THE GEOMETRIC PROPERTIES OF QUANTUM CODES

by
OUSSAMA AMIR

Submitted to the Graduate School of Natural Sciences
in partial fulfillment of
the requirements for the degree of Master of Science

Sabancı University
July 2023

ABSTRACT

THE GEOMETRIC PROPERTIES OF QUANTUM CODES

OUSSAMA AMIR

Mathematics M.S. THESIS, JULY 2023

Thesis Supervisor: Prof. Michel Lavrauw

Keywords: Quantum Codes, Linear codes, Stabilizer Codes

Quantum stabilizer codes are a family of quantum codes that embed qubits in vector spaces based on the error channel that is affecting these qubits. Such codes make use of many classical mathematical theories like spectral theory and group theory. Lately, a new theory emerged where linear codes were used to study stabilizer codes which open the door to the use of finite geometrical tools and interpretations, introducing a new perspective to the problem at hand. This manuscript aims to survey the theory behind the use of geometry in finding stabilizer codes and determining their minimal distances using geometrical arguments. We also introduce some constructions, where we showcase the feasibility of extending an existing code while preserving the minimal distance. We also introduce a new argument to construct minimal distance 2 stabilizer codes in arbitrary dimensions. The work in this manuscript extends naturally to the endeavor of finding new generalized constructions that have monotonic minimal distance.

ÖZET

KUANTUM KODLARININ GEOMETRIK ÖZELLIKLERI

OUSSAMA AMIR

Matematik YÜKSEK LİSANS TEZİ, MAYIS 2023

Tez Danışmanı: Prof. Dr. Michel Lavrauw

Anahtar Kelimeler: Kuantum Kodları, Doğrusal kodlar, Sabitleyici Kodları

Kuantum sabitleyici kodları, bu kübitleri etkileyen hata kanalına bağlı olarak vektör uzaylarına kübitleri gömen bir kuantum kodları ailesidir. Bu tür kodlar, spektral teori ve grup teorisi gibi birçok klasik matematiksel teoriden yararlanır. Son zamanlarda, doğrusal kodların, sonlu geometrik araçların ve yorumların kullanımına kapı açan ve eldeki soruna yeni bir bakış açısı getiren dengeleyici kodları incelemek için kullanıldığı yeni bir teori ortaya çıktı. Bu makale, geometrik argümanları kullanarak stabilizatör kodlarını bulmada ve minimum mesafelerini belirlemede geometrinin kullanılmasının arkasındaki teoriyi incelemeyi amaçlamaktadır. Ayrıca, minimum mesafeyi korurken mevcut bir kodu genişletmenin fizibilitesini sergilediğimiz bazı yapılar da sunuyoruz. Ayrıca, rastgele boyutlarda minimum mesafe 2 sabitleyici kodları oluşturmak için yeni bir argüman sunuyoruz. Bu el yazmasındaki çalışma, doğal olarak, monotonik minimum mesafeye sahip yeni genelleştirilmiş yapılar bulma çabasına uzanmaktadır.

ACKNOWLEDGEMENTS

I am extremely thankful of the help and support of everyone that contributed to this achievement: Family, Professors, Friends,...

"Mathematics is what is, what ought to be, and what can't be."
Oussama Amir

TABLE OF CONTENTS

1. INTRODUCTION

Throughout the past century, the field of coding theory has developed to satisfy the demand for reliable information storage and transmission. The foundation for modern coding theory, which includes quantum coding theory, was laid by classical coding theory, which was created in the middle of the 20th century. This is a brief historical account, starting with classical codes and ending with quantum codes.

Classical coding theory has its roots in Claude Shannon's introduction of information theory in the 1940s. Error-correcting codes, which are able to identify and fix errors in data transmission, were built on Shannon's work. (Shannon (1948)) In the 1950s, Richard Hamming was one of the pioneering scientists to create error-correcting codes. The Hamming codes, which bear his name, can fix single-bit data errors. (Hamming (1950))

Reed-Solomon codes were first used in the 1960s. These codes were used to fix issues with magnetic tape storage, satellite communication, and other digital communication systems.(Reed & Solomon (1960)) Convolutional codes were also a novel class of codes created in the 1970s. Convolutional codes, which can rectify multiple-bit errors, are built on the idea of convolution. NASA launched the Voyager space probes in the 1970s where they employed convolutional coding. (Viterbi (1971))

A new class of codes known as turbo codes emerged in the 1980s. Turbo codes have a high error correction performance and are based on iterative decoding techniques. The International Telecommunications Union (ITU) standardized turbo codes in the early 2000s after Claude Berrou first proposed them in the late 1980s. Digital television, mobile communication, satellite communication, and other types of communication systems all use turbo codes. (Berrou, Glavieux & Thitimajshima (1993))

Robert Gallager first suggested Low-Density Parity-Check (LDPC) codes in the 1960s. Unfortunately, the widespread adoption of LDPC codes in digital communication systems did not occur until the 1990s. LDPC codes offer great error correction performance and are based on the idea of parity-check matrices. Digital television, mobile communication, satellite communication, and other communication technologies all employ LDPC codes. (Gallager (1962))

In the area of coding theory, quantum error-correcting codes are a relatively recent innovation. They are intended to minimize errors brought on by noise and other sorts of interference. The development of quantum error-correcting codes began in the early 1990s when physicists and computer scientists started looking at the characteristics of quantum information and the difficulties associated with properly transmitting and storing it.

A quantum algorithm for factoring big numbers was presented in a ground-breaking study written by AT&T Bell Labs mathematician Peter Shor in 1995. This algorithm showed how quantum computing can be used to tackle issues that traditional computers were unable to handle. The technique, however, also highlighted a significant obstacle to quantum computing: the fragility of quantum states. (Shor (1997))

In traditional computing, errors can be fixed by adding redundancy to the data or by utilizing error-correcting codes that do so. The use of conventional error-correcting codes is, however, impossible in quantum computing since the act of measuring (observing) a quantum state has the potential to destroy it. As a result, quantum error-correcting codes were created.

A work by Oxford University physicist Andrew Steane from 1996 introduced the first quantum error-correcting code. The Steane algorithm uses nine qubits (quantum bits) to safeguard one qubit of data against mistakes. Using an error detection and repair method, the Steane code operates by encapsulating the qubit of information in a larger group of qubits. The development of quantum error-correcting codes advanced significantly in the years that followed. They created new, more effective, and reliable codes and investigated the fundamental quantum physics concept of quantum entanglement, which enables the exchange and manipulation of quantum information. (Steane (1998))

Currently, physics, mathematics, and computer science researchers are still working

on quantum error-correcting codes. Researchers are exploring new uses for quantum information, such as quantum communication and quantum cryptography, and are creating new codes that are more effective and reliable. Despite these developments, it is still extremely difficult to create useful quantum error-correcting codes. Large-scale quantum error-correcting codes are challenging to construct due to the fragility of quantum states and the complexity of quantum systems and the low number of qubits that is currently achievable. Quantum error-correcting codes, however, might be crucial in enabling the creation of real-world quantum computing systems in the future with further study and development.

This thesis aims to study a type of quantum error-correcting codes namely quantum stabilizer codes. Stabilizer codes were introduced in the late 1990s by Gottesman based on the previous work of Knill, Laflamme, et al. Many papers were written studying these codes algebraically but it wasn't until recently that some finite geometrical relations were discovered and this study is meant to dive into these geometrical aspects. (Gottesman (1997))

1.1 Basic Notions

In classical computing, one would have bits as a base layer which we use to store information that is then transformed and manipulated to achieve some computations. In quantum computing, we use qubits (quantum bits) that are fundamentally different from classical bits. We start by some notations and definition as found in Nielsen & Chuang (2010). A typical qubit is referred to as a vector in the Hilbert space $\mathbb{C}^2$:

$$\alpha = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \alpha_0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \alpha_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

We then define a new way to write these vectors called the braket notation. This notation simplifies our formulas and makes them more visually appealing.
In the braket notation, a column vector α is referred to as a ket:

$$|\alpha\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$$

And the conjugate transpose of the vector α is referred to as a bra:

$$\langle\alpha| = \begin{bmatrix} \bar{\alpha_0}, \bar{\alpha_1} \end{bmatrix}$$

Moreover, the inner product of two vectors α and β is written as:

$$\langle\alpha, \beta\rangle = \langle\alpha|\beta\rangle = \begin{bmatrix} \bar{\alpha_0}, \bar{\alpha_1} \end{bmatrix} . \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \bar{\alpha_0}\beta_0 + \bar{\alpha_1}\beta_1$$

From now on, we will refrain from using column and row matrices to refer to the vectors in our study, and rather use the braket notation. Hence, a vector α will be written as:

$$|\alpha\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$$

A qubit can be seen in both states $|0\rangle, |1\rangle$ simultaneously. Therefore, we say that the qubit is in a state of superposition. Furthermore, the qubit is found in state $|0\rangle$ with probability $\alpha_0\bar{\alpha_0}$, and similarly can be found in the state $|1\rangle$ with probability $\alpha_1\bar{\alpha_1}$. Being in a state refers to the outcome of measuring the qubit which we shall explain further in this introduction. The two probabilities presented should add up to 1:

$$\alpha_0\bar{\alpha_0} + \alpha_1\bar{\alpha_1} = 1$$

Where, $\{|0\rangle, |1\rangle\}$ is an orthonormal basis for $\mathbb{C}^2$.

Therefore, $\langle\alpha|\alpha\rangle$ should be equal to 1. Hence, qubits are represented by unit vectors of $\mathbb{C}^2$. The operations that allow us to manipulate these qubits in order to achieve a certain computation are unitary transformations (or unitary gates). A unitary transformation U of $\mathbb{C}^2$, is a non-singular matrix such that:

$$\forall |\alpha\rangle, |\beta\rangle \in \mathbb{C}^2, \quad \langle U\alpha|U\beta\rangle = \langle\alpha|\beta\rangle$$

Hermitian operators are also useful in the endeavor of studying qubits and qubit evolution. Hermitian operators are generally important and widely used in the context of quantum physics. We start by defining the Hermitian conjugate $M^\dagger$ of the linear operator M which is a linear operator satisfying $\langle M\alpha|\beta\rangle = \langle\alpha|M^\dagger\beta\rangle$, $\forall |\alpha\rangle, |\beta\rangle \in \mathbb{C}^2$. Then, M is Hermitian if and only if $M = M^\dagger$.
We finally define the trace of an operator which will often be handy in our studies.

let $\mathcal{B}$ be an orthonormal basis, then

$$tr(M) = \sum_{|\psi\rangle \in \mathcal{B}} \langle \psi | M\psi \rangle$$

Remark. *The map tr() is independent of the choice of* $\mathcal{B}$.

1.1.1 Pauli Operators

Some of the integral operators in studying quantum error-correcting codes are the Pauli operators which are defined as follows:

$$I = \sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, X = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

$$Z = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, Y = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

Remark. *Some of the properties of the Pauli operators are:*

- *The Pauli matrices are Unitary, Hermitian linear transformations of* $\mathbb{C}^2$*, they also form a basis of* $\mathcal{M}_{2\times 2}(\mathbb{C})$.
- *Moreover, the Pauli matrices are orthogonal w.r.t the Hilbert-Schmidt inner product* $\langle A, B \rangle = tr(A^\dagger B)$

1.1.2 Measurements and Density operators

A measurement is represented with a hermitian operator whose eigenvectors are an orthogonal basis of our Hilbert space. Furthermore, these eigenvectors span all possible states of the qubits. These eigenvectors also yield projectors Π_i of the corresponding eigenspace.
Let $|\alpha\rangle = \sum_i p_i |\phi_i\rangle$, we denote the expectation value of A as:

$$\langle \hat{A} \rangle = \langle \alpha | A\alpha \rangle = \sum_i p_i \langle \phi_i | A\phi_i \rangle = tr(A |\alpha\rangle \langle \alpha|) = tr(A \sum_i p_i |\phi_i\rangle \langle \phi_i|)$$

We denote the operator $\rho = \sum_i p_i |\phi_i\rangle \langle\phi_i|$ as the density operator of the quantum system at hand. Note that, ρ is equal to $\rho^\dagger$, the trace of ρ is equal to 1, and $\langle\psi|\rho\psi\rangle \geq 0$ for all $|\psi\rangle$.

Remark. *Here are some of the properties of density operators:*

- *A density operator is a positive semi-definite operator with a trace equal to 1.*
- *For each measurement that can be defined, the probability distribution over the different outcomes of that measurement can be computed from the density operator using the projectors to the eigenspaces:* $p_i = tr(\Pi_i \rho)$ *(*$\pi_i = |\phi_i\rangle \langle\phi_i|$*)*

Hence, density operators can be considered as generalisation of vectors that give a more descriptive view of the states of a qubit.

1.2 A multiple qubit system

In order to deal with multiple qubit systems, we use the tensor product of individual Hilbert spaces of each qubit. In other words, a system of n qubits lies in the n-fold tensor product space:

$$(\mathbb{C}^2)^{\otimes n} = \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$$

Furthermore, the time evolution of an isolated qubit is given by a unitary operator:

$$|\alpha\rangle \longmapsto U(t) |\alpha\rangle$$

Hence, on a closed quantum system of n qubits, the time evolution is given by unitary operators acting on $\mathcal{H}_{system} = (\mathbb{C}^2)^{\otimes n}$ ($2n \times 2n$ matrices). In case our system interacts with its environment (Not a closed system) the unitaries act on a larger system:

$$\mathcal{H}_{system} \otimes \mathcal{H}_{environment}$$

One can also use the Kauss decomposition to describe the evolution of a multiple qubit system, where:

$$|\alpha\rangle \longmapsto \sum_i K_i |\alpha\rangle \langle\alpha| K_i^\dagger, \quad \sum_i K_i^\dagger K_i = I$$

The operators K_i's are known as the Kauss operators.

The map $|\alpha\rangle \longmapsto \sum_i K_i |\alpha\rangle \langle\alpha| K_i^\dagger$ is known as a quantum channel or a completely positive map and it represents the most general form of the evolution of a qubit system. Hence, we can use the Kauss decomposition to describe noise affecting a system. Noise can be characterized as any error in the transmission of the qubit system as we can see in the following remark.

