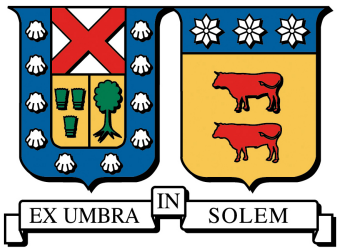


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE MATEMÁTICA

Algebraic Geometric Codes Applied to Quantum Information

Memoria de Título presentada por:

Lucas Montero

como requisito parcial para optar al título de

Ingeniero Civil Matemático

Otorgado por la Universidad Técnica Federico Santa María

Profesor Guía
Dr. Pedro Montero
Profesor Co-Guía
Dr. Dardo Goyeneche

Mes 202X.

TÍTULO DE LA TESIS:

Algebraic Geometric Codes Applied to Quantum Information.

AUTOR:

Lucas Montero

TRABAJO DE TESIS, presentado en cumplimiento parcial de los requisitos para el título de Ingeniero Civil Matemático de la Universidad Técnica Federico Santa María.

Dr. Profesor Guía _____

Dr. Profesor Co-Guía _____

Dr. Revisor 1 _____

Dr. Revisor 2 _____

Valparaíso, Chile, Mes Año.

Dedicatoria
Parrafo

Más palabras

Texto
Autor

AGRADECIMIENTOS

(Agradecimientos)

ABSTRACT

This thesis explores the intersection of algebraic geometry and quantum information theory through the study of AG-codes and their application to quantum error correction. Beginning with a review of algebraic tool needed for classical error-correcting codes and examples, we extend these concepts to the quantum setting. Stabilizer codes and the CSS construction are examined as foundational tools in quantum error correction. We then introduce key results from algebraic geometry, including the theory of algebraic curves to construct AG codes with desirable properties. Finally, we demonstrate how these codes can be adapted for quantum systems, providing explicit examples and circuits capable of correcting arbitrary single-qubit errors. this work highlights the power of geometric method in advancing the design of reliable quantum error-correcting codes with explicit quantum circuits.

Keywords: *Error-Correcting Codes, Algebraic Geometry, Quantum Information, Quantum Error-Correcting Codes (QECC), Stabilizer Codes, Algebraic Geometric (AG) Codes, Quantum Circuit.*

INDEX

INTRODUCTION

The current times of information theory has brought forward the urgent need for robust algorithms to detect and correct errors in quantum systems. While classical error-correcting codes have a well-established theoretical foundation rooted in algebra, the extension of these ideas to quantum system present new and profound challenges due to the principles of entanglement. Quantum error correction not only requires preserving fragile quantum states but also demands thecniques that respect the no-cloning theorem and the probabilistic nature of quantum measurements.

This thesis explores the application of algebraic geometry, a branch of mathematics that studies solutions (zeroes) of multivariate polynomials, to the construction of quantum error-correcting codes (QECC). In particular it focuses on how algebraic geometric (AG) codes, originally developed in the context of classical communication, can be used to build quantum codes that are capable of correcting any type of quantum errors.

Through this work, we aim to bridge the abstract mathematical structures of algebraic geometry with the practical demands of quantum error correction, highlighting not only the theoretical elegance of the approach but also its potential for real-world quantum technologies.

Chapter 1

ERROR CORRECTING CODES.

This chapter is a recollection of all the basic concepts of group theory based on the lectures from [9], which was a course taught by my advisor.

1.1 Preliminaries

In this section we focused on the basic algebraic tools that make error-correcting codes possible, we will go from binary operations in sets to fields and multilinear algebra. Everything in this chapter will be used to construct the algebraic geometric codes for chapter 4, but chapter 3 will make heavy use of this chapter?s material.

1.1.1 Group Theory

Definition 1.1.1 (Binary Operation)**.** A binary operation over a set S is a function (frequently denoted by $\cdot$)

$$\begin{aligned} \cdot : S \times S &\to S \\ (a, b) &\mapsto a \cdot b = c \end{aligned}$$

where $a, b, c \in S$.

Definition 1.1.2 (Equivalence Relation, Equivalence Class)**.** An equivalence relation is a subset K of a product of a set S that has the following properties

1. (Reflexivity) $(a, a) \in S^2$

2. (Symmetry) if $(a,b) \in S^2 \Rightarrow (b,a) \in S^2$
3. (Transitivity) if (a,b) and $(b,c) \in S^2 \Rightarrow (a,c) \in S^2$

if a pair $(a,b) \in S^2$ then it's said that a is related to b, and all of the elements related to an element a is called the equivalence class of a, denoted by $[a]$.

Definition 1.1.3 (Group)**.** A group is a pair $(G,\cdot)$ of a set G and a binary operation $\cdot$ that the following *group axioms* are satisfied

1. $\forall g,h,k \in G \,|\, g\cdot(h\cdot k) = (g\cdot h)\cdot k$
2. $\exists\, e \in G \,|\, g\cdot e = g\,, \forall\, g \in G$
3. $\forall g \in G$, $\exists g^{-1} \in G \,|\, g\cdot g^{-1} = g^{-1}\cdot g = e$

If the binary operation also satisfies the commutative property then the group is called *abelian* or commutative, if the binary operation isn't important or is assumed then the group is denoted just by the set of its elements G.

Observation 1.1.1. A pair $(S,\cdot)$ is called a semi-group if it fulfills the first group axiom (associativity), if it also fulfills the second group axiom then it is called a monoid.

Definition 1.1.4 (Group Morphism)**.** A map $f : G_1 \to G_2$ between two groups $(G_1,\cdot)$ and $G_2,\star)$ is called a group morphism if

$$f(a\cdot b) = f(a)\star f(b)$$

Definition 1.1.5 (Kernel of a Group Morphism, Image of a Group Morphism)**.** Let f be a group morphism between G,H groups, then

$$\text{Kernel of f} = \text{Ker}(f) := \{g \in G \,|\, f(g) = e_H\}$$
$$\text{Image of f} = \text{Im}(f) := \{h \in H \,|\, h = f(g), g \in G\}$$

Definition 1.1.6 (Group Monomorphism, Group Epimorphism, Group Isomorphism)**.**

Definition 1.1.7 (Subgroup, Normal Subgroup)**.** A subset H from a group G is called a subgroup if it's a group with the same binary operation and it's denoted by $H \leq G$, if $HG = GH$ then H is called a normal subgroup of G and it's denoted by $H \trianglelefteq G$.

Definition 1.1.8 (Coset, Quotient group)**.** Let $H \leq G$, then $gH = \{gh \,|\, h \in H\}$ is the left coset of H by g and $Hg = \{hg \,|\, h \in H\}$ the right coset of H by g, if the left and right coset are equal as a set for all $g \in G$ then they themselves as a set are called the quotient group $G/H = \{gH \,|\, g \in G\}$, the cardinality (number of elements in a set) of G/H is denoted by $[G : H]$

Observation 1.1.2. The right and left coset of a subgroup $H \leq G$ are equal if and only if $H \trianglelefteq G$.

Observation 1.1.3. The set G/H has a group structure with the same operation as G.

1.1.2 Field Theory

Definition 1.1.9 (Ring, Field)**.** A ring is a triad $(R, +, \cdot)$ of a set R and two binary operation $+$ and $\cdot$ satisfying the following properties

1. $(R, +)$ is an abelian group.
2. $(R, \cdot)$ is a semi-group.
3. $\forall a, b, c \in R \,|\, a \cdot (b + c) = a \cdot b + a \cdot c$

If the binary operation $\cdot$ also satisfies the commutative property then the ring is called *abelian* or commutative, if the binary operation isn't important or is assumed then the group is denoted just by the set of its elements R.
A ring R is called a field if $R \neq \{0\}$ and $(R - \{e_+\}, \cdot)$ is a group.

Definition 1.1.10 (Ideal)**.** A subset I from a ring $(A, +, \cdot)$ is called an ideal if it fulfills the following properties

1. $(I, +)$ is a subgroup of $(A, +)$.
2. $\forall\, a \in A, \forall\, b \in I \,|\, ab \in I$

Observation 1.1.4. A subring is a subset H of a ring R that it is a ring by itself with the same binary operation as R, but this structure isn't very useful for constructing quotients, we study these ideals because they are equivalent to a normal subgroup in group theory and we can construct a quotient ring with them.

Definition 1.1.11 (Quotient ring)**.** Let I and ideal of R a ring, the quotient group $(R/I, +)$ is called the quotient ring, let's see if the equivalence classes can be operated with both binary operations

1. $(a + I) + (b + I) = (a + b) + I$.
2. $(a + I) \cdot (b + I) = (ab + I)$.

we can see that the equivalence classes are well defined.

Definition 1.1.12 (Finite Field)**.** Let p be a prime and $q = p^n$ for some $n \in \mathbb{N}$, we will denote the finite field of q elements as $\mathbb{F}_q$.

Example 1.1.1. The easiest example to view is the $\mathbb{F}_2$ with the elements $(0, 1)$ and the operations $+$ and $\cdot$, then it is simple to construct the addition and multiplication tables

$+$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

then one can simply assume that the next finite field is the $(\mathbb{Z}_3, +, \cdot)$ and it would be right, but $\mathbb{F}_4$ is not isomorphic to $\mathbb{Z}_4$ with the usual addition and multiplication, we can see the tables

$+$	0	1	x	x+1
0	0	1	x	x+1
1	1	0	x+1	x
x	x	x+1	0	1
x+1	x+1	x	1	0

$\cdot$	0	1	x	x+1
0	0	0	0	0
1	0	1	x	x+1
x	0	x	x+1	1
x+1	0	x+1	1	x

here we can see that the finite field of 4 elements clearly cannot be the usual $\mathbb{Z}_4$, then the only other group of 4 elements is $\mathbb{Z}_2 \times \mathbb{Z}_2$, from here we got an idea about the structure and differences of fields with a prime number of elements and fields with a number of elements equal to a power of a prime.

Observation 1.1.5. Let p be a prime, then $\mathbb{F}_p$ is isomorphic to $\mathbb{Z}_p$ with the usual binary operations.

1.1.3 Multi-linear Algebra

Definition 1.1.13 (Vector space, Subspace)**.** A vector field is a triplet $(V, \mathbb{K}, +, \cdot)$ of a set $V = \mathbb{K}^n$ where $\mathbb{K}$ is a field with two binary operation $+$ and $\cdot$ satisfying the following properties

1. $\exists\, e \in V \,|\, \forall\, v \in V : v + e = e$.
2. $\forall\, v, w \in V \,|\, v + w \in V$.
3. $\forall v \in V \,|\, kv \in V, \forall\, k \in \mathbb{K}$

The vector space is called *over* a field $\mathbb{K}$ when the field is important to notice. A subset $U \subset V$ is called a subspace if it is a vector space over the same field and with the same binary operations.

