happens we say $\mathbf{x}$ and $\mathbf{y}$ are (r, s)-connected.

Furthermore, we need a notion of connection for a *cluster* C—or set of syndromes. Firstly, we define connected paths in C. A path γ in C is an ordered subset of C, such that $\gamma = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ...\mathbf{x}^{(n+1)}\}$, and we define it to be (r,s)-path-connected if for all j, $\mathbf{x}^{(j)}$ and $\mathbf{x}^{(j+1)}$ are (r,s)-connected. Now, we say the cluster is (r,s)-cluster-connected if for all $\mathbf{x},\mathbf{y} \in C$ there exists an (r,s)-connected-path in C starting at $\mathbf{x}$ and ending at $\mathbf{y}$. The intuition behind defining regions is this way is to take into account some of the degeneracy in the errors creating the syndromes. We will discuss this point in more detail shortly.

These geometric concepts can be used to explain the decoding scheme. Given measurement data for all the plaquettes, recall that we say a plaquette is charged if it has a non-trivial outcome. If plaquette at $\mathbf{x}$ has measurement outcome $m_{\mathbf{x}}$, then information is conveyed by the ordered pair $(\mathbf{x}, m_{\mathbf{x}})$, and the full list of charged plaquettes is $\mathcal{W} = \{(\mathbf{x}, m_{\mathbf{x}}) | m_{\mathbf{x}} \neq 1)\}$. Similarly a charged cluster is a subset of the full charge distribution, $\mathcal{C} \subset \mathcal{W}$, where we use a different script to indicate the presence of charge information. A charged cluster is said to be *neutral* if the *total* charge is zero, so that $\sum_{\mathcal{C}} m_{\mathbf{x}} = 0$ modulo d. Neutral clusters can always be annihilated by transporting and fusing the syndromes within the cluster until the total charge disappears. When doing so, we update the plaquette information from $\mathcal{W}$ to $\mathcal{W}'$, such that the annihilated neutral cluster $\mathcal{C}$ is no longer contained in $\mathcal{W}'$. In fact, if the size of a cluster is very small in comparison to the lattice size, then it is very likely that the cluster has been generated by local errors within the cluster. The intuition behind the HDRG decoder is that if such small clusters are annihilated locally, then the resultant correction chains, combined with the actual error chain, will form a trivial loop of errors. By trivial loop, we mean a loop which does not wrap around the torus, such operators are stabilizer elements and so equivalent to the identity on the code-space. Topologically, such chains are homologically trivial loops.

The complete set $\mathcal{W}$ can always be partitioned into a set of *disjoint* clusters $\tilde{\mathcal{W}} = \{\mathcal{C}_1, \mathcal{C}_2, ...\mathcal{C}_m\}$, for some m, and where $\mathcal{W} = \mathcal{C}_1 \cup \mathcal{C}_2 \cup \cdots \cup \mathcal{C}_m$. We say a particular partition $\tilde{\mathcal{W}}$ is a (r,s)-partition if both the following conditions are met:

i. *connectivity:* every charged cluster in the partition is (r,s)-cluster-connected;

i. *maximality:* for any distinct pair of charged clusters in the partition, $\mathcal{C}_j$ and $\mathcal{C}_{k \neq j}$, we find that $\mathcal{C}_j \cup \mathcal{C}_k$ is not (r,s)-cluster-connected.

Maximality ensures that there is no suitable path between the disjoint clusters, and so they could not be merged into a single cluster. Furthermore, whenever the connectivity condition is met, but maximality fails, there exists another partition that does fulfil both conditions using fewer charged clusters.

As stated previously, the HDRG decoder involves multiple levels of decoding. Each decoding level l is associated with a choice of regions $\mathcal{R}_{r,s}$, At the first level we begin with $(r, s) = (1, 0)$. The parameters increase iteratively, such that for $l + 1$, first we try to increase s by 1, but if $s = r$ we instead increase r by one and reset s to zero. The relation between the level number and the distance parameters can be determined with simple calculation to be $l = (s(r+1)/2+r)$. The decoder performs the following, beginning with the first level $l = 1$:

1. *Clustering*: Find a (r, s)-partition of $\mathcal{W}_l$ into disjoint charged clusters;

2. *Neutral Annihilation*: For every neutrally charged cluster in the partition, $\mathcal{C}_j$, find an Pauli correction e_j that annihilates all the syndromes by fusing them *arbitrarily* with their nearest neighbours in $\mathcal{C}_j$;

3. *Refresh:* Record the collective Pauli correction $E'_l = \prod_j e_j$ and update the syndrome information to $\mathcal{W}_{l+1}$. If $\mathcal{W}_{l+1}$ is non-empty, then repeat at next level $l = l + 1$.

It is helpful to refer to individual levels of the decoder as sub-protocols that we label $\mathfrak{D}_l$. Any charged cluster that cannot be annihilated completely by $\mathfrak{D}_l$, is therefore left for the next higher level of decoding. The higher levels will have larger regions and therefore any charged clusters will eventually be combined inside bigger neutral clusters which can then be annihilated. Also, notice that in the HDRG construction the correction chains are determined during the annihilation step at every level of decoding. In classical coding theory, this is a typical feature of what is known as a *hard-decisions decoder* [115, 126].

There are few crucial differences between our version of the HDRG decoder described above and the original decoder by Bravyi and Haah [25]. First, the distance measure in [25] is the Max distance $D^{(\infty)}$. Recall that a ball of radius r in this the Max distance is denoted B_r^{∞}, and Bravyi and Haah use such a region at level-r of the decoder. We also use such regions, since $\mathcal{R}_{r,0} = B_r^{\infty}$, but our protocol is more refined and uses additional levels of decoding. Finally, their decoder declares failure and aborts if the area of a cluster is larger than half the lattice size. The idea behind this requirement is that annihilating such large clusters would very likely lead to a logical error. However, in our decoder we did not enforce this requirement because, as we will see, in higher dimensions the syndrome tend to percolate if the physical error probability is high, and we would like to investigate how this decoder behaves in such regimes. Finally, in their decoder, the Max distance scales exponentially with the decoding levels, whereas in our case the scaling is linear.

In the qubit case, for all practical purposes (best case instances) the run-time complexity of the decoder by Bravyi and Haah is $O(L^2 \log L)$, where L is the lattice size [25]. In higher dimensions the number of syndromes increases with the qudit dimension. To understand this behaviour consider the following example. In the qubit case if two neighbouring errors occur then the shared syndrome will not be detected. But in the qudit case, the probability that two neighbouring errors will have equal and opposite weights will diminish quickly as the dimension increases. Hence, the shared syndrome will almost always be detected. The consequence of this observation is that for a given error rate the *density* of the syndromes will approach the density of the errors as the dimension increases. Therefore the exact run-time complexity needs to capture the relation between the number of syndromes and the qudit dimension, which is not a trivial task. But our numerical analysis (omitted here) shows that the speed of HDRG has a very small dependence on the dimension physical qudits, and for most practical purposes it can be safely neglected. Moreover, our refined distance measure $\mathcal{R}_{r,s}$ has linear scaling with the decoding levels. As a consequence the run-time complexity is $O(L^3)$ in the best case, which is slower than that of Bravyi and Haah [25].

4.3.2 Numerical Estimates of the Threshold

In this section we present the results of the Monte Carlo simulation for the HDRG decoder. We begin with the qubit case before moving to higher dimensions. We plot the success probability curves for the qubit case in Fig. 4.6. Using the fitting given in Eq. (4.7) we estimate the threshold to be 8.4% ± 0.01. This value is slightly higher than that of SDRG decoder in [48] of 8.2%.

Recall that the threshold achieved by the original HDRG decoder in [25] was 6.7%. The improvement in the threshold achieved by our HDRG decoder is mainly due the refined set of regions $\mathcal{R}_{r,s}$ we

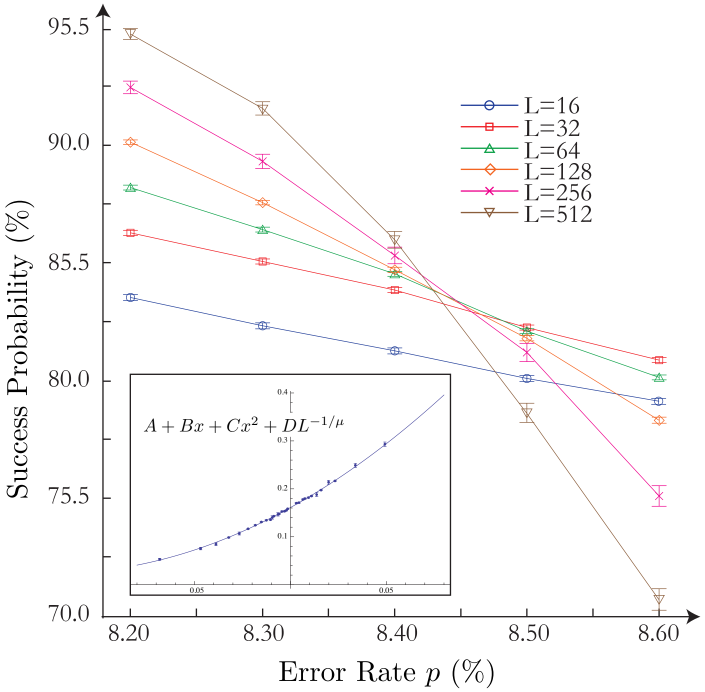

Figure 4.6: The success probability of the HDRG decoder for the qubit case. The data points are generated with 10^5 samples for $L \in \{16, 32, 64, 128\}$ and 10^4 samples for $L \in \{256, 512\}$. The error bars are taken to be 2σ. The boxed plot shows the data fitting, where $x = (p - p_{th})L^{1/\nu}$, $\nu = 1.85 \pm 0.04$ and $\mu = 0.46 \pm 0.06$.