Remark. *In classical information theory a fundamental error is simply a bit-flip. In the case of quantum information theory any non-identity unitary transformation or non-identity quantum channel is considered an error. Furthermore, a noisy channel can be written as* $\xi(-) = \sum_i K_i(-)K_i^\dagger$. *We often abuse notation and refer to the set of operators* $\{K_i\}_i$ *as* ξ *too.*

Note that we can decompose any unitary or quantum channel in terms of a matrix basis. Hence, all errors can be decomposed as the linear combination of the elements of a basis of $\mathcal{M}_{2n\times 2n}(\mathbb{C})$. A good candidate for the basis would be a generalization of the Pauli matrices.
Let's consider all tensors of the Pauli matrices alongside the phases: $\pm 1, \pm i$. This gives rise to the Pauli group $\mathcal{P}_n$. The Pauli group is a non-abelian group consisting of the 4^n tensor products of $\{\sigma_0, \sigma_x, \sigma_z, \sigma_y\}$. After considering the four phases, the size of $\mathcal{P}_n$ becomes 4^{n+1}.

Now, we move to present a more comprehensive and rigorous framework that explains quantum error correction. This framework requires a precise definition of what error correction is and a characterization of what errors can be correctable in a certain code. We discuss these topics and more in the following section.

1.3 Quantum error-correction

We start this section by defining what a classical error-correcting code is.

Definition 1.3.1. Given a finite set A called the alphabet, a classical code is a subset of A^n, where n is the length of the code.

Example 1 (Repetition code)**.**

$$a \in A, \quad a \longmapsto (a, a, \ldots, a) \qquad \text{(a n-times)}$$

For n=3:

$$0 \longmapsto 000 \quad , \quad 1 \longmapsto 111$$

In this case, one can correct up to one error by taking a majority decision. For instance, 101 can be corrected to 111 since there are two 1's and only one zero. Therefore, the majority is 1.

Unfortunately, no such quantum code can be found such that:

$$|\alpha\rangle \longmapsto |\alpha\rangle \otimes |\alpha\rangle \otimes |\alpha\rangle$$

because of the following theorem:

Theorem 1.3.1. *There is no linear map which maps* $|\alpha\rangle$ *to* $|\alpha\rangle \otimes |\alpha\rangle$, $\forall |\alpha\rangle \in (\mathbb{C}^2)^{\otimes n}$

Proof. Suppose not, then $\forall |\alpha\rangle, |\beta\rangle$:

$$|\alpha\rangle \longrightarrow |\alpha\rangle \otimes |\alpha\rangle$$

$$|\beta\rangle \longrightarrow |\beta\rangle \otimes |\beta\rangle$$

but,

$$|\alpha\rangle + |\beta\rangle \longrightarrow (|\alpha\rangle + |\beta\rangle) \otimes (|\alpha\rangle + |\beta\rangle) \neq |\alpha\rangle \otimes |\alpha\rangle + |\beta\rangle \otimes |\beta\rangle$$

Hence, the map is not linear □

Now we define what quantum error-correction means.

Definition 1.3.2. A quantum error-correcting code is a linear subspace Q of $(\mathbb{C})^{\otimes n}$ into which a number of logical qubits can be encoded such that all errors of a certain type can be detected and/or corrected.

Now, given a noisy channel ξ, is there a recovery channel $\mathcal{R}$, such that every density matrix ρ, for which $\xi(\rho)$ in Q can be mapped back to ρ? In other words,

$$\forall \rho, \text{ (density matrix)} \qquad \mathcal{R} \circ \xi(\rho) = a\rho$$

And hence, ρ can be recovered by normalizing $\mathcal{R} \circ \xi(\rho)$.

It is often the case that classical error-correcting codes are used as inspiration to develop quantum codes. Instead of repeating the actual qubit as was done earlier in the classical case, we introduce redundancy to the basis elements.

Repetition type code:

Let $|iii\rangle$ stand for $|i\rangle \otimes |i\rangle \otimes |i\rangle$ for $i = 0, 1$, we then introduce the map:

$$\alpha_0 |0\rangle + \alpha_1 |1\rangle \longrightarrow \alpha_0 |000\rangle + \alpha_1 |111\rangle$$

In this case, one can correct a bit flip (i.e. σ_x) by taking a majority decision.

$$(\sigma_x \otimes \sigma_0 \otimes \sigma_0)(\alpha_0 |000\rangle + \alpha_1 |111\rangle) = \alpha_0 |100\rangle + \alpha_1 |011\rangle \xrightarrow{\text{majority decision}} \alpha_0 |000\rangle + \alpha_1 |111\rangle$$

On the other hand, we can't correct a phase error (i.e. σ_z).

$$(\sigma_z \otimes \sigma_0 \otimes \sigma_0)(\alpha_0 |000\rangle + \alpha_1 |111\rangle) = \alpha_0 |000\rangle - \alpha_1 |111\rangle \xrightarrow{\text{majority decision}} \alpha_0 |000\rangle - \alpha_1 |111\rangle$$

Shor's code was the first code to deal with the sign error.

Example 2 (Shor's code)**.** The coding space for the Shor's code is $(\mathbb{C}^2)^{\otimes 9}$ and a qubit is encoded as:

$$|\alpha\rangle \longrightarrow |\alpha_L\rangle$$

Where:

$$|0_L\rangle = (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle)$$

$$|1_L\rangle = (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle)$$

By linearity:

$$\alpha_0 |0\rangle + \alpha_1 |1\rangle \longrightarrow \alpha_0 |0_L\rangle + \alpha_1 |1_L\rangle$$

This code ensures that we can correct bit flips (i.e σ_x) by taking majority decisions on $|iii\rangle$ for i in {0,1}, and we can fix sign errors (i.e σ_y) by taking majority decisions on the signs inside these tensors (i.e. $|000\rangle \pm |111\rangle$). Therefore, we can also correct the error σ_z since it is equal to $i\sigma_y\sigma_x$.

After these examples, one can already see some of the difficulties facing quantum error-correction. Some of the problems when transmitting an (Unknown) quantum system over a noisy channel are; measurement disturbance, meaning a qubit system automatically updates the system. Hence, when using measurement to detect errors, the system is altered. The set of errors is continuous and not discrete as in the classical case. Unknown quantum states cannot be copied, which prevents us from introducing redundancy as in the classical repetition code. One can mitigate these problems by choosing the measurements such that they stabilize the set of quantum states that consist of the code. Hence, the codes remain unchanged while errors might still change after a measurement (The change is reversible). The

linearity property of density operators allows all errors to be corrected in the span of a finite number of errors to be corrected. The encoded quantum information is distributed amongst many systems and is therefore away from noisy channels. Hence, no copying is needed.

Before we start presenting some mathematical theorems that outline the conditions to obtain quantum error-correcting codes, we first remind the reader of some basic facts of Linear Algebra.

- Let Q be a subspace of $(\mathbb{C}^2)^{\otimes n}$ then, $Q^{\perp}$ is Q's orthogonal subspace with respect to the standard inner products.
- Any vector $|\psi\rangle$ can be written uniquely as the sum of a vector $P|\psi\rangle \in Q$ and $P^{\perp}|\psi\rangle \in Q^{\perp}$.
- The map $|\psi\rangle \longmapsto P|\psi\rangle$ is called the orthogonal projection onto Q.

Lemma 1.3.2. *If $\{|\psi_1\rangle, \ldots, |\psi_k\rangle\}$ is an orthonormal basis for Q then,*

$$P = \sum_{i=1}^{k} |\psi_i\rangle \langle\psi_i|$$

Lemma 1.3.3. *If P is a linear Hermitian operator for which $P^2 = P$ and which image is Q. Then, P is the orthogonal projection onto Q*

We also introduce some notation that will help us formulate the characterization theorem of correctable errors as introduced by Knill, Laflamme & Viola (2000).
Let, $\mathcal{N}(.) = \sum_{\mu} E\mu(.)E_{\mu}^{\dagger}$, where: $\sum_{\mu} E_{\mu}^{\dagger}E_{\mu} = I$ be a quantum channel. Given the channel $\mathcal{N}$, with a code Q, is there a recovery channel $\mathcal{R}$, such that:

$$\mathcal{R} \circ \mathcal{N}(\rho) = \rho, \quad \forall \rho = \sum_{i} p_i |\phi_i\rangle \langle\phi_i|$$

where: $|\phi_i\rangle \in Q$
Let $\xi = \{E_{\mu}\}_{\mu}$ be a set of errors, the fidelity of a ξ-error correcting code $(Q, \mathcal{R})$ is determined by the fidelity of the composition $\mathcal{R} \circ \xi$ restricted on Q. The fidelity of the ξ-error correcting code $(Q, \mathcal{R})$ is defined as follows:

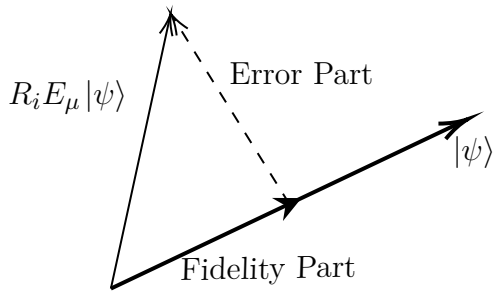
$$F(Q, \mathcal{R}\xi) = \min_{|\psi\rangle \in Q} F(|\psi\rangle, \mathcal{R}\xi) = \min_{|\psi\rangle \in Q} \sum_{r,\mu} |\langle\psi| R_r E_{\mu} |\psi\rangle|^2$$

In the case when $\sum_{\mu} E_{\mu}^{\dagger}E_{\mu} \neq I$, meaning that the errors are not normalized. The

fidelity above becomes not normalized. Then, we define the error of the code:

$$E(Q,\mathcal{R}\xi) = \max_{|\psi\rangle\in Q} \sum_{r,\mu} |(R_r E_\mu - (\langle\psi| R_r E_\mu |\psi\rangle))|\psi\rangle|^2$$

We illustrate the meanning of the fidelity and the error with following diagram:



The pair $(Q,\mathcal{R})$ is an ξ-error correcting code if $E(Q,\mathcal{R}\xi) = 0$. We have:

$$E(Q,\mathcal{R}\xi) = 0 \iff E(Q,\mathcal{R}E_\mu) = 0, \forall E_\mu \in \xi$$

This theorem then presents a condition for an error to be correctable.

Theorem 1.3.4. *(Knill et al. (2000))*

$(Q,\mathcal{R})$ *is an* ξ*-error correcting code if and only if when restricted to* Q*:*

$$R_r E_\mu = \lambda_{r\mu} I, \quad \forall R_r \in \mathcal{R}, \forall E_\mu \in \xi$$

Proof. Let $(Q,\mathcal{R})$ be an ξ-error correcting code. Then,

$$E(Q,\mathcal{R}E_\mu) = |(R_r E_\mu - (\langle\psi| R_r E_\mu |\psi\rangle))|\psi\rangle| = 0, \forall E_\mu \in \xi$$

Let $\lambda_{r\mu}^{|\psi\rangle} = \langle\psi| R_r E_\mu |\psi\rangle$. Then, $\forall |\psi\rangle \in Q, \quad \lambda_{r\mu}^{|\psi\rangle} |\psi\rangle = R_r E_\mu |\psi\rangle$

Here $\lambda_{r\mu}^{|\psi\rangle}$ means that the constant $\lambda_{r\mu}$ depends on $|\psi\rangle$. We then prove that this constant is independent of the choice of the $|\psi\rangle$. All the vectors in Q are eigenvectors of $R_r E_\mu$, which means that Q is embedded in an eigenspace of $R_r E_\mu$. Hence, $\lambda_{r\mu}$ is constant for all vectors $|\psi\rangle$. The inverse of the implication is trivial. □

Theorem 1.3.5. *(Knill et al. (2000))*

Let Q *be a subspace of* $(\mathbb{C}^2)^{\otimes n}$*. The channel* $\mathcal{N}(.) = \sum_\mu E\mu(.)E_\mu^\dagger$ *can be corrected by*

a code Q if and only if:

$$\forall |\phi\rangle, |\psi\rangle \in Q, \ \ \forall E_\mu, E_\sigma \ \textit{errors} \ : \quad \langle\phi| E_\mu^\dagger E_\sigma |\psi\rangle = c_{\mu\sigma} \langle\phi|\psi\rangle$$

for some $c_{\mu\sigma} \in \mathbb{C}$.

The condition in the theorem implies the following properties:

- Orthogonal code states remain orthogonal under the action of errors.
- The expectation value of $E_\mu^\dagger E_\sigma$ is constant for all code states.

Proof. " $\Rightarrow$ " : Let $\mathcal{R} = \{R_r\}_r$ be the set of recovery operators.

$$\begin{aligned}
\langle\phi| E_\mu^\dagger E_\sigma |\psi\rangle &= \langle\phi| E_\mu^\dagger (\sum_r R_r^\dagger R_r) E_\sigma |\psi\rangle \\
&= \sum_r \langle\phi| (E_\mu^\dagger R_r^\dagger)(R_r E_\sigma) |\psi\rangle \\
\text{(by theorem 1.3.4)} \quad &= \sum_r \langle\phi| \bar{\lambda_{\mu r}} \lambda_{r\sigma} |\psi\rangle \qquad (1.1) \\
&= \sum_r \bar{\lambda_{\mu r}} \lambda_{r\sigma} \langle\phi|\psi|\phi|\psi\rangle \\
&= c_{\mu\psi} \langle\phi|\psi|\phi|\psi\rangle
\end{aligned}$$

" $\Leftarrow$ " :

Assume:

$$\forall |\phi\rangle, |\psi\rangle \in Q, \ \ \forall E_\mu, E_\sigma \in \mathcal{N} \text{ errors} \ : \quad \langle\phi| E_\mu^\dagger E_\sigma |\psi\rangle = c_{\mu\sigma} \langle\phi|\psi\rangle \qquad (1.2)$$

Let V^i be the space spanned by $\{E_\mu |\phi_i\rangle\}_{E_\mu \in \mathcal{N}}$ and $|\phi_i\rangle \in Q$. Notice that the spaces V^i's are orthogonal by the equation (1.2).
Let $\{|v_{ir}\rangle\}_r$ be an orthogonal basis for V^i.Then,

$$\exists V_r \text{ (unitary), such that:} \quad V_r |v_{ir}\rangle = |\phi_i\rangle$$

The existence of V_r is guaranteed since both $\{|v_{ir}\rangle\}_i$ and $\{|\phi_i\rangle\}_i$ are both orthonormal sets of the same size. The recovery operators are given by:

$$\mathcal{R} = \{\mathcal{O}, R_i, \ldots, R_r, \ldots\}$$

Where $\mathcal{O}$ is the projection onto $(\bigoplus_i V^i)^\perp$ and $R_r = V_r \sum_i |v_{ir}\rangle \langle v_{ir}|$.