Definition 1.1.14 (Linear Bilinear and Multilinear Map)**.** A map $f : V \to W$ between two vector spaces $(V, +, \cdot)$ and $(W, \oplus, \star)$ over the same field $\mathbb{K}$ is called a linear map if

$$\begin{aligned} f(a+b) &= f(a) \oplus f(b) \qquad \forall\, a, b \in V \\ f(k \cdot a) &= k \star f(a) \qquad \forall\, a \in V,\, \forall k \in \mathbb{K} \end{aligned}$$

Every function that fulfills these two properties is called *linear*, a bilinear map is a map from $f : U \times V \to W$ with U, V and W vector spaces over the same field $\mathbb{K}$ that satisfied the following properties

$$\begin{aligned} f((u_1, v_1) + (u_2, v_2)) &= f((u_1, v_1)) + f((u_2, v_2)) \qquad \forall\, (u_1, v_1), (u_2, v_2) \in U \times V \\ f((u, kv)) = f((ku, v)) &= f(k(u, v)) = kf((u, v)) \qquad \forall\, (u, v) \in U \times V,\, \forall k \in \mathbb{K} \end{aligned}$$

with that in mind, a multilinear map is a map $f : \Pi_{i=0}^{n} V_i \to W$ where $\{V_i\}_{i=0}^{n}$ are all vector spaces for $i \in \{0, \ldots, n\}$ over a field $\mathbb{K}$ that satisfied the properties

$$\begin{aligned} f(v + w) &= f(v) + f(w) \qquad \forall\, v, w \in \Pi_{i=0}^{n} V_i \\ f(k(v_0, \ldots, v_n)) = f((kv_0, \ldots, v_n)) &= \cdots = f((v_0, \ldots, kv_n)) = kf((v_0, \ldots, v_n)) \\ &\qquad \forall\, (v_0, \ldots, v_n) \in \Pi_{i=0}^{n} V_i,\, \forall k \in \mathbb{K} \end{aligned}$$

Theorem 1.1.1. Let $f : V \to W$ a linear map between vector spaces over the field $\mathbb{K}$, then exists $T \in \mathbb{M}_{\dim(V) \times \dim(W)}(\mathbb{K})$ (the space of matrices with dimension $\dim(V) \times \dim(W)$ with entries in $\mathbb{K}$) such that f can be represented by the linear map

$$x \mapsto Tx, \qquad \forall\, x \in V.$$

Definition 1.1.15 (Eigenvalue, Eigenvector)**.** Let $T : V \to V$ a linear map from a vector space V to itself, and let $\lambda \in \mathbb{K}$. We say that λ is an eigenvalue of T if there is a non-zero vector $v \in V$ such that $T(v) = \lambda v$. Moreover, in that case we will say that v is an eigenvector associated to the eigenvalue λ.

Definition 1.1.16 (Quotient Space)**.** Let V be a vector space and U a subspace of V then the quotient space denoted by V/U is defined as the equivalence classes of the form $[v] = v + U$, with

$$[u] = [v] \iff u - v \in U.$$

Theorem 1.1.2. Let V, W vector spaces and U a subspace of V, there exists a linear transformation $T : V \to W$ with $\text{Ker}(T) = U$.

Definition 1.1.17 (Tensor product)**.** For this text a tensor product is an operation between two matrices that output a matrix where each entry is a multiplication of entries from the first matrix with every entry from the second matrix.
A simple example would be:

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}$$

1.2 Error-Correcting Codes

Codes were invented to correct error on noisy communication channels [8]. The main idea is to formalize the notion of a language and add redundancy to messages so errors that happened in the transmission of it, can be detected and corrected.

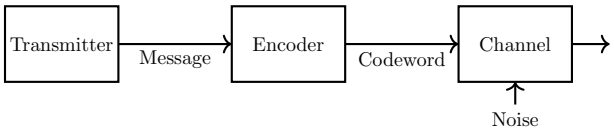


Figure 1.1. Diagram of a Message

Using this idea, we use the redundancy in a similar way we correct words in our mind. Take for example the word *Algeba*, we instinctively know that it is missing the 'r', because those words feel closer to each other, this notion of closeness is what error-correcting codes formalize, constructing languages that has words well spaced in a vector space so a computer could recognize possible errors and fix them.

1.2.1 Linear Codes

Definition 1.2.1 (Alphabet, Letters, Code, Word)**.** An arbitrary set A is called an alphabet and its elements are called letters, a code C is a subset of A^n and words are the elements in it, tuples of letters.

Example 1.2.1. A classic example can be a robot in the moon, let's say that we need to move it with the following commands:

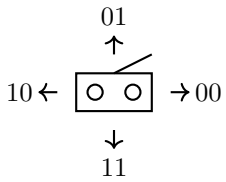


Figure 1.2. Robot in Mars

With this we have an alphabet with 0 and 1 and 4 words 00 - left, 01 - backward, 10 - right, 11 - forward.

Definition 1.2.2 (Linear Code, Generator Matrix)**.** A linear code is a subset of field $\mathbb{F}_p^n$ (with p prime) whose elements, called codewords, can be viewed as words embedded in some $\mathbb{F}_p^m$ with $m > n$ then

$$T : \mathbb{F}_p^n \hookrightarrow \mathbb{F}_p^m$$

Where T is a linear transform, then T can be represented by a matrix because of 1.1.1, this matrix is called the generator matrix of C and it's denoted by G_C.

Definition 1.2.3 (Hamming Distance, Weight of a Word)**.** The distance between two words c_1 and c_2 in the code C and length n is

$$d(x, y) = |\{i \text{ such that } 1 \leq i \leq n \text{ and } c_{1_i} \neq c_{2_i}\}|$$

and it is called the Hamming distance, the weight of a word it the Hamming distance from a word to the 0-word (the 0 vector of the space).

Definition 1.2.4 ($[n, k, d]_q$-Code)**.** A linear code C with a length of words n, dimension k and minimum Hamming distance between words d with alphabet $\mathbb{F}_q$ is denoted by $[n, k, d]_q$-code.

Definition 1.2.5 (Parity-Check Matrix, Syndrome)**.** The parity-check matrix of a $[n, k, d]_q$-code is a matrix that has the codewords as its kernel and the errors are mapped outside of it because of 1.1.2, the image of the errors are called syndromes. There is a construction for parity-check matrices when the generator matrix has a standard form i.e. $G_C = (I|A)$ then the parity-check matrix can we written as $H_C = (A^T|I)$.

Theorem 1.2.1. An $[n, k, d]_q$-code can correct up to $\lfloor \frac{d}{2} \rfloor$ errors.

Example 1.2.2. With the robot example we could say that the communication with it could be compromised by space radiation, then we need a bit of redundancy for the word to be able to be corrected, let's try constructing a $[5, 2, 3]_2$-code, let's use the matrix

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

then we get the code words:

$$\begin{aligned} (0,0) \cdot G &= (0,0,0,0,0) \\ (0,1) \cdot G &= (0,1,0,1,1) \\ (1,0) \cdot G &= (1,0,1,0,1) \\ (1,1) \cdot G &= (1,1,1,1,0) \end{aligned}$$

Note that the codewords have a minimum hamming distance of 3 between them, then we made a $[5,2,3]_2$-code and we can correct an error.
Let's say that we want the robot to move forward, we select the codeword x =(1,1,1,1,0) and send it, but in space it got changed to x' =(1,0,1,1,0), we can correct it with the parity-check matrix:

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} = (I|A)$$

$$H = (A^T|I) = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Now we need to determine the syndromes of this code to identify the possible errors we could correct, let's take (0,0,0,0,0) and do

$$\begin{aligned} H \cdot (1,0,0,0,0)^T &= (1,0,1) \\ H \cdot (0,1,0,0,0)^T &= (0,1,1) \\ H \cdot (0,0,1,0,0)^T &= (1,0,0) \\ H \cdot (0,0,0,1,0)^T &= (0,1,0) \\ H \cdot (0,0,0,0,1)^T &= (0,0,1) \\ \Rightarrow H \cdot x'^T &= (0,1,1) = e \end{aligned}$$

Note that we got the syndrome of the code word that is linked to the error (0,1,0,0,0) making us be able to restore the original message.

1.2.2 Dual Codes

Definition 1.2.6 (Dual Code)**.** A code C has a generator matrix G_C and a parity-check matrix H_C, the code generated by the parity-check matrix is the dual code of C denoted by $C^\perp$, then

$$G_C = H_{C^\perp}$$
$$H_C = G_{C^\perp}$$

Example 1.2.3. Let's have the classic Hamming code C, a $[7,4,3]_2$-code with the generator matrix

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

With the codewords

$$\begin{array}{cccc} (0,0,0,0,0,0,0) & (1,0,0,0,1,1,1) & (0,1,0,0,0,1,1) & (1,1,0,0,1,0,0) \\ (0,0,1,0,1,0,1) & (1,0,1,0,0,1,0) & (0,1,1,0,1,1,0) & (1,1,1,0,0,0,1) \\ (0,0,0,1,1,1,0) & (1,0,0,1,0,0,1) & (0,1,0,1,1,0,1) & (1,1,0,1,0,1,0) \\ (0,0,1,1,0,1,1) & (1,0,1,1,1,0,0) & (0,1,1,1,0,0,0) & (1,1,1,1,1,1,1) \end{array}$$

And with a parity-check

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Taking H as a generator matrix we get the codewords

$$(0,0,0,0,0,0,0) \quad (1,0,1,1,1,0,0) \quad (1,1,0,1,0,1,0) \quad (0,1,1,0,1,1,0)$$

$$(1,1,1,0,0,0,1) \quad (0,1,0,1,1,0,1) \quad (0,0,1,1,0,1,1) \quad (1,0,0,0,1,1,1)$$

Note that this is a subset of the codewords from C making a $[7,3,3]_2$-code, this is a special case where $C^{\perp} \subset C$.

Chapter 2

QUANTUM COMPUTERS.

In this chapter we will go from what a qubit is to constructing a quantum error-correcting code, what are errors in this type of systems and how this codes identifies them.

2.1 Quantum Computers

Most of the content in this chapter is based on [11].

2.1.1 From Bit to Qubit

A bit is a portmanteau of binary digit representing two logical states, 0 and 1, true and false, yes and no, and it is the fundamental concept of classical computation and classical information.
Quantum computers have a different information unit, one that can take the value 0 or 1, as well as any superposition of these states, and this quantum information unit is called a qubit.

Definition 2.1.1 (Bit, Qubit, Bloch Sphere, State)**.** A bit is a variable in $\mathbb{Z}_2$ and a qubit is a variable in the unitary sphere in $\mathbb{C}^2$ called the Bloch Sphere. The state of a bit or qubit is the value of this variable in its respective space.

Example 2.1.1. A state is the logical value of an information unit, bits have two states previously named 0 and 1 as the elements of $\mathbb{Z}_2$ while qubits have infinite states and all of them can be expressed as a linear superposition of two orthogonal quantum states [11], which form a base of a 2-dimensional subspace of the Bloch sphere, let $|\phi\rangle$ be a quantum state, then

$$|\phi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,.$$

With $|\alpha|^2 + |\beta|^2 = 1$ and the states are defined as

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

The notation $|0\rangle$ is called bra-ket where bras are $\langle 0| := (10)$ and kets are $|0\rangle = (10)^T$ and bra-ket $\langle 0|0\rangle := (10)\cdot(10)^T = 1$.