have adopted in comparison to Ref. [25], where regions based on solely the Max distance were considered. To demonstrate this point consider the simple examples depicted in Fig (4.7). First, Fig (4.7a) shows two plaquettes created by one and two errors. Clearly the single error is more likely to occur in comparison to two neighbouring errors. However, with the D^{∞}-metric the plaquettes in both cases will be connected at the first level. In contrast, the regions $\mathcal{R}_{r,s}$ distinguish between the two cases, and they will be connected at two separate decoding levels, namely $\mathfrak{D}_1$ and $\mathfrak{D}_2$. Also, Fig. (4.7b) shows two cases of two plaquettes created by two errors. For the first case, there are two errors for which the set of successful recovery operations are identical. Hence, the first case is more probable since it has double the degeneracy. The regions $\mathcal{R}_{r,s}$ better account for this degeneracy by again treating these cases into two separate levels, namely $\mathfrak{D}_2$ and $\mathfrak{D}_3$. The overall effect of such refinement is to create finer clusters which would lead to better error correction during the annihilation step.

The above observations suggest that to improve our decoder further one can consider a different sequence of regions. An optimal ordering of regions would always first connect syndromes that can be created by fewer errors and higher degeneracy. It is not hard to see that such improvement would switch, for example, level $\mathfrak{D}_5$ with level $\mathfrak{D}_6$, because the latter will connect syndromes created by fewer errors as shown in Fig. (4.7c). Our approach, however, was easier to implement, and we leave such further improvement for a future investigation.

The thresholds of the remaining prime dimensions are plotted in Fig. (4.8). To demonstrate that a numerical estimate of the threshold can be obtained for any dimension we have chosen the 1000th prime number d = 7919 to represent the limit of high d. As can be seen from Fig. (4.8), the threshold increases monotonically with qudit dimension and reach a saturating value of about 18.0%. We have discovered that the reason for this behaviour is due to what we call the "syndrome" percolation effect, which we will discuss next.

It was pointed out in the last section that for a given error rate the density of syndromes increases as the dimension of the qudits increases. In fact, as we will show in the next section, for any given prime dimension $d \geq 3$, there exists a unique threshold error rate at which the syndromes percolate the lattice. In other words, above this threshold the syndromes will span the lattice in a single connected cluster. We

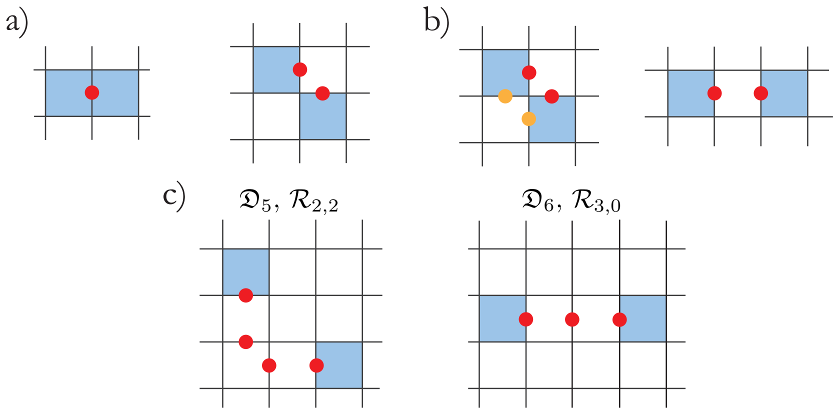
a)
b)
c)
$\mathfrak{D}_5$, $\mathcal{R}_{2,2}$
$\mathfrak{D}_6$, $\mathcal{R}_{3,0}$

Figure 4.7: The distance $\mathcal{R}_{r,s}$ distinguishes between a) and b), whereas the D^∞ distance does not. c) The optimal distance measure will switch levels $\mathfrak{D}_5$ and $\mathfrak{D}_6$.

refer to this threshold as the *syndrome percolation threshold* denoted by $p_{\text{syn}}^{\text{th}}$. We will provide numerical estimates of this threshold in the next section. We will find that it decreases as the dimension increases until it reaches a constant value of about 18.0% in the limit of high d, see Fig. (4.9).

Syndrome percolation has severe consequences for the HDRG decoder. For any error rate $p > p_{\text{synd}}^{\text{th}}$ there will be one percolating neutral cluster at the first level $\mathfrak{D}_1$ of decoding. The HDRG will try to annihilate the syndromes arbitrarily and will most likely fail. This suggests that we cannot expect the HDRG decoder to achieve a threshold higher than the percolation threshold, because the success probability curves must diminish above the percolation threshold. Indeed this is what we observe in the limit of high d, as illustrated by the boxed plot in Fig. (4.8). The point of intersection of the curves (which defines the threshold) intersects x-axis at the value of the percolation threshold. The actual curves (omitted here) are too noisy around the percolation threshold, for this reason we have indicated by the red error bar the range at which the actual curves intersects. Our numerical analysis shows that if we ignore the small lattice sizes, then the curves of the large lattice sizes clearly cross at single point around $18.0\% \pm 0.1$.

The conclusion of the above discussion is that in the limit of high d the syndrome percolation threshold is a tight upper-bound on the threshold achieved by the HDRG decoder.

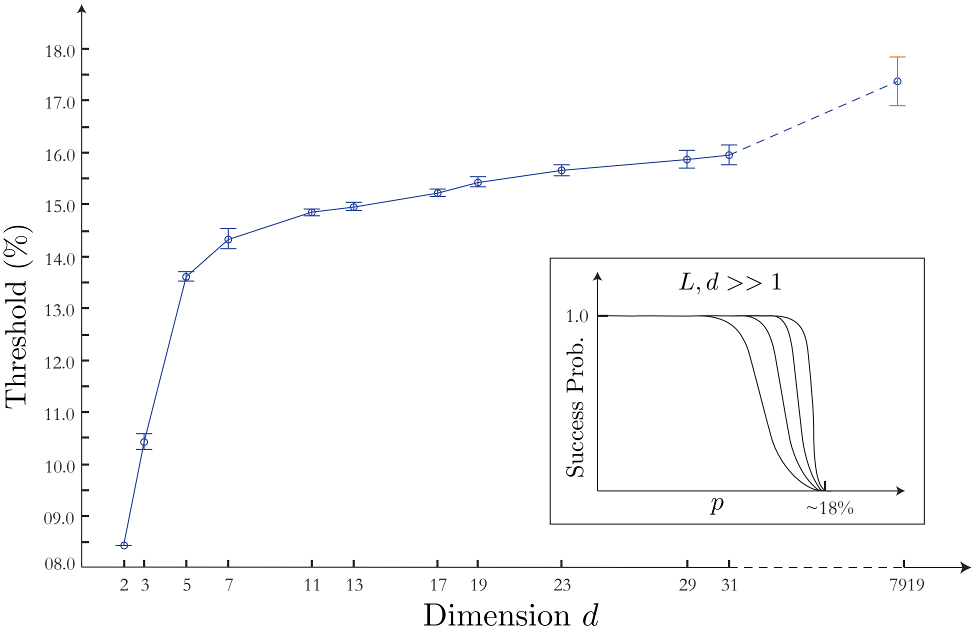

Figure 4.8: The threshold values of the HDRG decoder for prime dimensions with 2σ error bars. The boxed plot is illustrative figure of the behaviour the success probability in the limit of high d.

4.4 Beating the Percolation Upper bound

4.4.1 Syndrome Percolation

Percolation theory is the study of connectivity and transport on random graphs [30, 69, 148]. A standard percolation model consists of a random graph whose nodes (vertices) are distributed in space and the links (edges) connect neighbouring nodes only. We are mainly interested in the percolation behaviour on a 2D regular lattice, and in particular the square lattice. There are typically two stochastic mechanisms associated with each lattice structure: either the vertices of the lattice are fixed in space and edges are created randomly between them, or vertices are random in space and the edges connect neighbouring vertices.

For instance, in the random site (or vertex) model, each site is "empty" with IID probability p and otherwise it is "occupied". For each instance, we say percolation occurs if there is a nearest neighbour path that spans the lattice using only occupied vertices. The key result of percolation theory is that there exists a threshold, $p^{\mathrm{th}}_{\mathrm{site}} = 59.27\%$, above which the probability of percolation approaches unity with increasing lattice size, and below threshold the percolation probability vanishes in the large lattice limit. A similar phenomenon occurs when lattice edges (or bonds) are randomly removed, which has a threshold of $p^{\mathrm{th}}_{\mathrm{bond}} = 50\%$.