To ensure that $\mathcal{R}$ recovers the state, we need a unitary U_i such that:

$$\forall r, U_i |v_{0r}\rangle = |v_{ir}\rangle \quad \text{and} \quad \forall E_\mu, U_i E_\mu |\phi_0\rangle = E_\mu |\phi_i\rangle$$

First, we prove that: $dim(V^i) = dim(V^j)$ for all i, j
Let:

$$V^i = \{E_\mu |\phi_i\rangle\}_\mu \quad \text{and} \quad V^j = \{E_\mu |\phi_j\rangle\}_\mu$$

Suppose that $dim(V^i) < dim(V^j)$. Then, $\exists \mu$ such that:

$$E_\mu |\phi_i\rangle = \sum_{\sigma \neq \mu} \alpha_\sigma E_\sigma |\phi_i\rangle \quad \text{and} \quad \{E_\mu |\phi_j\rangle\} \perp \{E_\sigma |\phi_j\rangle\}_{\sigma \neq \mu}$$

Then,

$$\langle\phi_i| E_\mu^\dagger \sum_{\sigma \neq \mu} \alpha_\sigma E_\sigma |\phi_i\rangle = 1 \quad \text{and} \quad \langle\phi_j| E_\mu^\dagger \sum_{\sigma \neq \mu} \alpha_\sigma E_\sigma |\phi_j\rangle = 0$$

which is a contradiction since:

$$\langle\phi_i| E_\mu^\dagger \sum_{\sigma \neq \mu} \alpha_\sigma E_\sigma |\phi_i\rangle = \langle\phi_j| E_\mu^\dagger \sum_{\sigma \neq \mu} \alpha_\sigma E_\sigma |\phi_j\rangle = \sum_{\sigma \neq \mu} \alpha_\sigma c_{\mu\sigma}$$

Hence, $dim(V^i) = dim(V^j)$ which guarantees the existence of a unitary map U_i satisfying:

$$\forall r, \;\; U_i |v_{0r}\rangle = |v_{ir}\rangle$$

Since, $\{|v_{0r}\rangle\}_r$ and $\{|v_{ir}\rangle\}_r$ are both orthonormal sets of the same size. For the second part, let:

$$E_\mu |\phi_0\rangle = \sum_r \alpha_{0r} |v_{0r}\rangle \quad \text{and} \quad E_\mu |\phi_i\rangle = \sum_r \alpha_{ir} |v_{ir}\rangle$$

And let $B' = \{|v'_{ir}\rangle\}$ be another orthonormal basis of V_i such that:

$$E_\mu |\phi_i\rangle = \sum_r \alpha_{0r} \left|v'_{ir}\right\rangle$$

This is possible since:

$$\langle\phi_0| E_\mu^\dagger E_\mu |\phi_0\rangle = \langle\phi_i| E_\mu^\dagger E_\mu |\phi_i\rangle = \sum_r |\alpha_{0r}|^2 = \sum_r |\alpha_{ir}|^2$$

Meaning that the norms are equal which guarantees the existace of a unitary map in V^i, mapping $[\alpha_{i0}, \alpha_{i1}, \ldots]$ to $[\alpha_{00}, \alpha_{01}, \ldots]$.

WLOG, we identify $\{|v'_{ir}\rangle\}$ and $\{|v_{ir}\rangle\}$. Therefore,

$$U_i E_\mu |\phi_i\rangle = \sum_r \alpha_{0r} U_i \left|v'_{ir}\right\rangle = \sum_r \alpha_{0r} v_{0r} = E_\mu |\phi_0\rangle$$

This concludes the existance of U_i. Now, let $|\psi\rangle \in Q$. Then,

$$\begin{aligned} E_\mu |\psi\rangle &= E_\mu \sum_i \alpha_i |\phi_i\rangle \quad (|\psi\rangle = \sum_i \alpha_i |\phi_i\rangle) \\ &= \sum_i \alpha_i E_\mu |\phi_i\rangle \\ &= \sum_i \alpha_i U_i E_\mu |\phi_0\rangle \quad \text{(by the properties of } U_i\text{)} \\ &= \sum_{i,r} \alpha_i U_i \beta_{0,\mu,r} |v_{0r}\rangle \\ &= \sum_{i,r} \alpha_i \beta_{0,\mu,r} |v_{ir}\rangle \end{aligned} \tag{1.3}$$

The factors α_i and $\beta_{0,\mu,r}$ are obtained from the expansion in terms of the corresponding basis elements. U_i here allows for an expansion in terms of the basis elements with β not depending on i.
Now, let $R_r \in \mathcal{R}$. Then, we get:

$$\begin{aligned} R_r E_\mu |\psi\rangle &= \sum_i V_r |v_{ir}\rangle \langle v_{ir}| \sum_{j,s} \alpha_j \beta_{0,\mu,s} |v_{j,s}\rangle \\ &= \sum_i \alpha_i \beta_{0,\mu,r} V_r |v_{i,r}\rangle \\ &= \sum_i \beta_{0,\mu,r} \alpha_i |\phi_i\rangle \\ &= \beta_{0,\mu,r} |\psi\rangle \end{aligned} \tag{1.4}$$

Hence, $R_r E_\mu$ is a multiple of the identity on Q. Furthermore, $\mathcal{O}$ is null on $E_\mu |\phi_i\rangle$. Therefore, $\mathcal{R}$ is a recovery operator for Q. □

What this theorem allows us to do, is to characterize simply what an error-correcting code is. Furthermore, it provides a construction of the recovery channel.

The next section introduces the concept of classical codes and some geometrical interpretations that we will revisit in Chapter 3.

1.4 linear codes over a finite field

Let A be a finite set that we call an Alphabet. Then, a code C of length n is a subset of A^n. We also define the distance between two elements of A^n as the number of coordinates in which they are different. The minimum distance of a code C is then defined as the minimal distance between any two elements of C.
Let A be a finite abelian group with identity element 0. A code C is said to be additive if:

$$\forall u, v \in C, \quad u + v \in C$$

For an additive code, the weight of an element of C (codeword) u is the number of non-zero coordinates of u.

Lemma 1.4.1. *If C is an additive code over a finite abelian group then the minimum distance d of C is equal to the minimum non-zero weight w.*

Proof. Since A is a finite abelian group, every $u \in C$ has finite order. Therefore, adding u finitely many times would yield $(0, \ldots, 0) \in C$. Similarly, $-u \in C$.
Suppose that u is an element of minimum weight w. Then since $0 \in C$, we have $w \geq 0$.
If u and v are two elements differing in exactly d coordinates. Then $u - v \in C$ is of weight d and therefore, $d \geq w$. □

Let $A = \mathbb{F}_q$, where $\mathbb{F}_q$ is the finite field of order $q = p^h$.

- If $av \in C$ for all $a \in \mathbb{F}_p$ then C is called additive.
- If $av \in C$ for all $a \in \mathbb{F}_q$ then, C is called linear.

1.4.1 Geometrical interpretation

In this geometrical interpretation, we will be mainly interested in linear codes over $\mathbb{F}_q$. We start by stating some basic facts from Linear Algebra. Let $\mathbb{F}_q^k$ be the k-dimensional vector space over $\mathbb{F}_q$ and G be an $k \times n$ matrix. Then we have the following:

- For a row vector $a^t \in \mathbb{F}_q^k$, the expression a^tG is a linear combination of the rows of G.
- Similarly, for a column vector $b \in \mathbb{F}_q^n$, the expression Gb is a linear combination of the columns of G.

Let C be a k-dimensional linear code over $\mathbb{F}_q$ of length n. Then C can be viewed as a k-dimensional subspace of $\mathbb{F}_q^n$. Furthermore, C can be obtained from a $k \times n$ matrix G with row space is C. Therefore,

$$\forall u \in C, \quad \exists a^t \in \mathbb{F}_q^n, \quad u = a^tG$$

The geometrical aspect of this code is obtained by considering the set of columns of this matrix which we denote as $\mathcal{X}$ (i.e. n vector of $\mathbb{F}_q^k$).
The ith coordinate of $u = a^tG$ is zero if and only if

$$a.z = \sum_{i=1}^{k} a_i z_i = 0$$

where z is the ith column of G. Note that any scalar multiple of the vector z would yield the same result. Therefore, one can consider the set of vectors $\mathcal{X}$ as n points of the finite projective space $PG(k-1,q)$ of dimension k-1 over $\mathbb{F}_q$.
For the following theorem, we will need more information about subspaces of $PG(k-1,q)$

Theorem 1.4.2. *An $[n,k,d]$ linear code over $\mathbb{F}_q$ is equivalent to a set of points $\mathcal{X}$ in $PG(k-1,q)$ in which every hyperplane of $PG(k-1,q)$ contains at most $n-d$ points of $\mathcal{X}$ and some hyperplane contains exactly $n-d$ points of $\mathcal{X}$.*

Proof. Let G be a $k \times n$ matrix with row space C a $[n,k,d]$ code.
Let $\mathcal{X}$ be the set of points from $PG(k-1,q)$ obtained from G's columns.
Recall that $u = a^tG$ has a zero in its ith coordinate if and only if:

$$a.z = \sum_{i=1}^{k} a_i z_i = 0$$

Where z is the ith column of G.
Furthermore, the kernel of the form $a_1X_1 + a_2X_2 + \cdots + a_kX_k$ defines a hyperplane π_a of $PG(k-1,q)$.
The codeword $u = a^tG$ has weight w if and only if u has exactly $n-w$ zero coordinates. And this happens when π_a is incident with $n-w$ elements of $\mathcal{X}$.
By lemma 1.4.1, The minimum distance of this code is equal to the minimum weight

of its elements. Therefore, if d is the minimum distance of C. $n-d$ is the maximal number of points of $\mathcal{X}$ laying in a hyperplane of $PG(k-1,q)$. □

$$a_1X_1+\cdots+a_rX_r$$

In the next chapter, we will start by introducing the concept of quantum stabilizer codes. Furthermore, we shall dive into some of the characteristics of this type of code and derive some conclusions. We also present examples of such codes.

2. Quantum stabilizer codes

There are many types of Quantum error-correcting code each with their advantages and drawbacks. In this section, we discover one of these codes; Quantum stabilizer codes. We start this section by defining a Quantum stabilizer code.

Definition 2.0.1. A qubit stabilizer code Q(S) is the intersection of eigen spaces with eigenvalue 1 of the elements of an abelian subgroup S of $\mathcal{P}_n$ not containing -I. The subgroup S is known as the stabilizer.

We provide some explanations for the choices made in this definition. S is defined as the subgroup generated by the set $\{M_1, \ldots, M_{n-k}\}$ of n-k elements. Where $M_1, \ldots, M_{n-k} \in \mathcal{P}_n$, hence:

$$S = \langle M_1, \ldots, M_{n-k} \rangle$$

-I is not an element of S, since otherwise $Q(S) = \{0\}$. (Since -I has no eigenvalue 1)

The reason behind the choice of the eigenvalue 1 is the fact that we want the M_i's to stabilize all the states in Q(S). Furthermore, we shall assume that there is no position in the tensor where all M_i's have σ_0. Otherwise, that position is redundant; hence, the code can be embedded in a smaller space. Note also that the phase of every element in S (constant multiple) is ± 1. Otherwise, this yields the following contradiction:

$$M = \pm i\sigma_1 \otimes \cdots \otimes \sigma_n \in S \implies M^2 = -I \in S$$

Example 3 (A stabilizer in $\mathcal{P}_3$)**.** In this example, we construct a stabilizer for $n = 3$ and $k = 1$.
Let,

$$\begin{cases} M_1 = \sigma_x \otimes \sigma_0 \otimes \sigma_z \\ M_2 = \sigma_y \otimes \sigma_y \otimes \sigma_x \end{cases}$$

We have,

$$M_1M_2 = (\sigma_x\sigma_y)\otimes(\sigma_0\sigma_y)\otimes(\sigma_z\sigma_x) = (i\sigma_z)\otimes(\sigma_y)\otimes(i\sigma_y) = -\sigma_z\otimes\sigma_y\otimes\sigma_y$$

Also,

$$M_2M_1 = (\sigma_y\sigma_x)\otimes(\sigma_y\sigma_0)\otimes(\sigma_x\sigma_z) = (-i\sigma_z)\otimes(\sigma_y)\otimes(-i\sigma_y) = -\sigma_z\otimes\sigma_y\otimes\sigma_y$$

Hence,

$$M_1M_2 = M_2M_1 \neq I$$

Therefore, the subgroup $S = \langle M_1, M_2\rangle$ generated by M_1 and M_2 is a stabilizer.
Now we find the 1-eigenvectors of the elements of S.
Let $[x_1,\ldots,x_6]^t \in Q(S)$ meaning that the vector x is in the joint 1-eigenspace of the elements of S. Then, we have:

$$\begin{cases} M_1[x_1,\ldots,x_6]^t = (\sigma_x\otimes\sigma_0\otimes\sigma_z)[x_1,\ldots,x_6]^t = [x_1,\ldots,x_6]^t \\ M_2[x_1,\ldots,x_6]^t = (\sigma_y\otimes\sigma_y\otimes\sigma_x)[x_1,\ldots,x_6]^t = [x_1,\ldots,x_6]^t \\ M_1M_2[x_1,\ldots,x_6]^t = (-\sigma_z\otimes\sigma_y\otimes\sigma_y)[x_1,\ldots,x_6]^t = [x_1,\ldots,x_6]^t \end{cases}$$

Then,

$$\begin{cases} [x_2,x_1,x_3,\ldots,x_5,-x_6]^t = [x_1,x_2,x_3,\ldots,x_5,x_6]^t \\ [ix_2,-ix_1,ix_4,-ix_3,x_6,x_5]^t = [x_1,x_2,x_3,\ldots,x_5,x_6]^t \\ [-x_1,x_2,-ix_4,ix_3,-ix_6,ix_5]^t = [x_1,x_2,x_3,\ldots,x_5,x_6]^t \end{cases}$$

This implies that

$$x_1 = x_2 = x_5 = x_6 = 0 \quad x_4 = -ix_3$$

Hence, $Q(S) = \langle[0,0,1,-i,0,0]^t\rangle$ which is the subspace of $\mathbb{C}^6$ generated by the vector $[0,0,1,-i,0,0]^t$

It is often tough to calculate the basis of the eigenspace the higher our dimension gets. That is why we tend to use projections onto the space Q(S) rather than the basis of this space in our calculation.

2.1 The projection onto Q(S)

Let S be an abelian subgroup of $\mathcal{P}_n$ and let Q(S) be the subspace as defined earlier. Let P = P(S) be the orthogonal projection onto the subspace Q(S). We write the projection in terms of the elements of S in the following lemma.