Definition 2.1.2 (Quantum Logic Gate)**.** A quantum logic gate (simply as gate) is a unitary matrix 2×2.

Example 2.1.2. Let's define some important quantum gates. First the Pauli gates:

$$I := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

And the Hadamard and Phase gates

$$H := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad S := \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

now we can see what a gate does to a state

$$X \cdot |0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$$

$$H \cdot |0\rangle = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle$$

$$H \cdot |1\rangle = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix} = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle$$

$$S \cdot |0\rangle = \begin{pmatrix} 1/\sqrt{2} \\ i/\sqrt{2} \end{pmatrix} = \frac{|0\rangle + i\,|1\rangle}{\sqrt{2}} = |+i\rangle$$

$$S \cdot |1\rangle = \begin{pmatrix} 1/\sqrt{2} \\ -i/\sqrt{2} \end{pmatrix} = \frac{|0\rangle - i\,|1\rangle}{\sqrt{2}} = |-i\rangle$$

we can see that the gates work by rotating points in the Bloch sphere by some angle or by changing the coordinates of the basis of the state or changing the base itself

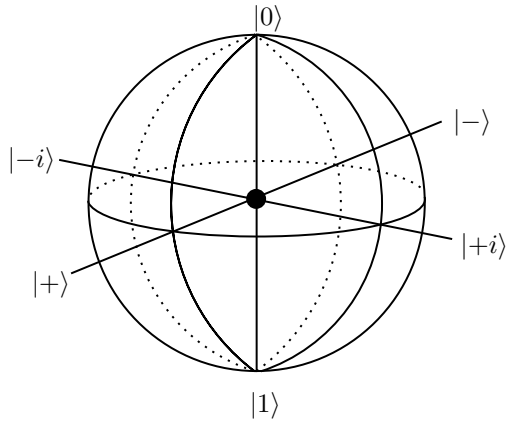


Figure 2.1. Bloch Sphere

The Bloch sphere lives in a 4-dimensional space. Such a surface is (n-1)-dimensional then the sphere is a 3-dimensional surface and its circumferences are lines, and those are (n-2)-dimensional and they turn out to be 2-dimensional so $|0\rangle$ and $|1\rangle$ can be a base of a subspace of the sphere.

2.1.2 Quantum Circuits

In this text we will refer to circuits as a collection of bits or qubits that can be altered using logical gates individually and collectively. This is modeled using tensors. Let's see it with an example

Example 2.1.3. First, let's see what happens with the elements of the basis $|0\rangle$ and $|1\rangle$ when they are tensorized with each other

$$|0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} := |00\rangle$$

$$|0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} := |01\rangle$$

$$|1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} := |10\rangle$$

$$|1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} := |11\rangle$$

the other named states work the same way, as a general example let $|\varphi\rangle$ and $|\theta\rangle$ two arbitrary states from two distinct qubits, then the tensor of them can be described as a linear combination of the tensorized component of their basis.

$$\begin{aligned} |\varphi\rangle \otimes |\theta\rangle &= (a\,|0\rangle + b\,|1\rangle) \otimes (c\,|0\rangle + d\,|1\rangle) \\ &= \left(a \begin{pmatrix} 1 \\ 0 \end{pmatrix} + b \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) \otimes \left(c \begin{pmatrix} 1 \\ 0 \end{pmatrix} + d \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) \\ &= \begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} \\ &= \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix} \end{aligned}$$

$$= ac \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + ad \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + bc \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} + bd \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$
$$= ac\,|00\rangle + ad\,|01\rangle + bc\,|10\rangle + bd\,|11\rangle$$

Example 2.1.4. The previous example can easily be extended to higher number of qubits defining a simple way of working with larger number of qubits with the tensorized basis of choice.
Gates work the same way in multiple qubits, let's see how the first example with gates works with two qubits

$$(X \otimes I) \cdot |00\rangle = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |10\rangle$$

Here we see that the X gate worked as intended in the first qubit while the I gate worked on the second.
A key difference between classical and quantum computers is that the logical gates are reversible in quantum computers. It is easy to see this with the rotation interpretation of gates. This quality makes the implementation of boolean functions from classical computing (which irreversible) possible at the cost of an extra qubit. This extra qubit is the ancilla bit.

Definition 2.1.3 (Ancilla Bit)**.** Extra bits in circuits are called ancilla bits.

Example 2.1.5. The control gates are logic gates that act on 2 or more qubits when a certain qubit act as a control for some operation, an important control gate is the controlled Pauli-X usually named CNOT or CX

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

let's see why it's called a control gate by applying the CNOT gate to the base $|++\rangle, |+-\rangle, |-+\rangle, |--\rangle$ also called the Hadamard basis

$$\text{CNOT} \cdot |++\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{pmatrix} = |++\rangle$$

$$\text{CNOT} \cdot |+-\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1/2 \\ -1/2 \\ 1/2 \\ -1/2 \end{pmatrix} = \begin{pmatrix} 1/2 \\ -1/2 \\ -1/2 \\ 1/2 \end{pmatrix} = |--\rangle$$

$$\text{CNOT} \cdot |-+\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1/2 \\ 1/2 \\ -1/2 \\ -1/2 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \\ -1/2 \\ -1/2 \end{pmatrix} = |-+\rangle$$

$$\text{CNOT} \cdot |--\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1/2 \\ -1/2 \\ -1/2 \\ 1/2 \end{pmatrix} = \begin{pmatrix} 1/2 \\ -1/2 \\ 1/2 \\ -1/2 \end{pmatrix} = |+-\rangle$$

It can be noticed that the CNOT gate only change the first qubit (left one) if the second one (right one) have the state $-$, that is the reason it is called control gate because the first qubits is controlled by the state of the second one which is usually an ancilla bit.

Example 2.1.6. Circuits in quantum computers aren't typically denoted by multiplication of matrices, vector and bra-kets but by the classic circuit representation. Lets say we have 2 qubits in a circuit, qubits are always initialized as $|0\rangle$, and we want to construct the state $|\varphi\rangle = (1/\sqrt{2})\,|00\rangle + (1/\sqrt{2})\,|11\rangle$, first we want to do it the way we were doing it

$$(H \otimes H) \cdot |00\rangle = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$= \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = |\alpha\rangle$$

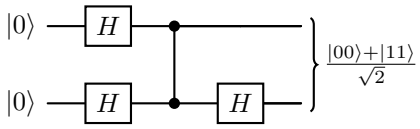
$$\Rightarrow \mathrm{CZ} \cdot |\alpha\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \cdot \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$= \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \end{pmatrix} = |\beta\rangle$$

$$\Rightarrow (I \otimes H) \cdot |\beta\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} \cdot \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \end{pmatrix}$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = (1/\sqrt{2})\,|00\rangle + (1/\sqrt{2})\,|11\rangle$$

We got the desired state, now we will use the circuit notation to represent the exact algorithm we did previously



2.2 Quantum Codes

Because of the non-deterministic way quantum computers work, for a long time the idea of error-correction codes was thought to be impossible until Shor proved otherwise [13].

2.2.1 Stabilizer Codes

The stabilizer formalism is a way of correcting codes in quantum systems, error correction faced a big problem with the non-cloning theorem [16] so for this context we do not simply repeat the codewords; instead we expand the space in which those words reside [13], let's have a single qubit $|\varphi\rangle \in \text{span}\{|0\rangle, |1\rangle\} =: \mathbb{H}_2$ then we have

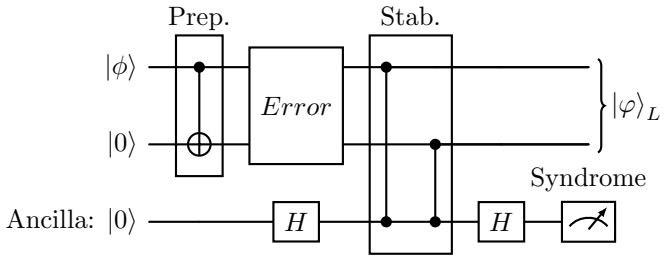
$$\begin{aligned} |\varphi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \to |\varphi\rangle_L &= \alpha\,|00\rangle + \beta\,|11\rangle \\ &= \alpha\,|0\rangle_L + \beta\,|1\rangle_L \end{aligned}$$

Note that $|\varphi\rangle_L = \alpha\,|00\rangle + \beta\,|11\rangle \neq |\varphi\rangle \otimes |\varphi\rangle$ so we are not cloning the state and $|\varphi\rangle_L \in C = \text{span}\{|00\rangle, |11\rangle\} \subset \mathbb{H}_4$ where C is the code subspace.
What we just did is called 'preparing the state' and it's the quantum equivalent to the redundancy applied to the words by the generator matrix in classical codes.
Let's say we have an error like this $X_1\,|\varphi\rangle_L = \alpha\,|10\rangle + \beta\,|01\rangle$, with X_1 being a X-gate on the first qubit, note that this can be interpreted as a bit flip because the first qubit changed from 0 to 1 and that we escape the C space where the original qubit was, now $X_1\,|\varphi\rangle_L \in E = \mathbb{H}_4 - C$ where E is the error subspace.

Note the following property:

$$
\begin{aligned}
Z_1 Z_2 \left|\varphi\right\rangle_L &= Z_1 Z_2(\alpha \left|00\right\rangle + \beta \left|11\right\rangle) = (+1) \left|\varphi\right\rangle_L \\
Z_1 Z_2 X_1 \left|\varphi\right\rangle_L &= Z_1 Z_2(\alpha \left|10\right\rangle + \beta \left|01\right\rangle) = (-1) X_1 \left|\varphi\right\rangle_L \\
Z_1 Z_2 X_2 \left|\varphi\right\rangle_L &= Z_1 Z_2(\alpha \left|01\right\rangle + \beta \left|10\right\rangle) = (-1) X_2 \left|\varphi\right\rangle_L \\
Z_1 Z_2 X_1 X_2 \left|\varphi\right\rangle_L &= Z_1 Z_2(\alpha \left|11\right\rangle + \beta \left|00\right\rangle) = (+1) X_1 X_2 \left|\varphi\right\rangle_L
\end{aligned}
$$

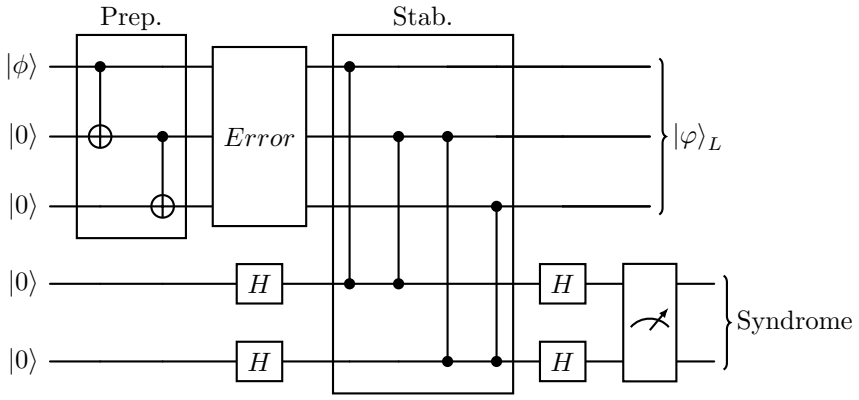
here the $Z_1 Z_2$ operator is said to stabilize the qubit $\left|\varphi\right\rangle_L$ as it leaves it unchanged but maps the X_1 and X_2 to the (-1) eigenspace [2]. We can extract that information using control gates on an extra qubit, these control gates on ancilla qubits are called the ?stabilizer?, and they are equivalent to the parity-check matrix from classical codes. We use an ancilla because we want to measure it to know if any error has occurred without destroying the word, let's see an example of this simple quantum code:



This code can identify 1 error [12], let's use the same table format from classical codes

Error	Syndrome
XI	1
IX	1

In the end, this code can only identify X-errors on one of the two qubits. We can make the code a bit more complex so that it not only identifies X-errors but also corrects one.