On the square lattice of the toric code, the bonds correspond to the qudits and the sites correspond to the vertex/plaquette operators. The bond percolation threshold tells us that above a 50% error rate there will exist (in the thermodynamics limit, $L \to \infty$) a percolating chain of bonds (or edges). With periodic lattice boundaries, such a chain does not necessary correspond to a non-trivial closed loop, and there must exist another slightly higher threshold for the existence of a 'percolating' non-trivial loop, which we denote by $p^{\mathrm{th}}_{\mathrm{loop}}$, such that $p^{\mathrm{th}}_{\mathrm{loop}} \gtrsim p^{\mathrm{th}}_{\mathrm{bond}}$. We identify straight away the threshold $p^{\mathrm{th}}_{\mathrm{loop}}$ to be the error rate above which a logical error is bound to exist, and hence no stabilizer measurement can detect it, and in turn no decoder can correct it. Hence, the threshold $p^{\mathrm{th}}_{\mathrm{loop}}$ represents a hard upper-bound to the optimal threshold for the qudit toric code, and we expect it to be very close to $p^{\mathrm{th}}_{\mathrm{bond}}$.

In our discussion that follows, we will be interested in the *syndrome* percolation threshold of the toric code. This is not equivalent to the site percolation threshold because the syndromes are created in pairs by qudit errors (on the bonds). Given a syndrome $\mathcal{W}$ we say that it percolates the lattice, if there is a nearest neighbour path in $\mathcal{W}$ that spans the lattice. In more general terminology, a nearest neighbour path is a $(1,0)$-connected-path in $\mathcal{W}$. There have been studies of site percolation with distant neighbouring interactions [105, 104], but to our knowledge there have not been investigations where the bonds interact with the sites in the manner defined by the toric code. Also, there does not appear to be an analytic method that can determine the syndrome percolation threshold for any dimension d from the known theory on the bond and site percolation. In the limit of high $d \to \infty$, we can make a crude assumptions to derive an upper bound for the syndrome percolation threshold.

When d is very high, we can safely assume that the only possible situation in which the charge of a plaquette (or a vertex) is zero if all its qudits are error-free. This occurs with a probability of $(1-p)^4$, and hence the probability of a plaquette with a non-trivial charge is $1-(1-p)^4$. If we make the crude assumption that such plaquettes are distributed equally, then we can equate this probability to the site percolation threshold, $p_{\text{site}}^{\text{th}} = 0.5927 = 1-(1-p)^4$. Solving for p, gives a hard upper-bound for the syndrome threshold of $p = 20.11\%$. We expect the actual threshold to be slightly smaller than this value.

To estimate the syndrome percolation threshold for qudit dimension d, we resort to numerical evaluation via a Monte Carlo simulation. The simulation is straightforward and it is very similar to that described in Sec. 4.2.2 in estimating the error correction threshold. For a given dimension d, error rate p, and lattice size L, we generate a qudit lattice such that each qudit suffers an error with probability p. The syndromes are then calculated. If the syndromes percolate (span the lattice) then the simulation will be declared successful, otherwise it is a failure. This procedure is then repeated N times, and the success probability is evaluated as the fraction of times the simulation has succeeded. The simulation is then repeated for a fixed range of p and different lattice sizes. The threshold is determined as the point of intersection of the different success probability curves (omitted here).

The syndrome percolation thresholds obtained are presented in Fig. (4.9). As can be seen from this figure, there does not exist a syndrome percolation threshold for the qubit case. This can be explained to be due to the fact that the number of non-trivial syndromes is completely symmetric about the 50% error

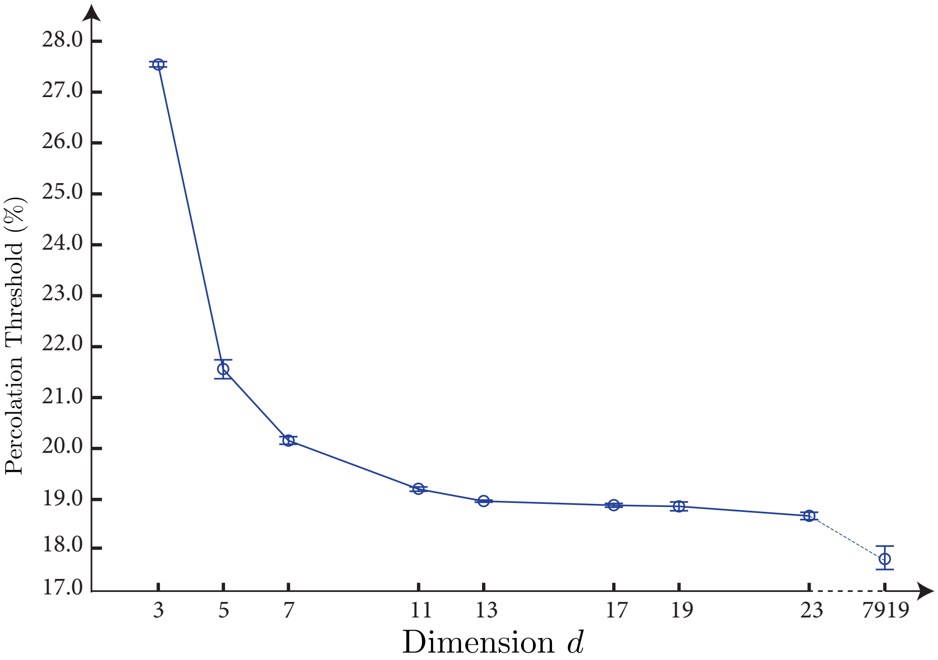

Figure 4.9: Syndrome percolation threshold for prime dimensions with 2σ error bars.

rate. For example, in the limit of large lattice size, the average number of non-trivial syndromes is the same if $p = 0.5 - \alpha$ or $p = 0.5 + \alpha$, where $0 \leq \alpha \leq 0.5$ [5]. This implies that there does not exist a threshold above which the syndromes *always* percolates. However, for the remaining prime dimensions, such symmetry does not exists and we always observe a threshold. We see that the syndrome percolation threshold decreases monotonically with the qudit dimension, and in the limit of high d it reaches a constant value of about 18%. confirms the suggestion of the last section that the syndrome percolation threshold is an upper-bound for the HDRG decoder. In the next section we will show how the HDRG decoder can be enhanced to beat this upper-bound.

[5] This fact can be understood by considering the probability that a plaquette (or a vertex) is non-trivial. In the qubit case, this occurs with probability $4p(1-p)^3 + 4p^3(1-p)$, which is symmetric about $p = 0.5 \pm \alpha$, hence indicating that the profile of the curve of the success probability of syndrome percolation versus the probability p has a bell shape about $p = 0.5$, which prohibits the existence of a unique threshold point.

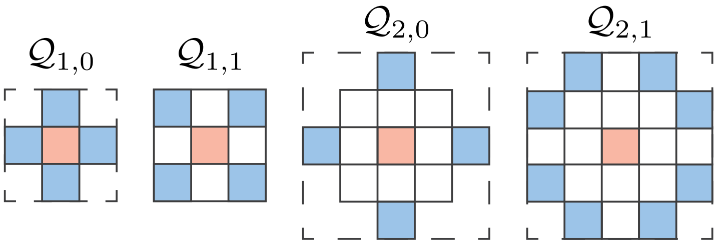

Figure 4.10: The first four levels of $\mathcal{Q}_{r,s}$. The red square is the syndrome s and the blue squares are the syndromes at the outer layer of $\mathcal{R}(r,s)$.

4.4.2 Enhanced HDRG Decoder

To beat the percolation threshold we consider an *initialization step* $\mathcal{I}$ that enhances the performance of the HDRG decoder. This step is not efficient, but we will show that it can boost the threshold to about 30% at a computationally feasible cost. The initialization step is designed to dissect a percolating cluster into a more sparse set of clusters *before* running the HDRG decoder (i.e. before running the first level of decoding $\mathfrak{D}_1$). It achieves this by using a brute force method to find any neutral sub-clusters within the percolating cluster. The sub-clusters are then annihilated before running the HDRG decoder. We have constructed this initialization step to search for the sub-clusters systematically utilizing similar concepts as those used in the HDRG decoder.

More formally, the initialization step consists of multiple levels $\mathcal{N}$ of searches for neutral sub-clusters. Let $\mathcal{Q}_{r,s}$ be the *set* of syndromes at the outer layer of $\mathcal{R}_{r,s}$, such that

$$\mathcal{Q}_{r,s}(\mathbf{x}) = \begin{cases} \text{for } s = 0, \mathcal{R}_{r,s} \smallsetminus \mathcal{R}_{r-1,r-1}, \\ \text{for } s > 0, \mathcal{R}_{r,s} \smallsetminus \mathcal{R}_{r,s-1}, \end{cases} \tag{4.11}$$

where "$A \smallsetminus B$" just means in A but not in B. This is more easily shown by the examples in Fig. (4.10). Then each initialization level $\mathcal{N}_{r,s}$ is associated with sets of syndromes of the form $\mathcal{Q}_{r,s}$. We denote the elements of the set $\mathcal{Q}_{r,s}$ by q_j, and by definition, each set has either 4 or 8 syndromes, see Fig. (4.10).