Lemma 2.1.1. *The orthogonal projection P(S) is:*

$$P(S) = \frac{1}{|S|} \sum_{E \in S} E$$

Proof. Let $F = \frac{1}{|S|} \sum_{E \in S} E$ Since S is an abelian subgroup. One has:

$$MF = FM = F \quad , \forall M \in S$$

Let $|\psi\rangle \in Q(S)$. Then, $F|\psi\rangle = |\psi\rangle$. Therefore, $|\psi\rangle \in Im(F)$
Now let $|\psi\rangle \in Im(F)$. Then, $|\psi\rangle = F|\phi\rangle$. Hence, $\forall M \in S$

$$M|\psi\rangle = MF|\phi\rangle = F|\phi\rangle = |\psi\rangle$$

Therefore, $|\psi\rangle$ is an element Q(S). Which means that $Q(S) = Im(F)$.
Furthermore, $E^\dagger = E$ for all E in $\mathcal{P}_n$ which implies that $F^\dagger = F$. Moreover,

$$F^2 = F\frac{1}{|S|} \sum_{E \in S} E = \frac{1}{|S|} \sum_{E \in S} FE = \frac{1}{|S|} \sum_{E \in S} F = F$$

To summarize, we have proven that: $F^2 = F$ and $F^\dagger = F$ and finally that $Im(F) = Q(S)$. These mean that F is the projection onto Q(S) (i.e. P(S) = F) □

Before we state the next theorem which calculates the dimension of Q(S). One must note a property of the trace function: $tr(\sigma_1 \otimes \cdots \otimes \sigma_n) = tr(\sigma_1) \ldots tr(\sigma_n)$. This implies that for all E in $\mathcal{P}_n$ different from $\pm I$ we have: tr(E) = 0. Also, $tr(I) = 2^n$.

Theorem 2.1.2. *The stabilizer code Q(S) which is the joint 1-eigenspace of an abelian subgroup S generated by n-k independent elements has dimension* 2^k

Proof. We have the orthogonal projection onto Q(S):

$$P(S) = \frac{1}{|S|} \sum_{M \in S} M$$

Since P is hermitian, then P is diagonolizable. Furthermore, $P^2 = P$ which implies that the eigen values of P are either 0 or 1.

Note that the trace of P is equal to the sum of its eigenvalues. Therefore,

$$dim(Q(S)) = dim(Im(P)) = tr(P) = \frac{1}{|S|}\sum_{M\in S} tr(M) = \frac{1}{|S|}tr(I) = \frac{2^n}{2^{n-k}} = 2^k$$

□

Remark. *When k is 0, Q(S) is then a 1-dimensional subspace. Therefore, such codes are often not practical to store quantum information.*

Example 4. Using the stabilizer constructed in Example 3, we can construct the projection onto Q(S) using Lemma 2.1.1.
We have,

$$P(S) = \frac{1}{|\ S\ |}\sum_{E\in S} E = \frac{1}{4}(I + M_1 + M_2 + M_1M_2)$$

Hence,

$$P(S) = \frac{1}{4}(\sigma_0\otimes\sigma_0\otimes\sigma_0 + \sigma_x\otimes\sigma_0\otimes\sigma_z + \sigma_y\otimes\sigma_y\otimes\sigma_x - \sigma_z\otimes\sigma_y\otimes\sigma_y)$$

Calculating the row space of P(S) will yield Q(S).
Let

$$v_1 = [1,0,0,1,1,0,0,-1] \quad , \quad v_2 = [0,1,-1,0,0,-1,-1,0]$$

Q(S) is then the space generated by v_1 and v_2.
The calculation is done through Sage and the code can be checked in Appendix.

2.2 Correctable errors in Q(S)

In the case of Quantum stabilizing codes, correctable errors are a subset of detectable errors. The following lemma explicitly states which elements of $\mathcal{P}_n$ are undetectable errors.

Lemma 2.2.1. *E is an undetectable error for Q(S) if and only if* $E \in C_{\mathcal{P}_n}(S) - S$

Proof. We state the proof by proving both directions of the equivalence statement.
" $\Rightarrow$ " :
Suppose that E is undetectable and that $E \notin C_{\mathcal{P}_n}(S) - S$. If $E \notin C_{\mathcal{P}_n}(S)$, then

$$\exists M \in S, \quad EM = -ME,$$

This is the case since elements of $\mathcal{P}_n$ either commute or anti-commute.
Let $|\psi\rangle, |\phi\rangle \in Q(S)$ with $\langle\psi|\phi\rangle = 0$. Then,

$$\langle\psi|E\phi\rangle = \langle\psi|ME\phi\rangle = \langle\psi|-EM\phi\rangle = -\langle\psi|E\phi\rangle$$

Therefore, $\langle\psi|E\phi\rangle = 0$
Now if $E \in S$, Then,

$$\langle\psi|E\phi\rangle = \langle\psi|\phi\rangle = 0$$

Hence, by Knill-La Flamme theorem E is correctable which means it is detectable, which is a contradiction.

" $\Leftarrow$ " :
Suppose that $E \in C_{\mathcal{P}_n}(S) - S$ and that E is detectable. Also, let $|\psi\rangle \in Q(S)$. $E \in C_{\mathcal{P}_n}(S)$ implies that:

$$ME|\psi\rangle = EM|\psi\rangle = E|\psi\rangle \quad ,\forall M \in S$$

Hence, $E|\psi\rangle \in Q(S)$. Now, extend $\{|\psi\rangle\}$ to an orthonormal basis $\mathcal{B}$ for Q(S). Then, $\mathcal{B} = \{|\psi\rangle\} \cup \{|\phi_i\rangle\}_i$
Since E is detectable we have:

$$\langle\phi_i|E|\psi\rangle = 0, \quad |\phi_i\rangle \in \mathcal{B} - \{|\psi\rangle\}$$

Then, $E|\psi\rangle \in (\langle\mathcal{B} - \{|\psi\rangle\}\rangle)^{\perp}$. Note that the space $(\langle\mathcal{B} - \{|\psi\rangle\}\rangle)^{\perp}$ has basis $\{|\psi\rangle\}$. Therefore,

$$E|\psi\rangle = \lambda_\psi|\psi\rangle, \quad \lambda_\psi \in \mathbb{C}$$

Hence, $|\psi\rangle$ is an eigen vector of E. By Knill-Laflamme theorem, $\langle\phi|E|\phi_i\rangle = \lambda_E$ for all $|\phi\rangle$ in $\mathcal{B}$. Since, for all $|\alpha\rangle$ in Q(S) $\langle\alpha|\alpha\rangle = 1$ then $\lambda_\alpha = \lambda_E$. Then, $E|\alpha\rangle = \lambda_E|\alpha\rangle$. Now note that $E \notin S$. Hence, λ_E is not equal to 1. Therefore, the subgroup generated by S and λ_E^{-1} defines a smaller stabilizer code which is not possible. Therefore, there is $|\alpha_i\rangle$ in Q(S) such that:

$$\lambda_E^{-1}E|\psi\rangle \neq |\psi\rangle$$

Which is a contradiction Of the properties of E stated above. Hence, E is not detectable. □

Remark. *If S' is a subgroup of S, the code $Q(S')$ can be also be studied. Note that:*

$$(C_{\mathcal{P}_n}(S') - S') \subseteq (C_{\mathcal{P}_n}(S') - S')$$

The weight of an operator E is defined as the number of the non-trivial components. (i.e. $wt(\sigma_x \otimes \sigma_y \otimes \sigma_0) = wt(\sigma_z \otimes \sigma_0 \otimes \sigma_x) = 2$). In the next theorem we showcase that the minimal distance of Q(S) will be exactly what one expects it to be.

Theorem 2.2.2. *If $k \geq 1$, the minimal distance of the stabilizer code Q(S) is the minimum weight of the errors in $C_{\mathcal{P}_n}(S) - S$*

Proof. By the previous lemma, Q(S) can detect all errors which are not elements of $C_{\mathcal{P}_n}(S) - S$.
In particular, it can detect all errors of weight less than the minimum weight of errors in $C_{\mathcal{P}_n}(S) - S$. □

Now we introduce the following notation for quantum codes:

- $((n, K, d))$ denotes a quantum code in $(\mathbb{C}^2)^{\otimes n}$ of dimension K and minimum distance d.
- $[[n, k, d]]$ denotes a quantum code in $(\mathbb{C}^2)^{\otimes n}$ of dimension 2^k and minimum distance d.

Example 5. In this example, we will find the minimum distance of the code introduced in Example 3. Recall that the stabilizer of that code was generated by two errors:

$$\begin{cases} M_1 = \sigma_x \otimes \sigma_0 \otimes \sigma_z \\ M_2 = \sigma_y \otimes \sigma_y \otimes \sigma_x \end{cases}$$

Since Q(S) is not trivial, the minimum distance is at least 1. Furthermore, note that $\alpha = \sigma_0 \otimes \sigma_y \otimes \sigma_0$ commutes with M_1 and M_2. Hence, $\alpha \in C_{\mathcal{P}_3}(S) - S$. And since α has weight 1, the minimum distance of this code is 1. This code is then noted as $[[3, 2, 1]]$.

The fact that the code from example 5 is of minimum distance 1 is a special case of lemma 2.2.3 which we will state now. We first introduce the following details to derive a conclusion on the minimal distance of a stabilizer code. Let S be the stabilizer generated by the following operators:

$$\begin{cases} M_1 = \sigma_{11} \otimes \sigma_{12} \otimes \cdots \otimes \sigma_{1n} \\ M_2 = \sigma_{21} \otimes \sigma_{22} \otimes \cdots \otimes \sigma_{2n} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ M_{n-k} = \sigma_{(n-k)1} \otimes \sigma_{(n-k)2} \otimes \cdots \otimes \sigma_{(n-k)n} \end{cases}$$

We define A_i as the following:

$$A_i = \mid \{\sigma_{1i}, \sigma_{2i}, \ldots, \sigma_{(n-k)i}\} - \{\sigma_0\} \mid$$

Lemma 2.2.3. *A pure code Q(S) has a minimal distance larger than 1 if and only if* $\min_i(A_i) \geq 2$

Proof. We state the proof by proving both directions of the equivalence statement.
" $\Rightarrow$ " : Assume that Q(s) has minimal distance larger than 1
And assume for a contradiction that there is an i such that $A_i \leq 1$. WLOG, we assume that $A_i = 1$ because otherwise the code can be embedded in a smaller dimension by simply removing the component i since it is trivial. Therefore,

$$A_i = 1 \implies \sigma_{ki} = \sigma_i \quad \forall k \in \{1, \ldots, (n-k)\}$$

for some fixed $\sigma_i \in \{\sigma_x, \sigma_z, \sigma_y\}$. Hence, $\sigma_0 \otimes \cdots \otimes \sigma_i \otimes \sigma_0 \otimes \cdots \otimes \sigma_0$ is an element of $C_{\mathcal{P}_n}(S) - S$ or S. If $\sigma_0 \otimes \cdots \otimes \sigma_i \otimes \sigma_0 \otimes \cdots \otimes \sigma_0 \in C_{\mathcal{P}_n}(S) - S$, then since the code is pure the minimal distance is 1 (by theorem 2.2.2) which is a contradiction.
" $\Leftarrow$ " : Assume that $\min_i(A_i) \geq 2$
Assume for a contradiction that T is an element of $C(\mathcal{P}_n)(S) - S$ of weight 1. Since T has weight 1, then

$$T = \sigma_0 \otimes \cdots \otimes \sigma_i \otimes \sigma_0 \otimes \cdots \otimes \sigma_0$$

Note that σ_i only commutes with σ_0 and itself. And since A_i is larger than 1, there is an element M in S with component i different from σ_i. Hence, M doesn't commute with T. which is a contradiction. Hence, elements of $C(\mathcal{P}_n)(S) - S$ have a weight of at least 2. □

3. Quantum stabilizer codes and Linear codes

In order to understand most of the geometrical arguments made in this thesis, one should be familiar with projective geometry and more specifically finite geometry.

3.1 Preliminaries from finite projective spaces

In this part of the chapter, we present some necessary definitions and lemma while providing some proofs when it's reasonable to do so, otherwise, the reader can be referred to textbooks where such proofs are available. We start by defining a finite projective space $PG(k-1,q)$.

Definition 3.1.1. Let $\mathbb{F}_q^k$ be the k dimensional vector space over the field $\mathbb{F}_q$. We also define the following equivalence relation for $a, b \in \mathbb{F}_q^k - \{0\}$

$$a \sim b \iff a = c \cdot b$$

where $c \in \mathbb{F}_q$. PG(k-1, q) can be then defined as the quotient of this equivalence relation.

$$PG(k-1,q) = \mathbb{F}_q^k / \sim$$

We note that the points of $PG(k1,q)$ correspond to the one-dimensional subspaces of $\mathbb{F}_k^q$. Generally, the (i-1) dimensional subspaces of $PG(k-1,q)$ are the i-dimensional subspaces of $\mathbb{F}_k^q$.

Remark. *We refer to the zero-dimensional, one-dimensional, and co-dimensional 1 subspaces of $PG(k-1,q)$ respectively.*

Furthermore, the projective space $PG(k-1,q)$ has the following properties:

- Any two points are joined by a line.

- Every two hyperplanes have a non-empty intersection.

Definition 3.1.2. We say that two subspaces of $PG(k-1,q)$ are said incident if one is included in the other. Furthermore, if these two subspaces have an empty intersection then they are said to be skew.
Now let $\{x_1,\ldots,x_r\}$ be a set of points in $PG(k-1,q)$. These points are said to be independent if the subspace generated by them spans an $(r-1)$-dimensional subspace. These points are said dependent if they are not independent.

We also state and prove the following statements:

- The number of r-tuples of linearly independent vectors of $\mathbb{F}_q^k$ is:

$$(q^k-1)(q^{k-1}-1)\ldots(q^{k-r+1}-1)$$

Proof. We choose the first vector $x_1 \in \mathbb{F}_q^k$ randomly and therefore we have (q^k-1) choices. The next vector we choose x_2 should not be a multiple of x_1 and there are q such multiples. Hence, we have $(q^{k-1}-1)$ such choices. Applying the same argument inductively yields the result above. □

- the number of r-dimensional subspaces of $\mathbb{F}_q^k$ is:

$$\begin{bmatrix} k \\ r \end{bmatrix}_q := \frac{(q^k-1)(q^{k-1}-1)\ldots(q^{k-r+1}-1)}{(q^r-1)(q^{r-1}-1)\ldots(q-1)}$$

Proof. The number of r-dimensional subspaces of $\mathbb{F}_q^k$ corresponds to the number of r-tuples of linearly independent vectors of $\mathbb{F}_q^k$. But we have to be wary of redundancies. Every basis of an r-dimensional subspace has r elements. the choice of these r elements is done in $(q^r-1)(q^{r-1}-1)\ldots(q-1)$ many ways. (by the same argument as the previous proof)
Hence, we get the statement. □

- Therefore, the number of points in $\mathbb{F}_q^k$ is:

$$\frac{q^k-1}{q-1} = q^{k-1}+\cdots+q+1$$

Proof. This is a straightforward application of the previous statement. □

- We denote the duality on $PG(k-1,q)$ as the mapping of points $(a_1,\ldots,a_k) \in PG(k-1,q)$ to the hyperplanes defined by the linear form

$$a_1X_1+\ldots,a_kX_k=0$$

This also implies that the number of hyperplanes is exactly equal to the number of points in a projective space.

