universität innsbruck

Master thesis

A Mathematical Approach to Quantum Information & Error Correction

Sandro Mignelli

Leopold-Franzens-Universität Innsbruck
Institute for Mathematics

Supervised by Univ.-Prof. Dr. Tim Netzer

September 24, 2025

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Tim Netzer, for his invaluable guidance, support, and encouragement throughout the course of my thesis. His insights and feedback have been instrumental in shaping this work.

I am deeply thankful to my girlfriend, Katy, whose patience, understanding, and unwavering support have been a constant source of strength and motivation.

I also want to extend my heartfelt appreciation to my family for their continuous support throughout my academic journey. Their belief in me has been a fundamental pillar of my success.

To my colleagues, thank you for the stimulating discussions and shared moments that enriched this experience. A special thanks goes to Stefan and Julia, whose friendship and encouragement made this process all the more rewarding with bonds that go far beyond university.

Abstract

Quantum computing introduces fundamentally new challenges in information theory, driven by the fragile nature of qubits and the restrictions imposed by quantum mechanics, such as the No-Cloning Theorem. This thesis presents a mathematically rigorous yet accessible exploration of quantum error correction (QEC), structured to be approachable for readers with a background in mathematics. Starting from the foundations of quantum mechanics and qubit representations, the work progresses through quantum circuits and measurements over a brief introduction to classical error correction to quantum error correction techniques. Illustrative examples, such as the Bloch sphere and quantum teleportation, ground the theoretical framework. Particular emphasis is placed on the three-qubit bit-flip, three qubit phase-flip and Shor code, showcasing how quantum information can be reliably preserved. The thesis concludes with a discussion of the fundamental limitations posed by No-Cloning and No-Broadcasting Theorems.

Contents

1 Introduction

Quantum computing represents a powerful new paradigm that challenges our classical understanding of computation and information. At the heart of this revolution lies the qubit, a quantum analog of the classical bit, whose behavior is governed by the laws of quantum mechanics. This fundamental difference allows us to solve certain problems much faster, such as integer factorization (Shor's algorithm), it also introduces new and significant challenges, most notably, the issue of noise and error correction.

In classical information theory, noise is handled by using well established error correcting codes. However, quantum information theory faces additional hurdles. Qubits cannot be cloned (No-Cloning Theorem) and any measurement typically disturbs the system, making the direct translation of classical techniques infeasible. As a result, the field of quantum error correction (QEC) has developed its own rich mathematical structure.

This thesis aims to introduce and explore QEC from a mathematical viewpoint, making the subject approachable for readers with little or no background in physics. The focus lies on developing the foundational formalism, building up to key quantum error correction schemes including the three-qubit bit-flip, phase-flip and the Shor code. The goal is not only to understand how QEC works but also why it is possible within the constraints of quantum mechanics.

Later on, we will also consider abstract formulations of fundamental limitations such as the No-Cloning and No-Broadcasting Theorems, which offer deeper insights into the structure of quantum information. These results not only reinforce the challenges posed by quantum noise but also illuminate the constraints any error correction scheme must respect.

2 Mathematical framework

This section takes a mathematically rigorous approach to introducing qubits, qubit states, and quantum circuits, ensuring that a mathematician with no prior knowledge of physics can grasp the concepts. It covers all the necessary preliminaries required for understanding quantum error correction (QEC) later on. We start by defining a pure state qubit and like the mathematicians that we are, use that as an excuse to accurately define the necessary notation.
A classical bit has two possible states, it can either be 0 or 1. When one looks at qubits the world gets a bit more complicated. A qubit state can be any complex linear combination of the states $|0\rangle$ and $|1\rangle$ with the constraint $|\alpha|^2 + |\beta|^2 = 1$

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,. \tag{1}$$

If you come from a purely mathematical background like myself, then you are probably wondering what these brackets are for. This is what we call the *Bra-ket* notation, also known as Dirac notation. Paul Dirac is considered as one of the founders of quantum mechanics. In 1950 Dirac published a three pages long paper called *A new notation for quantum mechanics* [5] in which he suggested the *Bra-ket* notation and its advantages. Now let us introduce this notation carefully so we can see why it is so helpful.
We need the first postulate of quantum mechanics [10]

Postulate 1. *Associated to any isolated quantum system is a Hilbert space H known as the state space.*

Therefore we have a scalar product $\langle\cdot|\cdot\rangle$. Note that in this thesis we are only interested in finite dimensional Hilbert spaces and we define the dual space as $H^* = Lin(H, \mathbb{C}) = \{f : H \to \mathbb{C} : f \text{ is linear}\}$. It is known that in the finite case $dim(H) = dim(H^*)$. A powerful result which made the *Bra-ket* notation possible is the Riesz Representation Theorem

Theorem 1 (Riesz)**.** *If T is a bounded linear functional on a Hilbert space H then there exists a unique $g \in H$ such that for every $f \in H$ we have*

$$T(f) = \langle g|f\rangle$$

With this we can identify every element ψ of the Hilbert space H with a unique element L_ψ of the dual space. And this is where the *Bra-ket* notation shows its brilliance. For elements ϕ of the dual space H^* use $\langle\phi|$ (Bra) and for elements ψ of the Hilbert space H use $|\psi\rangle$ (Ket). When applying a linear operator L_ϕ to $\psi \in H$ instead of writing $L_\phi(\psi)$ the Dirac notation suggests writing $\langle\phi|\psi\rangle$. Where $\phi \in H$ is the unique element suggested by the Riesz Representation Theorem.
The usefulness comes from the similarity to the scalar product. It turns out that many calculations within quantum mechanics are far less complicated and easier to write down with this notation. For better understanding we consider an easy example.

Example 2. *Consider an orthonormal basis $(|\phi_1\rangle, \ldots, |\phi_n\rangle)$ of H and express an arbitrary element $|\psi\rangle \in H$*

$$|\psi\rangle = \sum_i^n \alpha_i\,|\phi_i\rangle\,.$$

An elementary calculation shows that $\alpha_i = \langle\phi_i|\psi\rangle$.

$$\alpha_j = \sum_{i=1}^{n} \alpha_i \langle\phi_j|\phi_i\rangle = \left\langle \phi_j \middle| \sum_{i=1}^{n} \alpha_i \phi_i \right\rangle = \langle\phi_j|\psi\rangle \tag{2}$$

Where the first equal sign arises due to the orthonormality of the basis. This yields

$$\begin{aligned} |\psi\rangle &= \sum_{i=1}^{n} \langle\phi_i|\psi\rangle \, |\phi_i\rangle \\ &= \sum_{i=1}^{n} |\phi_i\rangle \, \langle\phi_i|\psi\rangle \\ &= \sum_{i=1}^{n} |\phi_i\rangle \, \langle\phi_i| \, |\psi\rangle \\ &= \left(\sum_{i=1}^{n} |\phi_i\rangle \, \langle\phi_i| \right) |\psi\rangle \, . \end{aligned}$$

But this just proves that for any orthonormal basis $(|\phi_1\rangle, \ldots, |\phi_n\rangle)$ *the sum* $\sum_{i=1}^{n} |\phi_i\rangle \, \langle\phi_i|$ *is equal to the identity operator* I*. Try proving the same without the Dirac notation and you will immediately see the advantages.*

2.1 Quantum States

Now that it has been established how the *Bra-ket* notation works we can discuss qubits and their states. States of qubits are generally speaking unit vectors in the Hilbert space $\mathbb{C}^2$. In quantum mechanics one abbreviates $(1,0)^\intercal$ with $|0\rangle$ and $(0,1)^\intercal$ with $|1\rangle$, we call these the computational basis states. So (1) is actually a complex linear combination of the unit vectors in $\mathbb{C}^2$. Since we want qubits to stay on the unit circle we need the additional restriction $\langle\psi|\psi\rangle = 1$ which translates via the orthogonality of the basis to

$$\begin{aligned} \langle\psi|\psi\rangle &= (\alpha\,|0\rangle + \beta\,|1\rangle)^*(\alpha\,|0\rangle + \beta\,|1\rangle) \\ &= (\overline{\alpha}\,\langle 0| + \overline{\beta}\,\langle 1|)(\alpha\,|0\rangle + \beta\,|1\rangle) \\ &= \overline{\alpha}\alpha\langle 0|0\rangle + \overline{\alpha}\beta\langle 0|1\rangle + \overline{\beta}\alpha\langle 1|0\rangle + \overline{\beta}\beta\langle 1|1\rangle \\ &= \overline{\alpha}\alpha + \overline{\beta}\beta = |\alpha|^2 + |\beta|^2 = 1. \end{aligned}$$

In order to understand quantum circuits and error correction schemes we need a firmer grip on the different qubit states and how to express them in mathematical terms.

2.1.1 Pure state and mixed state qubits

Firstly one has to mention, that for this part of the thesis it is favourable to use $\mathbb{C}^2$ as our Hilbert space. It is by all means possible, to define qubits and their states in arbitrary Hilbert spaces. We will consider these abstractions in the later parts of this thesis. A qubit $|\psi\rangle$ in a pure state can be written as $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ with $|\alpha|^2 + |\beta|^2 = 1$, so

it is described by a single state vector in $H = \mathbb{C}^2$. A mixed state qubit is a positive semi-definite matrix (*psd*) $\rho \in Psd_2(\mathbb{C})$ with the additional condition $trace(\rho) = 1$. Every mixed state qubit can be represented via the density matrix / density operator

$$\rho = \sum_i p_i \left|\phi_i\right\rangle \left\langle\phi_i\right|.$$

Where the $\left|\phi_i\right\rangle$ are pure states and the $p_i \geq 0$ are probabilities which sum to one. This condition on the p_i is equivalent to the density operator having trace one which is easily obtained by looking at the matrix $\left|\phi_i\right\rangle \left\langle\phi_i\right|$

$$\begin{aligned} \left|\phi_i\right\rangle \left\langle\phi_i\right| &= (\alpha \left|0\right\rangle + \beta \left|1\right\rangle) \cdot (\alpha \left|0\right\rangle + \beta \left|1\right\rangle)^* \\ &= \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \cdot \begin{pmatrix} \alpha^* & \beta^* \end{pmatrix} \\ &= \begin{pmatrix} |\alpha|^2 & \alpha\beta \\ \alpha\beta & |\beta|^2 \end{pmatrix}. \end{aligned} \tag{3}$$

Therefore, all summands in a mixed-state sum automatically have trace one. To ensure that the resulting density matrix also has trace one, we must assign appropriate weights p_i to the pure state products $\left|\phi_i\right\rangle \left\langle\phi_i\right|$. In this fashion we can also obviously express every pure state as a *psd* density matrix with trace one. The pure state $\left|0\right\rangle$ can for example be translated to

$$\left|0\right\rangle \equiv 1 \cdot \left|0\right\rangle \left\langle 0\right| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

Using (3) we can generally identify every pure state $\left|\phi\right\rangle = \alpha \left|0\right\rangle + \beta \left|1\right\rangle$ with

$$\left|\phi\right\rangle \equiv 1 \cdot \left|\phi\right\rangle \left\langle\phi\right| = \begin{pmatrix} |\alpha|^2 & \alpha\beta \\ \alpha\beta & |\beta|^2 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \cdot \begin{pmatrix} \overline{\alpha} & \overline{\beta} \end{pmatrix}$$

So we can express any pure state as vv^* with a suitable vector v. We know from linear algebra that vv^* is always a rank one matrix. Therefore, density matrices for pure states are by construction rank 1. Mixed states, on the other hand, have rank 2.
But what are these probabilities p_i in mixed states intuitively speaking? Imagine trying to somehow generate a pure state in a physics lab, if we succeed we know the state of the system (qubit). But maybe our tools and devices cannot work as precise as we might want them to. Hence, instead of generating the qubit $\left|\psi\right\rangle$ we might have generated a different qubit $\left|\phi\right\rangle$ each with probability $\frac{1}{2}$. So this brings another type of probability in the equation. On one hand we have the quantum probability of the qubit, deciding what state it is in when measured and on the other hand we also have the classical probability that is modeled by the concept of a mixed state.

2.1.2 Multipartite state qubits

Now what if one wants to handle multiple qubits at the same time? The most obvious mathematical construction for combining vector spaces is the tensor product. When talking about multipartite states we again have to differentiate between pure and mixed states but there is also a new notion of entanglement and separability that comes along.