For any plaquette u and $q_j \in \mathcal{Q}_{r,s}$, we construct a *search rectangle* $\mathcal{T}$ as the minimum rectangle that encloses syndromes u and q_j. In other words, the plaquette u and q_j form the opposite corners of the search rectangle. Inside $\mathcal{T}$, we define a *search-path* τ as any minimum size $(1,0)$-path-connected in $\mathcal{T}$ that starts at u and ends at q_j. There are many such paths, and by construction, they will contain $|\tau| = (r+s+1)$ total syndromes. We denote the set of all possible search-paths in $\mathcal{T}$ by $T = \{\tau_1, \ldots, \tau_{|T|}\}$, where $|T|$ is the total number of possible sub-clusters. Geometrically, in general, if the size of $\mathcal{T}$ is an $A \times B$ rectangle, then $|T| = (A+B)!/A!B!$. This expression was calculated by considering the equivalent problem of finding all the minimum paths between two points on a Manhattan (or taxi-cab) geometry (see [61], page 162).

The aim here is to treat each search-path as a sub-cluster. We then annihilate any neutral sub-clusters at each level of initialization, and the search for the sub-clusters is performed for each syndrome u from the set of all L^2 syndromes of the lattice regardless of whether it is trivial or charged. Based on the above definitions, we now summarise the search routine of an initialization level $\mathcal{L}_{r,s}$ as follows. For each plaquette $u_j \in \mathcal{U}$ (starting with u_1):

1. Choose an element $q_j \in \mathcal{Q}_{r,s}$, and construct a search rectangle $\mathcal{T}$;

2. Search for all possible sub-clusters $\tau_j \in T$ within $\mathcal{T}$ systematically. If any sub-cluster τ_j is found to be neutral, then annihilate τ_j and stop the search. Then start step 1 with the next plaquette $u_{j+1} \in \mathcal{U}$; Else

3. If no neutral sub-cluster were found, choose the next element $q_{j+1} \in \mathcal{Q}_{r,s}$ and repeat steps 1 and 2; Else

4. If there are no remaining syndromes $q_j \in \mathcal{Q}_{r,s}$, then the search has ended without finding a neutral sub-clusters for plaquette u_j. Start step 1 with the next plaquette $u_{j+1} \in \mathcal{U}$.

The above procedure is repeated until all the plaquettes $u_j \in \mathcal{U}$ have been searched. The overhead of this search procedure is proportional to the size of the search rectangle $|T|$, which is factorial in r and s. More precisely, for each initialization level $\mathcal{L}_{r,s}$, in the worst case scenario (where no neutral sub-clusters are found), the search takes αL^2 steps, with the constant overhead $\alpha = (r+s)!/r!s!$. Although that seems to be inefficient (in the depth of search), the parameters r and s increase polynomially with the number of initialization levels, and hence for the first few levels the overhead α is small enough. As a

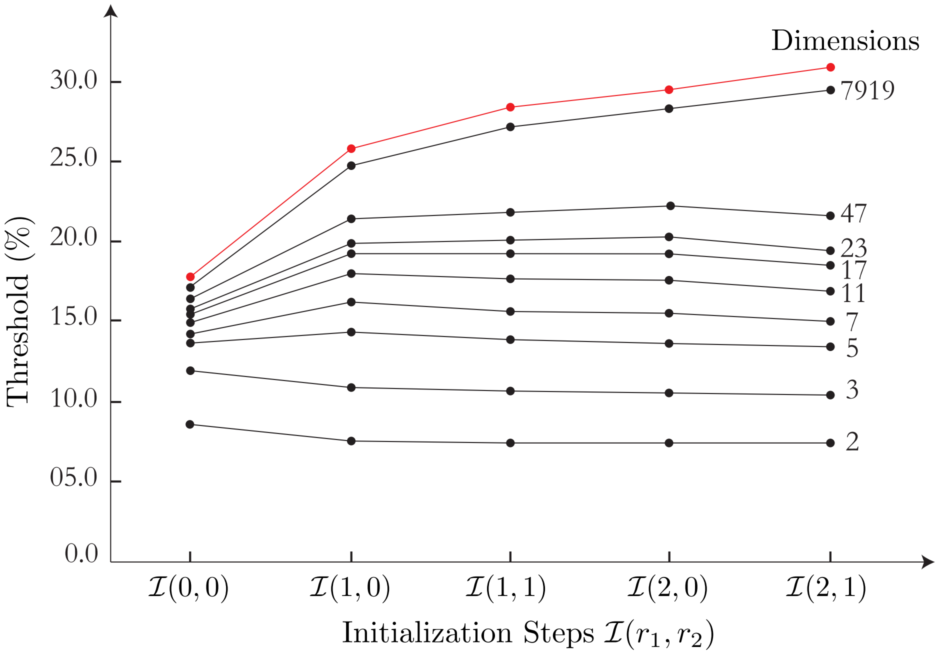

Figure 4.11: The thresholds for the *enhanced*-HDRG decoder with the first four initialization steps $\mathcal{I}(r,s)$. The error bars and data of some prime dimensions are not included for clarity. The red curve is the enhanced-syndrome-percolation threshold in the limit of high d.

result, running the above procedure for the first few initialization levels is still a computationally feasible task. It is important to notice that for each plaquette u_j the procedure stops once a neutral sub-cluster is found, and the worst case of not finding any neutral sub-clusters happens only when the dimension d and the error rate p are sufficiently high. The depth of searching for the neutral sub-clusters increases as the initialization levels increase in size. An initialization step is to *depth* (r,s), denoted by $\mathcal{I}(r,s)$, if it consists of running all the initialization levels in ascending order $\mathcal{I}(r,s) = \{\mathcal{N}_{1,0}, \mathcal{N}_{1,1}, \ldots, \mathcal{N}_{r,s}\}$. We propose an *enhanced*-HDRG decoder with depth (r,s) that runs the initialization step $\mathcal{I}(r,s)$ followed by the HDRG decoder described in Sec. 4.3.

The numerical estimates for the thresholds achieved by the enhanced-HDRG decoder for the first four initialization steps are summarized in Fig. (4.11). The thresholds for $\mathcal{I}(0,0)$ correspond to the

HDRG decoder without any enhancement. For the qubit and qutrit case we see that the thresholds decrease after initialization steps are introduced. This is because for these low dimensions, finding a neutral sub-cluster is very probable, and hence the initialization step is in fact too destructive. As a result the clusters are divided into a very sparse set of smaller clusters, and running the HDRG will end up connecting these sparse sets of clusters and causing more logical errors.

However, we start to observe an improvement in the thresholds above the qutrit case. Notice that for all the first listed prime dimensions ($d = 2, \ldots, 47$), after some initialization step the thresholds start to decrease. This is also because after some depth of searching the initialization step becomes too destructive. In the limit of high d, we see that a threshold just under 30% can be achieved. In addition, note that the shape of the curve indicates a potential increase in threshold with initialization step beyond $\mathcal{I}(2,1)$. We leave such investigations for future work.

Finally, in the limit of high d, the saturating thresholds of the enhanced-HDRG decoder can also be explained by the syndrome percolation effect. We introduce the *enhanced*-syndrome-percolation threshold which is determined by simply running the initialization step $\mathcal{I}(r,s)$ followed by the syndrome percolation simulation described in Sec. 4.4.1. The numerical estimates for the enhanced-syndrome-percolation thresholds are indicated in Fig. 4.11 by the red curve. Our numerical analysis shows that the enhanced-HDRG decoder can reach the upper-bound of the red line by ignoring small size effects and considering large lattice sizes only.

4.5 Summary and Open Problems

In this chapter we have presented an efficient decoder for the qudit toric code based on the HDRG decoder of [25]. This decoder has the convenient feature of having a run-time complexity independent from the dimension of the physical qudits. This has allowed us to obtain numerical estimates of the threshold for any desired dimension. We have shown how the original decoder of [25] can be refined to obtain higher thresholds, at the cost of linear increase of decoding levels. Although this means that our refined version of the the HDRG decoder is slower (by taking many additional decoding levels), such a slow down disappears as the dimension increases. This is because the density of the syndrome increases with increasing dimension, and hence both our refined version and the original HDRG decoder would only require the first few levels of decoding.

Under the IID generalised physical bit-flip noise model, our un-enhanced decoder achieves a threshold increasing with d, saturating around 18% in the limit of high dimension d. We have provided numerical evidence and this threshold is in fact tightly upper-bounded by the syndrome percolation threshold. To beat this upper-bound we have included an initialization step, that can boost the threshold to about 30% with computational feasible run-time overhead. There are many open problems that need to be addressed in a future investigation.

First, in the original HDRG decoder [25] the authors managed to prove the *existence* of a threshold for this decoder. Can we prove the existence of a threshold for our decoder given that the decoding levels increase linearly? In fact, it is not clear that we need this linear scaling for high decoding levels—one could consider the case where for the first few levels of decoding we adopt our refined decoding levels and then switch to the exponential scaling of [25]. The HDRG decoder does not seem to be suitable for correlated noise models. An open question is then, can we modify the distance measure somehow to account for correlated errors?

Finally, we have only considered the case of perfect decoding where we consider errors in the physical qudits only. For future work, we plan to consider a more realistic noise model where each element of the stabilizer measurement circuits in Fig. (4.3) is prone to errors. This means that the outcome of the stabilizer measurements are themselves faulty. For this case, we will have to create a 3D history of the stabilizer syndrome measurements. More precisely, we will have to record the *changes* (difference in outcome measurements modulo d) of the syndromes between time slices. The threshold that will be obtained from this simulation will be the actual fault-tolerant threshold below which qudits can be robustly stored by the qudit toric code.