Note that in our study we mainly consider the case when q is 2. Therefore,

- the number of points is: $2^k - 1$
- the number of lines is: $\frac{(2^k-1)(2^{k-2}-1)}{3}$

Lastly we prove a lemma that we use in the body of the thesis.

Lemma 3.1.1. *The number of $(r-1)$-dimensional subspaces of $PG(k-1,q)$ containing a fixed $(s-1)$-dimensional subspace is*

$$\begin{bmatrix} k-s \\ r-s \end{bmatrix}_q$$

Proof. Let U be an s-dimensional subspace of $\mathbb{F}_q^k$, the quotient space $\mathbb{F}_q^k/U$ is a $(k-s)$-dimensional vector space. The r-dimensional subspaces containing U are exactly the $(r-s)$-dimensional subspaces in the quotient space. If we now consider the projective spaces corresponding to these spaces and change the vector space dimension to the projective space dimension we obtain our claim. □

As we have established in the previous chapter it is often tedious to check if a set of operators from $\mathcal{P}_n$ generates a stabilizer (i.e. the operator inter-commute). Furthermore, we would like to determine fairly quickly if these operators are independent. Finally, determining the minimum distance of these codes remains a computationally heavy task if done by calculating the stabilizer. For the following reasons, we define a map τ that maps the error operators to classical binary vectors.
Let $\mathbb{F}_2$ denote the finite field of 2 elements. Then, consider the following map:

$$\tau : \quad \{\sigma_0, \sigma_x, \sigma_y, \sigma_z\} \longrightarrow \mathbb{F}_2^2$$

such that:

$$\tau : \begin{cases} \sigma_0 \longmapsto (0|0) \\ \sigma_x \longmapsto (1|0) \\ \sigma_z \longmapsto (0|1) \\ \sigma_y \longmapsto (1|1) \end{cases}$$

In order to extend the map to $\mathcal{P}_n$, we apply τ to an element of $\mathcal{P}_n$ coordinate-wise such that the image of the j-th position in an element of $\mathcal{P}_n$ is the j-th and the

(j+n)-th coordinate of a vector in $\mathbb{F}_2^{2n}$. For example,

$$\tau(\sigma_x \otimes \sigma_0 \otimes \sigma_y) = (101|001)$$

(Note that the map τ is phase agnostic i.e. $\tau(\lambda M) = \tau(M), \quad \forall \lambda \in \{\pm 1, \pm i\}$)

Lemma 3.1.2. *(Ball, Centelles & Huber (2021))*

$$\forall M, N \in \mathcal{P}_n/\{\pm 1, \pm i\}, \quad \tau(MN) = \tau(M) + \tau(N)$$

Proof. We note the following property of the Pauli matrices:

$$\sigma_i \sigma_j = \lambda \sigma_k$$

where distinct i, j, and k are in $\{x, y, z\}$. It is then straightforward to check that: $\tau(\sigma_i \sigma_j) = \tau(\sigma_i) + \tau(\sigma_j)$ for all i, j in $\{x, y, z\}$.
Applying the result component-wise on elements of $\mathcal{P}_n/(iI)$ yields the result we want. □

Therefore, there is an isomorphism between $\mathcal{P}_n/(iI)$ and $\mathbb{F}_n^{2n}$. Hence, a subgroup S of $\mathcal{P}_n$ is isomorphic to a subspace of $\mathbb{F}_n^{2n}$. Using this isomorphism, this section aims to derive results about stabilizer codes.
In order to introduce the next lemma, we start by defining an alternating form of $u, w \in \mathbb{F}_2^{2n}$:

$$(u, w)_a = \sum_{j=1}^{n} (u_j w_{j+n} - u_{j+n} w_j)$$

Lemma 3.1.3. *(Ball et al. (2021))*

For $M, N \in \mathcal{P}_n/(iI)$

$$MN = NM \iff (\tau(M), \tau(N))_a = 0$$

Proof. Let $u = \tau(M)$ and $\tau(N)$. Then, $\Delta_j = u_j W_{j+n} - u_{j+n} w_j$ is zero if and only if the Pauli matrices in the j-th component of M and N commute.

M	N	$(u_j\|u_{j+n})$	$(w_j\|w_{j+n})$	$\Delta_j \in \mathbb{F}_2$
σ_x	σ_y	$(1\|0)$	$(1\|1)$	1
σ_x	σ_z	$(1\|0)$	$(0\|1)$	1
σ_y	σ_z	$(1\|1)$	$(0\|1)$	1
σ_0	σ_i	$(0\|0)$	$(a\|b)$	0

Therefore, the operators M and N commute if and only if there are an even number of positions where the Pauli matrices do not commute. This is then equivalent to the case when there is an even number of coordinates satisfying:

$$u_j w_{j+n} - u_{j+n} w_j = 1$$

Hence, $(\tau(M), \tau(N))_a = 0$ □

We now define the symplectic weight of a vector $v \in \mathbb{F}_2^{2n}$ as:

$$| \{ i \in \{1, \ldots, n\} \mid (v_i, v_{i+n}) \neq (0,0) \} |$$

Lemma 3.1.4. *(Ball et al. (2021)) The weight of $M \in \mathcal{P}_n$ is equal to the symplectic weight of $\tau(M)$*

Proof. The weight of M is the number of components in M different from σ_0. Each one of these components is represented by $(i,j) \neq (0,0)$ in $\tau(M)$. Hence, $wt(M)$ is equal to the symplectic weight of $\tau(M)$. □

Let C be a subspace of $\mathbb{F}_2^{2n}$, we define $C^{\perp_a}$ as:

$$C^{\perp_a} = \{ u \in \mathbb{F}_2^{2n} \mid (u,w)_a = 0 \quad , \forall w \in C \}$$

Theorem 3.1.5. *(Ball et al. (2021))*

S is a subgroup of $\mathcal{P}_n$ generated by $(n-k)$ independent, mutually commuting elements if and only if $C = \tau(S)$ is a $(n-k)$-dimensional subspace of $\mathbb{F}_2^{2n}$ for which $C \leq C^{\perp_a}$.

- *If $k \neq 0$ then the minimum distance of Q(S) is equal to the minimum symplectic weight of the elements of $C^{\perp_a} - C$.*
- *If $k = 0$, then the minimum distance of Q(S) is equal to the minimum symplectic weight of the non-zero elements of $C = C^{\perp_a}$*

Proof. By lemma 3.0.3, $C \leq C^{\perp_a}$. Furthermore, if $k \neq 0$. The minimum distance is equal to the minimum weight of the images of the elements of $C_{\mathcal{P}_n}(S)$ under τ but are not elements of the image of S.
Since $C = \tau(S)$ and $C^{\perp_a} = \tau(C_{\mathcal{P}_n}(S))$. The theorem then follows.
For $k = 0$, by be definition, the minimum distance is equal t the minimum weight of the images of the elements of S under τ. □

Since $C = \tau(S)$ is a subspace of $\mathbb{F}_2^{2n}$, C can then be generated by a $(n-k) \times 2n$ matrix that we call $G(S)$. The i-th row of $G(S)$ is $\tau(M_i)$.

Lemma 3.1.6. *S is a subgroup of $\mathcal{P}_n$ generated by $(n-k)$ independent elements if and only if the matrix $G(S)$ has rank n-k.*

Proof. The rank of $G(S)$ is n-k if and only if the rows of this matrix are linearly independent. In other words, for every proper subset $J \subset \{1, \ldots, n-k\}$ we have:

$$\sum_{j \in J} \tau(M_j) \neq 0$$

By lemma 3.0.1,

$$\sum_{j \in J} \tau(M_j) \neq 0 \quad \iff \quad \prod_{j \in J} M_j \neq I$$

(note that $\tau(I) = 0$)
Therefore, the lemma follows. □

Example 6. We now showcase an example of a generator matrix using the same $[[3,1,1]]$ code from Example 3 (for simplicity). Similar codes can be found in Grassl (2007). Recall that the stabilizer of this code is generated by

$$\begin{cases} M_1 = \sigma_x \otimes \sigma_0 \otimes \sigma_z \\ M_2 = \sigma_y \otimes \sigma_y \otimes \sigma_x \end{cases}$$

Hence, the rows of the generator matrix are the two vectors u_1 and u_2

$$\begin{cases} u_1 = (100 \mid 001) \\ u_2 = (111 \mid 110) \end{cases}$$

Therefore, the generator matrix is

$$G(S) = \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{array}\right)$$

It is then trivial to see that G(S) has rank two.
To make computing the matrix seamless we provide a function doing exactly that built on Sage in the Appendix.

Corollary 1. *Let A be an $n \times n$ symmetric matrix, then the code generated by $G(S) = (I \mid A)$ is a $[[n, 0, d]]$ code.*

Proof. $G(S) = (I \mid A)$ generates a quantum code if and only if $(u, v)_a = 0$ for any

two rows u, v of $G(S)$. This is the case if the following statement holds:

$$(I \mid A)(\frac{A^t}{I}) = 0$$

Since A is symmetric, then

$$(I \mid A)(\frac{A^t}{I}) = A^t + A = A + A = 0$$

Therefore, $G(S)$ generates a code $[[n, k, d]]$. The fact that k is 0 is trivial since $G(S)$ has full rank. □

3.2 Geometrical interpretation of qubit quantum codes

The geometry of qubit quantum codes is naturally derived from the geometry of linear codes (subsection 1.4.1) since the image of a quantum code under the map τ is a linear code over $\mathbb{F}_q$ where $q = 2^h$.
Recall that a quantum stabilizer code is equivalent to a binary linear code $C = \tau(S)$ with length 2n and minimal distance equal to the minimal symplectic weight of $C^{\perp_a} - C$, such that $C \leq C^{\perp_a}$.

Example 7. We often start from a stabilizer and then obtain a generator matrix which can provide us with more insights into our code. On the other hand, one can start from an arbitrary $(n-k) \times 2n$ matrix with rows having a null alternating form and generate a stabilizer from this matrix. We showcase exactly that with the aid of some code that we provide in the Appendix.
Let

$$G(S) = \left(\begin{array}{cccccccccccc|cccccccccccc}
0&0&0&0&0&0&0&1&0&0&0&1 & 1&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&1&0&0&0&1&0 & 0&1&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&1&0&0&0&1&0&0 & 0&0&1&0&0&0&0&0&0&0&1&0\\
0&0&0&0&1&0&0&0&1&0&0&0 & 0&0&0&1&0&0&0&0&0&0&0&1\\
0&0&0&1&0&0&0&0&0&0&0&0 & 0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0 & 0&0&0&0&0&1&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&1&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&1&0&0&0&0
\end{array}\right)$$

be a 8×24 matrix. The matrix does indeed satisfy the condition that its rows have a

null symplectic product, this can be checked using the code in the Appendix. That means that we can obtain a stabilizer from $G(S)$ by using τ^{-1}.
Using the code in the Appendix we can turn the following rows into operators in $\mathcal{P}_{12}$

$$M_1 = ZIIIIIIXZIIX \qquad M_2 = IZIIIIXIIZXI \qquad M_3 = IIZIIXIIIXZI$$
$$M_4 = IIIZXIIIXIIZ \qquad M_5 = IIIXZIIIIIII \qquad M_6 = IIXIIZIIIIII$$
$$M_7 = IXIIIIZIIIII \qquad M_8 = XIIIIIIZIIII$$

S is the subgroup generated by the M_i's

$$S = \langle M_1, M_2, \ldots, M_8 \rangle$$

Remark. *Let S be a subgroup of $\mathcal{P}_n$ generated by the following $n-k$ elements $M_1, \ldots, M_{n-k}$. Then, we have the following:*

- *For $i \in \{1, \ldots, n\}$ we get a line by considering the span of the ith and $(i+n)$-th column of the generator matrix $G(S)$.*
- *Given n lines in $PG(n-k-1,2)$, we can construct an $(n-k) \times 2n$ matrix. Then, using τ^{-1} on the rows of this matrix we obtain the generators of S.*

Note that obtaining the generators of S from a tuple of lines in $PG(n-k-1,2)$ is up to a permutation of the coordinates of these generators. This is the case since we can permute the column of the induced matrix and still obtain a quantum stabilizer code since the commutativity of the operators is unaffected.
stabilizer codes obtained from permuting the columns in the ith and $(n+i)$th position of the generator matrix are considered equivalent.
We will also assume that the factor on all these generator operators is 1. Changing these factors would change the induced code which is something we shall discuss later on in Chapter 4.

Lemma 3.2.1. *(Ball et al. (2021)) The span of the ith and $(i+n)$th column of the generator matrix $G(S)$ is a line of $PG(n-k-1,2)$ for all $i \in \{1, \ldots, n\}$ if and only if the minimum non-zero weight of $C_{\mathcal{P}_n}(S)$ is at least two.*

Proof. The ith and the $(n+i)$th columns of the matrix of $G(s)$ don't induce a line in $PG(n-k-1,2)$ if these columns are either the same non-zero vector or one or both of them is the zero vector.
Hence, in the ith position of all the operators in S, there is either σ_0 or a fixed element $\sigma_i \in \{\sigma_x, \sigma_y, \sigma_z\}$. The statement then becomes equivalent to lemma 2.2.3. □

The next theorem presents a geometrical interpretation of the fact that $S \subset C_{\mathcal{P}_n}(S)$ or equivalently that $C \leq C^{\perp_a}$.