- **Pure multipartite states** So if we want to consider exactly n pure state qubits at the same time we consider the space $\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2 =: (\mathbb{C}^2)^{\otimes n} \cong \mathbb{C}^{2^n}$. The elements of this space are linear combinations of tensor products

$$|\psi\rangle = \sum_{i_1=0}^{1} \cdots \sum_{i_n=0}^{1} \alpha_{i_1,\ldots,i_n} \left|\gamma_{i_1}^{(1)}\right\rangle \otimes \cdots \otimes \left|\gamma_{i_n}^{(n)}\right\rangle$$

where $\left(\left|\gamma_{i_k}^{(k)}\right\rangle\right)_{i_k=0}^{1}$ are pure state qubits which each form an orthonormal basis for $\mathbb{C}^2$ and $\alpha_{i_1,\ldots,i_n} \in \mathbb{C}$. We still have the condition $\|\psi\|^2 = \sum_{i_1=0}^{1} \cdots \sum_{i_n=0}^{1} |\alpha_{i_1,\ldots i_n}|^2 = 1$.

  - **Seperable or entangled** We call a multipartite state separable if we can write it as a tensor product of elements of the individual spaces.

$$|\psi\rangle = |\psi_1\rangle \otimes \cdots \otimes |\psi_n\rangle$$

    and entangled if we need a sum of length more than one.

- **Mixed multipartite states** Mixed multipartite states on the other hand are elements $\psi \in Her_2(\mathbb{C})^{\otimes n} \cong Her_{2^n}(\mathbb{C})$, again with the constraint $tr(\psi) = 1$ and $\psi \geq 0$ (psd). Similarly to the pure multipartite states we can represent an arbitrary mixed multipartite state as

$$\psi = \sum_i \rho_i^{(1)} \otimes \cdots \otimes \rho_i^{(n)}.$$

  Where the $\rho_i^{(k)}$ are matrices in $Her_2(\mathbb{C})$ that are not necessarily psd.
  - **Seperable or entangled** We call ψ separable if there is a representation with $\rho_i^{(k)} \geq 0$ for all i and entangled otherwise.

It is important to note that one generally abbreviates $|\psi_1\rangle \otimes \cdots \otimes |\psi_n\rangle$ with $|\psi_1 \ldots \psi_n\rangle$. A famous example for pure bipartite entangled states are the Bell-states.

$$\begin{aligned}
\left|\Phi^+\right\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}} \\
\left|\Phi^-\right\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}} \\
\left|\Psi^+\right\rangle &= \frac{|01\rangle + |10\rangle}{\sqrt{2}} \\
\left|\Psi^-\right\rangle &= \frac{|01\rangle - |10\rangle}{\sqrt{2}}
\end{aligned}$$

As mentioned before, one is not obligated to use $\mathbb{C}^2$ as the representative Hilbert space of a quantum system (qubit). Hence, we can generalize this much more. The generalization from this point on is very easy, but since we first focus on QEC we are not going to give this any more attention right now. If you are interested in another rather mathematical approach to these concepts and more, a nice read is [4] by Gemma De les Coves.

2.2 Quantum circuits

The aim of this section is to explain all basic concepts that we will need in the future to describe QEC codes and quantum circuits. We already know about the computational basis $|0\rangle$ and $|1\rangle$ of $\mathbb{C}^2$ but this is obviously not the only choice. There is for example the Hadamard basis

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{4}$$

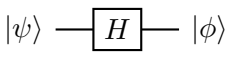
The unitary transformation between this basis and the standard (or computational) basis is called Hadamard gate or Hadamard transform and it is defined as follows

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

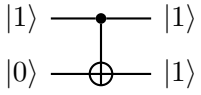
Notice how $H|0\rangle = |+\rangle$ and $H|1\rangle = |-\rangle$. Further important gates are

$$\begin{aligned} &\text{The Pauli X-gate} \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ &\text{The Pauli Y-gate} \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \\ &\text{The Pauli Z-gate} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \end{aligned} \tag{5}$$

So what exactly is a quantum circuit and how is it interpreted? We draw quantum circuits from left to right where the x-axis is time. Quantum gates are symbols similar to those in an electrical circuit that make the qubits engage with each other. We can for example let our initial qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ go through a Hadamard gate transforming it to $|\phi\rangle = \alpha|+\rangle + \beta|-\rangle$.



Another widely used concept is the controlled not gate.



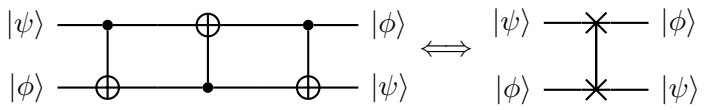
The $CNOT$ gate allows the control qubit to flip the target qubit iff the control qubit is in state $|1\rangle$. In this scenario we need at least two qubits and therefore the space $\mathbb{C}^2 \otimes \mathbb{C}^2$. The $CNOT$ gate can be written in matrix form as

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The following easy calculations show all possible outcomes

$$\begin{aligned} CNOT\,|00\rangle &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = |00\rangle \\ CNOT\,|01\rangle &= \ldots = |01\rangle \\ CNOT\,|10\rangle &= \ldots = |11\rangle \\ CNOT\,|11\rangle &= \ldots = |10\rangle\,. \end{aligned}$$

With the help of the $CNOT$ gate we can introduce the SWAP gate which is a combination of three $CNOT$ gates.



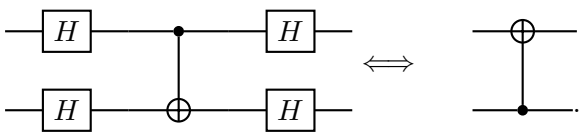
Where the inverted $CNOT$ gate flips the first bit iff the second bit is one. If we look at this through the scope of vectors and matrices we can easily define the corresponding matrix to the inverted $CNOT$ gate by considering the basis vectors.

$$\begin{aligned} &|00\rangle \to |00\rangle \\ &|01\rangle \to |11\rangle \\ &|10\rangle \to |10\rangle \\ &|11\rangle \to |01\rangle \end{aligned} \qquad \Longrightarrow \qquad CNOT_H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

We could now check if the circuit diagram swaps the qubits by the following elementary calculation

$$CNOT \cdot CNOT_H \cdot CNOT \cdot |\psi\phi\rangle = |\phi\psi\rangle$$

Since there is not really a name for the reversed or flipped $CNOT$ gate we called it $CNOT_H$ since we can also express it via a normal $CNOT$ gate and Hadamard transformations



2.3 Quantum measurements

In this section we want to understand quantum measurements in a mathematical sense. For this purpose we use [10, p.84] once again.
Generally speaking we want to measure some observable A of the quantum system. Just for the sake of some physical intuition, an observable could be the spin (see Stern-Gerlach experiment), the position, the energy or the angular momentum of a particle. But for this thesis an observable is simply an hermitian operator $A = A^*$.
The possible outcomes of the measurement correspond to the eigenvalues of the operator A. We start by looking at some very basic linear algebra facts about hermitian operators. Since A is hermitian we can write

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^*. \tag{6}$$

Where the λ_i are eigenvalues and the v_i are the corresponding eigenvectors which form an orthonormal basis. Now since the v_i form an ONB we can express our qubit $|\psi\rangle$ as the linear combination of these

$$|\psi\rangle = \sum_{i=1}^{n} \alpha_i |v_i\rangle \qquad \text{with } \alpha_i \in \mathbb{C}. \tag{7}$$

We already know that $\alpha_j = \langle v_j|\psi\rangle$ by (2), so we can write $|\psi\rangle = \sum_{i=1}^{n} \langle v_i|\psi\rangle |v_i\rangle$. Now we can discuss measurements. When we measure the qubit in the ONB corresponding to the observable we get a projection onto one of the elements of the ONB (eigenstates of A). So if we take the observable (6) then we project the qubit on the axis of a v_i.

$$v_i v_i^* |\psi\rangle \tag{8}$$

Since the outcome does not always lie on the unit sphere we have to normalize to get the state after the measurement

$$\frac{1}{|\alpha_i|} v_i v_i^* |\psi\rangle .$$

As mentioned before physics tells us that a qubit collapses on an eigenstate v_i of A immediately after measuring it. As one can see from the above calculation (7) the α_i are the weights of the orthogonal projections on the corresponding eigenstates v_i. This is a small motivation for the following postulate of quantum mechanics.

Postulate 2. *The measurement of an observable A in a normalized system $|\psi\rangle$ gives an eigenvalue λ_i of A with probability $|\langle v_i|\psi\rangle|^2$.*

Differently from classical information theory, in quantum information theory we cannot predict the result of a measurement. What we can do is assign probabilities to every possible outcome.
From the calculations before we know that $|\alpha_i|^2 = |\langle v_i|\psi\rangle|^2$. So the probability of the state collapsing to v_i is $|\alpha_i|^2$.

Since we cannot know the result of a measurement before the measurement itself we are often times interested in the expectation value of an observable.

Definition 3. *If a system is in state $|\psi\rangle$, the expectation value of the observable A is given by*

$$\langle\psi|A|\psi\rangle$$

We can also check if this definition agrees with our work from before. The expectation value is generally speaking defined as the sum over all possibilities times their probabilities which in our case would lead to

$$\begin{aligned}
\mathbb{E}(A) &= \sum_{i=1}^{n} \lambda_i |\alpha_i|^2 \\
&= \sum_{i=1}^{n} \lambda_i (\langle v_i|\psi\rangle)^* (\langle v_i|\psi\rangle) \\
&= \sum_{i=1}^{n} \lambda_i \psi^* v_i v_i^* \psi \\
&= \psi^* \left(\sum_{i=1}^{n} \lambda_i v_i v_i^* \right) \psi \\
&= \langle\psi|A|\psi\rangle.
\end{aligned}$$

Summarizing, we can decide what observable we want to measure (e.g. the spin) by choosing the correct hermitian operator A. Expressing the qubit in the corresponding ONB and then physically measuring the observable will lead to the collapse of the qubit to one of the eigenstates of the observable (e.g. spin up or spin down). We can determine the probability of each outcome and the state of the qubit after the measurement as showed above. To strengthen all these new realisations like qubit states, circuits and measurements we discuss two insightful applications in the next chapter.

3 Illustrative applications

Two very well known applications of quantum mechanics are the Bloch sphere and quantum teleportation. The following two subsections will introduce these concepts and make use of all the prior theory. The Bloch sphere is a nice way to visualize qubits and how they change by applying different gates to them. In quantum teleportation we find a way to teleport information (namely a single qubit) from one place to another, of course with some necessary preparations.

3.1 The Bloch sphere

The Bloch sphere is physically speaking a unit sphere in $\mathbb{R}^3$ which helps a lot with visualisation of qubit operations such as the Hadamard gate. In this section, we present a mathematical analysis of the Bloch sphere and derive it in a clear and elegant manner.
We first take a closer look at $Her_2(\mathbb{C})$. An arbitrary matrix $A \in Her_2(\mathbb{C})$ can be represented as

$$A = \begin{pmatrix} z_1 & x+iy \\ x-iy & z_2 \end{pmatrix}.$$

With $z_1, z_2, x, y \in \mathbb{R}$, so we can identify $Her_2(\mathbb{C}) \cong \mathbb{R}^4$. We slowly want to make our way from these matrices to an arbitrary qubit state so we introduce the additional constraint $trace(A) = 1$. This allows us to describe the matrix A with one less variable

$$A = \begin{pmatrix} \frac{1}{2}+z_1 & x+iy \\ x-iy & \frac{1}{2}-z_1 \end{pmatrix}.$$

Which leads us to $\{A \in Her_2(\mathbb{C}) \mid tr(A) = 1\} \cong \mathbb{R}^3$. Mathematically speaking, imposing a fixed trace creates a hyperplane in the space $Her_2(\mathbb{C})$. We are going to use this fact later on.

We already know from the chapters before, that qubits are *psd* matrices. It is well known that these matrices form a convex cone in the space of hermitian matrices.

Definition 4. *A subset S of a vectorspace V is a convex cone iff $S+S \subset S$ and $\mathbb{R}_{\geq 0} \cdot S \subset S$*

This can be seen by an elementary calculation using a characterisation of *psd* matrices. A matrix $A \in Her_m(\mathbb{C})$ is *psd* iff $\forall v \in \mathbb{C}^m : v^*Av \geq 0$. If $A, B \in Her_m(\mathbb{C})$ and $\lambda \geq 0$ then it follows trivially that

$$v^*(A+B)v \geq 0 \quad \text{and} \quad v^*(\lambda A)v \geq 0.$$

The final step of our construction is intersecting the convex cone of *psd* matrices with the three dimensional hyperplane of hermitian matrices with $tr(A) = 1$.

