

# Modern Approaches to Topological Quantum Error Correction



Swansea University  
Prifysgol Abertawe

Pedro Parrado Rodríguez

Department of Physics

Swansea University

Submitted to Swansea University in fulfilment  
of the requirements for the degree of

*Doctor of Philosophy*

2021

## Abstract

The construction of a large-scale fault-tolerant quantum computer is an outstanding scientific and technological goal. It holds the promise to allow us to solve a variety of complex problems such as factoring large numbers, quick database search, and the quantum simulation of many-body quantum systems in fields as diverse as condensed matter, quantum chemistry, and even high-energy physics. Sophisticated theoretical protocols for reliable quantum information processing under imperfect conditions have been developed, when errors affect and corrupt the fragile quantum states during storage and computations. Arguably, the most realistic and promising approach towards practical fault-tolerant quantum computation are topological quantum error-correcting codes, where quantum information is stored in interacting, topologically ordered 2D or 3D many-body quantum systems. This approach offers the highest known error thresholds, which are already today within reach of the experimental accuracy in state-of-the-art setups. A combination of theoretical and experimental research is needed to store, protect and process fragile quantum information in logical qubits effectively so that they can outperform their constituting physical qubits. Whereas small-scale quantum error correction codes have been implemented, one of the main theoretical challenges remains to develop new and improve existing efficient strategies (so-called decoders) to derive (near-)optimal error correction operations in the presence of experimentally accessible measurement information and realistic noise sources. One main focus of this project is the development and numerical implementation of scalable, efficient decoders to operate topological color codes. Additionally, we study the feasibility of implementing quantum error-correcting codes fault-tolerantly in near-term ion traps. To this end, we use realistic modeling of the different noise sources, computer simulations, and most modern quantum information approaches to quantum circuitry and noise suppression techniques.

---

**Declaration**


This work has not previously been accepted in substance for any degree and is not being concurrently submitted for any degree.

Signed: .....  ..... (candidate)

Date: 20-12-2021 .....

**Statement 1**

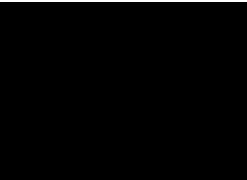
This thesis is the result of my own investigations, except where otherwise stated. Where correction services have been used, the extent and nature of the correction is clearly marked in a footnote(s). Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed: .....  ..... (candidate)

Date: 20-12-2021 .....

**Statement 2**

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed: .....  ..... (candidate)

Date: 20-12-2021 .....

---

## Authorship Declaration

The following people and institutions contributed to the publication of work undertaken as part of this thesis:

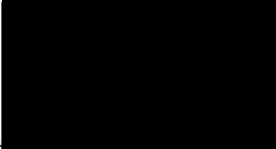
	Name	Institutions
Candidate	Pedro Parrado-Rodríguez	Department of Physics, Swansea University, Singleton Park, Swansea SA2 8PP, United Kingdom
Author 1 (PhD supervisor)	Markus Müller	Department of Physics, Swansea University, Singleton Park, Swansea SA2 8PP, United Kingdom  Institute for Quantum Information, RWTH Aachen University, D-52056 Aachen, Germany  Peter Grünberg Institute, Theoretical Nanoelectronics, Forschungszentrum Jülich, D-52425 Jülich, Germany
Author 2	Ciarán Ryan-Anderson	Department of Physics, Swansea University, Singleton Park, Swansea SA2 8PP, United Kingdom  Current affiliation: Honeywell Quantum Solutions, 303 S. Technology Ct., Broomfield, Colorado 80021, USA
Author 3	Alejandro Bermúdez	Departamento de Física Teórica, Universidad Complutense, 28040 Madrid, Spain.
Author 4	Manuel Rispler	Institute for Quantum Information, RWTH Aachen University, 52056 Aachen, Germany  Peter Grünberg Institute, Theoretical Nanoelectronics, Forschungszentrum Jülich, D-52425 Jülich, Germany



---

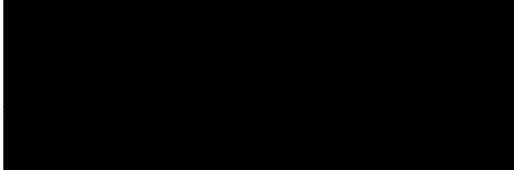
“Crosstalk Suppression for Fault-tolerant Quantum Error Correction with Trapped Ions”, Pedro Parrado-Rodríguez, Ciarán Ryan-Anderson, Alejandro Bermúdez, and Markus Müller, *Quantum* 5, 487 (2021).

- Located in Chap. 6.
- Candidate contributions (65%): programming and execution of numerical simulations; analysis, interpretation and visualization of obtained numerical data; derivation of effective error model for crosstalk suppression via refocusing; derivation of refined error models; main contribution to writing of the manuscript.
- Author 1 contributions (10%): original idea; supervision; critical revision of the manuscript.
- Author 2 contributions (15%): development of the simulation platform (PECOS) used for the numerical simulations; compilation of quantum circuits; contribution to the manuscript.
- Author 3 contributions (10%): derivation of the microscopic AC-Stark shift crosstalk model; contribution to the manuscript.
- We the undersigned agree with the above stated “proportion of work undertaken” for each of the above published peer-reviewed manuscripts contributing to this thesis:

Signed Candidate ..... 

Author 1 ..... 

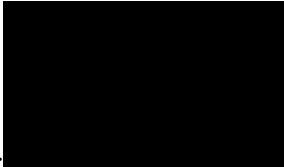
Author 2 ..... 