Theorem 3.2.2. *(Ball et al. (2021)) The following statements are equivalent:*

- *There is a $[[n,k,d]]$ stabilizer code $Q(S)$, where S is a subgroup generated by $n-k$ independent commuting elements of $\mathcal{P}_n$ and whose centralizer contains no element of weight one.*
- *There is a tuple of n lines $\mathcal{X}$ spanning $PG(n-k-1,2)$ with the property that every co-dimension 2 subspace is skew to an even number of lines of $\mathcal{X}$.*

Proof. We state the proof by proving both directions of the equivalence statement.
" $\Rightarrow$ " : Assume there is a $[[n,k,d]]$ stabilizer code $Q(S)$, where S is a subgroup generated by $n-k$ independent commuting elements of $\mathcal{P}_n$ and whose centralizer contains no element of weight one.
Let $C = \tau(S)$, and let G be the $(n-k) \times 2n$ generator matrix of C. As previously shown G has a full rank (i.e. n-k). Geometrically, this implies that the columns of G span $PG(n-k-1,2)$.
We also denote with $\mathcal{X}$ the tuple of lines spanned by the ith and the (i+n)th columns of G.
Let $u, v \in C$, then there is $a, b \in \mathbb{F}_2^{n-k}$ such that:

$$u = (a_1, \ldots, a_{n-k})G \qquad \textbf{and} \qquad w = (b_1, \ldots, b_{n-k})G$$

As shown earlier $C \subset C_{\perp_a}$ if and only if

$$(u,w)_a = \sum_{j=1}^{n} = (u_j w_{n+j} = w_j u_{n+j}) = 0 \qquad\qquad ,\forall u, w \in C$$

Let x and y be the jth and (j+n)th columns of G respectively. Then

$$u_j w_{n+j} = w_j u_{n+j} = (a \cdot x)(b \cdot y) - (a \cdot y)(b \cdot x)$$

This is zero if and only if the following matrix had a null determinant, therefore, having rank 1.

$$\begin{pmatrix} a \cdot x & a \cdot y \\ b \cdot y & b \cdot x \end{pmatrix}$$

This is also equivalent to the existance of $\lambda, \mu \in \mathbb{F}_2$ such that

$$a \cdot (\lambda x + \mu y) = b \cdot (\lambda x + \mu y) = 0$$

Now if we consider the two hyperplanes that are the zero loci of the following forms

$$\pi_a : a \cdot X = a_1X_1 + \cdots + a_{n-k}X_{n-k} \qquad \textbf{and} \qquad \pi_b : b \cdot X = b_1X_1 + \cdots + b_{n-k}X_{n-k}$$

Then, the point $\lambda x + \mu y$ lies in $\pi_a \cap \pi_b$ the intersection of both hyperplanes. Hence, the line spanned by x and y is incident with $\pi_a \cap \pi_b$.
So, for $(u,w)_a$ to be null, the sum should include an even number of ones. This implies that for a given a and b, the total number of lines of $\mathcal{X}$ skew to $\pi_a \cap \pi_b$ is even. Note that any co-dimension 3 subspace of $PG(n-k-1,2)$ can be constructed with such hyperplanes (i.e. as the intersection of π_a and π_b). The implication then follows.
" $\Leftarrow$ " : Assume there is a tuple of n lines $\mathcal{X}$ spanning $PG(n-k-1,2)$ with the property that every co-dimension 2 subspace is skew to an even number of lines of $\mathcal{X}$.
Let G be the matrix with ith and (i+n)th columns being points spanning the ith line of $\mathcal{X}$. Let C be the code generated by G. Then C is (n-k)-dimensional since $\mathcal{X}$ spans $PG(n-k-1,2)$ as shown earlier in this proof.
Furthermore, the fact that every co-dimension 2 subspace is skew to an even number of lines of $\mathcal{X}$ implies that $(u,w)_a = 0$ for any u and w in C (Using the same argument as in the first implication).
By Lemma 3.0.2, $\tau^{-1}(C)$ is an abelian subgroup S of $\mathcal{P}_n$ and by Lemma 3.0.5, S is generated by $n-k$ pairwise commuting elements of $\mathcal{P}_n$. □

In order to deduce more results about the minimum distance of stabilizer codes, we introduce the following definitions.

Definition 3.2.1. Let $\mathcal{X}$ be a tuple of projective lines and let $\Theta(\mathcal{X})$ be the space spanned by the lines of $\mathcal{X}$.
$\mathcal{X}$ is a quantum tuple of lines if we have every co-dimension 2 subspace of $\Theta(\mathcal{X})$ is skew to an even number of lines of $\mathcal{X}$.

Before we introduce the next definition, we recall that r points are independent if they span an $(r-1)$-dimensional subspace.

Definition 3.2.2. We define the parameter $d(\mathcal{X})$ as the following:
If $\Theta(\mathcal{X}) \neq |\ X\ | -1$, the parameter $d(\mathcal{X})$ is the minimum number of dependent points on distinct lines of $\mathcal{X}$; not including the cases for which there is a hyperplane of $\Theta(\mathcal{X})$ which satisfy both:

- contains all the lines of $\mathcal{X}$ which do not contain the dependent points
- contains all the dependent points

If $\Theta(\mathcal{X}) =\mid \mathcal{X} \mid -1$ the parameter $d(\mathcal{X})$ is the minimum d for which there is a hyperplane of $\Theta(\mathcal{X})$ containing $\mid \mathcal{X} \mid -d$ lines of $\mathcal{X}$. In other words, it is the minimum number of dependent points that can be found on distinct lines of $\mathcal{X}$.

Remark. *In the first condition where $\Theta(\mathcal{X}) \neq\mid \mathcal{X} \mid -1$, the parameter $d(\mathcal{X}) = r$ means that r is minimal such that there is a tuple of dependent points $\{x_1, \ldots, x_r\}$, with each element x_i incident with a line $l_i \in \mathcal{X}$ and the lines l_i's are distinct. Furthermore, there is no hyperplane containing the lines $\mathcal{X} - \{l_1, \ldots, l_r\}$ and the points $\{x_1, \ldots, x_r\}$.*

From now on we only consider stabilizer codes that induce quantum lines.

Theorem 3.2.3. *(Ball et al. (2021))*

There is a $[[n, k, d]]$ stabilizer code if and only if there is a quantum tuple of lines $\mathcal{X}$ for which $d(\mathcal{X}) = d$ and $\Theta(\mathcal{X}) = PG(n-k-1, 2)$.

Proof. In order to prove the theorem we only have to prove that the minimal distance of code d is $d(\mathcal{X})$. The rest of the statement is covered by Theorem 3.1.2.
" $\Rightarrow$ " : Assume there is a $[[n, k, d]]$ stabilizer code Q(S) where S is a stabilizer.
Let $C = \tau(S)$ and let G be the $(n-k) \times 2n$ generator matrix of C. Furthermore, let

$$\mathcal{X} = \{l_j \mid j = 1, \ldots, n\}$$

be the tuple of lines generated by the jth and the (j+n)th columns of G. If $k \neq 0$, then the minimal distance d is the minimum symplectic weight of $C^{\perp_a} - C$. Let $v \in C^{\perp_a}$ with symplectic weight w, also we define W as

$$W = \{j \in \{1, \ldots, n\} \mid (v_j, v_{j+n} \neq (0, 0))\}$$

Note that $\mid W \mid = w$. Let x_j be the jth column of G. With $v = [v_1, \ldots, v_{2n}]$ in $C^{\perp_a}$ we have

$$\sum_{j \in W} (v_{n+j} x_j - v_j x_{j+n}) = 0 \tag{3.1}$$

Each summand in the previous sum corresponds to a point in l_j since it is a linear combination of x_j and x_{j+n}. Since the sum is equal to zero, this provides us with $w =\mid W \mid$ dependent points each one laying in a distinct line l_j where $j \in W$.
These dependent points are only considered when $v \notin C$ since d is the minimum symplectic weight of $C^{\perp_a} - C$.

We characterize elements of C as the elements written as $v = aG$ for some $a \in \mathbb{F}_2^{n-k}$. Therefore, $v_j = a \cdot x_j$ for all $j = 1, \ldots, 2n$.
Let j be a position of v such that $j \notin W$, then $v_j = a \cdot x_j = 0$ and $v_{j+n} = a \cdot x_{j+n} = 0$. Furthermore, $v_j = a \cdot x_j = 0$ and $v_{j+n} = a \cdot x_{j+n} = 0$ if and only if the line l_j is incident with the hyperplane $\pi_a : a \cdot X = 0$. Therefore,

$$\{l_j \mid j \in (\{1, \ldots n\} - W)\} \subseteq \pi_a$$

Now let $j \in W$, then

$$a \cdot (v_{n+j}x_j - v_j x_{j+n}) = v_{n+j}(a \cdot x_j) - v_j(a \cdot x_{j+n}) = v_{n+j}v_j - v_j v_{n+j} = 0$$

Therefore, these dependent points defined by $v_{n+j}x_j - v_j x_{j+n}$ are incident with the hyperplane π_a. For the dependent points that are obtained from an element with a weight equal to d, the description above coincides with the definition of $d(\mathcal{X})$.
Now, assume that $k = 0$. Then d is the minimum non-zero symplectic weight of C. Let $v \in C$ such that its symplectic weight is equal to d. Since $v \in C$, then $v = aG$ with $a \in \mathbb{F}_2^{n-k}$. Therefore, $v_j = a \cdot x_j$ for all $j \in \{1, \ldots, 2n\}$.
We define the set W as in the previous case (i.e. the set of positions that contribute to the symplectic weight of v).
Assume now that $j \notin W$, then $a \cdot x_j = a \cdot x_{j+n} = 0$. This means that the line l_j is incident with the hyperplane π_a (Note that π_a is the zero locus of $a \cdot X$).
Hence, π_a is incident with $\mid \mathcal{X} \mid -d$ lines of $\mathcal{X}$. Recall that this is exactly the definition of $d(\mathcal{X}$ when k is 0.

" $\Rightarrow$ " : Assume there is a quantum tuple of lines $\mathcal{X}$ for which $d(\mathcal{X}) = d$ and $\Theta(\mathcal{X}) = PG(n-k-1, 2)$.
We define G as the $(n-k) \times 2n$ generator matrix of a code C. The matrix G is constricted by taking the ith and (i+n)th columns to be two points spanning the ith line in $\mathcal{X}$. We then define S as $\tau^{-1}(C)$ and Q(S) to be our stabilizer code. By Theorem 3.1.2. we see that Q(S) is a $[[n, k, d]]$ stabilizer code, since $\Theta(\mathcal{X}) = PG(n-k-1, 2)$. □

Example 8 (Shor's code)**.** In order to illustrate some of the geometrical aspects we have discussed so far, we will apply this knowledge to one of the first codes we encountered in Chapter 1; Shor's code.
Recall that Shor's code was described as follows; The coding space was $(\mathbb{C}^2)^{\otimes 9}$ and a qubit is encoded as:

$$|\alpha\rangle \longrightarrow |\alpha_L\rangle$$

Where:

$$|0_L\rangle = (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle)$$

$$|1_L\rangle = (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle)$$

Consider the following elements of $\mathcal{P}_9$

$$M_1 = ZZIIIIIII \qquad M_2 = IZZIIIIII \qquad M_3 = IIIZZIIII$$
$$M_4 = IIIIZZIII \qquad M_5 = IIIIIIZZI \qquad M_6 = IIIIIIIZZ$$
$$M_7 = XXXXXXIII \qquad M_8 = IIIXXXXXX$$

Applying τ to these operator gives us

$$\tau(M_1) = (000000000 \mid 110000000) \qquad \tau(M_2) = (000000000 \mid 011000000)$$
$$\tau(M_3) = (000000000 \mid 001100000) \qquad \tau(M_4) = (000000000 \mid 000110000)$$
$$\tau(M_5) = (000000000 \mid 000011000) \qquad \tau(M_6) = (000000000 \mid 000000011)$$
$$\tau(M_7) = (111111000 \mid 000000000) \qquad \tau(M_8) = (000111111 \mid 000000000)$$

Therefore, the generator matrix G(S) of the code $C = \tau(S)$ is

$$G(S) = \left(\begin{array}{ccccccccc|ccccccccc} 0&0&0&0&0&0&0&0&0 & 1&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 & 0&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 & 0&0&1&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 & 0&0&0&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 & 0&0&0&0&1&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0 & 0&0&0&0&0&0&0&1&1 \\ 1&1&1&1&1&1&0&0&0 & 0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&1&1&1&1 & 0&0&0&0&0&0&0&0&0 \end{array}\right)$$

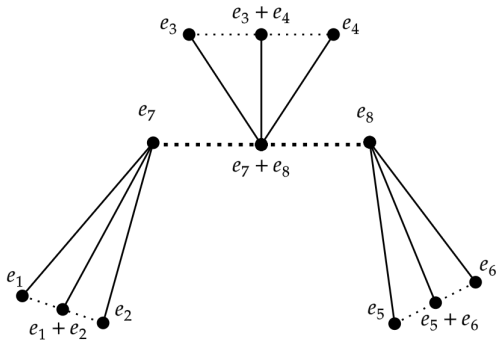
Now let $\{e_i\}_i$ be the canonical of $\mathbb{F}_2^8$. Then, from the matrix G(S) we deduce that the tuple of quantum lines $\mathcal{X}$ for this code is

$$\{\langle e_1, e_7\rangle, \langle e_1 + e_2, e_7\rangle, \langle e_2, e_7\rangle, \langle e_3, e_7 + e_8\rangle, \langle e_3 + e_4, e_7 + e_8\rangle,$$

$$\langle e_4, e_7 + e_8\rangle, \langle e_5, e_8\rangle, \langle e_5 + e_6, e_8\rangle, \langle e_6, e_8\rangle\}$$

where $\langle e_i, e_j\rangle$ is the line generated by the points e_i and e_j. We also illustrate these points and lines in the figure below.

Figure 3.1 Tuple of quantum lines coming from Shor's code



At first glace one notices that e_7 is incident with $\langle e_1, e_7\rangle$ and $\langle e_1+e_2, e_7\rangle$, taking the dependent point to be e_7 from both lines. Therefore, one can try to prove that $d(\mathcal{X}) = 2$. But the rest of the lines in $\mathcal{X}$ span a six dimentional space, since the two planes $\langle e_3, e_4, e_7+e_8\rangle$ and $\langle e_5, e_6, e_8\rangle$ span a 5 dimentional space and the line $\langle e_2, e_7\rangle$ extends it to a six dimentional space that contains the point e_7 containing all dependent points (i.e. $\{e_7\}$). Hence, the case with these dependent points (i.e. $\{e_7\}$) is not counted as in Theorem 3.1.3.

Now consider the dependent points $\{e_1, e_2, e_1+e_2\}$, the six lines that are not incident with these points are included in the hyperplane $\langle e_3, e_4, e_5, e_5, e_7, e_8\rangle$. Note that this hyperplane is not incident with $\{e_1, e_2, e_1+e_2\}$. Hence, $d(\mathcal{X}) = 3$.

Shor's code seems to be a special case in which a code come from a planar pencils of lines. We start by defining a planar pencil of lines.