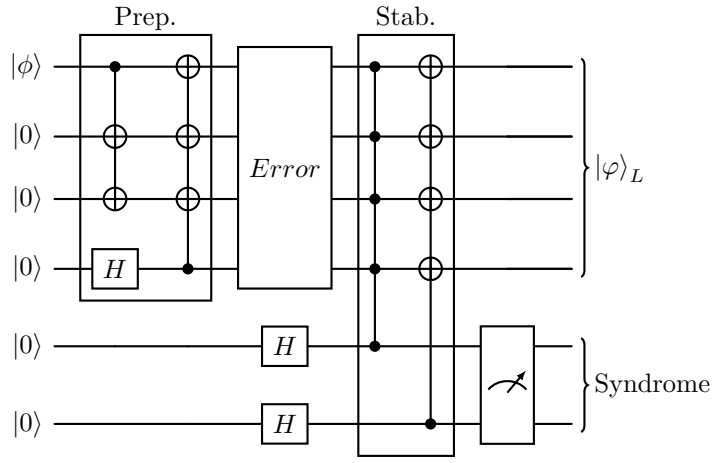


Error	Syndrome
XII	01
IXI	11
IIX	10

Now this is a more functional code because we can correct the X-errors, but what about Y and Z-errors?

First Y-errors are XZ errors because $X \cdot Z = Y$, does this mean that correcting X-errors correct Y-errors too? the answer is yes, but they have the same syndrome than the X-errors, so we can't differentiate between them in the last code that we made, also Z-errors can't be detected by that stabilizer.

A solution to that problem is to make a stabilizer for X-errors and Z-error, then we could identify each of them and if both have the same syndrome we can assume it was a Y-error. Let's see the next example:



Error	Syndrome	Error	Syndrome	Error	Syndrome
$XIII$	01	$ZIII$	10	$YIII$	11
$IXII$	01	$IZII$	10	$IYII$	11
$IIXI$	01	$IIZI$	10	$IIYI$	11
$IIIX$	01	$IIIZ$	10	$IIIY$	11

This code can't correct any errors but it can identify X-errors and Z-errors and by extension Y-errors.

Lastly we can construct a quantum error correcting code that can correct a single arbitrary error.

One can construct stabilizer codes from classic codes, one of them is the Calderbank-Shor-Steanne codes or CSS codes for short.

Proposition 2.2.1. Let C_1 a $[n, k_1, d]$ code and C_2 a $[n, k_2, d]$ code with $C_2 \subseteq C_1$ and $k_2 \leq k_1$. If C_1 and $C_2^{\perp}$ can correct up to d errors, then there exists a $[n, k_1 - k_2, d]$ quantum code that can correct d bit flips and d phase flip where

$$|x + C_2\rangle = \frac{1}{2^{k_2/2}} \sum_{y \in C_2} |x + y\rangle , \qquad x \in C_1.$$

Example 2.2.1. This way we can construct the preparation state for a code that can correct d errors. With H^* being the parity check matrix of the C_1 code with reversed row order, we can build

$$H = \begin{pmatrix} H^* & 0 \\ 0 & H^* \end{pmatrix}$$

and then we can construct the stabilizer of the code as following, for every 0 in the top half of the H matrix we will have an I-gate and for every 1 an X-gate, for the bottom half we will have for every 0 an I-gate and for every 1 a Z-gate.
With that we can use a control gate for the correspond gate type, between the ancilla qubit as the rows and the words qubits as the columns.
Let's have an example, let's have the following parity check matrix for a $[7, 4, 3]$ code:

$$H^* = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

$$H = \begin{pmatrix} H^* & 0 \\ 0 & H^* \end{pmatrix}$$

$$S = \begin{Bmatrix} I & I & I & X & X & X & X \\ I & X & X & I & I & X & X \\ X & I & X & I & X & I & X \\ I & I & I & Z & Z & Z & Z \\ I & Z & Z & I & I & Z & Z \\ Z & I & Z & I & Z & I & Z \end{Bmatrix}$$

With the parity check matrix we can construct the generator matrix of the code:

$$G = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

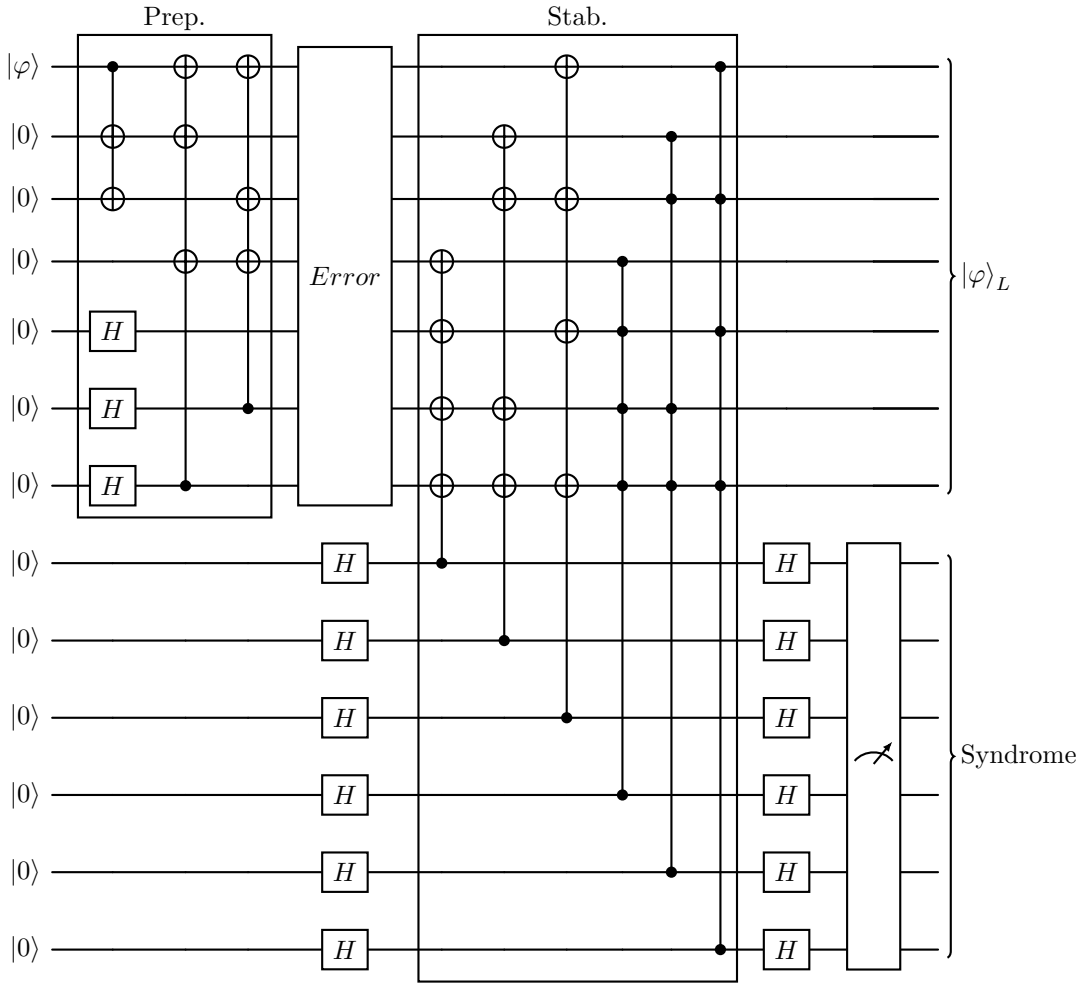
then we have the code words:

$$
\begin{array}{cccc}
(0,0,0,0,0,0,0) & (1,0,1,0,1,0,1) & (0,1,1,0,0,1,1) & (0,0,0,0,1,1,1) \\
(1,1,0,0,1,1,0) & (1,0,1,1,0,1,0) & (0,1,1,1,1,0,0) & (1,1,0,1,0,0,1) \\
(1,1,1,1,1,1,1) & (0,1,0,1,0,1,0) & (1,0,0,1,1,0,0) & (0,0,1,1,0,0,1) \\
(1,1,1,0,0,0,0) & (0,1,0,0,1,0,1) & (1,0,0,0,0,1,1) & (0,0,1,0,1,1,0)
\end{array}
$$

where the top half are the code words for C_2 and by using the last proposition we have the states:

$$
\begin{aligned}
|0\rangle_L = \frac{1}{\sqrt{8}} (&|0000000\rangle + |1010101\rangle + |0110011\rangle + |0000111\rangle + \\
&|1100110\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle) \\
|1\rangle_L = \frac{1}{\sqrt{8}} (&|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + \\
&|1110000\rangle + |0100101\rangle + |1000011\rangle + |0110110\rangle)
\end{aligned}
$$

Now we can use the Gottesman's algorithm [1] to find the gates that prepare this state, there is a python library that can compute that for you, it is called stac, either by doing it manually or with computer help we get the following circuit



Error	Syndrome	Error	Syndrome	Error	Syndrome
X_0	100000	Z_0	000100	Y_0	100100
X_1	010000	Z_1	000010	Y_1	010010
X_2	110000	Z_2	000110	Y_2	110011
X_3	001000	Z_3	000001	Y_3	001001
X_4	101000	Z_4	000101	Y_4	101101
X_5	011000	Z_5	000011	Y_5	011011
X_6	111000	Z_6	000111	Y_6	111111

Finally we made an error correcting code for 1 arbitrary error for a quantum circuit.

Chapter 3

ALGEBRAIC GEOMETRY.

In this chapter we will develop the mathematical background to construct algebraic geometric codes for their useful properties in the application for quantum error-correcting codes

3.1 Preliminaries

For this section it is important to understand some basic concepts from topology and geometry that will be useful in the understanding of more advanced concepts in this chapter. For more advanced topics in topology, [10] is an outstanding first course on this topic.

3.1.1 Topology

Definition 3.1.1 (Power Set)**.** Let S be a set, the power set of S denoted $\mathcal{P}(S)$ is the set of all the subsets of S.

Definition 3.1.2 (Topology, Open Set, Neighborhood, Closed Set)**.** A topology $\mathcal{T}$ on S is a subset of the power set $\mathcal{P}(S)$ with the following properties:

(a) $S \in \mathcal{T}$ and $\emptyset \in \mathcal{T}$.