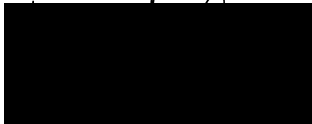
Author 3 ..... 


---

“Rescaling decoder for 2D topological quantum color codes on 4.8.8 lattices”, Pedro Parrado-Rodríguez, Manuel Rispler, and Markus Müller. (2021) arXiv:2112.09584

- Located in Chap. 8.
- Candidate contributions (75%): Original idea; development of the code; finding solutions to the problems of adapting the decoder to the new lattice; programming, execution and analysis of the numerical simulations; main contribution to writing of the manuscript.
- Author 1 contributions (10%): supervision; critical revision of the manuscript.
- Author 4 contributions (15%): co-supervision; contribution to and critical revision of the manuscript.
- We the undersigned agree with the above stated “proportion of work undertaken” for the above published (currently under peer-review) manuscript contributing to this thesis:

Signed Candidate ..... 

Author 1 ..... 

Author 4 ..... 

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>I</b>	<b>Background</b>	<b>11</b>
<b>2</b>	<b>Quantum computing</b>	<b>13</b>
2.1	Qubit systems . . . . .	14
2.2	Gates . . . . .	17
2.3	Noise modelling . . . . .	19
<b>3</b>	<b>Quantum error correction</b>	<b>23</b>
3.1	Stabilizer codes . . . . .	26
3.1.1	Basic concepts . . . . .	27
3.1.2	Circuits for stabilizer measurement . . . . .	31
3.1.3	Fault tolerance . . . . .	32
3.2	Topological codes . . . . .	36
3.2.1	Surface code . . . . .	38
3.2.2	Color codes . . . . .	40
3.2.2.1	Steane code and flag-based readout techniques . . . . .	42
3.3	Decoders . . . . .	45
<b>4</b>	<b>Simulation of quantum circuits</b>	<b>49</b>
4.1	State-vector simulation . . . . .	52
4.2	Stabilizer simulation . . . . .	53
4.3	Other simulation techniques . . . . .	56

## CONTENTS

---

<b>5</b>	<b>Ion traps</b>	<b>59</b>
5.1	Ion traps for quantum computation . . . . .	60
5.2	Scalability approaches . . . . .	61
5.3	Qubits in the ion trap . . . . .	63
5.4	Qubit control . . . . .	64
<b>II</b>	<b>Research results</b>	<b>69</b>
<b>6</b>	<b>Feasibility study of QEC with trapped ions</b>	<b>71</b>
6.1	Background . . . . .	73
6.2	Error model . . . . .	75
6.2.1	Idle errors . . . . .	76
6.2.2	Operational noise . . . . .	81
6.2.3	Crosstalk . . . . .	83
6.3	Adversity of crosstalk on QEC . . . . .	88
6.3.1	Coherent vs. incoherent crosstalk . . . . .	88
6.3.2	Impact of entangling crosstalk noise on QEC performance . . . . .	91
6.3.3	Impact of Stark-shift crosstalk noise on QEC performance . . . . .	94
6.4	Active suppression of crosstalk with refocusing pulse sequences . . . . .	95
6.4.1	Analytical suppression of residual noise after refocusing . . . . .	96
6.5	Numerical analysis of the performance of QEC with active crosstalk suppression . . . . .	100
6.6	Summary and outlook . . . . .	102
<b>7</b>	<b>Union-find decoder for color codes</b>	<b>105</b>
7.1	Background . . . . .	106
7.1.1	Union find decoder for surface codes . . . . .	108
7.1.1.1	Growing and merging clusters . . . . .	108
7.1.1.2	Peeling decoder . . . . .	112
7.1.2	Results and analysis . . . . .	115
7.2	Extension to color codes . . . . .	118
7.2.1	Growing and merging clusters . . . . .	119
7.2.2	Peeling decoder on color codes . . . . .	121

7.2.3	Implementation details . . . . .	123
7.3	Performance of the decoder . . . . .	125
7.4	Summary and outlook . . . . .	129
<b>8</b>	<b>Rescaling decoder</b>	<b>131</b>
8.1	Background . . . . .	132
8.1.1	Introduction to the rescaling decoding approach . . . . .	135
8.1.2	Motivation and previous work . . . . .	137
8.2	The decoder algorithm for the 4.8.8 color code lattice . . . . .	138
8.2.1	Minimal cell . . . . .	143
8.2.2	Belief propagation . . . . .	144
8.2.3	Corner updates . . . . .	149
8.2.4	Splitting the stabilizers . . . . .	150
8.2.5	Rescaling of the cells . . . . .	156
8.2.6	Decoding of the final lattice . . . . .	159
8.3	Results . . . . .	160
8.4	Summary and outlook . . . . .	162
<b>9</b>	<b>Conclusions and outlook</b>	<b>165</b>
	<b>References</b>	<b>173</b>

## Acknowledgements

First and foremost, I would like to thank my supervisor Markus Müller, who has never failed to give helpful advice and guidance, and he is always ready to find time in his busy schedule to help. He has been a great inspiration and an example of passion, hard work, creativity, and leadership. His open mind, invaluable feedback, and his endless support were crucial for the development and finalization of this thesis.