Chapter 5

Summary and Outlook

Building a scalable quantum computer is one of the major challenges in modern physics. The main obstacle that face any physical realisation of a quantum computer is the problem of quantum decoherence, where due to the delicate nature of quantum systems, any unwanted interaction with the environment can destroy the coherence (or superposition) of the quantum states, causing an instant loss of any useful information stored. Despite this serious problem, over the past two decades the theory of quantum fault-tolerance was developed to combat quantum decoherence and to show how an error-free computation can be accomplished. This was proven by the threshold theorem, which asserts that arbitrary accurate and long computations can be performed in a fault-tolerant manner provided that the rate at which each computational element fails is below a certain threshold.

A typical quantum fault-tolerant scheme consists of two crucial components. First, it must include a fault-tolerant implementation of an error correction code to protect the information during all steps of the computation. Second, it must provide the means to achieve universal quantum computation. Arguably, the most promising fault-tolerant scheme known to date is that introduced by Raussendorf *et al.* [132, 131] for the measurement-based cluster-state architecture [129, 130, 128]—achieving a fault-tolerant threshold of about 1%. The two central components of this fault-tolerant scheme are the topological error correction codes[1] for robust protection against noise and magic state distillation for achieving

[1] Note that although this scheme borrows many techniques from the topological computational model (including, for exam-

quantum universality. Our main focus in this thesis has been to investigate the advantages of using higher dimensional quantum systems (qudits) instead of the conventional two-level quantum systems (qubits), as candidates for fault-tolerant quantum computation. We have chosen the scheme by Raussendorf *et al.* as the testbed in our qudit generalisation. Since the model of measurement-based quantum computation generalises naturally to higher dimensions [172], what remains is the generalisation of magic state distillation protocols and topological error correction, which is our contribution in this thesis.

In our investigation of higher dimensional magic state distillation we have considered two approaches which were independently presented in chapters 2 and 3. In chapter 2, we introduced a numerical approach to study the distillation properties of any qudit stabilizer code [7]. We showed how the generic distillation map of a distillation protocol can be evaluated when only the stabilizers of the code are known. By numerically searching the state space as input states to the distillation map, the distillable magic states (attractors of the distillation map) can be determined. We used this approach to study the distillation capability of the five-qutrit code and identified, for the first time, new families of magic states (the $H-$ and H^2–type magic states) beyond the qubit case. We then showed how such states can be converted into a more suitable form to perform a non-Clifford qutrit gate, thus achieving universality. This approach has the drawback of being inefficient for large codes, but nevertheless, it is very useful when little is known about the nature of magic states that a small code can distil, and thus it was very convenient to us as a first investigation.

In chapter 3, we adopted a more analytic approach by studying the distillation properties of the family of Reed-Muller codes in all prime dimensions [35]. In the qubit case, it was known that Reed-Muller codes have the remarkable property of having a transversal non-Clifford gate. For example the 15-qubit code has a transversal $\pi/8$ gate—a property that plays a central role in its magic state distillation performance [27]. Using this property as a guideline, we developed a generalisation of the $\pi/8$ gate to all prime dimensions. By exploiting the transversality property we were able to employ techniques from classical coding theory that allowed us to analytical evaluate distillation thresholds for various error

ple, the braiding concept to implement logical gates via code deformation [131, 21]), it is not actually topological in nature, i.e. there are no non-Abelian anyons involved in the computation.

models and determine the yield of the protocol. In particular, we found a small five-dimensional code, $\mathcal{QRM}_5(1)$, which has a superior performance in comparison to many qubit protocols. With respect to depolarising noise, it achieves an error thresholds of 36.3%—the highest distillation threshold known to date. In addition, it has a very high yield (γ^* = 2)—outperforming its qubit counterpart by many orders of magnitude.

In chapter 4, we studied the higher dimensional toric code[2] serving as a quantum memory [6]. In our study presented in this thesis, we assumed that the syndrome measurements are noise-free. We chose a very fast hard-decisions renormalization group (HDRG) decoder [25], and we refined its construction to improve its threshold performance. This decoder has a run-time complexity that is almost independent of the physical qudit dimensions of the code—a feature that allowed us to numerically estimate the threshold for any dimension of the qudit toric code. We saw that the thresholds obtained by this decoder increases as the qudit dimension increase, and reach a saturating value of about 18%. We discovered that this behaviour was due to a syndrome percolation effect, such that the percolation thresholds always upper bound the thresholds achieved by the HDRG decoder. To beat this upper bound, we introduced a special procedure (the initialization step) which can disrupt the percolation effect and can boost the threshold to about 30% for a sufficiently high qudit dimensions. In a future work, we plan to extend our investigation by applying our HDRG decoder to the more realistic 2D surface code—which is the same as the toric code, but without the periodic boundaries—in addition to allowing for noisy syndrome measurements. Given a fully realistic noise model, the thresholds that will be obtained in that study will be the actual 'fault-tolerant' threshold below which information can be stored for arbitrary long time[3].

An important problem that needs to be addressed is the physical implementation of qudits. It is commonly known that many current physical realisations of qubits [156] are in fact multi-level quantum systems (with two sets of multi-level regimes that distinguish the two levels of a qubit). In the case of the Raussendorf *et al.* qubit scheme, the computation is carried out on a 2D lattice with only local and nearest-neighbour (ferromagnetic Ising-type) interactions are required. Such a model can be poten-

[2] Also known as the $\mathbb{Z}_d$ Kitaev's code.

[3] Note that because logical gates are performed via code deformation in the scheme of Raussendorf *et al.*, the fault-tolerant threshold for the quantum memory that will be obtained is similar to that for the fault-tolerant computation.

tially realised in experimental setting where short-range interactions are readily available. Promising examples include cold atoms in optical lattices [68, 106, 107, 29, 80], trapped ions [19, 113], photons [102] and solid-state systems (such as quantum dots and superconducting circuits) [155, 164, 22]. For the remaining components of the scheme, small scale experiments have been recently demonstrated for magic state distillation (with NMR quantum processor [146]) and topological error correction (with an eight-photon cluster state [167]). In the case of qudit implementations, the work of Zhou *et al.* [172] showed the Hamiltonian needed to create a qudit cluster states is simply the 2D spin-$\frac{d-1}{2}$ Ising model. Higher spins Ising models are studied both theoretically and experimentally in condensed matter theory [168, 109, 79, 78, 153, 145, 38, 12, 54, 97]. We speculate that exploring such models can provide the key for an experimental realisation of the qudit scheme of Raussendorf *et al.*

Our study in this thesis can be extended in two ways. First, throughout we assumed that the qudit dimension is always a prime number. This was mainly to simplify the arguments by using the structure of finite fields in our construction of quantum codes that were used for magic state distillation. We predict that the generalisation to include dimensions that are prime powers should be straightforward as the same language of finite fields can be used (with some modifications to our definitions in chapter 1 [8, 9]). However, the generalisation for the remaining dimensions could prove to be more technical and would require a more careful investigation. Second, the noise models that we used to evaluate the error thresholds for magic state distillation and the qudit toric code were not physically motivated. Instead we considered the natural generalisation of the qubit depolarizing noise channel and the uncorrelated bit-flip noise model in our studies. In reality, noise models are more likely to be correlated and asymmetric with respect to bit- and phase-flip errors. In order to give a fairer comparison between the noise thresholds of qudit systems ($d > 2$) and those achieved in the qubit case, it would be perhaps more constructive to consider a noise model that is motivated by an experimental implementation of qudit systems. For example, we can consider an experiment implementation that involves a nuclear spin and electron spin system such as the one in [116, 13]. Without going into the experimental details, the system in this experiment is effectively a 20-dimensional qudit. However, a closer look reveals that in such a system the allowed operations are restricted and the noise involved is very specific. For instance, the 20-level system is divided into two regions each containing 10-levels, and within each region only nearest-level transitions are allowed and across the two regions only certain transitions are allowed. If such aspects can be incorporated in our qudit model of computation, then the noise thresholds that will be obtained can be directly compared to those of qubit systems.

It is still far from clear what will be the winning physical implementation in the race to build a quantum computer, and what will be the key feature(s) that will make quantum computers a physical reality one day. We hope that this thesis has offered persuasive theoretical evidence that higher-dimensional systems have some intrinsic properties that give them advantage over two-level systems, and we hope that the our results can serve as a motivation for more intensive investigations considering higher-dimensional systems as building blocks for fault-tolerant quantum architectures.

Bibliography

[1] S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, **70**:052328, (2004).

[2] L.M. Adleman, J. Demarrais, and M.D.A. Huang. Quantum computability. *SIAM J. Comput.*, **26**:1524–1540, (1997).

[3] D. Aharonov and M. Ben-Or. Fault-tolerant quantum computation with constant error. *In Proc. 28th Ann. ACM Sympo. Theo. Comp. (STOC'97)*, pages 176–188, (1997).

[4] J.T. Anderson. On the power of reusable magic states. *arXiv:1205.0289*, (2012).

[5] I. Andriyanova, D. Maurice, and J.-P. Tillich. New constructions of CSS codes obtained by moving to higher alphabet. *arXiv:1202.3338*, (2012).

[6] H. Anwar, B. Brown, E.T. Campbell, and D.E. Browne. Efficient decoders for the qudit toric code. *arXiv:1311.4895*, (2013).