(b) $\forall \{U_i\}_{i\in\Delta} \subseteq \mathcal{T} : \bigcup_i U_i \in \mathcal{T}$ with Δ an arbitrary set of indices.

(c) $\forall \{U_i\}_{i=1}^{n\in\mathbb{N}} \subseteq \mathcal{T} : \bigcap_{i=1}^{n} U_i \in \mathcal{T}$.

We call the elements of $\mathcal{T}$ open sets and if an open set U contains an element $x \in S$ then we call U a neighborhood of x (sometimes called open-neighborhood). The complement of an open set is called a closed set.

Observation 3.1.1. A topology can be defined by it's close sets, doing it this way the properties (b) and (c) change union by intersections and vice-versa by De Morgan's law [10].

3.1.2 Tangent Space

This is the only section that requires prior knowledge (specifically, of differentiation in the analytic sense), because the purely algebraic notion is too cumbersome for our purposes (though, in its defense, it is useful in other contexts).
The few things we want from calculus is the derivation operator, as the derivation of a polynomial, and the notion that the derivation of a function in a point yields the slope of a hyperplane (a plane in n-dimensions) that is tangent to the original function in that point. The geometrical notion of a tangent is important because it's a specific kind of intersection between functions because if we look really closely at the point, the tangent and the original function become indistinguishable, we say that the tangent space behaves like the function near the intersection and can be linearly model by it [7].

Definition 3.1.3 (Tangent Space)**.** Let V a vector space and $f : V \to V$ a function, the tangent space of f in the point p is:

$$T_p(f) := \sum_i^n \left.\frac{df}{dx_i}\right|_{p_i} \cdot (x_i - p_i), \qquad p \in \mathrm{dom}(f)$$

Example 3.1.1. First, this text would only have polynomials so we will limit ourselves to them but this method works with other types of analytic functions as well. Lets have $z = x^2 + y^2$ where $x, y, z \in \mathbb{R}$ and the point $p = (0, 0, 0)$

$$T_p(f) = \left.\frac{d(x^2+y^2-z)}{dx}\right|_0 (x-0) + \left.\frac{d(x^2+y^2-z)}{dy}\right|_0 (y-0) + \left.\frac{d(x^2+y^2-z)}{dz}\right|_0 (z-0)$$
$$T_p(f) = z$$

Then the tangent space of that function in $\mathbb{R}^3$ is the plane $z = 0$.

Definition 3.1.4 (Smooth Point, Smooth Function)**.** A point p in a function f is called smooth if $\dim T_p(f) = \dim_p f$. If every point of a function is smooth, the function is called smooth.

Example 3.1.2. Lets have the function $f : y^2 = x^3 + 1$, then the tangent space will be:

$$T_p(f) = -3p_1^2 x + 3p_1^3 + 2p_2 y - 2p_2^2$$

and the tangent space has dimension 1 for every point that solves f, then f is smooth.

Definition 3.1.5 (Singular Point)**.** If a point is not smooth, then it is called singular, then a function with a singular point is called a singular function.

Example 3.1.3. Following the previous example, let $g : y^2 = x^3$. Its tangent space $T_p(g)$ equals $T_p(f)$, but at the point $p = (0, 0)$ we have $T_{(0,0)}(g) = 0$, so g is singular at p because $\dim(0) = 0 \neq \dim(T_p(f)) = 1$.

Observation 3.1.2. We will normally use non-singular instead of smooth because smoothness does not describe well other situations where the dimension of the tangent space differs from the dimension of the curve like self intersections.

3.2 From Varieties to Differentials

This section will be based mostly on the algebraic geometric background from [5] and [14] that is needed for the Riemann-Roch theorem and the construction for the algebraic geometry codes.

3.2.1 Varieties

Definition 3.2.1 (Affine n-Space, Zero Set, Algebraic Set)**.** Let $\mathbb{K}$ be a field, we define an affine n-space over $\mathbb{K}$ the set of n-tuples of elements in $\mathbb{K}$ denoted by $\mathbb{A}^n_{\mathbb{K}}$ or simply $\mathbb{A}^n$ and we will call an element P in $\mathbb{A}^n$ a point or place. Let $A = \mathbb{K}[x_1, \ldots, x_n]$ be a polynomial ring in n variables over $\mathbb{K}$, we will interpret the elements of A as functions of the affine n-space to $\mathbb{K}$ by defining

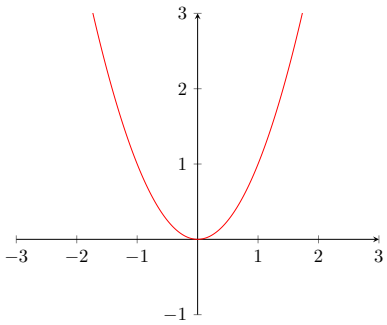
$$f(P) = f(p_1, \ldots, p_n),\ f \in A,\ P \in \mathbb{A}^n$$

The set of zeros (or zero locus) of f is $Z(f) = \{P \in \mathbb{A}^n \mid f(P) = 0\}$ and more generally the zero set of $T \subset A$

$$Z(T) = \{P \in \mathbb{A}^n \mid f(P) = 0\, \forall f \in T\}$$

We will call $Y \subset \mathbb{A}^n$ an algebraic set if there exists a subset $T \subset A$ such that $Z(T) = Y$.

Example 3.2.1. Let's have the field $\mathbb{R}$, then $\mathbb{A}^2_{\mathbb{R}}$ would be the set of 2-tuples of the form (a, b) where $a, b \in \mathbb{R}$, then $A = \mathbb{R}[x_1, x_2]$ would be the polynomial ring in two variables over $\mathbb{R}$, if we pick the function $y = x^2$ then its zero locus would be $\mathbb{Z}(y - x^2) = \{(a, b) \in \mathbb{A}^2 \mid b - a^2 = 0\}$. It is clear that the zero locus coincides with the graph of the function:



The set of points in red can be named Y, then it is clear that Y is an algebraic set of $\mathbb{A}^2$ because we found $f \in T \subset A$ which $Z(T) = Y$.

Example 3.2.2. In the last example we do not have to use a single function set, we could have $T = \{y - x^2, y - x^4\}$, then the zero locus of that would be

$$Z(T) = \{(a, b) \in \mathbb{A}^n \mid f(a, b) = 0 \; \forall \, f \in T\}$$

leaving the intersection of the graphs $Z(T) = \{(0, 0), (1, 1), (-1, 1)\}$.

Definition 3.2.2 (Ideal of an Algebraic Set, Irreducible Algebraic Set)**.** Let Y be an algebraic set of $\mathbb{A}^n$, then the ideal of Y is:

$$I(Y) := \{f \in A \mid f(P) = 0 \; \forall \, P \in Y\}$$

then Y is called irreducible if $I(Y)$ is prime.

Example 3.2.3. The idea of irreducible algebraic set, is to differentiate between sets that can be constructed by intersecting two or more sets and ones that are constructed with only one. we use the last two examples: $\{(0, 0), (1, 1), (-1, 1)\}$ is not ideal because there are no polynomials that have this three points as their only solutions, on the other side we have $\{(a, b) \in \mathbb{A}^2 \mid b - a^2 = 0\}$ which is the whole solution of a polynomial and then an algebraic set that is irreducible.

Observation 3.2.1. Note that the ideal of an algebraic set is an ideal of A, first $I(Y)$ is an aditive subgroup of A, let $f, g \in I(Y)$:

$$f + g := h, \quad h(P) = (f + g)(P) = f(P) + g(P) = 0 \, \forall P \in Y$$

and the inverses, let $f \in I(Y)$, $-f(P) = 0 \Rightarrow -f \in I(Y)$, lastly let $f \in I(Y)$ and $g \in A$:

$$f \cdot g := h, \quad h(P) = (f \cdot g)(P) = f(P) \cdot g(P) = 0 \cdot g(P) = 0 \, \forall P \in Y$$

then, the ideal of an algebraic set is in fact an ideal.

Definition 3.2.3 (Zariski's Topology)**.** The Zariski's topology on $\mathbb{A}^n$ by taking the open subsets to be the complement of the algebraic sets.

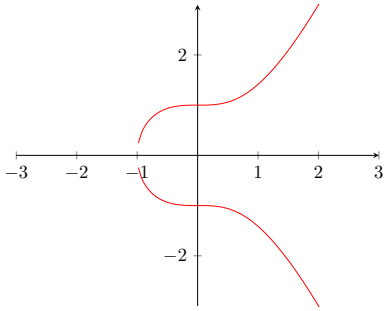
Observation 3.2.2. Zariski's tolopogy define the zero locus of intersection of polynomials would be the base of close sets and opens would be their complement.

Definition 3.2.4 (Affine Variety, Quasi-Affine Variety)**.** An affine algebraic variety or simply affine variety is an irreducible closed subset of $\mathbb{A}^n$ with the induced topology, an open subset of an affine variety is a quasi-affine variety.

Observation 3.2.3. Note that an affine variety can be 'named' as the polynomial whose zero locus is the algebraic set that defined the variety.

Definition 3.2.5 (Affine Algebraic Curve, Affine Algebraic Plane Curve)**.** An affine variety of a two-variable polynomial is called an affine algebraic curve or curve and the three-variable polynomial version is called an affine algebraic plane curve or plane curve.

Example 3.2.4. Let's take the usual field $\mathbb{R}$ and the polynomial $C_1 : y^2 = x^3 + 1$ then $Z(C_1)$ is an affine variety of a two-variable polynomial and its degree is 3, then it's an elliptic curve that can be plotted as:



Another example could be the curve $C : xy(x+y)(x+z) + xz^2(x+z) + y^2z(y+z) = 0$ over the field $\mathbb{F}_2$, $Z(C)$ is a plane curve with degree 3, an elliptic plane curve.

Definition 3.2.6 (Projective Space, Projective Point)**.** A projective space over $\mathbb{K}$, denoted by $\mathbb{P}^n_{\mathbb{K}}$ or simply $\mathbb{P}^n$ is defined by the equivalence:

$$(x_0, \ldots, x_n) \sim \lambda \cdot (x_0, \ldots, x_n), \qquad (x_0, \ldots, x_n) \in \mathbb{A}^{n+1} - \{0\}, k \in \mathbb{K}^\times$$

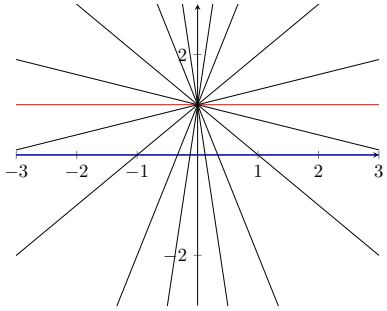
and its elements are called projective points or simply points.

Example 3.2.5. Let have $\mathbb{K}$ be field. Then

1) $\mathbb{P}^0 = \frac{\mathbb{K}^1 - \{0\}}{\sim} \cong \{1\}$

2) $\mathbb{P}^1 = \frac{\mathbb{K}^2-\{0\}}{\sim} = \frac{(x_0,x_1)\in\mathbb{A}^2 \,|\, (x_0,x_1)\neq(0,0)}{\sim} \cong \{(1:a) \,|\, a \in \mathbb{K}\} \cup \{(0:1)\} := \mathbb{A}^1 \cup \infty$ Where $a = \frac{x_1}{x_0}$ if $x_0 \neq 0$.