Secondly, I would like to thank the present and past members of our group: David Amaro, Fernando Martínez, Mauricio Gutiérrez, Amit Kumar Pal, Davide Vodola, Ciarán Ryan-Anderson, Farid Shahandeh, Jemma Bennett, Alejandro Bermúdez, Manuel Rispler, Thomas Botzung, Eliana Fiorelli, Lorenzo Cardarelli, Sascha Heußen, David Locher, and Lukas Bödeker. Working with them has been a fantastic experience, filled with enlightening discussions and good memories. It has been an absolute pleasure to be part of this wonderful group. Special thanks go to Fernando Martínez, who was always there to help with all sorts of problems and proved to be a great colleague and friend along the way.

Third, I want to thank all my friends, both the old and the new ones I made during my time at Swansea, Aachen, and Poland, for their encouragement, support, and the very good times together. In this line, I would like to mention my friend Svart Langsomme and his particular sense of humor.

I want to acknowledge the computational resources provided by Supercomputing Wales, as well as the help from the support team. The courses provided on the use of the cluster and the feedback from Edd Bennett greatly improved the efficiency of the use of resources from my simulations.

Finally, I want to thank Joanna Kuśnierek for being there every day, helping me push through the hardest moments, and accompanying me in the best.

Por supuesto, no podría acabar sin agradecer a mis padres, hermanos, y al resto de mi familia por estar siempre ahí para apoyarme.

## CONTENTS

---



# Chapter 1

## Introduction

## 1. INTRODUCTION

---

The use of tools for computation has been present in humankind since early history. Ancient cultures such as Greeks, Babylonians, Mayans or Aztecs used parallel lines on a table and placed pebbles or kernels of grain for counting and small operations. Our modern words *calculate* and *calculus* come from the word *calculi*, the name that ancient Romans had for the pebbles used on their counting boards. With the development of mathematics and science, new tools were required for increasingly complex calculations, and inventions like Napier's bones, Pascal's calculator or the slide rule became enormously successful [3].

The increasing necessity of computational tools leads to computers' development, in a revolution that we are still living today. From the early ideas of Charles Babbage on his first mechanical computer and the notion of the universal machine by Alan Turing, to Von Neumann's model of computing architecture [4–6], modern computers have evolved through the development of electronics, transistors, integrated circuits, and all the underlying physics and engineering knowledge.

The development of modern computers and information science has had a massive impact on society. Today, desktop computers and smartphones are a part of our everyday life, used for all sorts of tasks, from communication and entertainment to accounting or office work. In science, the access to the computational power offered by computers has made possible the development of new methods to study nature. Simulations and numerical analysis are now a fundamental aspect of science and engineering that allows people to solve increasingly complex problems [7]. Most universities and research centers have access to supercomputing centers, with clusters of specialized computing nodes available as a computational hammer to crack problems that would be unthinkable a few decades ago.

However, some challenges are still beyond the possibilities of current computational devices. Feynman already foresaw this problem in the early 80's, when he pointed out that simulating quantum systems with classical computers is not efficient, as the number of computational resources needed to simulate a quantum system does not scale well with the size of the system we want to simulate [8]. As illustrated by his famous quote:

*Nature isn't classical, dammit, and if you want to make a simulation of Nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem because it doesn't look so easy.*

---

Feynman planted the seed of an idea leading to a fundamentally new type of computing machine. A quantum computing device would exploit the complexity of the wavefunction of a many-particle quantum system to solve computational problems. One fundamental problem that has been used as an example of the computational power derived from quantum computation is the factorization of large numbers [9, 10]. The complexity of solving this task with a classical computer is the stone over which data encryption is built [11], as it could take on the order of several years even for a cluster of supercomputers to find these prime factors, ensuring the safety of the encryption. However, a quantum computer using Shor’s factorization algorithm [12, 13] could find the solution in a much shorter time, which would have significant implications for the future of classical encryption. This promise has served as a great motivation for developing quantum technologies.

The power to efficiently factorize numbers is only one example of the potential capabilities of a quantum device. Harnessing the interactions of many-particle entangled states can open the door to the efficient simulation of quantum systems [8, 14–33], which was the task that originally inspired the development of quantum computation. Over the last years, the first proposals of quantum simulation have been demonstrated experimentally in several physical platforms, like ion traps, ultracold atoms, or superconducting qubits [28, 34–39]. These new powerful methods could play a key role in applications like computational quantum chemistry [18, 19], which can lead to the development of multiple fields like nanotechnology, pharmaceuticals, agriculture (nitrogen fixation process), and sustainability (carbon sequestration, energy storage and production). The power of quantum simulations would also greatly improve our understanding of the behavior of biological molecules, opening a new branch of nano-engineering technologies.

Nevertheless, the potential applications of quantum technologies reach far beyond the simulation of physical systems. Quantum communication and cryptography, which could enable secure exchange of information beyond the limits of classical communication channels [40–44]. Quantum metrology, that allows measurements with a precision capable of detecting even gravitational waves [45–52]. Quantum machine learning techniques could offer a speedup over classical neural networks [53–58], or work in combination with them (or other classical algorithms) as a quantum processing unit [59–64]. Recently, applications of quantum computers to solve optimization problems

## 1. INTRODUCTION

---

have become increasingly popular in the literature e.g. [18, 19, 54, 64–78]. Using principles from adiabatic quantum computation, the solution for the optimization problem can be encoded in the ground state of a multi-qubit state [79]. Algorithms like Quantum Approximate Optimization Algorithm (QAOA) or the recent Filtering Variational Quantum Eigensolver [80, 80–85] allow even near term noisy devices to find approximations to this ground state, many times using a hybrid quantum-classical approach. These algorithms could find an application in the near future, which has led to multiple companies, startups and universities investing time and resources in the topic [71, 86].