[7] H. Anwar, E.T. Campbell, and D.E. Browne. Qutrit magic state distillation. *New J. Phys.*, **14**:063006, (2012).

[8] D.M. Appleby. SIC-POVMs and the extended Clifford group. *J. Math. Phys*, **46**:052107, (2005).

[9] D.M. Appleby. Properties of the extended Clifford group with applications to SIC-POVMs and MUBs. *arXiv:0909.5233*, (2009).

[10] E.F. Assmus and J.D. Key. *Designs and their Codes*. Cambridge University Press, (1994).

[11] D. Bacon. Operator quantum error-correcting subsystems for self-correcting quantum memories. *Phys. Rev. A*, **73**:012340, (2006).

[12] L. Bahmad, A. Benyoussef, and H. Ez-Zahraouy. Order-disorder layering transitions of a spin-1 Ising model in a variable crystal field. *J. Magnetism and Magnetic Materials*, **251**:115–121, (2002).

[13] S.J. Balian, M.B.A. Kunze, M.H. Mohammady, G.W. Morley, W.M. Witzel, C.W.M. Kay, and T.S. Monteiro. Measuring central-spin interaction with a spin bath by pulsed endor: Towards suppression of spin diffusion decoherence. *Phys. Rev. B*, **86**:104428, (2012).

[14] A. Barenco. A universal two-bit gate for quantum computation. *Proc. R. Soc. London*, **449**:679–683, (1995).

[15] A. Barenco, C.H. Bennett, R. Cleve, D.P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J.A. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, **52**:3457–3467, (1995).

[16] I. Bengtsson, S. Weis, and K. Zyczkowski. Geometry of the set of mixed quantum states: An apophatic approach. *arXiv:1112.2347*, (2011).

[17] C.H. Bennett, H.J. Bernstein, S. Popescu, and B. Schumacher. Concentrating partial entanglement by local operations. *Phys. Rev. A*, **53**:2046–2052, (1996).

[18] C.H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W.K. Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Phys. Rev. Lett.*, **70**:1895–1899, (1993).

[19] R. Blatt and D. Wineland. Entangled states of trapped atomic ions. *Nature*, **453**:1008–1015, (2008).

[20] H. Bombin, R.S. Andrist, M. Ohzeki, H.G. Katzgraber, and M.A. Martin-Delgado. Strong resilience of topological codes to depolarization. *Phys. Rev. X*, **2**:021004, (2012).

[21] H. Bombin and M.A. Martin-Delgado. Quantum measurements and gates by code deformation. *J. Phys. A*, **42**:095302, (2009).

[22] M. Borhani and D. Loss. Cluster states from heisenberg interaction. *Phys. Rev. A*, **71**:034308, (2005).

[23] L.J. Boya and K. Dixit. Geometry of density matrix states. *Phys. Rev. A*, **78**:042108, (2008).

[24] P.O. Boykin, T. Mor, M. Pulver, V. Roychowdhury, and F. Vatan. On universal and fault-tolerant quantum computing: a novel basis and a new constructive proof of universality for Shor's basis. *In Proc. IEEE: 40th Ann. Sympo. Found. Comp. Sci. (FOCS'99)*, pages 486–494, (1999).

[25] S. Bravyi and J. Haah. Analytic and numerical demonstration of quantum self-correction in the 3D cubic code. *arxiv:1112.3252*, (2011).

[26] S. Bravyi and J. Haah. Magic-state distillation with low overhead. *Phys. Rev. A*, **86**:052329, (2012).

[27] S. Bravyi and A. Kitaev. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Phys. Rev. A*, **71**:022316, (2005).

[28] S.B. Bravyi and A.Y. Kitaev. Quantum codes on a lattice with boundary. *arXiv:quant-ph/9811052*, (1998).

[29] H.J. Briegel and R. Raussendorf. Persistent entanglement in arrays of interacting particles. *Phys. Rev. Lett.*, **86**:910–913, (2001).

[30] S.R. Broadbent and J.M. Hammersley. Percolation processes I. Crystals and mazes. *Math. Proc. Cambridge Philos. Soc.*, **53**:629–645, (1957).

[31] S.S. Bullock and G.K. Brennen. Qudit surface codes and gauge theory with finite cyclic groups. *J. Phys. A*, **40**:3481–3505, (2007).

[32] A.R. Calderbank, E.M. Rains, P.W. Shor, and N.J.A. Sloane. Quantum error correction and orthogonal geometry. *Phys. Rev. Lett.*, **78**:405–408, (1997).

[33] A.R. Calderbank and P.W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, **54**:1098–1105, (1996).

[34] E.T. Campbell. Catalysis and activation of magic states in fault-tolerant architectures. *Phys. Rev. A*, **83**:032317, (2011).

[35] E.T. Campbell, H. Anwar, and D.E. Browne. Magic-state distillation in all prime dimensions using quantum Reed-Muller codes. *Phys. Rev. X*, **2**:041021, (2012).

[36] E.T. Campbell and D.E. Browne. On the structure of protocols for magic state distillation. *In Proc. Theo. Quant. Comput. Comm. Crypt. 4th Workshop (TQC'09)*, page 20, (2009).

[37] E.T. Campbell and D.E. Browne. Bound states for magic state distillation in fault-tolerant quantum computation. *Phys. Rev. Lett.*, **104**:030503, (2010).

[38] Y. Canpolat, A. Torgürsül, and H. Polat. The magnetic properties of spin-1/2 and spin-1 Ising models in an applied magnetic field by introducing the effective-field approximation. *Phys. Scr.*, **76**:597–605, (2007).

[39] J.A. Smolin C.H. Bennett, D.P. DiVincenzo and W.K. Wootters. Mixed-state entanglement and quantum error correction. *Phys. Rev. A*, **54**:3824–3851, (1996).

[40] S. Popescu B. Schumacher J.A. Smolin C.H. Bennett, G. Brassard and W.K. Wootters. Purification of noisy entanglement and faithful teleportation via noisy channels. *Phys Rev Lett.*, **76**:722–725, (1996).

[41] H.F. Chau. Five quantum register error correction code for higher spin systems. *Phys. Rev. A*, **55**:R839–R841, (1997).

[42] X. Chen, H. Chung, A.W. Cross, B. Zeng, and I.L. Chuang. Subsystem stabilizer codes cannot have a universal set of transversal gates for even one encoded qudit. *Phys. Rev. A*, **78**:012353, (2008).

[43] S. Clark. Valence bond solid formalism for d-level one-way quantum computation. *J. Phys. A*, **39**:2701–2721, (2006).

[44] S.L.A. de Queiroz. Location and properties of the multicritical point in the gaussian and $\pm J$ ising spin glasses. *Phys. Rev. B*, **79**:174408, (2009).

[45] P. Delsarte, J.M. Goethals, and F.J. Mac Williams. On generalized Reed-Muller codes and their relatives. *Inf. Control*, **16**:403–442, (1970).

[46] E. Dennis. Toward fault-tolerant quantum computation without concatenation. *Phys. Rev. A*, **63**:052314, (2001).

[47] I. Devetak and A. Winter. Distillation of secret key and entanglement from quantum states. *Proc. R. Soc. A*, **461**:207–235, (2005).

[48] G. Duclos-Cianci and D. Poulin. Fast decoders for topological quantum codes. *Phys. Rev. Lett.*, **104**:050504, (2010).

[49] G. Duclos-Cianci and D. Poulin. A renormalization group decoding algorithm for topological quantum codes. *IEEE ITW*, page 1, (2010).

[50] G. Duclos-Cianci and D. Poulin. Fault-tolerant renormalization group decoder for abelian topological codes. *arXiv:1304.6100*, (2013).

[51] G. Duclos-Cianci and D. Poulin. Kitaev's $\mathbb{Z}_d$-codes threshold estimates. *arXiv:1302.3638*, (2013).

[52] B. Eastin and E. Knill. Restrictions on transversal encoded quantum gate sets. *Phys. Rev. Lett.*, **102**:110502, (2009).

[53] J. Edmonds. Paths, trees, and flowers. *Can. J. Math.*, **17**:449–467, (1965).

[54] R.S. Ellis, P.T. Otto, and H. Touchette. Analysis of phase transitions in the mean-field blumeemerygriffiths model. *Ann. Appl. Probab.*, **15**:1591–2254, (2005).

[55] A.G. Fowler, S.J. Devitt, and C. Jones. Surface code implementation of block code state distillatio. *Sci. Rep.*, **3**:1939, (2013).

[56] A.G. Fowler and K. Goyal. Topological cluster state quantum computing. *Quantum Inf. Comput.*, **9**:0721, (2009).

[57] A.G. Fowler, A.M. Stephens, and P. Groszkowski. High-threshold universal quantum computation on the surface code. *Phys. Rev. A*, **80**:052312, (2009).

[58] A.G. Fowler, A.C. Whiteside, and L.C.L. Hollenberg. Towards practical classical processing for the surface code: Timing analysis. *Phys. Rev. A*, **86**:042313, (2012).

[59] K.M. Svore G. Duclos-Cianci. A state distillation protocol to implement arbitrary single-qubit rotations. *arXiv:1210.1980*, (2012).

[60] F. Gaitan. *Quantum Error Correction and Fault Tolerant Quantum Computing*. CRC Press, (2008).