As $\mathbb{P}^1$ is defined over $\mathbb{A}^2$ then one visualization of this space can be done in the following way, let have $\mathbb{A}^2 = \mathbb{R}^2$ and the affine linear functions that intersect $(0,1)$ and their intersections with $y = 0$:



Now we can see that the line $y = 0$ can describe every line in $\mathbb{A}^2$ but the one $y = 1$, then the line $y = 0$ would be our $\mathbb{A}^1$ and $(0,1)$ our point to infinity. $\mathbb{A}^1$ is usually represented as a circle where the bottom represent $y = 0$ and the top $y = 1$.

2) $\mathbb{P}^2 = \{(x_0, x_1, 1) \in \mathbb{A}^3 \,|\, x_0, x_1 \in \mathbb{K}\} \cup \{(x_0, x_1, 0) \in \mathbb{A}^3 \,|\, (x_0, x_1) \neq (0,0)\} \cong \mathbb{A}^2 \cup \mathbb{P}^1$

This projective space can be visualize in the same way as the $\mathbb{P}^1$ but with lines being represented as planes and a sphere instead of a circle at the end.

Definition 3.2.7 (Projective Algebraic Set, Homogeneous Ideal, projective Variety)**.** We define the projective algebraic set of a set of polynomials S

$$Z(S) = \{P \in \mathbb{P}^n \,:\, f(P) = 0, \text{ for all homogeneous } f \in S\}$$

then an homogeneous ideal in $\bar{k}[x_0, \ldots, x_n]$ is a set of homogeneous polynomials (the same as in the ideal algebraic set), then a projective variety is an irreducible projective algebraic set in $\mathbb{P}^n$.

Definition 3.2.8 (Projective Curve, Projective Plane Curve)**.** A projective curve is a smooth projective variety of dimension one, if the dimension is two then it's a projective plane curve.

Example 3.2.6. Lets have $C : y^2 = x^2 + z^2$, the polynomial is homogeneous, its tangent space is

$$T_P(C) = -2p_1(x - p_1) + 2p_2(y - p_2) - 2p_3(x - p_3)$$

note that (0,0,0) would be a problem but the definition of projective space does not have the 0 element, then C is smooth in the projective space. We have proved that C is a projective plane curve.

Definition 3.2.9 (Regular Function)**.** A function $f : Y \to \mathbb{K}$ is regular at a point $P \in Y$ if there is an open neighborhood U of P, with $U \subset Y$ and polynomials $g, h \in A = \mathbb{K}[x_1, \ldots, x_n]$, such that h is nowhere zero on U and $f = g/h$ on U. We say that f is regular on Y if it is regular on every place in Y.

Observation 3.2.4. Por el teorema 3.2 del Heartshorne a posteriori el anillo de funciones regulares es de hecho un cociente de anillo de polinomios.

Definition 3.2.10 (Algebraic Function Field)**.** If Y is a variety, we define the function field $K(Y)$ of Y as follows: an element of $K(Y)$ is an equivalence class of pairs $\langle U, f\rangle$ where U is a nonempty subset of Y and f is a regular function on U, we identify two pairs $\langle U, f\rangle$ and $\langle V, g\rangle$ if

$$f = g \text{ in } U \cap V.$$

In particular, we always have a field extension $K(Y)/\mathbb{K}$, and we call the elements of $K(Y)$ rational functions on Y. Typically, we will consider $\mathbb{K} = \mathbb{F}_q$ and thus $F := K(Y)$ gives a field extension $F/\mathbb{F}_q$.

Example 3.2.7. Lets take our curve $Y : y^2 = x^3 + 1$ in $\mathbb{F}_2$, because curves are varieties then we can construct the function field

$$F = K(Y) = \frac{\mathbb{F}_2(x, y)}{y^2 - x^3 - 1}$$

as the fraction field of the ring of polynomials:

$$\mathcal{O}(Y) = \frac{\mathbb{F}_2[x, y]}{y^2 - x^3 - 1}$$

then $K(Y) := \{f = \frac{p}{q} \,|\, p, q \in \mathcal{O}(Y)\}$

Observation 3.2.5. For function fields over projective spaces there is an extra condition:

$$\mathbb{K}(Y) := \{f = \frac{p}{q} \,|\, p, q \in \mathcal{O}(Y), deg(f) = deg(g)\}$$

Definition 3.2.11 (Valuation Ring)**.** A valuation ring of the function field $F/\mathbb{K}$ is a ring $\mathcal{O} \subseteq F$ with the following properties:

(1) $\mathbb{K} \subsetneq \mathcal{O} \subsetneq F$.

(2) $\forall\, z \in F : z \in \mathcal{O} \vee z^{-1} \in \mathcal{O}$.

Example 3.2.8. Given an irreducible monic polynomial $p(x) \in K[x]$, in the case of a function field $K(x)$, consider de set:

$$\mathcal{O}_{p(x)} := \left\{ \frac{f(x)}{g(x)} \;\middle|\; f(x), g(x) \in K[x], p(x) \nmid q(x) \right\}$$

is a valuation ring of $K(x)$.

Definition 3.2.12 (Discrete Valuation, Discrete Valuation Ring)**.** A discrete valuation of $F/\mathbb{K}$ is a function $v : F \to \mathbb{Z} \cup \{\infty\}$ with the following properties:

(1) $v(x) = \infty \iff x = 0$

(2) $v(xy) = v(x) + v(y) \qquad \forall\, x, y \in F.$

(3) $v(x + y) \geq \min\{v(x), v(y)\} \qquad \forall\, x, y \in F.$

(4) $\exists\, z \in F : v(z) = 1.$

(5) $v(a) = 0 \qquad \forall\, a \in K : a \neq 0.$

Additionally $G := \{x \in F \,|\, v(x) \geq 0\} \subsetneq F$ is called the discrete valuation ring (DVR) of v.

Example 3.2.9. A classic algebraic example can be: Let's pick a prime $p \in \mathbb{Z}$ then

$$\mathbb{Z}_{(p)} := \{p^k \cdot \frac{a}{b} \in \mathbb{Q} \,|\, a, b \in \mathbb{Z}, p \nmid b\}$$

is called a localization of a Dedekind domain, it is a DVR and its discrete valuation is $v(q) = k$.
While a more geometric example could be previous given for a valuation ring applied to a curve, let's have $K(Y)$ where $Y : y^2 = x^3 + 1$ in $\mathbb{F}_2$ and the smooth point $p \in Y$ with its maximal ideal $\mathfrak{m}_p := \{f \in \mathcal{O}(Y) \,|\, f(p) \neq 0\}$, then $\mathfrak{m}_p = \langle h \rangle$ and therefore

$$\mathcal{O}_{Y,p} := \mathcal{O}_{\mathfrak{m}_p} = \left\{ h^k \frac{f}{g} \in K(Y) \;\middle|\; f, g \in \mathcal{O}(Y), h \nmid q \right\}$$

Is the local ring of Y at a point p and it is a DVR with its discrete valuation $v(a) = k$.

Observation 3.2.6. DVRs are rings unlike the ones seen in basic algebra courses because they come from a strong geometrical notion so they come across as unnatural from that perspective and while the last geometrical example wasn't much different and approachable at first glance, the geometric idea of DVRs is a selection of functions using a curve, that are defined at a point p and the valuation is the multiplicity of 0 as a solution.

Definition 3.2.13 (Place, Set of Places, Prime Element)**.** A place P in the function field $K(Y)$ is the maximal ideal of some valuation ring $\mathcal{O}$ of $K(Y)$. Every element $t \in P$ such

that $P = t\mathcal{O}$ is called a prime element of P. The set of places in the function field is denoted by

$$\mathbb{P}_{K(Y)} := \{P \mid P \text{ is a place of } K(Y)\}$$

Observation 3.2.7. The prime element of a place is often called a local parameter of the place.

Definition 3.2.14 (Valuation Ring of a Place, Residue Class Field, Residue Class Map)**.** If P is the maximal ideal of some valuation ring $\mathcal{O}$ then this ring is uniquely determined by P, namely

$$\mathcal{O}_P := \{z \in K(Y) \mid z^{-1} \notin P\}.$$

Now the residue class field of P is the quotient field

$$F_P := \mathcal{O}_P/P.$$

The map

$$\begin{aligned} K(Y) &\to F_P \cup \{\infty\} \\ f &\mapsto f + P \end{aligned}$$

Is called the residue class map with respect to P. Sometimes we would use the notation $f(P) := f + P$.

Definition 3.2.15 (Degree of a Place, Rational Place)**.** The degree of a place in $K(Y)$ is defined by

$$\deg(P) := [F_P : \mathbb{K}].$$

A place of degree 1 is called a rational place of $K(Y)$.

Example 3.2.10. In the function field $K(Y)$, given an irreducible monic polynomial $p \in \mathcal{O}(Y)$ we can build the valuation ring:

$$\mathcal{O}_p := \left\{ \frac{f}{g} \;\middle|\; f, g \in \mathcal{O}(Y), p \nmid g \right\}$$

then find its maximal ideal

$$P_p := \left\{ \frac{f}{g} \;\middle|\; f, g \in \mathcal{O}(Y), p | f, p \nmid g \right\}$$

In the particular case of $p = t - \alpha$ with $\alpha \in \mathbb{K}$ and t the local parameter of p, we can abbreviate $P_\alpha := P_{t-\alpha} \in \mathbb{P}_{K(Y)}$.
If P_α is a ration place, then the residue class map is given by $z(P_\alpha) = z(\alpha)$ for $z \in K(Y)$ and defined by $z = f/g$ with relatively prime polynomials $f, g \in \mathcal{O}(Y)$:

$$z(\alpha) = \begin{cases} f(\alpha)/g(\alpha) & , g(\alpha) \neq 0 \\ \infty & , g(\alpha) = 0 \end{cases}$$

3.2.2 Divisors

Definition 3.2.16 (Divisor Group, Divisor)**.** The Divisor group of $K(Y)$ is defined as the (additively written) free abelian group which is generated by the places of $K(Y)$, it is denoted by $\mathrm{Div}(Y)$.
The elements of $\mathrm{Div}(Y)$ are called divisors of $K(Y)$. In other words, a divisor is a formal sum

$$D = \sum_{P \in \mathbb{P}_{K(Y)}} n_P P, \text{ with } n_P \in \mathbb{Z}, \text{ almost all } n_P = 0.$$

Example 3.2.11. We saw that places are the maximal ideal of a local ring and can be represented by that point, let's have the curve $Y : y^2 = x^3 + 1$ then a divisor could be

$$D = 3(0, 1) + 2(0, -1)$$

Definition 3.2.17 (Support of a Divisor)**.** The support of D is defined as

$$\mathrm{supp}(D) := \{P \in \mathbb{P}_{K(Y)} \,|\, n_P \neq 0\}$$

Definition 3.2.18 (Prime Divisor)**.** A divisor of the form $D = P$ with $P \in \mathbb{P}_F$ is called a prime divisor.