The result is exactly the space of all possible pure and mixed state qubits. Hence, we can describe all qubits within this three dimensional space. What is left to do is make the connection from this rather abstract space to the Bloch sphere. The picture 1 visualizes the issue but keep in mind that the picture is drawn one dimension lower than the actual space.
At the tip of the cone there obviously is the zero matrix. From (5) we know that the Pauli

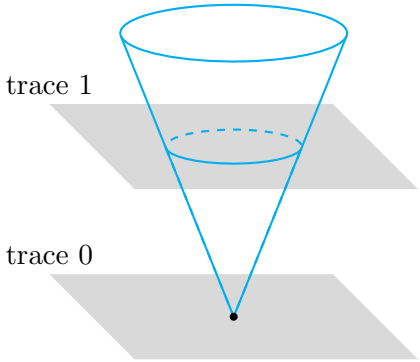


Figure 1: Representation of the cone of *psd* matrices intersected with the hermitian matrices of trace 1 and 0.

matrices all have trace 0 and are orthogonal to each other with respect to the Hilbert-Schmidt inner product. Hence, they form an orthogonal basis for the space

$$\{A \in Her_2(C) \mid tr(A) = 0\}.$$

Our goal is to find a basis representation for qubits with the Pauli matrices. Since linear combinations of them have trace 0 we need to lift the whole span up by $\frac{1}{2}I$ which lies in the center of the trace 1 hyperplane intersection with the cone.

A possible representation of an arbitrary qubit state ρ could therefore be

$$\rho = \frac{1}{2}(I + \alpha X + \beta Y + \gamma Z) \quad \text{with} \quad \alpha, \beta, \gamma \in \mathbb{C}.$$

We immediately get a unique triple (α, β, γ) for each qubit state. If we interpret this triple as coordinates in $\mathbb{R}^3$ we end up with the Bloch sphere. Important to notice is that in the Bloch sphere the computational basis states $|0\rangle$ and $|1\rangle$ are on opposite sides of the ball not orthogonal to each other. To finalize this chapter it is worth mentioning that there are many Bloch sphere online-tools that help with visualizing the action of quantum gates to a qubit.

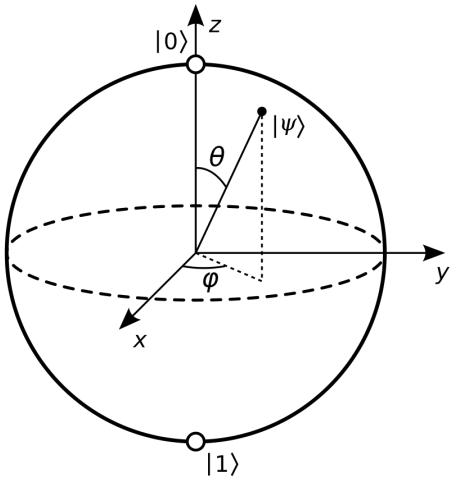


Figure 2: The Bloch sphere

3.2 Quantum teleportation

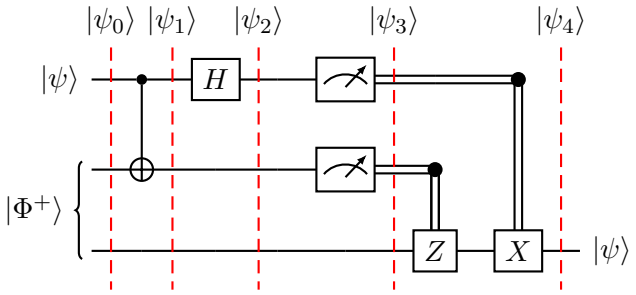


Figure 3: Quantum teleportation circuit

This section will explain how one can teleport quantum information over long distances. This is a nice example that contains all of the above, such as quantum circuits, quantum states and measurements.
Imagine the following situation, Alice and Bob have met each other a long time ago but now are very far apart. However, while they were together they created the Bell state

$$\left|\Phi^+\right\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

and each of them took one of the entangled qubits. This part is essential for the teleportation to work. Today Alice wants to send Bob an unknown qubit state $|\psi\rangle$ but she can only send classical information. This problem seems practically unsolvable since she would have to send an infinite amount of classical information to describe the qubit state to Bob. But Alice remembers that they still have the entangled qubit pair and discovers a clever idea.
Figure 3 helps a lot to understand the process of quantum teleportation. There are five lines sectioning the circuit in six parts which allow us to look at the qubits after each step of the procedure. Notice that the upper two qubits belong to Alice and the bottom one to Bob.

Alice wants to send the unknown qubit state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ to Bob. At the very beginning the three qubits are in the state

$$\begin{aligned}|\psi_0\rangle = |\psi\rangle \otimes \left|\Phi^+\right\rangle &= (\alpha\,|0\rangle + \beta\,|1\rangle) \otimes \left(\frac{|00\rangle + |11\rangle}{\sqrt{2}}\right) \\ &= \frac{1}{\sqrt{2}}\Big[\alpha\,|0\rangle\,(|00\rangle + |11\rangle) + \beta\,|1\rangle\,(|00\rangle + |11\rangle)\Big].\end{aligned}$$

After applying the $CNOT$ gate the three qubits are in the state

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}\Big[\alpha\,|0\rangle\,(|00\rangle + |11\rangle) + \beta\,|1\rangle\,(|10\rangle + |01\rangle)\Big].$$

As you can see, the second qubit gets flipped if the first qubit is in the state $|1\rangle$. By applying a Hadamard gate on the first qubit and pulling out $\frac{1}{\sqrt{2}}$ we receive

$$|\psi_2\rangle = \frac{1}{2}\Big[\alpha(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + \beta(|0\rangle - |1\rangle)(|10\rangle + |01\rangle)\Big].$$

After regrouping these terms we can get the following representation of $|\psi_2\rangle$

$$\begin{aligned}|\psi_2\rangle =&\frac{1}{2}\Big[\,|00\rangle\,(\alpha\,|0\rangle+\beta\,|1\rangle)+|01\rangle\,(\alpha\,|1\rangle+\beta\,|0\rangle)\\&+|10\rangle\,(\alpha\,|0\rangle-\beta\,|1\rangle)+|11\rangle\,(\alpha\,|1\rangle-\beta\,|0\rangle)\Big].\end{aligned}$$

Where we used the associative property of tensor products. Alice now measures her two qubits and receives one of the four basis states $|00\rangle, |10\rangle, |01\rangle, |11\rangle$.

$$\begin{aligned}|00\rangle &\rightarrow |\psi_3\rangle = |00\rangle\,(\alpha\,|0\rangle+\beta\,|1\rangle)\\|01\rangle &\rightarrow |\psi_3\rangle = |10\rangle\,(\alpha\,|1\rangle+\beta\,|0\rangle)\\|10\rangle &\rightarrow |\psi_3\rangle = |01\rangle\,(\alpha\,|0\rangle-\beta\,|1\rangle)\\|11\rangle &\rightarrow |\psi_3\rangle = |11\rangle\,(\alpha\,|1\rangle-\beta\,|0\rangle)\end{aligned}$$

According to Alice's result, Bob simply uses the corresponding operation on his qubit. If Alice receives $|00\rangle$ then Bob already has the correct qubit, hence we are done. If Alice receives $|01\rangle$, Bob needs to use a Pauli X-gate to flip his qubit. If Alice received $|10\rangle$, then Bob applies a Pauli Z-gate on his qubit and if Alice receives $|11\rangle$ Bob uses both gates.
The necessity of transmitting the measurement results of Alice's two qubits to Bob is the sole reason why this type of communication does not exceed the speed of light, which would otherwise contradict the No-Communication Theorem.
Note that this also does not violate the No-Cloning Theorem, as the state of Alice's original qubit must be destroyed in the process of transferring it to Bob.

4 Classical error correction - an overview

This section provides a brief overview of the challenges encountered in classical error correction and the most common strategies used to address them. Additionally, it highlights the key differences between classical and quantum error correction. The main source for this section was mainly prior knowledge but interested readers might want to consider [7] and the sources given there. So imagine Alice wants to send a message to Bob, the process can be split up into three steps

$$A \xrightarrow{\text{encoding}} x \xrightarrow{\text{noise}} \tilde{x} \xrightarrow{\text{decoding}} B.$$

Alice encodes her message using zeros and ones, expecting Bob to receive it without alterations. In reality, messages are often disrupted by noise, such as magnetic fields which can cause bits to flip.
At this point we notice the first difference to quantum error correction since bit-flips are the only error that can occur in classical information theory. To counteract this problem Alice makes use of error-correcting codes. The concept is simple, by adding redundancy to Alice's message, Bob can detect errors and even reconstruct the original message.

The simplest approach to this problem is the repetition code. Instead of sending her message just once, Alice transmits it three times. For example, if she wants to send 101 to Bob, she instead sends $111 - 000 - 111$, a scheme known as the three-bit repetition code. By adding redundancy, the code becomes more resistant to errors.
However, messages can still be disrupted by noise. A bit-flip can happen with probability $p < 1$ which leaves a probability of $1 - p$ for the bit to stay untouched. We want to emphasize here, that a bit-flip usually happens with a rather small probability $p << 1$. Suppose the received message is $110 - 011 - 111$ due to interference. To correct errors, Bob applies a majority vote to each three-bit group, selecting the most common bit. This process would reconstruct the message 111. Bob received the message 110 for the first bit. The probability that the original first message was 111 is $(1-p)^2 p$, while the probability that it was 000 is $p^2(1-p)$. If $p < \frac{1}{2}$ it is significantly more likely that 111 was the initial message. Furthermore notice that Bob also observed an error in the second bit group. However, using the majority vote Bob then gets the wrong result. As explained in figure 4 the repetition code is single error correcting and double error detecting, sometimes called *SECDEC*.
So what is the probability that we get the wrong correction by decoding with the majority vote? In this case that happens iff two or more bits have flipped in a three bit block. We assume the bit-flips happen independently and each bit can only flip once. The probability results in

$$\begin{pmatrix} 3 \\ 2 \end{pmatrix} (1-p)p^2 + \begin{pmatrix} 3 \\ 3 \end{pmatrix} p^3 \approx p^2.$$

Simplicity is good but the drawbacks are obvious. If more than one bit flips in any three-bit group, the correction process will lead to the wrong result. Additionally, repetition coding requires significant memory, as it triples the message length, making it highly inefficient.
Another simple error detecting code works by adding a parity bit to the message. You send your information decoded in zeros and ones and add a bit that only tells if the sum over all ones is odd or even. The parity bit is then set to zero if the sum is even and to one otherwise. This method cannot correct any errors but it can detect errors if the number of errors occurred is odd.

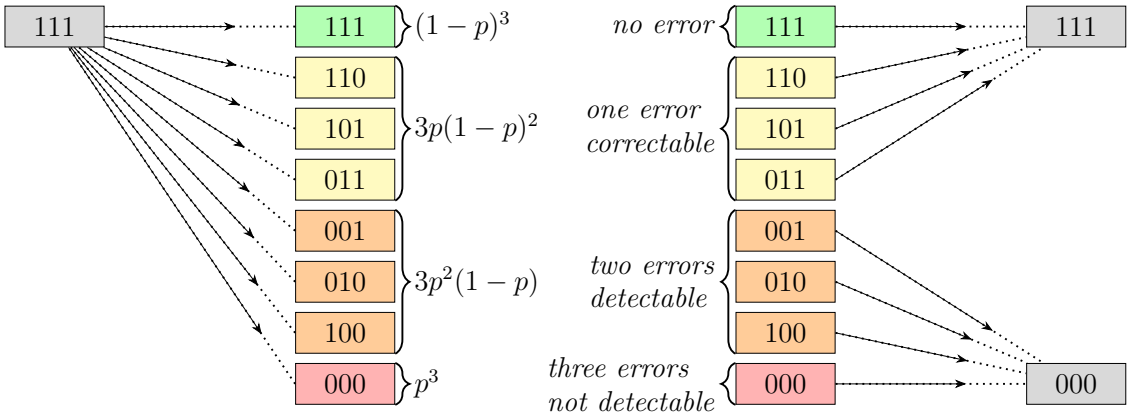


Figure 4: *Adapted from [6].* Alice encodes the first bit 1 of her message by the three-bit repetition code 111 which gets possibly altered by noise in the channel. Then Bob decodes the message using the majority vote.