With so many promising applications, the motivations to develop a quantum computing device are clear. Thus, the next question is: how can we build such a device? For a quantum computer, we typically want a quantum system in which we can define qubits, control their state, generate entanglement between different qubits, and measure the state [87]. Over the last decades, multiple platforms have been proposed and studied [88–94], many of which have demonstrated very promising capabilities. Prominent examples are the recent demonstrations of quantum advantage realized in superconducting [95] and photonic quantum devices [96, 97]. Ion traps, superconducting circuits, nitrogen-vacancy centers, quantum dots, Rydberg atoms and photonic devices are some of the leading platforms of qubit-based quantum computation [88].

The main challenge in the development of quantum computing platforms lies in the fragile nature of quantum states. Imperfections on the gates, fluctuations in the magnetic fields, or even spontaneous decay are just examples of different processes that can affect the information stored on a quantum device and corrupt the results of a computation. To achieve a quantum computing device capable of large scale computations with high fidelity is a challenge that not only requires the refinement of experimental techniques and quantum control, but also the development of quantum error correction (QEC) protocols that can detect and correct the detrimental effects of the different noise sources affecting a device [98–100]. Therefore, the use of QEC is a fundamental field in the development of large-scale devices capable of accessing the most powerful capabilities of quantum computation.

Quantum error-correcting protocols can be applied at different levels of computation. At a lower level, different techniques can be applied to maximize the fidelity of the operations [101–108] using particular properties of the noise to cancel its effects.

---

For instance, the correction of some coherent sources of noise can be done by applying a sequence of pulses that cancel the unwanted evolution [101–106]. While these techniques can greatly reduce the impact of different sources of noise, the power of QEC becomes a fundamental part of the development of quantum computing with the introduction of quantum error correcting codes (QECC's) [100, 109–114]. Similar to classical error-correcting codes, QECC's can encode the information of logical qubits into a multi-qubit entangled state. This encoding can, in turn, allow the detection of errors occurring on the physical qubits, making possible the application of a recovery operation that will restore the logical information encoded on the system. From this general idea, the development of QECC's started with small codes (e.g. repetition code, 5-qubit code, concatenated codes or the Bacon-Shor codes) capable of detecting and correcting errors through the measure of some parity checks in different Pauli basis, a concept described as *stabilizer codes* [100, 109, 115]. From there, numerous QECC's have been designed, often with a particular type of quantum noise as the target of the code. For instance, we can find codes specialized in qubit loss [116–118], continuous errors [119–121], amplitude damping [122] or quantum communication [43, 123]. For a practical application of QEC, the next milestone for QEC is the development of Fault Tolerant (FT) quantum error correction protocols, capable of preventing the unwanted propagation of errors through the circuit elements [124–128]. Fault Tolerant protocols underlie the construction of one of the most prominent results of QEC: the threshold theorem [127, 129, 130]. Provided that the physical error rate is lower than a critical value, the logical error rate on the encoded qubits can be arbitrarily reduced by scaling the size of the code. These ideas form the paradigm of Fault Tolerant Quantum Computation (FTQC), which aims to achieve a large-scale quantum computing device capable of realizing long computations with qubits that are encoded and error-corrected through the whole process [127]. Currently, topological codes are being studied as one of the main options for the future of FTQEC. These codes encode the logical qubits on the topological degrees of freedom of a scalable lattice [109–114, 131], and present high error thresholds [109, 131]. Furthermore, their implementation relies on purely local operations, which allows the embedding of the codes in 2D and 3D lattices. Surface codes [110, 111] and color codes [113, 132] are the most prominent topological codes in the literature, with the latter being one of the main focal points of this thesis.

## 1. INTRODUCTION

---

The realization of fully fledged FTQC is still out of reach of the current experimental devices. The state of quantum technologies is now popularly referred to as the NISQ era, which stands for Noisy Intermediate Scale Quantum technology [65]. State-of-the-art devices have on the order of tens of qubits and noisy gates, and have yet to overcome the daunting challenge of scaling from tens of qubits to the millions of physical qubits<sup>1</sup> that we expect to need for the most interesting applications [13]. Experimental devices are, nevertheless, progressing at a good pace [133], with high gate fidelities (e.g. superconducting circuits [134–136], silicon qubits [137, 138], ion traps [90–92, 139–145] or penning traps [146]) and demonstrations of QEC protocols already shown, e.g. using trapped ions [147–156], nuclear magnetic resonance [157, 158], superconducting circuits [93, 159–163], or nitrogen-vacancy centres [164, 165]. NISQ devices are already available (e.g. IBM quantum cloud computer [166]), and applications in the industry are already being found for these state-of-the-art technologies [68–78]. Over the next few decades, we could experience a new revolution in computing similar to the one originated by the discovery of transistors and the development of classical computation. During the writing of this thesis, some prominent landmarks have been achieved, like code state preparations of large surface codes [167], logical operations on encoded qubits [168, 169] fault tolerant stabilizer measurement [170], repeated QEC cycles [171–174] as well as fault tolerant magic state preparation and injection [169].

Beyond improvements in the experimental capabilities, the long-term implementation of FTQC requires solutions for additional theoretical challenges [175]. In order to find an optimal encoding strategy, we need realistic modeling of the existing hardware and a complete understanding of the error sources that affect the device. QEC protocols can then be adapted to more effectively target the noise and mitigate the damaging effects of errors in the computation. In this line, multiple works have been published studying different noise modelings for existing quantum devices, e.g. [149, 176–179].