Definition 3.2.19 (Degree of a Divisor)**.** The degree of a divisor is defined as

$$\deg(D) := \sum_{P \in \P_{K(Y)}} n_P \cdot \deg(P)$$

Definition 3.2.20 (Principal Divisor, Group of Principal Divisors, Divisor Class Group)**.** A divisor D is called *principal* if it is equal to a divisor of the form

$$\mathrm{div}(f) = \sum_{P \in \mathbb{P}_{K(Y)}} v_P(f) \cdot P$$

where $v_P(f) =$ number of zeros (positive) or number of poles (negative) at P of $f \in K(Y)$, the set principal divisors form a normal subgroup $\mathrm{Princ}(Y)$ of $\mathrm{Div}(Y)$ and we can define the

quotient group named the divisor class group.

$$\mathrm{Cl}(Y) = \frac{\mathrm{Div}(Y)}{\mathrm{Princ}(Y)}$$

Example 3.2.12. Let's take $K(Y)$ with $Y : y^2 = x^3 + 1$ over $\mathbb{P}^1$ with the function $f = x/y$ then

$$\begin{aligned} \{x = 0\} \cap \{y^2 - x^3 - 1 = 0\} &= (0:1) \\ \{y = 0\} \cap \{y^2 - x^3 - 1 = 0\} &= (1:0) \\ \Rightarrow \mathrm{div}(f) &= (0:1) - (1:0) \end{aligned}$$

Definition 3.2.21 (Riemann-Roch Vector Space Associated to a Divisor, Dimension of a Divisor)**.** Let D be a divisor of $K(Y)$, the Riemann-Roch Vector Space associated to the divisor D is defined as

$$\mathcal{L}(D) := \{f \in K(Y) \mid \mathrm{div}(f) + D \geq 0\} \cup \{0\}$$

and the dimension of this vector space is called the dimension of the divisor and it is denoted by

$$\ell(D) = \dim_{\mathbb{K}}(\mathcal{L}(D))$$

3.2.3 Differentials on a curve

For this subsection we will use [15] to guide us in the complex topic of differentials on algebraic curves.

Definition 3.2.22 (Derivation)**.** Let V be a vector space over $K(Y)$ with Y an irreducible smooth curve. An linear map

$$\begin{aligned} D : K(Y) &\to V \\ fg &\mapsto D(fg) = D(f)g + fD(g) \end{aligned}$$

is called a derivation, the set of all derivations is denoted by $\mathrm{Der}(Y, V)$ and $\mathrm{Der}(Y)$ if $V = K(Y)$.

Definition 3.2.23 (Differential)**.** A differential (or a rational differential form) on Y is a linear map

$$\begin{aligned} df : \mathrm{Der}(Y) &\to K(Y) \\ D &\mapsto df(D) = D(f) \end{aligned}$$

the set of all differential forms is denoted by $\Omega(Y)$.

Observation 3.2.8. Differentials are linked to a rational function by a local parameter, the relation is that $\omega = f dt_P$ where $\omega \in \Omega(Y)$ and t a local parameter in a place P on $K(Y)$, so the zeros and poles of the differential are the same as the rational function [15].

Definition 3.2.24 (Order of Valuation of a Differential)**.** Let $\omega \in \Omega(Y)$, P a place in Y. The order of valuation of ω in P is

$$\mathrm{ord}_P(\omega) = v_P(\omega) = v_P(f).$$

Definition 3.2.25 (Divisor of a Differential)**.** Let $\omega \in \Omega(Y)$. The divisor of ω is

$$(\omega) = \sum_{P \in \mathbb{P}_{K(Y)}} v_P(\omega) P.$$

Observation 3.2.9. Of course, for the divisor of a differential one must show that only a finitely many coefficients in (ω) are not 0.

Definition 3.2.26 (Residue of a Differential at a Point)**.** Let $\omega \in \Omega(Y)$, P a place in $K(Y)$, t a local parameter at P and $\omega = f dt_P$ the representation of ω. The rational function f can be written as $\sum_{i \in \mathbb{Z}} a_i t^i$. We define the residue of ω at P as

$$\mathrm{res}_P(\omega) = a_{-1}$$

3.3 Riemann-Roch theorem

Definition 3.3.1 (Genus)**.** The genus g of an algebraic function field $K(Y)$ is defined by

$$g := \max\{\deg(D) - \ell(D) + 1 \,|\, D \in \mathrm{Div}(Y)\}$$

Theorem 3.3.1 (Riemann-Roch)**.** Let W be a canonical divisor of $K(Y)$, then for each divisor $D \in \mathrm{Div}(Y)$

$$\ell(D) = \deg(D) + 1 - g + \ell(W - D)$$

Proof. The proof of this result is rather technical and it goes beyond the scope of the present thesis. See [5] for details. □

Definition 3.3.2 (Riemann-Roch Vector Space Associated to a Differential, Index of Speciality of a Divisor)**.** Let D be a divisor on a curve Y. We define the vector space

$$\Omega(D) = \{\omega \in \Omega(Y) \,|\, (\omega) - D \geq 0\}$$

and we denote the dimension of $\Omega(D)$ over $\mathbb{K}$ by $\delta(D)$, called the index of speciality of D

3.4 Algebraic Geometry codes

Proposition 3.4.1. Let $P_1, \ldots, P_n$ be a pairwise distinct rational places of $K(Y)$ with Y a curve over $\mathbb{F}_q$ and $D = P_1 + \cdots + P_n$. We choose a divisor G of $K(Y)$ such that $\text{supp}(D) \bigcap \text{supp}(G) = \emptyset$. The code $C_{\mathcal{L}}(D, G)$ defined as the image of the linear map:

$$\begin{aligned} ev_D : \mathcal{L}(G) &\to \mathbb{F}_q^n \\ f &\mapsto ev_D(f) = (f(P_1), \ldots, f(P_n)). \end{aligned}$$

And $C_{\mathcal{L}}(D, G)$ is a $[n, k, d]_q$-code with

$$\begin{aligned} k &= \ell(G) - \ell(G - D) \\ d &\geq n - \deg(G). \end{aligned}$$

If $2g - 2 < \deg(G) < n$ then $k = \deg(G) - g + 1$.

Proposition 3.4.2. Let $P_1, \ldots, P_n$ be a pairwise distinct rational places of $K(Y)$ with Y a curve over $\mathbb{F}_q$, $D = P_1 + \cdots + P_n$. We choose a divisor G of $K(Y)$ such that $\text{supp}(D) \bigcap \text{supp}(G) = \emptyset$ and $\omega \in \Omega(G - D)$. The code $C_{\Omega}(D, G)$ defined as the image of the map:

$$\begin{aligned} \text{res}_D : \Omega(G - D) &\to \mathbb{F}_q^n \\ \omega &\mapsto \text{res}_D(\omega) = (\text{res}_{P_1}(\omega), \ldots, \text{res}_{P_n}(\omega)). \end{aligned}$$

And $C_{\Omega}(D, G)$ is a $[n, k, d]_q$-code with

$$\begin{aligned} k &= n + g - 1 - \deg(G) \\ d &\geq \deg(G) - (2g - 2). \end{aligned}$$

Theorem 3.4.1. $C_{\mathcal{L}}(D,G)^{\perp} = C_{\Omega}(D,G)$. Additionally if we have a Weil differential η such that $v_{P_i}(\eta) = -1$ and $\eta_{P_i}(1) = 1$ for all $i = 1, \ldots, n$, then $C_{\mathcal{L}}(D, G^{\perp}) = C_{\Omega}(D,G)$.

Proof. The proof for this theorem can be found in [14] □

Chapter 4

AG CODES APPLIED TO QECC.

All the work in this thesis was aimed at constructing a simple example that takes an algebraic curve as input and outputs a circuit capable of correcting a single arbitrary error in a quantum system.

4.1 From Curves to QECC

We can follow the steps found in remark 27 from [3].
Let's consider the curve $\mathcal{X} : xy(x+y)(x+z) + xz^2(x+z) + y^2z(y+z) = 0$, and it's points over $\mathbb{F}_2$:

$$P_1 = [0:0:1]$$
$$P_2 = [0:1:0]$$
$$P_3 = [0:1:1]$$
$$P_4 = [1:0:0]$$
$$P_5 = [1:0:1]$$
$$P_6 = [1:1:0]$$
$$P_7 = [1:1:1]$$

Now we have the tangent L_i of the point P_i on $\mathcal{X}$ and the intersection divisors:

L_i	$L_i \cdot \mathcal{X}$
$L_1 : x = 0$	$2P_1 + P_2 + P_3$
$L_2 : z = 0$	$2P_2 + P_4 + P_6$
$L_3 : y + z = 0$	$2P_3 + P_4 + P_7$
$L_4 : y = 0$	$P_1 + 2P_4 + P_5$
$L_5 : x + z = 0$	$P_2 + 2P_5 + P_7$
$L_6 : x + y + z = 0$	$P_3 + P_5 + 2P_6$
$L_7 : x + y = 0$	$P_1 + P_6 + 2P_7$

Now we define the divisors $D = P_1 + P_2 \dots P_7$ and Q a place of degree 3 in $\mathcal{X}$ that corresponds to the orbit of $[\alpha^2 : \alpha : 1]$ over $\mathbb{F}_8$.
Then we have $C(\mathcal{X}, D, 2Q)$- an AG-code- and we must find a basis of $\mathcal{L}(2Q)$, so we define the following forms:

$$\begin{aligned} H_1 : x^3z + x^2y^2 + x^2z^2 + xy^3 + y^3z + yz^3 \\ H_2 : x^2 + y^2 + z^2 + xy \\ H_3 : y^2 + z^2 + xy + xz + yz \end{aligned}$$

Then we have the rational functions on $\mathcal{X}$:

$$\begin{aligned} f_1 : \frac{H_3 L_1 L_3}{H_1} \\ f_2 : \frac{L_3 L_5 L_7^2}{H_2} \\ f_3 : \frac{L_4 L_5^2 L_6}{H_3} \end{aligned}$$

Which form the basis $\{1, f_1, f_2, f_3\}$ of $\mathcal{L}(2Q)$, then we need the matrix representation of the linear map ev_D:

$$\begin{aligned}(1) &= 0\\ (f_1) &= P_1 + 2P_3 + P_4 + 2P_5 - 2Q\\ (f_2) &= 2P_1 + P_2 + P_4 + 2P_6 - 2Q\\ (f_3) &= P_1 + 2P_2 + P_3 + 2P_7 - 2Q\end{aligned}$$