In 1950 Richard Hamming published a much more efficient method, the Hamming code, which is a great example for the beginnings of error correction and generally known as the first efficient error correcting scheme.
To really grasp the concept we try a more founded mathematical approach.

Definition 5. *An alphabet Σ is a finite set and a word $x = (x_1, x_2, \ldots, x_k)$ of length k is an element of Σ^k where $x_i \in \Sigma$.*

So generally we will have a message of length k living in Σ^k and we will encode it to a message of length $n > k$ in Σ^n. The encoded message is called code word.

Definition 6. *A code $\mathcal{C}$ of dimension n is a subset of Σ^n containing all possible code words of length n.*

Definition 7. *The Hamming distance of two words $v = (v_1, \ldots, v_n), u = (u_1, \ldots, u_n)$ in an alphabet is the number of differences in the words.*

$$d(v,u) \coloneqq \#\{i | v_i \neq u_i, i = 1, \ldots, n\}$$

This immediately implies the more relevant minimal Hamming distance:

$$d_{min} = min(\{d(v,u) | v, u \in \mathcal{C} \ : v \neq u\})$$

The minimal Hamming distance gives the minimal amount by that two distinct words of a code differ. Now if all code words differ by at least d_{min} then the code can detect and correct $e = \lfloor \frac{d_{min}-1}{2} \rfloor$ many errors. If you think of the code words as points on a plane and create a circle around them with radius e, no circles will touch since d_{min} is the minimal distance of all points. Now say a bit in the code word gets flipped then the new emerged word has distance one of the original word. Notice that, if d_{min} is larger than three then the code can correct single bit-flips with certainty.

Definition 8. *A linear $[n, k]$ code is a k-dimensional linear subspace of Σ^n*

In this setting n is the total length of the code word and k is the length of the message itself. Encoding a message with a linear code can be looked at as a linear injective map:

$$\begin{aligned} \pi : \Sigma^k &\to \Sigma^n \\ x &\to Ax \end{aligned}$$

where $A \in Mat_{n,k}(\Sigma)$ is called the generating matrix of the code and x is the message we want to send. A is in standard form if

$$A = \begin{pmatrix} I_k \\ A' \end{pmatrix}$$

where $A' \in Mat_{n-k,k}$
For every generating matrix we also get the parity check matrix $H \in Mat_{n-k,n}(\Sigma)$ which can be defined as:

$$H = \begin{pmatrix} -A' & I_{n-k} \end{pmatrix}$$

With the parity check matrix we can verify whether a code word is in the code or not

$$x \in C \Leftrightarrow Px = 0.$$

Therefore, if we get a message and $Px \neq 0$ we know there is an error. Then we have to find the closest code word to the received message to get the correct message. There are clever methods on how to find that closest code word but this lies beyond the scope of this basic overview.
Returning back to Hamming codes, Hamming codes are generally linear $[2^m-1, 2^m-m-1]$ codes. The generator matrix in standard form for the $[7, 4]$ Hamming code is defined as:

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

From this, it is easy to derive the parity check matrix.

There are many more fascinating aspects to explore in classical error correction but we will conclude here. A really great and more in detail introduction to classical coding theory is [9]. For an excellent visual explanation of the Hamming code, it is recommended to watch this video by 3Blue1Brown.

5 Quantum error correction

When discussing quantum error correction (QEC) one might initially expect that the classical error correction schemes from classical information theory might also work for the quantum case. This is not the case, there are a few differences when considering qubits instead of bits. First and probably the most drastic is that we cannot copy information and send it multiple times, the repetition code is therefore not an option. This is due to the No-Cloning Theorem which essentially states that there is no valid function that can copy the state of an arbitrary qubit. Also relevant is the type of error that can occur. In classical error correction, errors are limited to bit flips, which change a bit from 0 to 1 and vice versa. However, in quantum error correction, noise can affect qubits in multiple ways, such as bit flips, phase flips or even more general decoherence processes. Imagine a unit vector, one could push it to any direction with any amount of force, leading to a continuous error space. These are obviously not all complications that occur but should give a brief understanding of the problems faced.
However, in the next subsections we are going into some of the basic QEC schemes and explain every step thoroughly and mathematically rigorous. In figure 5 one can see the general concept of quantum error correction which appears for all of the presented schemes. As mentioned in figure 5, we assume that noise only affects the qubit once it is encoded and all the processes are completely noise free. This of course is an idealisation.

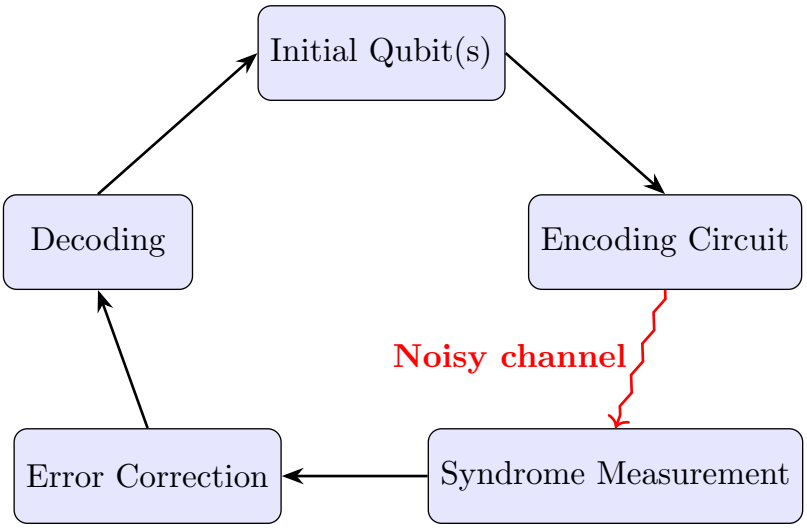


Figure 5: General concept of quantum error correction.

5.1 Bit-flip code

In this chapter we look at the first basic error correcting code, the three qubit bit-flip code. This scheme is able to detect and correct single bit-flips. We will assume in this chapter that this is the only error that happens to our qubit. Firstly we need to encode our qubit with the following circuit.

The initial qubit was $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, then we add two qubits in the basis state $|0\rangle$ to get $|\psi\rangle = \alpha|000\rangle + \beta|100\rangle$. By adding these two states we refer to applying the tensor product

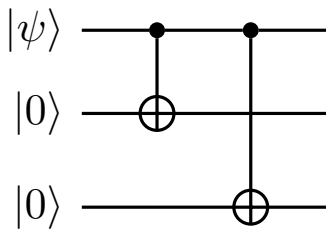


Figure 6: Encoding circuit for the three qubit bit-flip code.

$$|\psi\rangle \otimes |0\rangle \otimes |0\rangle = \alpha\,|000\rangle + \beta\,|100\rangle\,.$$

By the definition of pure multipartite states in chapter 2.1.2 we know that this is a separable state. However we want to entangle our qubits, so we make use of two $CNOT$ gates and we get $|\psi\rangle = \alpha\,|000\rangle + \beta\,|111\rangle$ just like in figure 6.
To see that this is an entangled state we can again use the definition and express $|\psi\rangle$ in the computational basis

$$\begin{aligned}|\psi\rangle = \alpha\,|000\rangle + \beta\,|111\rangle =& \alpha_0\beta_0\gamma_0\,|000\rangle + \alpha_0\beta_0\gamma_1\,|001\rangle + \alpha_0\beta_1\gamma_0\,|010\rangle + \alpha_1\beta_0\gamma_0\,|100\rangle \\ &+ \alpha_0\beta_1\gamma_1\,|011\rangle + \alpha_1\beta_0\gamma_1\,|101\rangle + \alpha_1\beta_1\gamma_0\,|110\rangle + \alpha_1\beta_1\gamma_1\,|111\rangle\,.\end{aligned}$$

By comparing coefficients we know that $\alpha_0\beta_0\gamma_0 = \alpha$ and $\alpha_1\beta_1\gamma_1 = \beta$. Hence, none of the coefficients can be zero. However, since the basis states are all orthogonal, the coefficient of $|001\rangle$ has to be zero. Therefore α_0, β_0 or γ_1 has to be zero and that yields a contradiction. So the state $\alpha\,|000\rangle + \beta\,|111\rangle$ is not separable, but entangled.
In a purely mathematical sense we can describe the encoding process simply by matrix multiplication. For this we need the matrix representation of the $CNOT$ gates with the first qubit as the control and the second qubit respectively the third qubit as the target. Analogously as in Chapter 2.2 we can find those matrices by looking at the basis of $\mathbb{C}^8$.

$$CNOT_{1\to2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \qquad CNOT_{1\to3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

These two $CNOT$ gates entangle our three qubits as follows

$$CNOT_{1\to2} \cdot CNOT_{1\to3} \cdot (\alpha\,|000\rangle + \beta\,|100\rangle) = \alpha\,|000\rangle + \beta\,|111\rangle\,.$$

Now that we encoded our qubit we can send it through a noisy channel. As mentioned above we assume that only single bit-flips can happen to our qubit. There is an easy procedure to check if one or less bits have been flipped. For this we consider the following projections which represent four measurements.

$$\begin{aligned} P_0 &= |000\rangle\langle 000| + |111\rangle\langle 111| \quad \text{No error} \\ P_1 &= |100\rangle\langle 100| + |011\rangle\langle 011| \quad \text{Bit flip on qubit one} \\ P_2 &= |010\rangle\langle 010| + |101\rangle\langle 101| \quad \text{Bit flip on qubit two} \\ P_3 &= |001\rangle\langle 001| + |110\rangle\langle 110| \quad \text{Bit flip on qubit three} \end{aligned} \tag{9}$$

The measurement result $\langle\psi|P_i|\psi\rangle \in \{0,1\}$ is called the error syndrome. In this case there are four different error syndromes corresponding to the projecting operators P_i. Notice that the error syndrome does not contain any information about the qubit state but only about what error has occurred. Hence, the qubit state is not destroyed by this measurement.
In a more technical way one can explain this using the structure of the P_i. Assuming that at most a single bit-flip occurs, the encoding ensures that each measurement preserves the state. There are only four states that the qubits can be in

$$\begin{aligned} |\psi\rangle &= \alpha\,|000\rangle + \beta\,|111\rangle \\ |\psi\rangle &= \alpha\,|001\rangle + \beta\,|110\rangle \\ |\psi\rangle &= \alpha\,|010\rangle + \beta\,|101\rangle \\ |\psi\rangle &= \alpha\,|100\rangle + \beta\,|011\rangle\,. \end{aligned}$$

That means, the encoded qubits generally live in $\mathbb{C}^8$, always lie within the span of two basis vectors. The projecting operators P_i form exactly those combinations of basis vectors such that each possible single bit-flip moves the qubits in the span of one of them. Hence, if any one of the three bits flip, or no error occurs, the system is already in an eigenstate of the measurement basis, preventing further collapse by any measurement.

We consider a short example that will help understanding the process of the three qubit bit-flip code.

Example 9. *Our initial qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ gets encoded to $\alpha\,|000\rangle + \beta\,|111\rangle$ through the circuit above. Then we send it through some noisy channel and a bit flip occurs on the second qubit, so we end up with $|\psi\rangle = \alpha\,|010\rangle + \beta\,|101\rangle$. Now we want to figure out what happened without destroying the information. Note the error syndrome $\langle\psi|\,P_2\,|\psi\rangle = 1$ in this case and $\langle\psi|\,P_i\,|\psi\rangle = 0$ for all $i \in \{0,1,3\}$. Hence, we know that the second qubit flipped and we can correct the error afterwards by flipping it again via a Pauli X-gate without destroying the state.*

After correcting the error the qubits are back in the entangled state $|\psi\rangle = \alpha\,|000\rangle + \beta\,|111\rangle$. This however makes it hard to further operate with our initial qubit, so we want our first qubit back that we put into the machinery. Hence, we need to decode the qubits. This is rather easy, we simply disentangle the state via the same $CNOT_{1\to 2}$ and $CNOT_{1\to 3}$ gates we used to entangle the states and end up with our initial qubit $|\psi\rangle = \alpha\,|000\rangle + \beta\,|100\rangle = (\alpha\,|0\rangle + \beta\,|1\rangle) \otimes |0\rangle \otimes |0\rangle$.

Looking at this code in such detail one has to ask, why does this idea with the projecting operators P_i work so well and why do we use exactly three qubits? Would it work with two? The next few lines are dedicated to these questions.