Definition 3.2.3. A planar pencil of lines in a projective space is defined as a set of lines that are all contained in some plane and are concurrent.

Lemma 3.2.4. *(Ball et al. (2021))*

The union modulo two of two quantum tuples of lines is a quantum tuple of lines.

Proof. Let $\mathcal{X}$ and $\mathcal{Y}$ be two quantum tuples of lines. We denote by $\Theta(\mathcal{X}), \Theta(\mathcal{Y})$ and $\Theta(\mathcal{X}\cup\mathcal{Y})$ the spaces spanned by the following tuples of lines respectively $\mathcal{X}, \mathcal{Y}$ and $\mathcal{X}\cup\mathcal{Y}$.
Now let π be a co-dimension 2 subspace of $\Theta(\mathcal{X})$. Therefore, π intersects with $\Theta(\mathcal{X})$ in either a co-dimension 2 subspace, in a hyperplane, or in $\Theta(\mathcal{X})$. The first case results in π being skew with an even number of lines of $\mathcal{X}$, and the other two cases imply that π is skew with no lines of $\mathcal{X}$. Hence, in all cases, π is skew to an even number of lines.

We also denote by $\bar{\mathcal{X}}$ the sub-tuple of lines skew to π in $\mathcal{X}$. Similarly, we define $\bar{\mathcal{Y}}$ as the sub-tuple of lines skew with π in $\mathcal{Y}$. After we take the union modulo two of the tuples of lines $\mathcal{X}$ and $\mathcal{Y}$, we can conclude that π is skew to $|\bar{\mathcal{X}}| + |\bar{\mathcal{Y}}| - 2|\bar{\mathcal{X}} \cap \bar{\mathcal{Y}}|$ many lines. (Note that this number is even)
Hence, all co-dimension 2 subspaces are skew to an even number of lines of $\mathcal{X} \cup \mathcal{Y}$. The lemma then follows. □

Lemma 3.2.5. *(Ball et al. (2021)) Let $\mathcal{X}$ be a quantum tuple of lines. There is a tuple D of dependent points such that each point of D is incident with a different line of $\mathcal{X}$.*

Proof. Let $\pi = \Theta(\mathcal{X})$ be the subspace spanned by the lines in the tuple $\mathcal{X}$. Let $l \in \mathcal{X}$ then we define $\pi' = \Theta(\mathcal{X} - \{l\})$ as the subspace spanned by the lines in $\mathcal{X}$ that are distinct from l. π' is then one of three: a co-dimension 2 subspace of π, a hyperplane of π, or π itself.
Since $\mathcal{X}$ is a quantum tuple of lines, any co-dimension 2 subspace is skew to an even number of lines of $\mathcal{X}$. This implies that $(\mathcal{X} - \{l\})$ cannot be a co-dimension 2 subspace since it is only skew to l.
We can construct a tuple of dependent points incident with distinct lines of $\mathcal{X}$ because every point in $\pi_\prime$ is the linear combination of points incident with the lines in $(\mathcal{X} - \{l\})$. In this tuple of points, if two of them are incident with the same line $l' \in \mathcal{X}$, then we can replace y and z by $l' - \{y, z\}$.
Finally, we get a tuple of dependent points, with each point incident with a different point of $\mathcal{X}$. □

Lemma 3.2.6. *(Ball et al. (2021)) A quantum tuple of three lines is a planar pencil of lines.*

Proof. Let $\mathcal{X} = \{l_1, l_2, l_3\}$ be a quantum tuple of lines. If $\mathcal{X}$ spans either PG(4,2) or PG(5,2) respectively. Then, the co-dimension 2 subspace spanned by l_1 and $x \in l_2$ or by l_1 and l_2 respectively is skew to l_3. This is a contradiction to the fact that a co-dimension 2 subspace is skew to an even number of lines in the tuple of quantum lines $\mathcal{X}$
Now assume that $\langle l_1, l_2, l_3 \rangle = PG(3, 2)$. If l_1 and l_2 are intersecting with a co-dimension 2 subspace (line), then they are both intersecting l_3 because $\mathcal{X}$ is a tuple of quantum lines as in the previous case.
Now, we proceed with a counting argument. There are 9 lines that intersect both l_1 and l_2 since both lines are incident with 3 points. These 9 lines are all intersecting l_3. Furthermore, we know that every point in $\{l_1, l_2, l_3\} \subset PG(3, 2)$ is incident with 7 lines, 4 of these lines are 3 lines out of the 9 lines we mentioned plus one line from $\mathcal{X}$.

Hence, the total number of lines we have is larger than or equal to $9(7-4)+3+9=39$. This is a contradiction since the lines of $PG(3,2)$ are 35 by Appendix lemma ??.
Therefore the only left case is that the space spanned by $\mathcal{X}$ is $PG(2,2)$. Since a co-dimension 2 subspace of $PG(2,2)$ is a point, that implies that the lines in $\mathcal{X}$ are concurrent. Furthermore, the lines in $\mathcal{X}$ are incident with all the points in the plane. Hence, $\mathcal{X}$ is a planar pencil of lines. □

Definition 3.2.4. We define an r-sputnik as a set of (r+1) concurrent lines in an r-dimensional subspace S such that r of these lines span S

Lemma 3.2.7. *(Ball et al. (2021))*

An r-sputnik is the union modulo 2 of planar pencils of lines. Hence, an r-sputnik is a quantum tuple of lines.

Proof. Since the case where r is 1 is trivial, assume this is true for an (r-1) sputnik. And let $\mathcal{X}$ be an r-sputnik and let $l_1, l_2 \in \mathcal{X}$. The (r-1)dimensional subspace spanned by $\mathcal{X} - \{l_1, l_2\}$ intersects the plane spanned by l_1 and l_2 in a line which we call l. The line l is the third line in the planar pencil $\{l, l_1, l_2\}$. Hence, adding this planar pencil of lines modulo 2 to $\mathcal{X}$ we get an (r-1) sputnik. Therefore, an r sputnik is the union modulo 2 of planer pencils of lines and by lemma 3.1.4 an r sputnik is a quantum tuple of lines. □

Theorem 3.2.8. *(Ball et al. (2021))*

A qubit stabilizer code with a minimum distance of at least three is equivalent to a quantum tuple of lines generated by the union modulo two of planar pencils of lines.

Proof. Let $\mathcal{C}$ be a quantum tuple of lines. Then by lemma 3.1.5 there is a minimal tuple of dependent points $x_1, \ldots, x_{r_1}$ of dependent points incident with the lines $l_1, \ldots, l_{r+1}$, respectively. Consider $x \in l_{r+1} - \{x_{r+1}\}$, and let l'_j be the line spanned by the point x and x_j for all j. Then, define $\mathcal{X}'$ as

$$\mathcal{X}' = \{l'_j \mid j \in \{1, \ldots, r+1\}\}$$

Let $\mathcal{L}_j$ be the planar pencil of lines spanned by l_j and l'_j. Then using the lemma 3.1.4 the union modulo 2 of

$$(\cup_{j+1}^{r} \mathcal{L}_j) \cup \mathcal{X} \cup \mathcal{X}'$$

is a quantum tuple of lines (it is a union modulo 2 of quantum tuples of lines). We also conclude that the tuple has $\mid \mathcal{X} \mid -1$ □

3.3 Some constructions

This section aims to provide some constructions that provide some interesting results. We start with an example to illustrate that:

Example 9. To illustrate the result in the next theorem we use the following two examples where we provide an argument for the minimal distance of the constructed code.
Let,

$$G_1 = \left(\begin{array}{cc|cc} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{array}\right), G_2 = \left(\begin{array}{cccc|cccc} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

The matrix G_1 and G_2 yield the following stabilizers respectively

$$S_1 = \langle ZX, XZ \rangle, S_2 = \langle ZXII, XZII, IIZX, IIXZ \rangle$$

This then gives us a $[[2,0,2]]$ and a $[[4,0,2]]$ code respectively.
These parameters are not arbitrary as we shall see in the next theorem, for now, we provide an argument for both minimal distances.
For $Q(S_1)$, the generators of the stabilizer have weight 2. Furthermore, the only non-identity element of S_1 remaining is YY which has weight 2 as well. By Theorem 2.2.2, the minimal distance is 2.
For the code $Q(S_2)$, we notice that the generator matrix of this code G_2 seems like two copies of G_1 and we use that to obtain that the minimal distance is once again 2. It is trivial that the minimal weight of any element of S_2 is 2 indeed.

Theorem 3.3.1. *Let $[[n,0,d]]$ be a code, then a $[2n,0,d]$ code exists.*

Proof. This theorem is a generalization of example 10.
Let A, B be two $n \times n$ matrices and O the $n \times n$ identity matrix. Let $[[n,0,d]]$ be a stabilizer code, and let $G = (A \mid B)$ be the generator matrix of this code. We claim that the matrix

$$\bar{G} = \left(\begin{array}{ccc|ccc} A & \vdots & O & B & \vdots & O \\ O & \vdots & A & O & \vdots & B \end{array}\right)$$

generates a $[[2n,0,d]]$ stabilizer code. If $\bar{G}$ is proven to generate a stabilizer code (i.e. its rows have alternating form zero) then we prove that the minimal distance is

d. Note that the minimal distance here refers to the minimal weight of the elements of $C^{\perp}$ which is C.
Any row of $\bar{G}$ would be of the form $(a \mid 0 \mid b \mid 0)$ or $(0 \mid a \mid 0 \mid b)$. Hence, the nullity of the alternating form is inherited from G.
Therefore, we deduce that $\bar{G}$ yields a $[[2n, 0, d']]$ stabilizer code.
The minimal distance is still d since we can embed the n lines induced from G in two disjoint, isomorphic subspaces of $PG(2n-2k-1)$. The two embedding map the tuple of lines $\mathcal{X}$ to two isomorphic copies of it. Hence, the construction of the dependency lines in the quantum tuple of lines remains the same. □

The last theorem created a new motivation to obtain such augmentation of codes where the dimension of the system is less than 2n while still retaining the same minimal distance.

Now consider this example

Example 10. In this example, we illustrate how one can get different codes from the same tuple of lines. Let $\mathcal{X} = \{\langle e_4 + e_5, e_1\rangle, \langle e_3, e_2 + e_5\rangle\}$ be a tuple of lines in $PG(4,2)$. Then, we can construct the following matrices:

$$G_2 = \left(\begin{array}{cccccc|cccccc} 0&0&0&1&0&0&1&0&0&0&1&1\\ 0&0&1&0&0&0&0&1&0&0&0&0\\ 0&1&0&0&0&0&0&0&1&0&0&0\\ 1&0&0&0&1&1&0&0&0&1&0&0 \end{array}\right)$$

$$G_3 = \left(\begin{array}{cccccccc|cccccccc} 0&0&0&1&0&0&0&1&1&0&0&0&1&0&0&0\\ 0&0&1&0&0&0&1&0&0&1&0&0&0&1&0&0\\ 0&1&0&0&0&1&0&0&0&0&1&0&0&0&1&0\\ 1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&1 \end{array}\right)$$

These matrices yield the following stabilizers:

$$S_2 = \langle ZIIXZZ, IZXIII, IXZIII, XIIZXX\rangle$$

$$S_3 = \langle ZIIXZIIX, IZXIIZXI, IXZIIXZI, XIIZXIIZ\rangle$$

These codes produce the following codes accordingly: $[[6,2,2]]$, $[[8,4,2]]$
Note that the minimal distance is the same for both these codes and we shall see why in theorem 3.1.4.

This theorem that we present now attempts to generalize example 10 where we construct minimal distance 2 codes in arbitrary dimensions.

Theorem 3.3.2. *Let $\mathcal{X}$ be a quantum tuple of lines for which $\Theta(\mathcal{X}) = PG(n-k-1,2)$. Then for every t, there is a $[[n+2t, k+2t, 2]]$*

Proof. Let $l \in \mathcal{X}$ and let G be a matrix obtained from the tuple $\mathcal{X}'$ a quantum tuple of lines equal to $\mathcal{X}$ plus the line l added 2t-times. Note that l will be counted $2t+1$ times as a line in $\mathcal{X}'$.
Now, we can easily construct a pair of linearly dependent points (a, b) such that $a = b \in l$. The fact that l is repeated at least 3 times in the tuple $\mathcal{X}'$ implies that there are no hyperplanes containing the other lines of $\mathcal{X}'$ while not containing l. Hence, the minimal distance is indeed 2. □

While the theorem might seem trivial, it does introduce a degeneracy in the geometry of stabilizer codes. Such repetition weren't considered previously and many theorem may fail when considering such cases when we have line repetition.

4. Conclusion

It should be clear to the reader that the study of the geometry of quantum stabilizer codes is a new topic that will be seeing many advancements in the future. Quantum stabilizer codes with arbitrary parameters is still an open question, the smallest unknown stabilizer code has the configuration $[[14, 3, 5]]$. This fact raises many questions.

- Can we construct other quantum stabilizer codes inductively that have monotonic minimal distance?
- How can stabilizer codes be generalized?
- Can stabilizer codes be obtained from parametrizing other geometrical structures?

The second question was partially answered by Ball et.al where two generalizations were introduced. The first one was by considering qudits (objects similar to qubits but with larger alphabets) instead of qubits which we won't discuss here. The second one is taking the direct sum of quantum stabilizer codes which produces a class of non-additive codes. We discuss this case briefly in the following subsection.

4.1 Direct sum of quantum stabilizer codes

As the title of this section suggests, we discuss some non-additive quantum codes that are constructed by taking the direct sum of quantum stabilizer codes. We recall that we construct quantum stabilizer codes by taking the intersection of $(+1)$-eigenspace of $n-k$ independent operators $M_1, M_2, \ldots, M_{n-k}$ from $\mathcal{P}_n$.

We also observe that $\pm M_1, \pm M_2, \ldots, \pm M_{n-k}$ commute if and only if

$M_1, M_2, \ldots, M_{n-k}$. Therefore, for $t = (t_1, t_2, \ldots, t_{n-k}) \in \mathbb{F}_2^{n-k}$ one can define a quantum stabilizer code $Q(S_t)$ as the intersection of the $(+1)$-eigenspace of the following operators

$$(-1)^{t_1} M_1, (-1)^{t_2} M_2, \ldots, (-1)^{t_{n-k}} M_{n-k}$$

Where,

$$Q(S_t) = \langle\ (-1)^{t_1} M_1, (-1)^{t_2} M_2, \ldots, (-1)^{t_{n-k}} M_{n-k}\ \rangle$$

Lemma 4.1.1. *(Ball & Puig (2021))*

Let $u, t \in \mathbb{F}_2^{n-k}$.
If there is a $j \in \{1, 2, \ldots, n-k\}$ with $u_j \neq t_j$ then, $Q(S_t)$ and $Q(S_u)$ are orthogonal.