Then we get the matrix:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1\\ 0 & 1 & 0 & 0 & 0 & 1 & 1\\ 0 & 0 & 1 & 0 & 1 & 0 & 1\\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Which is the generator matrix of the $[7,4,3]_2$ Hamming Code C, in standard form, and the parity-check matrix is:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1\\ 0 & 1 & 0 & 0 & 0 & 1 & 1\\ 0 & 0 & 1 & 0 & 1 & 0 & 1\\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}, \quad H = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0\\ 1 & 1 & 0 & 1 & 0 & 1 & 0\\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

with the codewords of C are:

$$\begin{array}{llll} (0,0,0,0,0,0,0) & (1,0,0,0,1,1,1) & (0,1,0,0,0,1,1) & (1,1,0,0,1,0,0)\\ (0,0,1,0,1,0,1) & (1,0,1,0,0,1,0) & (0,1,1,0,1,1,0) & (1,1,1,0,0,0,1)\\ (0,0,0,1,1,1,0) & (1,0,0,1,0,0,1) & (0,1,0,1,1,0,1) & (1,1,0,1,0,1,0)\\ (0,0,1,1,0,1,1) & (1,0,1,1,1,0,0) & (0,1,1,1,0,0,0) & (1,1,1,1,1,1,1) \end{array}$$

Now the check matrix for the Steane Code will be:

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Then the stabilizer will be:

$$XXXIIIX$$
$$XXIXIXI$$
$$XIXXXII$$
$$ZZZIIIZ$$
$$ZZIZIZI$$
$$ZIZZZII$$

Using the generator matrix we encode the preparation states:

$$\begin{aligned} |0\rangle_L = \frac{1}{\sqrt{8}} (&|0000000\rangle + |1011100\rangle + |1101010\rangle + |0110110\rangle + \\ &|1110001\rangle + |0101101\rangle + |0011011\rangle + |1000111\rangle) \\ |1\rangle_L = \frac{1}{\sqrt{8}} (&|1111111\rangle + |0100011\rangle + |0010101\rangle + |1001001\rangle + \\ &|0001110\rangle + |1010010\rangle + |1100100\rangle + |0111000\rangle) \end{aligned}$$

And using the Gottesman's algorithm [1] with help of the stac library from [6] with the following code:

```
import numpy as np
import stac

hamming2 = np.array([
[1, 0, 1, 1, 1, 0, 0],
[1, 1, 0, 1, 0, 1, 0],
[1, 1, 1, 0, 0, 0, 1]
], dtype=int)

zeroM = np.zeros(hamming2.shape, dtype=int)

Sx = np.concatenate((hamming2, zeroM))
Sz = np.concatenate((zeroM, hamming2))
code = stac.Code(Sx,Sz)

code.construct_standard_form()
code.construct_logical_operators()
enc_circ = code.construct_encoding_circuit()
enc_circ.draw()
```

Figure 4.1. Python code used for getting the encoding circuit for the preparation state.

Giving the following image as a result:

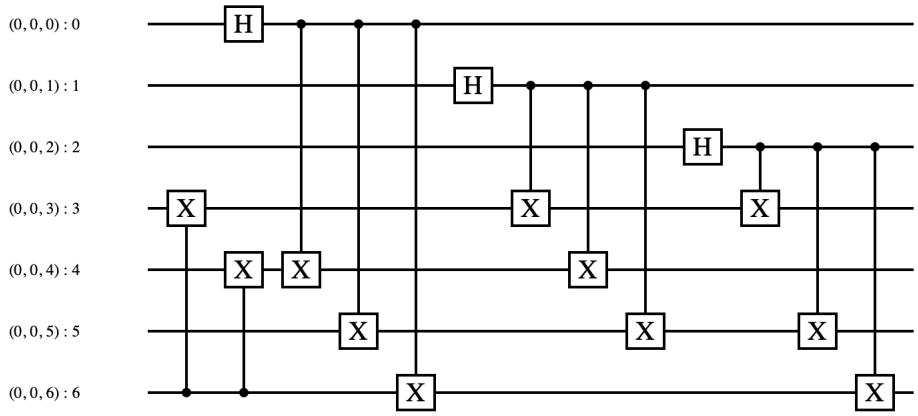
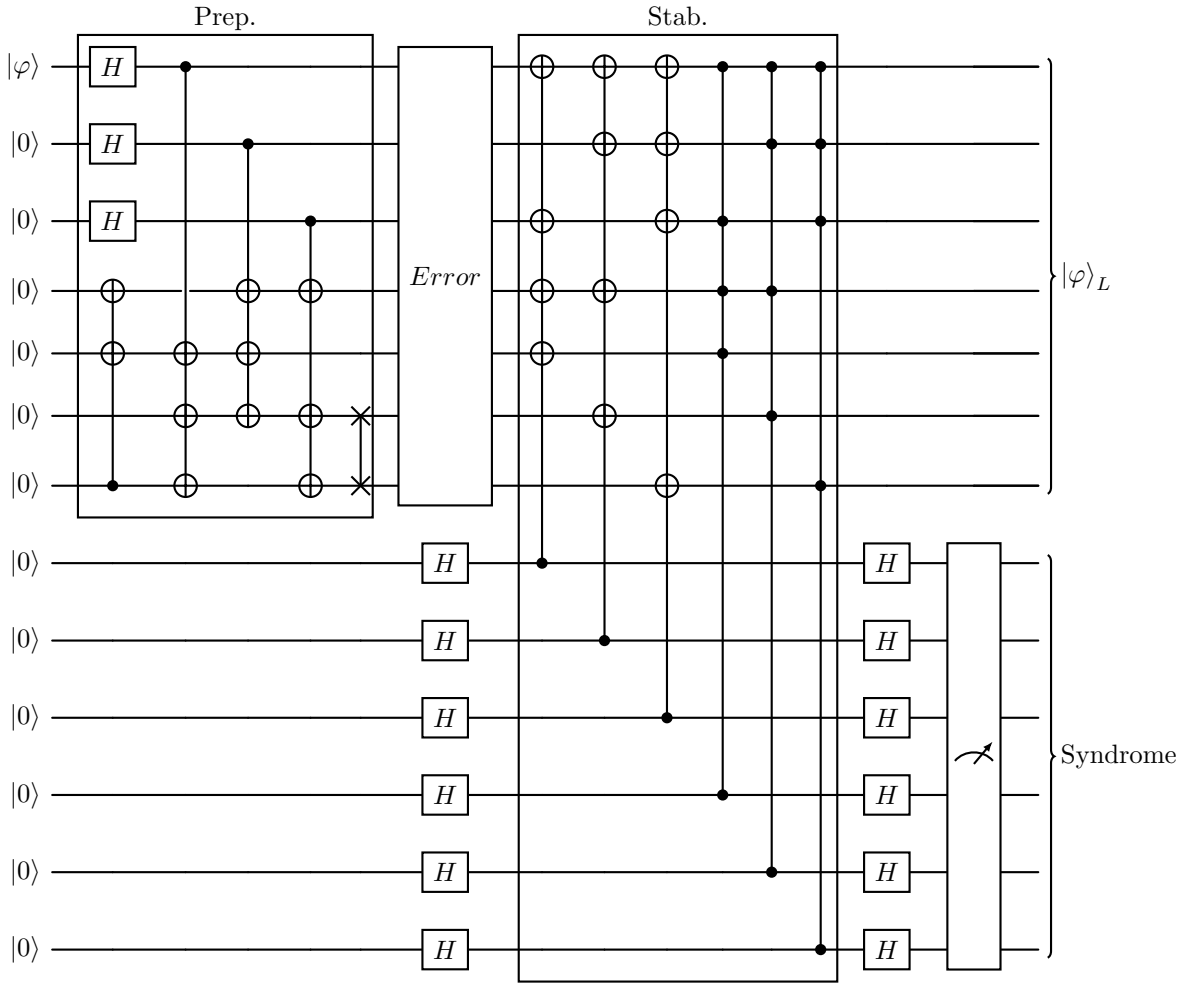


Figure 4.2. Output of the code 4.1

The author confirmed me that the code have minor bugs in the generation of encoding circuits and I did encounter an error, but I managed to correct it by swapping two bits, as seen in the circuit I managed to create:



Note that the stabilizer doesn't follow the orientation as seen in [12], I do not know why this code work in this specific why but it works. Here is the list of syndromes.

Error	Syndrome	Error	Syndrome	Error	Syndrome
X_0	111000	Z_0	000111	Y_0	111111
X_1	110000	Z_1	000110	Y_1	110110
X_2	101000	Z_2	000101	Y_2	101101
X_3	011000	Z_3	000011	Y_3	011011
X_4	001000	Z_4	000001	Y_4	001001
X_5	010000	Z_5	000010	Y_5	010010
X_6	100000	Z_6	000100	Y_6	100100

By the way this syndromes relate to the ones in [12], one can assume that there is a problem with the order in which the gates are arranged but I couldn't find the correct arrangement.

4.2 Final Remarks and Future Directions

In this project, we investigated an application of AG-codes to the field of quantum information with a focus on constructing a quantum error-correcting code capable of mitigating the effect of quantum noise. Starting from basic concepts in algebra, field theory, and classical coding theory we built towards the stabilizer formalism and the CSS construction, both essential tools in quantum error correction.

By taking advantage of the rich and complex structures of algebraic curves, we demonstrated how AG-codes can be adapted to create a quantum code with error correction capabilities. The constructed result offer insights for understanding these concepts from the abstract theory to the applied circuits.

Looking forward, I would like to following this last example to understanding fully why this experiment worked the way it does. I would also like to expand the example to more codes, like the ones found in [4]. Another step would be exploring the entanglement assisted paradigm for correcting errors that was in the scope initially for this thesis but eventually had to be taken out due to time constrains. Lastly it would be interesting to keep doing this work for any new or non-traditional computers like ternary ones.

In sum, this work provides a solid mathematical foundation for applying AG codes in quantum error correction and opens the door to more robust, scalable, and theoretically grounded quantum technologies and sparked the interest to follow this research in the future.

BIBLIOGRAPHY

[1] D. Gottesman. *Stabilizer codes and quantum error correction. Caltech Ph. D.* PhD thesis, Thesis, eprint: quant-ph/9705052, 1997.

[2] D. Gottesman. An introduction to quantum error correction. In *Proceedings of Symposia in Applied Mathematics*, volume 58, pages 221–236, 2002.

[3] B.-Z. S. G.R. Pellikan and G. van Wee. Which linear codes are algebraic-geometric? *IEEE Transactions on Information Theory*, 1991.

[4] M. Grassl. Code tables. https://www.codetables.de.

[5] R. Hartshorne. *Algebraic geometry*. Springer Science & Business Media, 2013.

[6] A. Khalid. Stac. https://github.com/abdullahkhalids/stac, 2022.

[7] J. M. Lee. *Smooth manifolds*. Springer, 2003.

[8] F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes*. Elsevier, 1977.

[9] P. Montero. Algebra abstacta. Lecture notes, 2019.

[10] J. Munkres. Topology james munkres second edition.

[11] M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[12] J. Roffe. Quantum error correction: an introductory guide. *Contemporary Physics*, 2019.

[13] P. W. Shor. Scheme for reducing decoherence in quantum computer memory. *Physical review A*, 1995.

[14] H. Stichtenoth. *Algebraic function fields and codes*. Springer Science & Business Media, 2009.

[15] J. T. Høholdt, J.H. van Lint and G. Pellikaan. *Algebraic geometry codes.* Elsevier, Netherlands, 1998.

[16] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 1982.