5.1.1 Finding the projection operators

We start by considering just one qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$. If we take the scalar product of the qubit and the computational basis $|0\rangle, |1\rangle$ we get

$$\begin{aligned}\langle 0|\psi\rangle &= \alpha\langle 0|0\rangle + \beta\langle 0|1\rangle = \alpha\\ \langle 1|\psi\rangle &= \alpha\langle 1|0\rangle + \beta\langle 1|1\rangle = \beta.\end{aligned}$$

If a bit-flip had occurred on $|\psi\rangle$, the state would become $|\psi\rangle = \alpha\,|1\rangle + \beta\,|0\rangle$. We could not tell since we don't know the actual values of α and β. The scalar product would give the exact same results just swapped. So let us try this with two qubits and see what happens. With a two qubit system we now consider the basis $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. We assume our two qubits to be encoded via a $CNOT$ gate so $|\psi\rangle = \alpha\,|00\rangle + \beta\,|11\rangle$. Taking the scalar product yields

$$\begin{aligned}\langle 00|\psi\rangle &= \alpha\\ \langle 01|\psi\rangle &= 0\\ \langle 10|\psi\rangle &= 0\\ \langle 11|\psi\rangle &= \beta.\end{aligned}$$

What would happen with these scalar products if a bit-flip happened on qubit one or two?

$$|\psi\rangle = \alpha\,|10\rangle + \beta\,|01\rangle \qquad\qquad |\psi\rangle = \alpha\,|01\rangle + \beta\,|10\rangle$$

$$\begin{aligned}\langle 00|\psi\rangle &= 0\\ \langle 01|\psi\rangle &= \beta\\ \langle 10|\psi\rangle &= \alpha\\ \langle 11|\psi\rangle &= 0\end{aligned} \qquad\qquad \begin{aligned}\langle 00|\psi\rangle &= 0\\ \langle 01|\psi\rangle &= \alpha\\ \langle 10|\psi\rangle &= \beta\\ \langle 11|\psi\rangle &= 0\end{aligned}$$

Different from the one qubit system we would be able to tell that an error happened by receiving a result > 0 when measuring via the basis vectors $|01\rangle$ and $|10\rangle$. But we still could not tell which bit has flipped. If we consider

$$\langle\psi|xy\rangle\langle xy|\psi\rangle = \overline{\langle xy|\psi\rangle}\langle xy|\psi\rangle = |\langle xy|\psi\rangle|^2 \quad \text{for } x, y \in \{0, 1\},$$

instead of just the scalar products, we can construct the measurement operators

$$\begin{aligned}Q_0 &= |00\rangle\langle 00| + |11\rangle\langle 11|\\ Q_1 &= |10\rangle\langle 10| + |01\rangle\langle 01|.\end{aligned} \tag{10}$$

Now if no error happened we get

$$\begin{aligned}\langle\psi|Q_0|\psi\rangle &= |\alpha|^2 + |\beta|^2 = 1\\ \langle\psi|Q_1|\psi\rangle &= 0.\end{aligned}$$

But if a single bit has flipped we get

$$\langle\psi|Q_0|\psi\rangle = 0$$
$$\langle\psi|Q_1|\psi\rangle = |\alpha|^2 + |\beta|^2 = 1.$$

This leads us in the right direction but does not quite solve our problem. Let us start by considering three qubits and observe what happens. With a three qubit system the computational basis becomes

$$|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle.$$

We again assume our qubits to be entangled via $CNOT$ gates so our qubit is in the state $|\psi\rangle = \alpha|000\rangle + \beta|111\rangle$. We once again consider the scalar products to get a better understanding of what has changed.

$$\begin{aligned} \langle 000|\psi\rangle &= \alpha & \langle 001|\psi\rangle &= 0 \\ \langle 010|\psi\rangle &= 0 & \langle 011|\psi\rangle &= 0 \\ \langle 100|\psi\rangle &= 0 & \langle 101|\psi\rangle &= 0 \\ \langle 110|\psi\rangle &= 0 & \langle 111|\psi\rangle &= \beta \end{aligned}$$

As before we consider the scalar products after a single bit-flip.

$$|\psi\rangle = \alpha|100\rangle + \beta|011\rangle \qquad |\psi\rangle = \alpha|010\rangle + \beta|101\rangle \qquad |\psi\rangle = \alpha|001\rangle + \beta|110\rangle$$

$$\begin{aligned} \langle 000|\psi\rangle &= 0 & \langle 000|\psi\rangle &= 0 & \langle 000|\psi\rangle &= 0 \\ \langle 001|\psi\rangle &= 0 & \langle 001|\psi\rangle &= 0 & \langle 001|\psi\rangle &= \alpha \\ \langle 010|\psi\rangle &= 0 & \langle 010|\psi\rangle &= \alpha & \langle 010|\psi\rangle &= 0 \\ \langle 011|\psi\rangle &= \beta & \langle 011|\psi\rangle &= 0 & \langle 011|\psi\rangle &= 0 \\ \langle 100|\psi\rangle &= \alpha & \langle 100|\psi\rangle &= 0 & \langle 100|\psi\rangle &= 0 \\ \langle 101|\psi\rangle &= 0 & \langle 101|\psi\rangle &= \beta & \langle 101|\psi\rangle &= 0 \\ \langle 110|\psi\rangle &= 0 & \langle 110|\psi\rangle &= 0 & \langle 110|\psi\rangle &= \beta \\ \langle 111|\psi\rangle &= 0 & \langle 111|\psi\rangle &= 0 & \langle 111|\psi\rangle &= 0 \end{aligned}$$

Different from the earlier cases we can now differentiate between the bit-flip errors. However, we actually get too much information by taking every scalar product. So we do the same as in (10) and construct measuring operators by adding the relevant basis vectors and we end up with the projection operators we already saw in (9).

$$\begin{aligned} P_0 &= |000\rangle\langle 000| + |111\rangle\langle 111| \\ P_1 &= |100\rangle\langle 100| + |011\rangle\langle 011| \\ P_2 &= |010\rangle\langle 010| + |101\rangle\langle 101| \\ P_3 &= |001\rangle\langle 001| + |110\rangle\langle 110| \end{aligned}$$

With this set of projection operators we are now able to identify any single bit-flip error. Hence, we now know why one considers exactly two additional qubits to $|\psi\rangle$ in the three

qubit bit-flip code.

After this we want to shortly consider a different view on the syndrome measurement part. Instead of measuring the four projectors P_0, P_1, P_2, P_3 we measure the observable Z_1Z_2 which is shorthand for $Z \otimes Z \otimes I$ where Z is the Pauli Z-gate. This measurement compares the first and second qubit. If they are different the measurement will result in -1 and 1 if they are the same. This is obvious after looking at this decomposition of Z_1Z_2

$$Z_1Z_2 = (|00\rangle\langle 00| + |11\rangle\langle 11|) \otimes I - (|01\rangle\langle 01| + |10\rangle\langle 10|) \otimes I.$$

Similarly, measuring the observable Z_2Z_3 tells us if qubit two and three are different or the same. Simply by combining these two measurements one can tell if a bit-flip happened or not. For example, if $Z_1Z_2 = 1$ and $Z_2Z_3 = -1$ then qubit three was flipped. Using this method one can draw the quantum circuit as in 7. Where the last two qubits take the roll of syndrome measurement. The fifth qubit compares the first two qubits and the forth qubit tells if the second and third qubit differ.

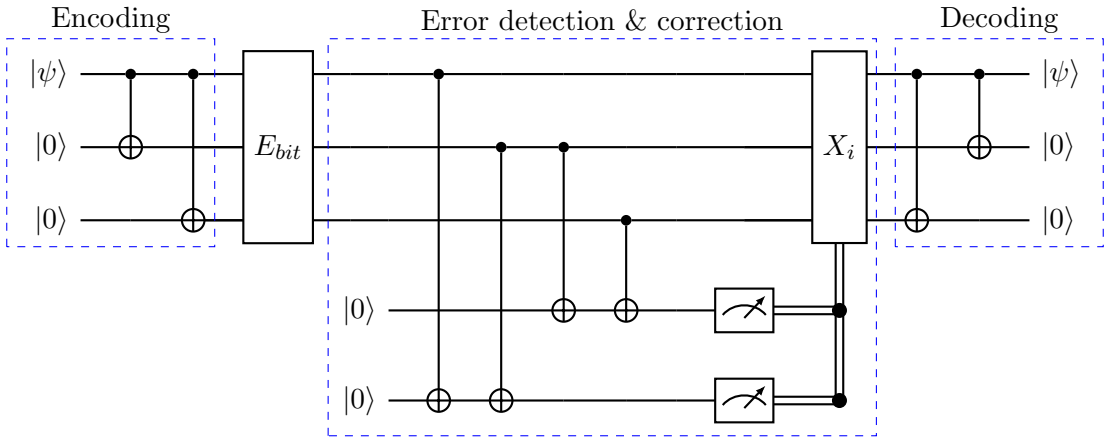


Figure 7: Complete three qubit bit-flip error correction code.

Another question arises naturally by looking at this code. Why do we consider only pure states in quantum error correcting codes? Doesn't this make the procedure less general? It turns out, that this does not restrict us in any way.

5.1.2 QEC for mixed states

Assume we have an initial qubit in the mixed state

$$\rho = p\,|0\rangle\langle 0| + (1-p)\,|1\rangle\langle 1|\,.$$

We do the analog to before but we have to adapt the two qubits we want to add to the correct dimension. So instead of adding $|0\rangle$ we add $|0\rangle\langle 0|$.

$$\rho \otimes (|0\rangle\langle 0|) \otimes (|0\rangle\langle 0|) = p\,|000\rangle\langle 000| + (1-p)\,|100\rangle\langle 100|$$

We continue doing the analog to the pure state case by entangling the three qubits with the correct $CNOT$ gates. Of course we have to adapt to the fact, that our qubits are now represented by matrices and not vectors.

$$CNOT_{1\to 2}\cdot CNOT_{1\to 3}\cdot (p\,|000\rangle\langle 000| + (1-p)\,|100\rangle\langle 100|)\cdot CNOT_{1\to 3}^{\intercal}\cdot CNOT_{1\to 2}^{\intercal}$$
$$= p\,|000\rangle\langle 000| + (1-p)\,|111\rangle\langle 111|$$

Now that our qubit is encoded we can send it through a noisy channel and perform the measurements afterwards. But how do we measure this object, that is now no longer a vector but a matrix?

Before we considered $\langle\psi| M |\psi\rangle$ as the measurement, where M was a projecting operator. If we naively do the same we would get a matrix and not a number, so we have to adapt this properly. The density matrix gives us a mixture of states $|\psi_i\rangle$ which in this case are pure states in $\mathbb{C}^8$. We can write our three qubit mixed state as follows

$$\rho = \sum_i p_i\,|\psi_i\rangle\langle\psi_i|\,.$$

Instead of measuring the density matrix as a whole we could measure the weighted sum of the individual pure states.

$$\sum_i p_i\,\langle\psi_i| M |\psi_i\rangle$$

An easy calculation shows, this expression is simply the trace of the product ρM

$$\begin{aligned} tr(\rho M) &= tr\left(\sum_i p_i\,|\psi_i\rangle\langle\psi_i| M\right)\\ &= \sum_i p_i tr(|\psi_i\rangle\langle\psi_i| M)\\ &= \sum_i p_i tr(\langle\psi_i| M |\psi_i\rangle)\\ &= \sum_i p_i\,\langle\psi_i| M |\psi_i\rangle\,. \end{aligned}$$

In the second line we used the linearity and in the third line the cyclic property of the trace function. So we just worked out that the same procedure can be done for mixed states we just need to adapt our techniques but ultimately we will get the same results. If the first qubit has flipped we can simply calculate $Tr(\rho P_1) = 1$ and $Tr(\rho P_i) = 0$ for all $i = 0, 1, 2, 3$. Doing the reverse operation, which in this case is flipping the first qubit again, we can reconstruct the original state without error. After this thorough analysis of the three qubit bit-flip code we want to discuss another elementary error correcting code, the three qubit phase-flip code.