However, testing and evaluating these new strategies can also be problematic due to the exponential scaling of the Hilbert space of quantum systems with the number of qubits. The use of efficient algorithms to simulate noisy quantum circuits is thus

---

<sup>1</sup>To factorize a number of  $N$  bits, we need around  $2N$  logical qubits. For instance, for a number of  $N = 2000$  bits (600 digits) we would need around  $2N = 4000$  logical qubits. Now, the number of physical qubits needed depends strongly on the physical error rate, as well as the circuit depth. Assuming an error rate of one tenth of the threshold rate, this leads to 14500 physical qubits required to encode each logical qubit. Thus, under this assumptions we need on the order of  $4000 \cdot 14500 = 58 \cdot 10^6$  physical qubits for the factorization of a 600 digit number [13].

---

an excellent tool for the development and understanding of QEC protocols on different platforms [180–182].

At this point, if we can scale the number of qubits while keeping under control the error rate of the gates, it is possible to reduce the logical error rate on our encoded qubits arbitrarily. Topological QEC codes provide a system capable of encoding logical qubits in a large lattice of physical qubits [109–114, 131]. With this encoding, it is possible to measure a set of *stabilizers* operators, which give valuable information about the position of errors that can be used to apply a recovery operation. However, the interpretation of the information from the measurement of the stabilizers, typically called the *syndrome*, is not straightforward. The problem of finding a recovery operation for an error case given the information from the syndrome is called the decoding problem, and finding an optimal decoding strategy (which always finds the recovery operation with the maximum success rate) is believed to be an NP-hard problem in general [183, 184]. For any practical application of topological QEC, a powerful classical computer would be needed to decode the information of the syndrome and apply a recovery operation. If the decoding algorithm is not fast enough, this process will slow down the quantum computation, leading to additional errors from decoherence and other processes<sup>1</sup>. Thus, finding efficient and scalable decoding algorithms is a problem of great importance for the future of QEC.

Furthermore, the success rate of the decoder is one limiting factor for the error rate threshold of the QEC protocol, which has important implications for the required error rate of the experimental device. Pushing the error rate below the threshold increases considerably the error suppression capabilities with the scaling of the lattice [127, 129, 130]. For a given error rate, achieving a higher threshold with an improved decoder can drop the logical error rate by even orders of magnitude as we scale the code. Multiple decoding algorithms have been proposed and studied for surface codes [185–188] and color codes [189–196]. Moreover, machine learning techniques have also been successfully applied to the decoding problem [197–204].

The contributions of this thesis are aimed at two fronts. First, we focus on near-term devices, studying the feasibility of small-scale QEC protocols applied to ion traps. We

---

<sup>1</sup>As discussed in [109], if the rate at which the syndrome information is generated is faster than the rate at which it is processed, this increasing backlog of syndrome data can lead to an exponential slow down during the computation

## 1. INTRODUCTION

---

use noise models based on a microscopic description of the different physical phenomena that can generate noise. Using numerical simulations, we study the performance of a seven qubit QEC code, the smallest fully-functioning color code, currently at the center of recent QEC efforts (e.g. [169, 171]). We simulate the circuits involved in the protocol, finding target experimental values to achieve a regime of beneficial error correction, in which the error correction capabilities of the protocol compensate the possible errors introduced by the gates of the circuit. Furthermore, we study the use of a refocusing pulse sequence to suppress crosstalk, a particularly damaging error source originated from undesired interactions between the target qubits of a quantum gate and the neighboring qubits in the ion chain.

The second front of the thesis is focused on the long-term development of QEC. We study the use of two decoding algorithms for color codes, a particular class of topological QECC's for which decoding has proven a particularly challenging task. The first decoder that we study is based on the ideas presented in [193] for an efficient decoder for surface codes with a computational complexity that scales almost linearly with the number of qubits. Using these ideas, we develop a decoding algorithm for color codes, and study its performance with numerical simulations. The second decoder that we study is built over the algorithm presented in [189] for a decoder based on the rescaling of the color code lattice. Initially presented for the hexagonal color code lattice, we extend the algorithm to the square-octagon (4.8.8) color code lattice. This lattice is particularly relevant, as it is the only 2D color code lattice that allows the transversal implementation of the entire Clifford group [113], which simplifies its FT implementation. Additionally, the 4.8.8 color code lattice requires the minimal number of data qubits  $n$  for a code of distance  $d$ . Using numerical simulations, we analyze the execution of the different steps of the decoding algorithm and study its performance. With these two projects, we aim to develop efficient decoding techniques, which can potentially be extended to more complex error models, including qubit losses and correlated noise.

This thesis is organized as follows. First, chapter 2 introduces the main basic concepts of quantum computation used through the entire work, including the notation for states and quantum operations, as well as the basics of noise modeling. In chapter 3, the main concepts of QEC are introduced, including stabilizer codes, fault tolerance, topological codes, and decoders. These concepts are the building blocks for the research work. In chapter 4, we introduce some of the main algorithms for simulating quantum



---

circuits. Chapter 5 serves as an introduction to ion trap quantum computing and is the last background chapter. The research part begins with chapter 6, where we describe the work done on the feasibility study of QEC with near term ion trap systems, published in [1], in which we analyze the effects of crosstalk noise on the performance of a QEC protocol, realized on a near term ion trap device with realistic error modeling. Chapter 7 describes the work realized in the development and analysis of an efficient decoding algorithm for color codes based on the pre-existing union-find decoder [193]. In chapter 8 we describe the work published in [2], where we present a decoder for the square-octagon color code lattice based on the rescaling-based approach presented in previous works [187–189]. The thesis ends in chapter 9 with the final conclusion.