Proof. Assume there is a $j \in \{1, 2, \ldots, n-k\}$ with $u_j \neq t_j$.
WLOG, assume that $t_j = 1$. Let $|a\rangle \in Q(S_t)$, $|b\rangle \in Q(S_u)$. Then

$$\langle a|b\rangle = \langle a|M_j b\rangle = \langle M_j a|b\rangle = -\langle a|b\rangle$$

Hence, $|a\rangle$ and $|b\rangle$ are orthogonal. □

Definition 4.1.1. Let T be a subset of $\mathbb{F}_2^{n-k}$. We define a direct sum stabilizer code also known as a union stabilizer code as

$$Q(S_T) = \bigoplus_{t \in T} Q(S_t)$$

We start the study of direct sum stabilizer codes by determining which errors are not detectable.

Definition 4.1.2. (Ball & Puig (2021))

Let S be a stabilizer, and G be the generator matrix of $C = \tau(S)$.
Let $t, u \in T - \{0\}$, and let $A_{t,u}$ be a $(n-k) \times (n-k)$ non-singular matrix with first two columns u and t. Then $A_{t,u}^{-1} G$ is also a generator matrix for the code C.
Subsequently, one can construct another set of generators of S

$$\{M_1', M_2', \ldots, M_{n-k}'\}$$

where M_i' is obtained from applying the map τ^{-1} to the ith row of $A_{t,u}^{-1} G$. We also call the subgroup of S generated by $\{M_3', \ldots, M_{n-k}'\}$, $S_{t,u}$.

Lemma 4.1.2. *(Ball & Puig (2021))*

Let $\left|\psi^t\right\rangle \in Q_t(S)$ *and* $|\psi^u\rangle \in Q_u(S)$*. Then, for all* $M \in S_{t,u}$*,*

$$M\left|\psi^t\right\rangle = \left|\psi^t\right\rangle \qquad \textit{and} \qquad M\,|\psi^u\rangle = |\psi^u\rangle$$

Proof. We notice that $Q_t(S)$ depends on the generators of S. Consider

$$S = \langle M'_1, \ldots, M'_{n-k}$$

$Q_t(S)$ is then $Q_{(1,0,\ldots,0)}(S)$ and $Q_u(S)$ is $Q_{(0,1,\ldots,0)}$.
Therefore, $M\left|\psi^t\right\rangle = \left|\psi^t\right\rangle$ and $M\,|\psi^u\rangle = |\psi^u\rangle$ for all $j = 3, \ldots, n-k$ □

Lemma 4.1.3. *(Ball & Puig (2021))*

Suppose $Q(S_T)$ *is unable to detect an error E. Then there is a pair* $t, u \in T$ *such that* $E \in C_{\mathcal{P}_n}(S_{t,u})$*.*

Proof. Suppose there is no pair $t, u \in T$ such that $E \in C_{\mathcal{P}_n}(S_{t,u})$. Then, for all $t, u \in T$ there is $M_{t,u} \in S_{t,u}$ such that E anti-commutes with $M_{t,u}$.
Let $\left|\psi^t\right\rangle \in Q_t(S)$ and $|\psi^u\rangle \in Q_u(S)$ be elements in an orthogonal basis of $Q(S_T)$. By lemma 4.1.2,

$$M_{t,u}\left|\psi^t\right\rangle = \left|\psi^t\right\rangle \quad \text{and} \quad M_{t,u}\,|\psi^u\rangle = |\psi^u\rangle$$

Therefore,

$$\left\langle\psi^t\right| E\,|\psi^u\rangle = \left\langle\psi^t\right| EM_{t,u}\,|\psi^u\rangle = -\left\langle\psi^t\right| M_{t,u}E\,|\psi^u\rangle = -\left\langle\psi^t\right| E\,|\psi^u\rangle$$

This implies that $\left\langle\psi^t\right| E\,|\psi^u\rangle = 0$. Therefore by Kill-Laflamme theorem E is detectable.

□

By lemma 3.1.1 and lemma 4.0.3, an error that is not detectable is an element of $C_{\mathcal{P}_n}(S_{t,u})$ for any $t, u \in T$.

Definition 4.1.3. (Ball & Puig (2021))

We define d_T as:

$$d_T = min\{d_{t,u} \mid t, u \in T\}$$

where $d_{t,u}$ is the minimum weight from the Pauli operators in $C_{\mathcal{P}_n}(S_{t,u})$

Theorem 4.1.4. *(Ball & Puig (2021))*

The subspace $Q(S_T)$ is an $((n, \mid T \mid 2^k, d_T))$ quantum code.

Proof. The theorem follows automatically from the parts above. Note that if E is undetectable then $E \in C_{\mathcal{P}_n}(S_{t,u})$ for some $t, u \in T$. □

Example 11. (The Rains, Hardin, Shor, Sloane non-additive quantum code)

This code was first introduced by Rains et al. In this example we present the code along side its geometric interpretation. Note that the errors of the code are elements of $\mathcal{P}_5$.

$$\begin{cases} M_1 = Z\ X\ Y\ Y\ X \\ M_2 = X\ Z\ X\ Y\ Y \\ M_3 = Y\ X\ Z\ X\ Y \\ M_4 = Y\ Y\ X\ Z\ X \\ M_5 = X\ Y\ Y\ X\ Z \end{cases}$$

We then construct the generator matrix G with rows $\tau(M_i)$.

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

If we delete any two rows of the matrix, we obtain a 3×10 matrix whose 5 pairs of columns define a quantum tuple of lines in $PG(2,2)$. This quantum tuple of lines gives us a stabilizer code with a minimum distance of 2.
Hence, if we let T be the following set

$$T = \{, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$$

then, by Theorem 4.1.4, $Q(S_T)$ is a $((5,6,2))$ quantum code.

The geometrical interpretation of the direct sum of stabilizer codes follows quite naturally from the geometrical interpretation of stabilizer codes.

We start by restricting the elements of T to singleton subsets and the empty set. Now, let $\mathcal{X}$ be the quantum tuple of lines of $PG(n-k-1,2)$ generating the

$[[n,k,d]]$ quantum stabilizer code $Q(S)$, where S is the stabilizer group generated by$M_1,\ldots,M_{n-k}$.
Let $P=\{e_1,\ldots,e_r\}$ be a tuple of linearly independent points in $PG(n-k-1,2)$, such that the projection from any two points $e_i,e_j\in P$ of the lines of $\mathcal{X}$ is a tuple of lines in $PG(n-k-3,2)$.
If this projection is a tuple of lines, it is then a quantum tuple of lines. We denote such a tuple of quantum lines by $\mathcal{X}_{ij}$.

Definition 4.1.4. (Ball & Puig (2021))

Let $d(\mathcal{X}_{ij})$ be the size of the smallest tuple of dependent points incident with distinct lines of $\mathcal{X}_{ij}$. We define the parameter d_T by:

$$d_T=\min\{d(\mathcal{X}_{ij})\mid i,j\in\{1,\ldots,r\}\}$$

Remark. *The previous geometric construction yields a* $((n,(r+1)2^k,d_T))$ *given that* $r\leq n-k$

BIBLIOGRAPHY

Ball, S., Centelles, A., & Huber, F. (2021). Quantum error-correcting codes and their geometries.

Ball, S. & Puig, P. (2021). The geometry of non-additive stabiliser codes.

Berrou, C., Glavieux, A., & Thitimajshima, P. (1993). Near shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Proceedings of ICC '93 - IEEE International Conference on Communications*, volume 2, (pp. 1064–1070 vol.2).

Gallager, R. (1962). Low-density parity-check codes. *IRE Transactions on Information Theory*, *8*(1), 21–28.

Gottesman, D. (1997). Stabilizer codes and quantum error correction.

Grassl, M. (2007). Bounds on the minimum distance of linear codes and quantum codes. Online available at `http://www.codetables.de`. Accessed on 2023-06-04.

Hamming, R. W. (1950). Error detecting and error correcting codes. *The Bell System Technical Journal*, *29*(2), 147–160.

Knill, E., Laflamme, R., & Viola, L. (2000). Theory of quantum error correction for general noise. *Physical Review Letters*, *84*(11), 2525–2528.

Nielsen, M. A. & Chuang, I. L. (2010). *Quantum Computation and Quantum Information*. Cambridge university press.

Reed, I. S. & Solomon, G. (1960). Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, *8*(2), 300–304.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*(3), 379–423.

Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *26*(5), 1484–1509.

Steane (1998). Quantum computing. *Reports on Progress in Physics*, *61*(2), 117.

Viterbi, A. (1971). Convolutional codes and their performance in communication systems. *IEEE Transactions on Communication Technology*, *19*(5), 751–772.

APPENDIX A : Computational tools (Sage + GAP)

For **Example 4** we use a function from Sage to calculate tensors of operators and later calculate the basis of the joint eigenspace. The code is as follows:

```
Id = matrix([[1,0],[0,1]])
X = matrix([[0,1],[1,0]])
Y = matrix([[0,-I],[I,0]])
Z = matrix([[1,0],[0,-1]])

V = VectorSpace(CC,8)

I3 = Id.tensor_product(Id).tensor_product(Id)
M1 = X.tensor_product(Id).tensor_product(Z)
M2 = Y.tensor_product(Y).tensor_product(X)
M12 = - Z.tensor_product(Y).tensor_product(Y)

P = I3 + M1 + M2 + M12
rows = P.rows()
QS = V.subspace(rows)
QS_basis = []

for b in QS.basis():
    b = [int(abs(p)) for p in b]
    b1 = tuple(b)
    b = b1
    QS_basis.append(b)

print(QS_basis)
```

Fucnctions used in the calculations; these are some of the functions that were built while working on the thesis topic. All function were built using (Python+Sage) and were routinely used throughout this thesis.

```
#This function returns the dimension n based on a vector that
                                    represents an operator.

def determine_n (w):
    if len(w) % 2 == 0:
        return len(w)/2
    else:
        return f" this is not a valid vector!"


#This function calculate the symplectic form of two given vectors
                                    of even length.
def symplectic_form(w,v):
    s = 0
    n = determine_n(w)
    for i in range(n):
        s += w[i]*v[n+i] - w[n+i]*v[i]
    return s

#This function takes a string representing an operator (i.e. XYYXI)
                                     and returns the vector
                                    corresponding to it.

def make_vector(st):
    st1 = [p.upper() for p in st]
    st = st1
    l = []
    r = []
    for i in range(len(st)):
        if st[i] == 'X':
            l.append(1)
            r.append(0)
        elif st[i] == 'Y':
            l.append(1)
            r.append(1)
        elif st[i] == 'Z':
            l.append(0)
            r.append(1)
        elif st[i] == 'I':
            l.append(0)
            r.append(0)
        else:
            print(f"We have a problem")
```

```
            break
    return vector(GF(2), l+r)


#This function takes a list of strings representing the potential
                                    stabilizer elements and then
                                    return whether they make a
                                    stabilizer of not (i.e. quantum
                                    code)

def check_if_code(ls):
    ls1 = [make_vector(p) for p in ls]

    for i in range(len(ls1)):
        for j in range(len(ls1[i:])):
            if symplectic_form(ls1[i],ls1[i+j]) != 0:
                return f"Not a quantum code! the problimatic
                                                    positions are {i+
                                                    1} and {j+1}"

    return f"It is a quantum code"



#This function takes a list of strings for the stabilizer
                                    generators, and return the
                                    generator matrix corresponding to
                                     it.

def gen_matrix(st):
    st1 = [make_vector(p) for p in st]
    G = matrix(GF(2), st1)
    return G


# This function return k from the generator matrix of a code [[n, k
                                    , d]].
def dim_code(G):
    r = G.rank()
    return r


#This function return the quantum set of lines coming from the
                                    columns of the generator matrix.

def chi_lines_gen(G):
    n = len(G[0])/2
```

```
    c = G.columns()
    l = c[:n]
    r = c[n:]
    chi_gen = [[l[i], r[i]] for i in range(n)]
    return chi_gen

#This function takes a generator matrix and returns the generators
                                    of the stabilizer of a code as a
                                    list of strings.

def matrix_to_stabilizer(st):
    S = []
    n = len(st[0]) / 2
    for j in range(len(st)):
        xyz = []

        for i in range(n):
            a_xyz = ""
            if st[j][i] == 0 and st[j][i+n] == 0:
                a_xyz += "I"
            elif st[j][i] == 1 and st[j][i+n] == 0:
                a_xyz += "X"
            elif st[j][i] == 1 and st[j][i+n] == 1:
                a_xyz += "Y"
            elif st[j][i] == 0 and st[j][i+n] == 1:
                a_xyz += "Z"
            else:
                print("error")
                break
            xyz += a_xyz

        S.append(xyz)


    return S


#This function takes the genrator matrix of a [[n,k,d]] code and
                                    returns the generator matrix of a
                                     [[n+2t,k+2t,2]] code.
def augment_code(A,t, rd):
    B = []
    n = len(A[0])/2
    for vec in A:
        a = vec
        B.append(a[:n] + [a[rd] for i in range(2*t)] + a[n:] +
        [a[n+rd] for i in range(2*t)])
```

```
    return B
```

calculating the minimal distance (GAP); this code has been provided by Pr.Simeon Michael Ball, and was quite useful to calculate the minimal distance of stabilizer codes in reasonable dimensions. We start with A which is a $n \times n$ matrix.

```
H:=Z(2)^0*A;
n:=Size(H[1])/2;
k:=n-Size(H);

Gam:=[];
for i in [1..2*n] do
    Gam[i]:=[];
    for j in [1..2*n] do
        Gam[i][j]:=0*Z(2);
                if j=((i+n-1) mod (2*n))+1 then
                    Gam[i][j]:=Z(2)^0;
                fi;

        od;
    od;

CentTrue:=NullspaceMat(TransposedMat(H*Gam));

mwt:=n;

for r in [1..(n+k)] do;
    for C in Combinations([1..n+k],r) do
                u:=0*Z(2)^0*H[1];
                for c in C do
                  u:=u+Z(2)^0*CentTrue[c];
                od;
                if CentTrue*(u*Gam)<>
            0*Z(2)*CentTrue*(u*Gam) then
                  wt:=0;
            for j in [1..n] do
                                if u[j]<>0*Z(2) or u[j+n]<>0*Z(2) then
```

```
                                        wt:=wt+1;
                                    fi;
                                od;
                                if wt<mwt then
                                        Print(u," ",C," ",wt,"\n");
                                        mwt:=wt;
                                fi;
                            fi;
                    od;
            od;

            mwt;
```