5.2 Phase-flip code

As mentioned before the bit-flip error is the only error that can occur in classical computers. In this section we assume that only a single sign-flip can happen to our encoded qubit state. So with some probability $p > 0$ a sign-flip occurs to a single qubit. That transforms $\alpha\,|0\rangle + \beta\,|1\rangle$ into $\alpha\,|0\rangle - \beta\,|1\rangle$. Now remember from the chapter before, we introduced the $|+\rangle, |-\rangle$ basis (4). A phase-flip transforms $|+\rangle$ to $|-\rangle$ and vice versa. So in this sense, the

phase-flip can be interpreted as a bit-flip in the Hadamard basis.
The trick behind the phase-flip code lies in the basis change. We can use the exact same method as for the bit-flip code but with respect to the $|+\rangle, |-\rangle$ basis. We start with our initial qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Then we entangle the qubits via the same $CNOT$ gates as before and get $|\psi\rangle = \alpha|000\rangle + \beta|111\rangle$. After that we apply a Hadamard gate to each qubit to get the encoded qubit state

$$|\psi\rangle = \alpha|+++\rangle + \beta|---\rangle .$$

Now we send this encoded qubit through a noisy channel and assume one or less bits experience a phase-flip. Then we are able to detect the error via the same projecting operators as before. But instead of calculating $\langle\psi| P_i |\psi\rangle$ we have to conjugate the measurements with the Hadamard gate first, hence we use $H^{\otimes 3} P_i H^{\otimes 3}$ as measurement operators. Checking this can be done as an easy but tedious exercise.

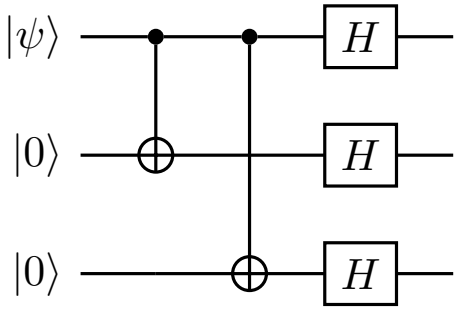


Figure 8: Encoding circuit for the three qubit phase-flip code.

5.3 Shor code

Now we got to know two interesting error correcting codes, both are rather unpractical in application. They cannot protect against an arbitrary error but only a specific type. The goal of this section is to understand the Shor code. The Shor code can protect a single qubit from an arbitrary error. But the two initial codes from before were no waste of time since this code is a combination of the two.
As before we first talk about the encoding of the initial qubit. We start again with our qubit $\alpha|0\rangle + \beta|1\rangle$ and do the same transformations as in the phase-flip code

$$|0\rangle \longrightarrow |+++\rangle \qquad \text{and} \qquad |1\rangle \longrightarrow |---\rangle .$$

Which leaves us with this entangled state of three qubits

$$|\psi\rangle = \alpha|+++\rangle + \beta|---\rangle .$$

And after that we add another six qubits to our system and entangle them via $CNOT$ gates, similar to the bit-flip encoding

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \longrightarrow \frac{(|000\rangle + |111\rangle)}{\sqrt{2}} \qquad \text{and} \qquad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \longrightarrow \frac{(|000\rangle - |111\rangle)}{\sqrt{2}}.$$

Thus, we obtain the following encoded nine qubit system

$$|0\rangle \to |0_L\rangle = \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}$$
$$|1\rangle \to |1_L\rangle = \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}.$$

As mentioned before the Shor code is able to protect a single qubit from phase-flips, bit-flips and even arbitrary errors.

✓ Bit-flip errors

Assume a bit-flip error happened on the second qubit. As discussed in the bit-flip chapter, we perform a measurement on Z_1Z_2 to find out if the first and second qubit are different. Where Z_1Z_2 is shorthand for $Z \otimes Z \otimes I^{\otimes 7}$. A quick reminder that measuring Z_1Z_2 means we calculate $\langle\psi| Z_1Z_2 |\psi\rangle$. If we get $+1$ the first and second qubit are equal and if we get -1 they differ. Since the error happened on the second qubit we will get -1. We now know that a bit-flip happened on qubit one or two. The measurement on Z_2Z_3 also results in -1. We can conclude that the error happened on qubit 2, hence we correct it via a Pauli X-gate.

✓ Phase-flip errors

Now assume a phase-flip error happened on the second qubit in the first block. Then the first block changes from $(|000\rangle + |111\rangle)$ to $(|000\rangle - |111\rangle)$. Notice that the same would have happened if the phase of any of the qubits in the first block had flipped. So we can correct any of these three errors analogously. The same holds of course for the other blocks. To detect this error we compare the sign of the first block to the second one and the sign of the second block to the third one. We can do this by measuring

$$H^{\otimes 9} Z_1Z_2Z_3Z_4Z_5Z_6 H^{\otimes 9} = X_1X_2X_3X_4X_5X_6$$
$$H^{\otimes 9} Z_4Z_5Z_6Z_7Z_8Z_9 H^{\otimes 9} = X_4X_5X_6X_7X_8X_9.$$

Since we assumed, that only one error occurred, we can conclude that the sign of the second block of qubits got flipped. One can undo a phase-flip error on any of the second block qubits by applying $Z_4Z_5Z_6$.

✓ Combined bit- and phase-flip errors

Finally we can even correct combined phase- and bit-flip errors. So assume a bit-flip and a phase-flip happened on qubit one. It is easy to see, that the bit-flip detection procedure would find the bit-flip and equally so for the phase-flip. So what is left to do is correcting the error. We can correct the combined error by applying X_1Z_1 to our qubits.

? Arbitrary errors

We already know that the error space is continuous and there are many other errors that could have happened. We now want to answer the question how the Shor code can correct arbitrary errors that affect only a single qubit.
First we have to define what an arbitrary error or so called noise is. How can we sensibly define such a general concept? Well noise, measurements, bit-flips, phase-flips, etc. can all be described in a more abstract setting as quantum operations.

6 Quantum operations

The goal of this section is to define quantum operations and identify the constraints necessary to ensure consistency with physical principles. One way to define quantum operations is via the following three axioms.

Definition 10. *A quantum operation is a map $\mathcal{E}$ from the set of all density operators to itself with the following properties.*

i) $0 \leq tr[\mathcal{E}(\rho)] \leq 1$

ii) $\mathcal{E}$ *is a linear map*

iii) $\mathcal{E}$ *is completely positive*

Usually one would expect a trace preserving map but *i)* only demands the trace to stay in $[0, 1]$. The reason for this can be seen in (8) as the outcome of a measurement does not always lie on the unit sphere and one has to normalize the result first. So we need this relaxation if we want to consider measurements as operations. If no measurements are done then we can only consider trace preserving maps, so if $tr[\rho] = 1 \implies tr[\mathcal{E}(\rho)] = 1$. These maps are called quantum channels.
We need the map to be positive since we want to map density operators to density operators and preserve the positive semi-definite property, so if $\rho \geq 0 \implies \mathcal{E}(\rho) \geq 0$. But completely positive is even more restrictive, we also want to be able to operate on subsystems without destroying the *psd* property. For example if we have multiple qubits but only use a quantum operation on the last two we still want the result to be *psd*.

Definition 11. $\mathcal{E}$ *is a completely positive map if for all* $n \in \mathbb{N} : Id_n \otimes \mathcal{E}$ *is a positive map.*

Now that we defined quantum operations which are able to model many actions in quantum mechanics we introduce a powerful theorem. The theorem of Choi and Kraus gives us a representation of quantum operations that will prove to be useful in the future [3].

Theorem 12 (Choi & Kraus)**.** *Let* $\Phi : Mat_n(\mathbb{C}) \to Mat_m(\mathbb{C})$ *be a linear map. The following are equivalent:*

i) Φ *is n-positive i.e.* $(Id_n \otimes \Phi)(A) \in Mat_n(\mathbb{C}) \otimes Mat_m(\mathbb{C})$ *is positive* (≥ 0) *whenever* $A \in Mat_n(\mathbb{C}) \otimes Mat_n(\mathbb{C})$ *is positive.*

ii) The Choi-matrix with operator entries $C_\Phi = (Id_n \otimes \Phi)\left(\sum_{i,j} E_{ij} \otimes E_{ij}\right) = \sum_{i,j} E_{ij} \otimes \Phi(E_{ij})$ *is psd. Where* $E_{ij} \in Mat_n(\mathbb{C})$ *is the matrix with* 1 *at entry* (i, j) *and zero anywhere else.*

iii) There are $V_i \in Mat_{m,n}(\mathbb{C})$ *such that for any matrix* $A \in Mat_n(\mathbb{C})$ *it holds that* $\Phi(A) = \sum_{i=1}^{nm} V_i A V_i^*$

iv) Φ *is completely positive.*

Proof. i) $\implies$ *ii)*: Consider the matrix $E = \sum_{i,j} E_{ij} \otimes E_{ij}$. We observe that

$$E^* = \left(\sum_{i,j} E_{ij} \otimes E_{ij}\right)^* = \sum_{i,j} E_{ij}^\mathsf{T} \otimes E_{ij}^\mathsf{T} = \sum_{j,i} E_{ji} \otimes E_{ji} = E$$

and $E^2 = nE$ according to

$$\begin{aligned} E^2 &= \left(\sum_{i,j} E_{ij} \otimes E_{ij}\right)^2 \\ &= \sum_{i,j,l,k} E_{ij}E_{kl} \otimes E_{ij}E_{kl} \\ &= \sum_{i,j,l,k} \delta_{jk}E_{il} \otimes \delta_{jk}E_{il} \\ &= \sum_{j} \sum_{i,l} E_{il} \otimes E_{il} \\ &= nE. \end{aligned}$$

Now $E = \frac{1}{n} EE^*$ which is *psd* and by the n-positivity of Φ we get *ii*).

ii) $\implies$ *iii*): Since C_Φ is *psd* we can express it as

$$C_\Phi = \sum_{i=1}^{nm} v_i v_i^*, \qquad \text{with eigenvectors } v_i \in \mathbb{C}^{nm}.$$

Now we define the following two mappings

$$\begin{aligned} P_k : \mathbb{C}^n \otimes \mathbb{C}^m &\to \mathbb{C}^m & \qquad P_k^* : \mathbb{C}^m &\to \mathbb{C}^n \otimes \mathbb{C}^m \\ \sum_i e_i \otimes w_i &\mapsto w_k & w &\mapsto e_k \otimes w. \end{aligned}$$

In the next step we show that $P_l C_\Phi P_k^* = \Phi(E_{lk})$. So we take an arbitrary vector $w \in \mathbb{C}^m$ and calculate

$$\begin{aligned} P_l \left(\sum_{i,j} E_{ij} \otimes \Phi(E_{ij})\right) P_k^* \cdot w &= P_l \left(\sum_{i,j} E_{ij} \otimes \Phi(E_{ij})\right) (e_k \otimes w) \\ &= P_l \left(\sum_{i,j} E_{ij}e_k \otimes \Phi(E_{ij})w\right) \\ &= P_l \left(\sum_{i} e_i \otimes \Phi(E_{ik})w\right) \\ &= \Phi(E_{lk})w. \end{aligned}$$

Using (6) we can thus write

$$\Phi(E_{lk}) = P_l C_\Phi P_k^* = \sum_{i=1}^{nm} P_l v_i (P_k v_i)^*$$

Defining the operators $V_i \in Mat_{m,n}(\mathbb{C})$ as $V_i e_k = P_k v_i$ we get

$$\Phi(E_{lk}) = \sum_{i=1}^{nm} V_i e_l e_k^* V_i^* = \sum_{i=1}^{nm} V_i E_{lk} V_i^*$$

and by linearity we get the desired expression

$$\Phi(A) = \sum_{i=1}^{nm} V_i A V_i^*.$$

iii) $\implies$ *iv)*: If Φ is defined in this way then it is easy to see that Φ is completely positive. Let r be an arbitrary natural number and $A \in Mat_{rn}(\mathbb{C}) \geq 0$, we get

$$\begin{aligned}(Id_r \otimes \Phi)(A) &= \sum_i (Id_r \otimes V_i) A (Id_r \otimes V_i)^* \\ &= \sum_i (Id_r \otimes V_i) C^* C (Id_r \otimes V_i)^* \\ &= \sum_i \left[C(Id_r \otimes V_i)^*\right]^* \left[C(Id_r \otimes V_i)^*\right].\end{aligned}$$

Where we used that all *psd* matrices A can be decomposed as C^*C. Obviously each summand is *psd* and the same holds for the sum.
iv) $\implies$ *i)*: This is trivially true. □

So now we have a reliable representation of a completely positive map. However, if we also require the map to be trace-preserving that leads to the additional requirement

$$\sum_{i=1}^{nm} V_i^* V_i = Id_n.$$

Which is called the completeness relation. An easy calculation shows that Φ is trace preserving under this condition.

$$\begin{aligned}tr(\Phi(A)) &= tr\left(\sum_{i=1}^{nm} V_i A V_i^*\right) \\ &= tr\left(\sum_{i=1}^{nm} V_i^* V_i A\right) \\ &= tr(A)\end{aligned}$$

Now that we know that we can express quantum noise as a trace preserving, completely positive linear map $\mathcal{E}$, more precisely as a finite sum of matrix products, we can finally explain why the Shor code protects single qubits from arbitrary errors.