[61] M. Gardner. *The Last Recreations*. Springer-Verlag New York Inc., (1997).

[62] D. Gottesman. Class of quantum error-correcting codes saturating the quantum hamming bound. *Phys. Rev. A*, **54**:1862–1868, (1996).

[63] D. Gottesman. Stabilizer codes and quantum error correction (phd thesis). *arXiv:quant-ph/9705052*, (1997).

[64] D. Gottesman. Theory of fault-tolerant quantum computation. *Phys. Rev. A*, **57**:127–137, (1998).

[65] D. Gottesman. Fault-tolerant quantum computation with higher-dimensional systems. *Chaos Solitons Fractals*, **10**:1749–1758, (1999).

[66] D. Gottesman and I.L. Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, **402**:390–393, (1999).

[67] S.K. Goyal, B.N. Simon, R. Singh, and S. Simon. Geometry of the generalized Bloch sphere for qutrit. *arXiv:1111.4427v1*, (2011).

[68] M. Greiner, O. Mandel, T.W. Hänsch, and I. Bloch. Collapse and revival of the matter wave field of a Bose-Einstein condensate. *Nature*, **419**:51–54, (2002).

[69] G. Grimmett. *Percolation*. Springer, (1989).

[70] J. Haah. Local stabilizer codes in three dimensions without string logical operators. *Phys. Rev. A*, **83**:042330, (2011).

[71] J.W. Harrington. Analysis of quantum error-correcting codes: symplectic lattice codes and toric codes. *PhD Thesis*, (2004).

[72] P. Heijnen and R. Pellikaan. Generalized hamming weights of $q-$ary Reed-Muller codes. *IEEE Trans. Inf. Theory*, **44**:181–196, (1998).

[73] A. Honecker, M. Picco, and P. Pujol. Universality class of the Nishimori point in the 2D $\pm J$ random-bond Ising model. *Phys. Rev. Lett.*, **87**:047201, (2001).

[74] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki. Quantum entanglement. *Rev. Mod. Phys.*, **81**:865–942, (2009).

[75] E. Hostens, J. Dehaene, and B.D. Moor. Stabilizer states and clifford operations for systems of arbitrary dimensions and modular arithmetic. *Phys. Rev. A*, **71**:042315, (2005).

[76] M. Howard and J. Vala. Qudit versions of the qubit $\pi/8$ gate. *Phys. Rev. A*, **86**:022316, (2012).

[77] P. Iyer and D. Poulin. Hardness of decoding quantum stabilizer codes. *arXiv:1310.3235*, (2013).

[78] N.Sh. Izmailian. A spin-3/2 Ising model on a square lattice. *J. Exp. Theor. Phys. Lett.*, **63**:290–295, (1996).

[79] N.Sh. Izmailian and N.S. Ananikian. General spin-3/2 Ising model in a honeycomb lattice: Exactly solvable case. *Phys. Rev. B*, **50**:6829–6832, (1994).

[80] D. Jaksch, H.J. Briegel, J.I. Cirac, C.W. Gardiner, and P. Zoller. Entanglement of atoms via cold controlled collisions. *Phys. Rev. Lett.*, **82**:1975–1978, (1999).

[81] T. Jochym-O'Connor, Y. Yu, B. Helou, and R. Laflamme. The robustness of magic state distillation against errors in Clifford gates. *Quant. Inf. Comput.*, **13**:361–378, (2013).

[82] C. Jones. Composite Toffoli gate with two-round error detection. *Phys. Rev. A*, **87**:052334, (2013).

[83] C. Jones. Distillation protocols for Fourier states in quantum computing. *arXiv:1303.3066*, (2013).

[84] C. Jones. Low-overhead constructions for the fault-tolerant Toffoli gate. *Phys. Rev. A*, **87**:022328, (2013).

[85] C. Jones. Multilevel distillation of magic states for quantum computing. *Phys. Rev. A*, **87**:042305, (2013).

[86] N.S. Jones and N. Linden. Parts of quantum states. *Phys. Rev. A*, **71**:012324, (2005).

[87] R.M. Karp. Reducibility among combinatorial problems. *Complexity Computer Computations, Plenum Press*, pages 85–103, (1972).

[88] T. Kasami, Shu Lin, and W. Peterson. New generalizations of the Reed-Muller codes–I: Primitive codes. *IEEE Trans. Inf. Theory*, **14**:189–199, (1968).

[89] A.Y. Kitaev. Quantum computations: algorithms and error correction. *Russ. Math. Surv.*, **52**:1191–1249, (1997).

[90] A.Y. Kitaev. Fault-tolerant quantum computation by Anyons. *Ann. of Phys.*, **303**:2–30, (2003).

[91] E. Knill. Fault-tolerant postselected quantum computation: Schemes. *arXiv:quant-ph/0402171*, (2004).

[92] E. Knill. Fault-tolerant postselected quantum computation: Threshold analysis. *arXiv:quant-ph/0404104*, (2004).

[93] E. Knill. Quantum computing with realistically noisy devices. *Nature*, **434**:39–44, (2005).

[94] E. Knill, R. Laflamme, and W. Zurek. Threshold accuracy for quantum computation. *arXiv:quant-ph/9610011*, (1996).

[95] E. Knill, R. Laflamme, and W.H. Zurek. Resilient quantum computation: error models and thresholds. *Proc. R. Soc. Lond. A*, **454**:365–384, (1998).

[96] V. Kolmogorov. Blossom V: a new implementation of a minimum cost perfect matching algorithm. *Math. Prog. Comp.*, **1**:43–67, (2009).

[97] J.J. Krebs, P. Lubitz, A. Chaiken, and G.A. Prinz. Magnetic resonance determination of the antiferromagnetic coupling of Fe layers through Cr. *Phys. Rev. Lett.*, **63**:1645–1648, (1989).

[98] K.-Y. Kuo and C.-C. Lu. On the hardnesses of several quantum decoding problems. *arXiv:1306.5173*, (2013).

[99] P. Kurzynski. Multi-bloch vector representation of the qutrit. *arXiv:0912.3155v1*, (2009).

[100] R. Laflamme, C. Miquel, J.P. Paz, and W.H. Zurek. Perfect quantum error correcting code. *Phys. Rev. Lett.*, **77**:198–201, (1996).

[101] A.J. Landahl and C. Cesare. Complex instruction set computing architecture for performing accurate quantum Z rotations with less magic. *arXiv:1302.3240*, (2013).

[102] D. Leibfried, E. Knill, S. Seidelin, J. Britton, R.B. Blakestad, J. Chiaverini, D.B. Hume, W.M. Itano, J.D. Jost, C. Langer, R. Ozeri, R. Reichle, and D.J. Wineland. Creation of a six-atom Schrödinger cat state. *Nature*, **438**:639–642, (2005).

[103] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, (1977).

[104] M. Majewski and K. Malarz. Square lattice site percolation thresholds for complex neighbourhoods. *Acta Phys. Pol.*, **B38**:2191, (2007).

[105] K. Malarz and S. Galam. Square-lattice site percolation at increasing ranges of neighbor bonds. *Phys. Rev. E*, **71**:016125, (2005).

[106] O. Mandel, M. Greiner, A. Widera, T. Rom, T.W. Hänsch, and I. Bloch. Coherent transport of neutral atoms in spin-dependent optical lattice potentials. *Phys. Rev. Lett.*, **91**:010407, (2003).

[107] O. Mandel, M. Greiner, A. Widera, T. Rom, T.W. Hänsch, and I. Bloch. Controlled collisions for multi-particle entanglement of optically trapped atoms. *Nature*, **425**:937–940, (2003).

[108] A. Mari and J. Eisert. Positive Wigner functions render classical simulation of quantum computation efficient. *Phys. Rev. Lett.*, **109**:230503, (2012).

[109] V. Matveev and R. Shrock. Zeros of the partition function for higher-spin 2D Ising models. *J. Phys. A: Math. Gen.*, **28**:L533, (1995).

[110] A.M. Meier, B. Eastin, and E. Knill. Magic-state distillation with the four-qubit code. *Quant. Inf. Comput.*, **13**:195–209, (2013).

[111] F. Merz and J.T. Chalker. Two-dimensional random-bond Ising model, free fermions, and the network model. *Phys. Rev. B*, **65**:054425, (2002).

[112] D.A. Meyer M.H. Freedman. Projective plane and planar quantum codes. *Foun. Comp. Math.*, **1**:325–332, (2001).

[113] D.L. Moehring, P. Maunz, S. Olmschenk, K.C. Younge, D.N. Matsukevich, L.-M. Duan, and C. Monroe. Entanglement of single-atom quantum bits at a distance. *Nature*, **449**:68–71, (2007).

[114] C. Monroe, D.M. Meekhof, B.E. King, W.M. Itano, and D.J. Wineland. Demonstration of a fundamental quantum logic gate. *Phys. Rev. Lett.*, **75**:4714–4717, (1995).

[115] J.C. Moreira and P.G. Farrell. *Essentials of Error-Control Coding*. Wiley, (2006).

[116] G.W. Morley, P. Lueders, M.H Mohammady, S.J. Balian, G. Aeppli, C.W.M. Kay, W.M. Witzel, G. Jeschke, and T.S. Monteiro. Quantum control of hybrid nuclear-electronic qubits. *Nature Materials*, **12**:103–107, (2013).