## 1. INTRODUCTION

---

**Part I**

**Background**



## Chapter 2

# Quantum computing

## 2. QUANTUM COMPUTING

---

This chapter introduces the basic concepts of quantum information and quantum computation that are used in the thesis. These concepts are a fundamental part of the background and form the basis over which the rest of the thesis is built. We begin by introducing qubits, the most basic unit of quantum information. Then, in Sec. 2.2 we introduce fundamental operations used to control states in quantum computation: gates and measurements. We then examine the concepts of a universal gate set and the restricted set of operations that can be simulated efficiently in a classical computer. Finally, in Sec. 2.3 we study how to model the effects of noise on qubit states, as well as some of the main error models used through the thesis.

We used [100] as the primary source material during the writing of this chapter.

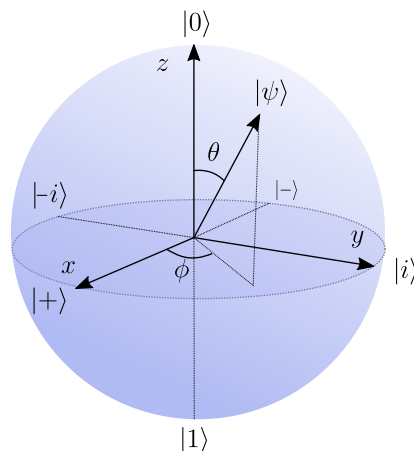
### 2.1 Qubit systems

Qubits comprise the elementary unit of quantum information and are the quantum counterpart of classical bits. In essence, qubits are two-level quantum systems. As such, the state of a qubit belongs to a two-dimensional Hilbert space. Using the Dirac notation, the two basis states of a qubit are typically represented as  $|0\rangle$  and  $|1\rangle$ , by analogy to the two states of a bit: 0 and 1. This orthonormal basis is typically referred to as the computational basis. However, the state of a qubit can be an arbitrary superposition  $|\psi\rangle$  of both states, which in general can be represented as

$$|\psi\rangle = e^{i\alpha}(\cos(\theta/2)|0\rangle + \sin(\theta/2)e^{i\phi}|1\rangle), \quad (2.1)$$

with  $\theta \in [0, \pi]$  and  $\phi \in [0, 2\pi]$ . The global phase  $e^{i\alpha}$  is usually neglected, as it is undetectable. Using these two parameters as angles, we can represent states as points on the surface of a unit sphere. This representation, called the Bloch sphere, is a useful and intuitive system to visually represent the state of a qubit and understand the effects of different operations. The two states of the computational basis,  $|0\rangle$  and  $|1\rangle$ , lie at the two ends of the  $z$  axis and correspond to the two eigenstates of the Pauli  $Z$  matrix.

An alternative formulation to represent the state of a quantum system is possible using the density matrix. This notation allows the description of a quantum system whose state is not completely known, but we know that it can be in a number of states  $|\psi_i\rangle$  with respective probabilities  $p_i$ . For such a system, the density matrix is defined



**Figure 2.1: Bloch sphere representation.** Pure states of a qubit are represented as points on the surface of a unit sphere, parametrized by the angles  $\theta$  and  $\phi$ . The states of the computational basis,  $|0\rangle$  and  $|1\rangle$ , lie at the two ends of the  $z$  axis, and are the eigenstates of the Pauli  $Z$  operator. Similarly, the eigenstates of the other Pauli matrices lie at the ends of their respective axes. Mixed states, on the other hand, are located inside the sphere, with the fully mixed state located at the origin.

by

$$\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i|. \quad (2.2)$$

The notation in terms of the density matrix is useful for QEC, as it allows the description of a system where an error might have happened. When a state in the density matrix formalism can be described by a single  $|\psi_0\rangle$  with probability  $p_0 = 1$ , it is said to be in a pure state, otherwise, it is called a mixed state. In the Bloch sphere (see Fig. 2.1), pure states of a qubit lie on the surface of the sphere, while mixed states are located inside the sphere. In particular, any single-qubit mixed state  $\rho$  can be written as

$$\rho = \frac{1}{2}(\mathbf{1} + \vec{r} \cdot \vec{\sigma}) = \frac{1}{2}(\mathbf{1} + xX + yY + zZ), \quad (2.3)$$

where  $\vec{r} = (x, y, z)$  is the Bloch vector, a real vector such that  $|\vec{r}| \leq 1$  that represents the state  $\rho$  in the Bloch sphere. States with  $|\vec{r}| = 1$  correspond to pure states. The vector  $\vec{\sigma} = (X, Y, Z)$  represents the three Pauli matrices. In general, the purity of a normalized state can be defined by the scalar  $\text{tr}(\rho^2)$ , which lies in the interval  $\frac{1}{d} \leq \text{tr}(\rho^2) \leq 1$ , where  $d$  is the dimension of the Hilbert space, and takes the value 1 for pure states. One example of a mixed state is the so-called maximally mixed state  $\rho_m = \frac{1}{2}\mathbf{1}$ , which lies at the center of the Bloch sphere.