If the encoded qubit is in the state $|\psi\rangle = \alpha |0_L\rangle + \beta |1_L\rangle$ before the noise, then it is in the state $\mathcal{E}(|\psi\rangle \langle\psi|) = \sum_i E_i |\psi\rangle \langle\psi| E_i^*$ after the noise. The matrices E_i of the corresponding quantum operation $\mathcal{E}$ are often called operation elements. Since we are only interested in single qubit errors, we assume without loss of generality and for the sake of simplicity that E_i only affects the first qubit.

Since the Pauli matrices form an orthogonal basis of the space of all 2×2 complex matrices we can expand the operation elements as

$$E_i = e_{i1} I + e_{i2} X_1 + e_{i3} Y_1 + e_{i4} Z_1.$$

Keep this in mind, for now we need some new definitions. Before any quantum error correcting code the qubits get encoded by some unitary matrix U to the code-space. The code-space is a subset of a Hilbert space. We additionally define the projector P onto the code-space.

$$P = |0_L\rangle \langle 0_L| + |1_L\rangle \langle 1_L| .$$

For the three qubit bit-flip code the projector would be $P = |000\rangle \langle 000| + |111\rangle \langle 111|$. Now we can define a strong condition for the existence of an error correcting operation and a follow up theorem that allows us to conclude [10, p.436] [8].

Theorem 13 (Knill-Laflamme)**.** *Let C be a quantum code and P the projector onto C. Suppose $\mathcal{E}$ is a quantum operation with operation elements E_i. A necessary and sufficient condition for the existence of an error correcting operation R, correcting $\mathcal{E}$ on C is that*

$$P E_i^* E_j P = \alpha_{ij} P,$$

for some complex hermitian matrix α.

Theorem 14. *Suppose C is a quantum code and R is the error-correction operation to recover from a noise process $\mathcal{E}$ with operation elements E_i. Suppose F is a quantum operation with operation elements F_j which are linear combinations of the E_i, that is $F_j = \sum_i m_{ji} E_i$ for some complex matrix m_{ji}. Then the error-correction operation R also corrects for the effects of the noise process F on the code C.*

We omit the proofs, as they do not contribute to our goal of showing that the Shor code corrects arbitrary single errors.

Using these strong theorems it is now enough to show the Knill-Laflamme condition for the Shor code only for the Pauli matrices. Theorem 14 then implies that all linear combinations of the Pauli matrices are also correctable with the same error correcting operation R. So we have to check

$$P \sigma_i \sigma_j P = \alpha_{ij} P \quad \text{with } \sigma_i \in \{I, X_1, Y_1, Z_1\}.$$

By expanding this equation and analyzing the coefficients, we arrive at a much simpler equation that is easier to verify.

$$(|0_L\rangle \langle 0_L| + |1_L\rangle \langle 1_L|) \sigma_i \sigma_j (|0_L\rangle \langle 0_L| + |1_L\rangle \langle 1_L|) = \alpha_{ij} (|0_L\rangle \langle 0_L| + |1_L\rangle \langle 1_L|)$$

Expanding the LHS results in

$$\begin{aligned} LHS = &|0_L\rangle \langle 0_L| \sigma_i \sigma_j |0_L\rangle \langle 0_L| \\ &+ |0_L\rangle \langle 0_L| \sigma_i \sigma_j |1_L\rangle \langle 1_L| \\ &+ |1_L\rangle \langle 1_L| \sigma_i \sigma_j |0_L\rangle \langle 0_L| \\ &+ |1_L\rangle \langle 1_L| \sigma_i \sigma_j |1_L\rangle \langle 1_L| . \end{aligned}$$

It is now easily obtained by comparing coefficients that our initial condition is equivalent to

$$\begin{aligned}\langle 0_L|\,\sigma_i\sigma_j\,|1_L\rangle &= \langle 1_L|\,\sigma_i\sigma_j\,|0_L\rangle = 0\\ \langle 0_L|\,\sigma_i\sigma_j\,|0_L\rangle &= \langle 1_L|\,\sigma_i\sigma_j\,|1_L\rangle\,.\end{aligned}$$

Since the product of two Pauli matrices is again a Pauli matrix multiplied by some complex number we can simplify this even more to

$$\begin{aligned}\langle 0_L|\,\sigma_i\,|1_L\rangle &= \langle 1_L|\,\sigma_i\,|0_L\rangle = 0\\ \langle 0_L|\,\sigma_i\,|0_L\rangle &= \langle 1_L|\,\sigma_i\,|1_L\rangle\,.\end{aligned}$$

To check all these equations is still a tedious task. Hence, we will only consider one short example for both equations. Let us assume $\sigma_i = Z_1$.

$$\begin{aligned}\langle 0_L|\,Z_1\,|0_L\rangle =& \left[\left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right)^{\otimes 3}\right]^* (Z\otimes I^{\otimes 8})\left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right)^{\otimes 3}\\ =&\frac{1}{8}(\langle 000| - \langle 111|)(\langle 000| + \langle 111|)(\langle 000| + \langle 111|)\\ &\cdot(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)\end{aligned}$$

Factorizing this product a bit differently yields

$$\begin{aligned}\langle 0_L|\,Z_1\,|0_L\rangle =&\frac{1}{8}(\langle 000| - \langle 111|)(|000\rangle + |111\rangle)\\ &\cdot(\langle 000| + \langle 111|)(|000\rangle + |111\rangle)\\ &\cdot(\langle 000| + \langle 111|)(|000\rangle + |111\rangle).\end{aligned}$$

By looking at the first factor of this product one can see that it is equal to 0. By reversing the signs in the last equation, we can immediately tell that the same holds for $\langle 1_L|\,Z_1\,|1_L\rangle$. For another example we consider $\sigma_i = X_1$.

$$\langle 0_L|\,X_1\,|1_L\rangle = \left[\left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right)^{\otimes 3}\right]^* (X\otimes I^{\otimes 8})\left(\frac{|000\rangle - |111\rangle}{\sqrt{2}}\right)^{\otimes 3}.$$

We once again multiply out and consider a better factorization that yields

$$\begin{aligned}\langle 0_L|\,X_1\,|1_L\rangle =&\frac{1}{8}(\langle 100| + \langle 011|)(|000\rangle - |111\rangle)\\ &\cdot(\langle 000| + \langle 111|)(|000\rangle - |111\rangle)\\ &\cdot(\langle 000| + \langle 111|)(|000\rangle - |111\rangle)\end{aligned}$$

Once again the first factor is 0 which implies that $\langle 0_L|\,X_1\,|1_L\rangle = 0$ and trivially the same holds for $\langle 1_L|\,X_1\,|0_L\rangle = 0$.

By checking the rest of these equations one can verify that the Shor code protects against arbitrary single qubit errors, since they are simply linear combinations of Pauli matrices. We already know from the pages before, that the Shor code is capable of protecting against Pauli errors and by theorem 14 we get the desired result.

7 Cloning and broadcasting

In this section we will first consider the classical No-Cloning Theorem and show two separate proofs of it. Later we will discover, that one can formulate this problem in a much more abstract sense and we will prove a No-Cloning Theorem and No-Broadcasting Theorem in a mathematical way. We will learn that broadcasting is only possible in classical theories which excludes not only quantum theories but much more than that. To understand what No-Cloning actually states we must first consider the following postulate of quantum mechanics [10].

Postulate 3. *The evolution of a closed quantum system is described by a unitary transformation. That is, the state $|\psi\rangle$ of the system at t_1 is related to the state $|\psi'\rangle$ of the system at time t_2 by a unitary operator U which depends only on the times t_1 and t_2*

$$\left|\psi'\right\rangle = U\left|\psi\right\rangle.$$

With this strong postulate which we unconsciously used throughout the whole thesis we can prove the quantum version of the No-Cloning Theorem.

Theorem 15 (No-Cloning)**.** *There exists no unitary operator U that can clone an arbitrary quantum state. Specifically, for an arbitrary qubit state $|\psi\rangle$ and some fixed initial state $|e\rangle$, there is no unitary U such that*

$$U(|\psi\rangle \otimes |e\rangle) = |\psi\rangle \otimes |\psi\rangle \qquad \textit{holds for all } |\psi\rangle.$$

Proof. So suppose the unitary transformation U can create an exact copy of the two unknown qubit states $|\psi\rangle$ and $|\phi\rangle$ and let $|e\rangle$ be some standard pure state. That would imply

$$\begin{aligned} U(|\psi\rangle \otimes |e\rangle) &= |\psi\rangle \otimes |\psi\rangle \\ U(|\phi\rangle \otimes |e\rangle) &= |\phi\rangle \otimes |\phi\rangle. \end{aligned}$$

By considering the scalar product of these we receive the following results

$$\begin{aligned} \langle U(\phi \otimes e)|U(\psi \otimes e)\rangle &= \langle \phi \otimes \phi|\psi \otimes \psi\rangle \\ \langle U(\phi \otimes e)|U(\psi \otimes e)\rangle &= \langle \phi \otimes e|\psi \otimes e\rangle. \end{aligned}$$

The first result follows directly from applying the copying transformation, whereas the second one arises from the invariance of unitary transformations in scalar products. Using this equality will lead us to the desired result.

$$\langle\phi|\psi\rangle^2 = \langle\phi \otimes \phi|\psi \otimes \psi\rangle = \langle\phi \otimes e|\psi \otimes e\rangle = \langle\phi|\psi\rangle\langle e|e\rangle = \langle\phi|\psi\rangle.$$

Where the first and third equal sign arises due to the compatibility of the scalar- and tensor product. This equality can only be true if

$$\langle\phi|\psi\rangle = 0 \quad \text{or} \quad \langle\phi|\psi\rangle = 1.$$

In the first case $|\psi\rangle$ and $|\phi\rangle$ are orthogonal and in the second case $|\psi\rangle = |\phi\rangle$. □

That means we cannot copy arbitrary non-orthogonal states with one unitary operator U. This is in particular the reason why we cannot use the repetition code from classical error correction to encode qubit messages. We want to give another proof that uses the linearity of U.

Proof. We are again in the situation where we want to copy our arbitrary pure state $|\psi\rangle$ onto $|e\rangle$. Remember that we can always express $|\psi\rangle$ in a basis. So let $\{\phi_i\}_{i=1}^n$ be an arbitrary basis such that

$$|\psi\rangle = \sum_i a_i |\phi_i\rangle .$$

By assumption on U we expect

$$U(|\psi\rangle \otimes |e\rangle) = |\psi\rangle \otimes |\psi\rangle = \sum_i a_i |\phi\rangle \otimes \sum_i a_i |\phi\rangle .$$

Since U copies an arbitrary state it must also hold for the basis states

$$U(|\phi\rangle \otimes |e\rangle) = |\phi\rangle \otimes |\phi\rangle .$$

But this implies

$$U(|\psi\rangle \otimes |e\rangle) = U\left(\sum_i a_i |\phi\rangle \otimes |e\rangle\right) = \sum_i a_i U(|\phi_i\rangle \otimes |e\rangle) = \sum_i a_i |\phi_i\rangle \otimes |\phi_i\rangle .$$

This is a contradiction since generally speaking it does not hold that

$$\sum_i a_i |\phi_i\rangle \otimes \sum_i a_i |\phi_i\rangle = \sum_i a_i |\phi_i\rangle \otimes |\phi_i\rangle .$$

□

Now that we have proven the No-Cloning Theorem in the quantum setting, we want to look at it in a more mathematical way and see what we can achieve there [2] [1]. We only need a finite dimensional vector space V and a linear map that does the following

$$\begin{aligned} T : V &\to V \otimes V \\ \omega &\mapsto \omega \otimes \omega. \end{aligned}$$

We will see that this is only possible for linearly independent vectors.