[117] D.E. Muller. Application of boolean algebra to switching circuit design and to error detection. *IRE Trans. Elect. Comput.*, **3**:6–12, (1954).

[118] G. Nebe, E.M. Rains, and N.J.A. Sloane. The invariants of the Clifford groups. *Des. Codes and Cryptogr.*, **24**:99–122, (2001).

[119] G. Nebe, E.M. Rains, and N.J.A. Sloane. *Self-Dual Codes and Invariant Theory*. Springer (Berlin), (2006).

[120] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, (2000).

[121] M. Ohzeki. Locations of multicritical points for spin glasses on regular lattices. *Phys. Rev. E*, **79**:021129, (2009).

[122] A. Paetznick and B.W. Reichardt. Universal fault-tolerant quantum computation with only transversal gates and error correction. *arXiv:1304.3709*, (2013).

[123] M. Plenio and S. Virmani. Upper bounds on fault tolerance thresholds of noisy Clifford-based quantum computers. *New J. Phys.*, **12**:033012, (2010).

[124] M.B. Plenio and S. Virmani. An introduction to entanglement measures. *Quant. Inf. Comput.*, **7**:1–51, (2007).

[125] J. Preskill. Reliable quantum computers. *Proc. R. Soc. A*, **454**:385–410, (1998).

[126] J. Proakis and M. Salehi. *Digital Communications*. McGraw-Hill Higher Education, (2000).

[127] N. Ratanje and S. Virmani. Generalised state spaces and non-locality in fault tolerant quantum computing schemes. *Phys. Rev. A*, **83**:032309, (2011).

[128] R. Raussendorf. Measurement-based quantum computation with cluster states. *Int. J. Quant. Inf.*, **07**:1053, (2009).

[129] R. Raussendorf and H.J. Briegel. A one-way quantum computer. *Phys. Rev. Lett.*, **86**:5188–5191, (2001).

[130] R. Raussendorf, D. Browne, and H. Briegel. The one-way quantum computer–a non-network model of quantum computation. *J. Mod. Optics*, **49**:1299–1306, (2002).

[131] R. Raussendorf and J. Harrington. Fault-tolerant quantum computation with high threshold in two dimensions. *Phys. Rev. Lett.*, **98**:190504, (2007).

[132] R. Raussendorf, J. Harrington, and K. Goyal. A fault-tolerant one-way quantum computer. *Ann. of Phys.*, **321**:2242–2270, (2006).

[133] R. Raussendorf, J. Harrington, and K. Goyal. Topological fault-tolerance in cluster state quantum computation. *New J. Phys.*, **9**:199, (2007).

[134] I.S. Reed. A class of multiple-error-correcting codes and the decoding scheme. *Trans. IRE Prof. Group Inf. Theo.*, **4**:38–49, (1954).

[135] I.S. Reed and G. Solomon. Polynomial codes over certain finite fields. *J. Soc. Indust. Appl. Math.*, **8**:300–304, (1960).

[136] B.W. Reichardt. Quantum universality from magic states distillation applied to CSS codes. *Quantum Info. Processing*, **4**:251–264, (2005).

[137] B.W. Reichardt. Quantum universality by state distillation. *Quant. Inf. Comput.*, **9**:1030–1052, (2006).

[138] J.M. Renes, F. Dupuis, and R. Renner. Efficient quantum polar coding. *Phys. Rev. Lett.*, **109**:050504, (2012).

[139] E. Rieffel and W. Polak. *Quantum Computing*. The MIT Press, (2011).

[140] P.K. Sarvepalli and A. Klappenecker. Nonbinary quantum Reed-Muller codes. *Intl. Symp. Inform. Theory (Adelaide, Australia)*, pages 1023–1027, (2005).

[141] B. Schumacher. Quantum coding. *Phys. Rev. A*, **51**:2738–2747, (1995).

[142] Y. Shi. Both Toffoli and controlled-NOT need little help to do universal quantum computation. *Quant. Inf. Comput.*, **3**:84–92, (2003).

[143] P.W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, **52**:R2493–R2496, (1995).

[144] P.W. Shor. Fault-tolerant quantum computation. *In Proc. 37th Conference on Found. Comp. Sci.*, pages 56–65, (1996).

[145] J. Sivardiére and M. Blume. Dipolar and quadrupolar ordering in S=3/2 Ising systems. *Phys. Rev. B*, **5**:1126–1134, (1972).

[146] A.M. Souza, J. Zhang, C.A. Ryan, and R. Laflamme. Experimental magic state distillation for fault-tolerant quantum computing. *Nature Communications*, **2**:169, (2011).

[147] F.M. Spedalieri and V.P. Roychowdhury. Latency in local, two-dimensional, fault-tolerant quantum computing. *Quant. Inf. Comp.*, **9**:666, (2009).

[148] D. Stauffer and A. Aharony. *Introduction to Percolation Theory*. CRC Press, (1994).

[149] A. Steane. Multiple-particle interference and quantum error correction. *Proc. R. Soc. Lond. A*, **452**:2551–2577, (1996).

[150] A.M. Steane. Error correcting codes in quantum theory. *Phys. Rev. Lett.*, **77**:793–797, (1996).

[151] A.M. Steane. Quantum Reed-Muller codes. *IEEE Trans. Info. Theo.*, **45**:1701–1703, (1999).

[152] A.M. Stephens and Z.W.E. Evans. Accuracy threshold for concatenated error detection in one dimension. *Phys. Rev. A*, **80**:022313, (2009).

[153] M. Suzuki. Relationship between d-dimensional quantal spin systems and (d+1)-dimensional Ising systems. *Prog. Theor. Phys.*, **56**:1454–1469, (1976).

[154] K.M. Svore, D.P. DiVincenzo, and B.M. Terhal. Noise threshold for a fault-tolerant two-dimensional lattice architecture. *Quant. Inf. Comput.*, **7**:297–318, (2007).

[155] T. Tanamoto, Y. x. Liu, S. Fujita, X. Hu, and F. Nori. Producing cluster states in charge qubits and flux qubits. *Phys. Rev. Lett.*, **97**:230501, (2006).

[156] R. Hughes *et al.* A quantum information science and technology roadmap part 1: Quantum computation. *http://qist.lanl.gov/*, **v2.0**, (2004).

[157] W. van Dam and M. Howard. Tight noise thresholds for quantum computation with perfect stabilizer operations. *Phys. Rev. Lett.*, **103**:170504, (2009).

[158] W. van Dam and M. Howard. Noise thresholds for higher-dimensional systems using the discrete Wigner function. *Phys. Rev. A*, **83**:032310, (2011).

[159] V. Veitch, C. Ferrie, D. Gross, and J. Emerson. Negative quasi-probability as a resource for quantum computation. *New J. Phys.*, **14**:113011, (2012).

[160] V. Veitch, S.A.H. Mousavian, D. Gottesman, and J. Emerson. The resource theory of stabilizer computation. *arXiv:1307.7171*, (2013).

[161] O. Viyuela, A. Rivas, and M.A. Martin-Delgado. Generalized toric codes coupled to thermal baths. *New J. Phys.*, **14**:033044, (2012).

[162] C. Wang, J. Harrington, and J. Preskill. Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory. *Ann. of Phys.*, **303**:31–58, (2003).

[163] D.S. Wang, A.G. Fowler, A.M. Stephens, and L.C.L. Hollenberg. Threshold error rates for the toric and surface codes. *Quant. Inf. Comput.*, **10**:456, (2010).

[164] Y.S. Weinstein, C.S. Hellberg, and J. Levy. Quantum-dot cluster-state computing with encoded qubits. *Phys. Rev. A*, **72**:020304(R), (2005).

[165] E. Weldon. New generalizations of the Reed-Muller codes–II: Nonprimitive codes. *IEEE Trans. Inf. Theory*, **14**:199–205, (1968).

[166] N.S. Yanofsky and M.A. Mannucci. *Quantum Computing For Computer Scientists*. Cambridge University Press, (2008).

[167] X.C. Yao, T.X. Wang, H.Z. Chen, W.B. Gao, A.G. Fowler, R. Raussendorf, Z.B. Chen, N.L. Liu, C.Y. Lu, Y.J. Deng, Y.A. Chen, and J.W. Pan. Experimental demonstration of topological error correction. *Nature*, **482**:489–494, (2012).

[168] J.M. Yeomans. *Statistical Mechanics of Phase Transitions*. Oxford University Press, (1992).

[169] C. Zalka. Threshold estimate for fault tolerant quantum computation. *arXiv:quant-ph/9612028*, (1997).

[170] B. Zeng, Andrew Cross, and I.L. Chuang. Transversality versus universality for additive quantum codes. *IEEE Trans. Inf. Theo.*, **57**:6272–6284, (2011).

[171] L. Zhang and I. Fuss. Quantum Reed-Muller codes. *arXiv:quant-ph/9703045v1*, (1997).

[172] D.L. Zhou, B. Zeng, Z. Xu, and C.P. Sun. Quantum computation based on d-level cluster state. *Phys. Rev. A*, **68**:062303, (2003).

[173] X. Zhou, D.W. Leung, and I.L.Chuang. Methodology for quantum logic gate construction. *Phys. Rev. A*, **62**:052316, (2000).