## 2. QUANTUM COMPUTING

---

Pauli matrices  $\sigma_i$  are a primary tool when working with qubits. They are Hermitian operators, and they form an orthogonal basis for the vector space of 2x2 Hermitian matrices when combined with the identity matrix. On the computational basis, the Pauli matrices can be written as:

$$\sigma_1 = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (2.4)$$

Throughout this thesis, we use the notation  $\sigma_i$  and  $X, Y, Z$  indistinctly to refer to the Pauli operators. The operator  $\sigma_0$  corresponds to the identity operator  $\mathbb{1}$ . The eigenstates of the Pauli  $X, Y$  and  $Z$  matrices lie at the ends of the  $x, y$  and  $z$  axis in the Bloch sphere, respectively (see Fig. 2.1), and they have eigenvalues  $\pm 1$ . We use the following notation:

$$\begin{aligned} X|+\rangle &= |+\rangle, & X|-\rangle &= -|-\rangle, \\ Y|i\rangle &= |i\rangle, & Y|-i\rangle &= -|-i\rangle, \\ Z|0\rangle &= |0\rangle, & Z|1\rangle &= -|1\rangle. \end{aligned}$$

The Hilbert space of an  $n$ -qubit pure state can be obtained by the tensor product of the  $n$  single qubit Hilbert spaces resulting in a Hilbert space of  $2^n$  dimensions. The basis of this Hilbert space can be defined with a combination of tensor products of the basis of each individual qubit. Typically, we write each of the elements in a single ket:

$$|x_1\rangle \otimes |x_2\rangle \otimes \dots \otimes |x_n\rangle = |x_1, x_2, \dots, x_n\rangle. \quad (2.5)$$

For instance, the state of a two-qubit system can be written as:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle. \quad (2.6)$$

One notable property of multi-qubit systems is the possibility to present quantum entanglement, which plays a fundamental role in some of the most interesting applications of quantum computation, like quantum teleportation, quantum error correction, quantum cryptography or fast quantum algorithms [100]. A quantum state is said to be entangled if it cannot be written as the product of the states of its constituents, i.e. if it is not a separable (or product) state. To be more precise, a state  $\rho_{ABC\dots}$  of



many elements  $A, B, C, \dots$  is said to be separable if it can be written in the form

$$\rho_{ABC\dots} = \sum_i p_i \rho_A^{(i)} \otimes \rho_B^{(i)} \otimes \rho_C^{(i)} \otimes \dots \quad (2.7)$$

where  $p_i$  is a probability distribution, and each  $\rho_S^{(i)}$  represents the state of element  $S$  [205]. The different aspects of entanglement are still an active field of research, including its characterization, manipulation, measurement and applications [100, 205–207]

## 2.2 Gates

Quantum gates and measurements are fundamental elements of quantum computation, as they allow us to realize operations and read the information stored in the qubits. In this section, we study the main gates used for quantum computation in this thesis and some important concepts, like the universal gate set, the Clifford group, or the Gottesman-Knill theorem.

Pauli matrices play a crucial role in understanding single-qubit gates. The operators obtained by the exponentiation of the Pauli matrices  $e^{i\frac{\theta}{2}\sigma_i}$  represent rotations around the different axes in the Bloch sphere. Any arbitrary single-qubit operation can be obtained by combining these rotations. Furthermore, it is possible to apply a single qubit rotation of arbitrary angle  $\theta$  and rotation axis  $\vec{n}$  as:

$$R_{\vec{n}}(\theta) = e^{-i\frac{\theta}{2}\vec{n}\cdot\vec{\sigma}} = \mathbb{1} \cos\left(\frac{\theta}{2}\right) - i \sin\left(\frac{\theta}{2}\right) \vec{n} \cdot \vec{\sigma}. \quad (2.8)$$

Note that the Pauli operators represent  $\pi$  rotations around their corresponding axis in the Bloch sphere (Fig. 2.1). Another interesting property of the Pauli operators is that the set of all Pauli operators and the identity, together with prefactors  $\pm 1$  and  $\pm i$ , form the Pauli group [100]:

$$\mathbf{P} = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}. \quad (2.9)$$

The general Pauli group on  $n$  qubits  $\mathbf{P}_n$  is defined to consist of all  $n$ -fold tensor products of Pauli matrices, including the multiplicative factors  $\pm 1, \pm i$ . However, using Pauli matrices alone would only allow us to reproduce unitaries applied to each of the individual qubits separately. Ideally, we want a quantum computer to be able to apply

## 2. QUANTUM COMPUTING

---

any arbitrary multi-qubit unitary, including operations that generate entanglement between the qubits (like the CNOT gate, which will be introduced below and which can generate entanglement between two qubits). One of the challenges of universal quantum computation is to find a reduced set of gates that allow a quantum computer to closely approximate any quantum operation using a finite sequence of gates from the set. Such a set is called a universal quantum gate set. The Solovay-Kitaev theorem [100, 208, 209] implies that an arbitrary single qubit gate may be approximated to an accuracy  $\epsilon$  using  $\mathcal{O}(\log^c(1/\epsilon))$  gates from this universal gate set, where  $c$  is a constant approximately equal to 2. Thus, to approximate a circuit containing  $m$  two qubit CNOT and single qubit unitaries to an accuracy  $\epsilon$  requires  $\mathcal{O}(m \log^c(m/\epsilon))$  gates from the discrete set, a polylogarithmic scaling that can be acceptable for most applications.

A common universal gate set is composed of the single-qubit Hadamard (H) and T gate, and the two-qubit CNOT gate. In matrix form, these gates have the following expression:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}, \quad CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (2.10)$$

Note that only one entangling gate is needed to obtain a universal gate set, in this case the CNOT gate, and it involves only two qubits. A universal gate set can also be obtained by exchanging the CNOT gate with the Mølmer-Sørensen gate [210]:

$$XX_{ij}(\theta) = \exp\left(-i\frac{\theta}{2}X_iX_j\right) = \cos(\theta/2)\mathbf{1} - i\sin(\theta/2)X_iX_j, \quad (2.11)$$

where  $X_iX_j$  is a tensor product of Pauli  $X$ -type operators on qubits  $i$  and  $j$ , and  $\theta$  is the MS gate angle, which is typically set as  $\pi/2$  to define the fully entangling MS gate  $XX_{ij}(\pi/2)$ .