Theorem 16 (No-Cloning)**.** *Let V be a vector space with $\{v_1, \ldots, v_n\} \subset V$ pairwise distinct. There is a linear map $T : V \to V \otimes V$ that clones the states $\{v_1, \ldots, v_n\}$ iff they are linearly independent.*

Proof. We first prove the *if*-direction. If $v_1, \ldots, v_n$ are linearly independent then we can expand these to a basis of the vector space and then simply define a linear map by demanding

$$\begin{aligned} T : V &\to V \otimes V \\ v_i &\mapsto v_i \otimes v_i. \end{aligned}$$

By the linear extension theorem this defines the linear map uniquely, hence we get a map that clones the states $v_1, \dots, v_n$.

Now for the other direction we assume that there is a map T that clones the states $v_1, \dots, v_n$ and we want to show that they are therefore linearly independent. Assume that the vectors $v_1, \dots, v_n$ are linearly dependent. Without loss of generality $v_1, \dots, v_{n-1}$ are linearly independent and $v_n = \sum_{i=1}^{n-1} \lambda_i v_i$. Furthermore, T clones all of them. Hence, we get the following two results

$$T(v_n) = T\left(\sum_{i=1}^{n-1} \lambda_i v_i\right) = \sum_{i=1}^{n-1} \lambda_i T(v_i) = \sum_{i=1}^{n-1} \lambda_i (v_i \otimes v_i)$$

$$T(v_n) = v_n \otimes v_n = \sum_{i=1}^{n-1} \lambda_i v_i \otimes \sum_{j=1}^{n-1} \lambda_j v_j = \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \lambda_i \lambda_j v_i \otimes v_j.$$

By comparing coefficients we immediately see that $\lambda_i \lambda_j = 0$ for $i \neq j$. Therefore there can only be one $\lambda_i \neq 0$. This implies that $v_n = v_i$ for some $i \in \{1, 2, \dots, n-1\}$ but we assumed them to be pairwise distinct, hence they were not linearly dependent. □

This proof works in a general setting we don't even need to assume a convex domain or any specific property of a quantum system.
After this result one can dive deeper into the limits of quantum theory by considering broadcasting. The No-Broadcasting Theorem generalizes the No-Cloning Theorem to mixed states. However the statement of the theorem is not obvious. The problem of the No-Broadcasting Theorem in terms of quantum mechanics can be stated as follows [2].

Two Hilbert spaces A and B form a joint quantum system AB where A is secretly prepared to be in state $\rho_s \in \{\rho_0, \rho_1\}$ and B is in some standard quantum state Σ. The initial state is $\rho_s \otimes \Sigma$. The question is, if there is a map T such that $T(\rho_s \otimes \Sigma) = \tilde{\rho_s} \in AB$ such that

$$tr_A(\tilde{\rho_s}) = \rho_s \quad \text{and} \quad tr_B(\tilde{\rho_s}) = \rho_s.$$

Where $tr_A()$ and $tr_B()$ denote the partial trace operation.

Different from the No-Cloning Theorem we will not discuss this problem in the quantum setting but rather in a more abstract way. We first have to define broadcasting in a mathematical sense. Consider again a finite dimensional real vector space V and a linear map $T : V \mapsto V \otimes V$. Moreover we need a linear functional $\varepsilon : V \mapsto \mathbb{R}$. A vector $v \in V$ gets broadcast by the linear map T if

$$(Id_V \otimes \varepsilon)(T(v)) = v \quad \text{and} \quad (\varepsilon \otimes Id_V)(T(v)) = v.$$

In quantum theory the linear functional ε represents the trace. We also need a cone that we want to broadcast. In quantum theory this cone would be the cone of *psd* matrices, but we consider arbitrary cones in the vector space. In this section we also did not use any concrete tensor product but an arbitrary one that lies in between the maximal and minimal tensor product.
We want to show that there is a linear map that broadcasts the entire cone iff the cone is a simplex. For this result we need the following Theorem.

Theorem 17. *Let V_1 and V_2 be two finite dimensional real vector spaces and $C_1 \subset V_1, C_2 \subset V_2$ cones. We also fix two linear functionals*

$$\varepsilon_1 \colon C_1 \to \mathbb{R} \quad \text{and} \quad \varepsilon_2 \colon C_2 \to \mathbb{R}.$$

Now let $\omega \in C_1 \overline{\otimes} C_2$ be an element of the maximal tensor product such that

$$(\varepsilon_1 \otimes \varepsilon_2)(\omega) = 1$$

and

$$\omega_1 = (Id_{V_1} \otimes \varepsilon_2)(\omega) \quad \text{and} \quad \omega_2 = (\varepsilon_1 \otimes Id_{V_2})(\omega).$$

If ω_1 (or ω_2) is an extreme ray in C_1 (resp. C_2) then $\omega = \omega_1 \otimes \omega_2$.

Proof. Choose an arbitrary linear functional $f : C_2 \to \mathbb{R}$ such that $0 \leq f \leq \varepsilon_2$ on C_2. Then define

$$w_{1,f} = \frac{(Id_{V_1} \otimes f)(\omega)}{f(\omega_2)} \in C_1.$$

It is easily seen that $\varepsilon_1(\omega_{1,f}) = 1$.

$$\begin{aligned}
\varepsilon_1(Id_{V_1} \otimes f)(\omega) &= \varepsilon_1(Id_{V_1} \otimes f) \left(\sum_i x_i \otimes y_i \right) \\
&= \varepsilon_1 \left(\sum_i x_i \otimes f(y_i) \right) \\
&= \sum_i \varepsilon_1(x_i) \otimes f(y_i) \\
&= f \left(\sum_i \varepsilon_1(x_i) \otimes y_i \right) \\
&= f(\varepsilon_1 \otimes Id_{V_2}) \left(\sum_i x_i \otimes y_i \right) \\
&= f(\varepsilon_1 \otimes Id_{V_2})(\omega) \\
&= f(\omega_2)
\end{aligned}$$

By construction of $\omega_{1,f}$ the following holds for arbitrary linear functionals

$$\forall g \in C^\vee \colon \quad g(\omega_{1,f}) f(\omega_2) = \frac{(g \otimes f)(\omega) f(\omega_2)}{f(\omega_2)} = (g \otimes f)(\omega). \tag{11}$$

If we can show (11) for w_1 instead of $w_{1,f}$ then we are done, since it holds for arbitrary functionals g. So let us take a closer look at ω_1

$$\begin{aligned}
\omega_1 &= (Id_{V_1} \otimes \varepsilon_2)(\omega) \\
&= (Id_{V_1} \otimes (f + \varepsilon_2 - f))(\omega) \\
&= (Id_{V_1} \otimes f)(\omega) + (id_{V_1} \otimes (\varepsilon_2 - f))(\omega) \\
&= f(\omega_2)\omega_{1,f} + (Id_{V_1} \otimes (\varepsilon_2 - f))(\omega).
\end{aligned}$$

In this representation ω_1 is a sum of two elements of C_1. But we remember that w_1 is extremal in C_1. Hence, both summands must be multiples of ω_1.
Now we can conclude as follows

$$(g \otimes f)(\omega) = g(\omega_{1,f}) f(\omega_2) = g(\omega_1) f(\omega_2).$$

Since this holds for arbitrary functionals g we can conclude that $\omega = \omega_1 \otimes \omega_2$. □

With this strong theorem we can prove the following rather concrete corollary.

Corollary 18. *Let V be a finite dimensional real vector space, $C \subset V$ a cone and $v \in C$ an extreme ray. If the linear map $T : V \to V \otimes V$ with $T(C) \subset C \bar{\otimes} C$ broadcasts v, then T clones v.*

Proof. Since T broadcasts v, there is by definition an already fixed linear functional ε such that

$$(Id_V \otimes \varepsilon)(T(v)) = v \quad \text{and} \quad (\varepsilon \otimes Id_V)(T(v)) = v.$$

Now we apply the strong theorem from above and get immediately that $T(v) = v \otimes v$ and we are done. □

With this corollary we can now finally prove that there is a linear map that broadcasts the entire cone iff the cone is a simplex.

Theorem 19. *A cone can be universally broadcast iff the cone is a simplex.*

Proof. If the entire cone gets broadcast by the linear map T, then T especially broadcasts the extreme rays of C. By Corollary 18 we know that they get cloned. That however implies by 16 that they are linearly independent and hence form a simplex. For the other direction assume that we have a simplex together with a linear functional ε. Then the extreme rays are linearly independent, hence they can be cloned by a linear map T. Every point in the simplex is a convex combination of said rays. Assume the extreme rays are $\{v_1, \ldots, v_n\}$ and choose v arbitrarily in the cone. It turns out that the map that clones the extreme rays broadcasts the vectors in between. To show that we can use T on v and get

$$T(v) = T\left(\sum_{i=1}^{n} \lambda_i v_i\right) = \sum_{i=1}^{n} \lambda_i T(v_i) = \sum_{i=1}^{n} \lambda_i (v_i \otimes v_i).$$

Now, it is easy to see that

$$(\varepsilon \otimes Id_V)(T(v)) = (\varepsilon \otimes Id_V)\left(\sum_{i=1}^{n} \lambda_i (v_i \otimes v_i)\right) = \sum_{i=1}^{n} \lambda_i v_i = v.$$

In the second equality we used that $\varepsilon(v_i) = 1$. We can always scale our extreme rays $\{v_1, \ldots, v_n\}$ such that this is true. Trivially the same can be done for $(Id_V \otimes \varepsilon)(T(v))$.
Hence, we found a linear map that broadcasts the entire simplex and we are done. □

8 Conclusion and future directions

This thesis has provided an accessible, mathematically grounded introduction to quantum error correction. Starting from the principles of quantum mechanics, we developed a formal understanding of qubits and measurements and quantum circuits, illustrated through examples like the Bloch sphere and quantum teleportation.

A central theme was the contrast between classical and quantum error correction. Quantum systems face unique challenges, most notably, the No-Cloning Theorem and the continuity of the error space. Yet, reliable correction is achievable, as demonstrated by codes like the bit-flip, phase-flip, and the Shor code.

In the final chapter of the thesis the important No-Cloning and No-Broadcasting Theorems were introduced in a quantum environment and later analysed in a general and mathematical setting. The results of this mathematical perspective are elegant theorems with broad implications.

Looking ahead, many intriguing directions open up beyond this thesis. One natural extension is the study of more advanced codes, such as stabilizer codes and topological codes, which can reliably correct errors across multiple qubits. These schemes are crucial for the development of large-scale, practical quantum systems.

In sum, QEC is essential for robust quantum information processing. As this thesis has shown, it lies at the intersection of physics, mathematics, and computer science, offering both deep insights and practical avenues for advancing quantum technology.

References

[1] Howard Barnum et al. *Cloning and Broadcasting in Generic Probabilistic Theories*. 2006. arXiv: quant-ph/0611295 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/0611295.

[2] Howard Barnum et al. "Noncommuting Mixed States Cannot Be Broadcast". In: *Physical Review Letters* 76.15 (Apr. 1996), pp. 2818–2821. ISSN: 1079-7114. DOI: 10.1103/physrevlett.76.2818. URL: http://dx.doi.org/10.1103/PhysRevLett.76.2818.

[3] Man-Duen Choi. "Completely positive linear maps on complex matrices". In: *Linear Algebra and its Applications* 10.3 (1975), pp. 285–290. ISSN: 0024-3795. DOI: https://doi.org/10.1016/0024-3795(75)90075-0. URL: https://www.sciencedirect.com/science/article/pii/0024379575900750.

[4] Gemma De les Coves. "Quantum information theory - A crash course for Modern Physics". In: (2023).

[5] P. A. M. Dirac. "A new notation for quantum mechanics". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 35.3 (1939), pp. 416–418. DOI: 10.1017/S0305004100021162.

[6] A Ekert et al. *Introduction to Quantum Information Science*. https://qubit.guide. Dec. 8, 2024.

[7] Nathan Kaplan. "Coding Theory Lecture Notes". In: (2011). URL: https://www.math.uci.edu/~nckaplan/teaching_files/kaplancodingnotes.pdf.

[8] Emanuel Knill, Raymond Laflamme, and Lorenza Viola. "Theory of Quantum Error Correction for General Noise". In: *Physical Review Letters* 84.11 (Mar. 2000), pp. 2525–2528. ISSN: 1079-7114. DOI: 10.1103/physrevlett.84.2525. URL: http://dx.doi.org/10.1103/PhysRevLett.84.2525.

[9] Jacobus Hendricus van Lint. *Introduction to coding theory*. eng. 3., rev. and expanded ed. Graduate texts in mathematics. 1999. ISBN: 3540641335.

[10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.