There is one more set of gates of particular relevance for this thesis: the Clifford group. The Clifford group is the group of unitary operators that map Pauli operators into Pauli operators (for any Clifford unitary  $U_C$ , the product  $U_C\sigma_iU_C^\dagger \in \mathbf{P}$ ). The Clifford group can be generated by three gates: Hadamard, CNOT and S gates. Although it is not enough to realize a universal gate set (this requires one gate outside of the Clifford group, e.g. the T gate), it is relevant for the study of quantum computation and

quantum information because of the Gottesman-Knill theorem [211, 212]. The main result from this theorem states that quantum circuits consisting only of Clifford gates and measurements in the computational basis can be perfectly simulated in polynomial time on a classical computer. This dramatically impacts the efficiency of quantum circuit simulations, particularly for quantum error correction circuits, as most of them utilize Clifford gates only. In chapter 4, we take a further look at this topic and study some algorithms that allow this efficient simulation of Clifford circuits.

## 2.3 Noise modelling

Harnessing the effects of noise on quantum computers is the main challenge along the path towards the implementation of quantum information processing devices. Imperfection on the implementation of the gates, fluctuations of electromagnetic fields, or coupling with the environment can corrupt the information stored in the qubits and reduce the fidelity of the gates applied. Thus, finding ways to control the effects of noise becomes a necessity for the development of quantum computation. The first step in this battle against noise is to understand our contender. In this section, we study some of the main noise models and the basic theory to analyze the effects of noise on quantum information.

To describe the dynamics of noise, we utilize the mathematical formalism of quantum operations, which is useful to describe both closed systems and systems coupled to their environment, thus providing a powerful tool that addresses a wide range of physical scenarios. This formalism describes the evolution of a system in state  $\rho$  to the final state  $\rho'$  as

$$\rho' = \varepsilon(\rho), \tag{2.12}$$

where both the initial and final states are described by density matrices, and the map  $\varepsilon$  is a *quantum operation*. This formalism can take into account the effects from both unitary transformations (i.e.  $\varepsilon_U(\rho) = U\rho U^\dagger$ ), measurements and stochastic processes [100]. An advantage of this formalism is that it allows a discrete-time analysis that does not require a complete understanding of the continuous-time description of the noise (described by master equations, Langevin equations, or stochastic differential equations), but rather the overall final effect on the state.

## 2. QUANTUM COMPUTING

---

Quantum operations can take into account the interactions of the system with an environment, which can be described by a unitary evolution  $U_{SE}$  (not necessarily known). This environment can be assumed to start in a pure state  $|e_0\rangle$  with no initial correlations between system and environment. By tracing over the environment, we can obtain a description of the evolution of the state that only concerns the effects on our system:

$$\varepsilon(\rho) = \sum_i \langle e_i | U_{SE}(\rho \otimes |e_0\rangle \langle e_0|) U_{SE}^\dagger |e_i\rangle \quad (2.13)$$

$$= \sum_i E_i \rho E_i^\dagger. \quad (2.14)$$

This description of the quantum operation is called a Kraus map, and the operators  $E_i = \langle e_i | U_{SE} |e_0\rangle$  are called the Kraus operators. These operators satisfy an important constraint that ensures that the normalization of the density matrix is preserved after the evolution, the completeness relation:

$$1 = \text{Tr}(\varepsilon(\rho)) = \text{Tr}\left(\sum_i E_i \rho E_i^\dagger\right). \quad (2.15)$$

Since this relationship is true for all  $\rho$ , it follows:

$$\sum_i E_i^\dagger E_i = \mathbb{1}, \quad (2.16)$$

an equation satisfied by quantum operations that are trace-preserving.

Throughout this thesis, we use the Kraus map formalism to describe the effects of different error channels. Additionally, we consider error channels that are completely uncorrelated both in time and space, which allows us to simplify the description of the different noise models.

We can illustrate this formalism with some of the simplest noise models considered in QEC. The first error model is a direct analogy from classical information: bit-flip noise. This error channel simulates the possibility of a qubit to suffer a transformation  $|0\rangle \iff |1\rangle$  with probability  $p$ . This transformation corresponds to the Pauli  $X$  operator and is typically referred to as an  $X$  error. The quantum channel for bit-flip noise can

be written in terms of Kraus operators as:

$$E_0 = \sqrt{1-p}\mathbb{1} \tag{2.17}$$

$$E_1 = \sqrt{p}X, \tag{2.18}$$

which leads to the following evolution:

$$\varepsilon(\rho) = (1-p)\rho + pX\rho X. \tag{2.19}$$

The following error model considers the possibility of a change in the phase, e.g.  $|+\rangle \iff |-\rangle$ . This error channel is called the dephasing channel, or the phase-flip channel, and can be written as

$$\varepsilon(\rho) = (1-p)\rho + pZ\rho Z. \tag{2.20}$$

These two channels are commonly studied in QEC, and are particularly useful for the study of some QEC codes that correct  $X$  and  $Z$  errors separately. The direct generalization of the Pauli channels is the depolarizing error channel, which simulates how a qubit can suffer one of the three Pauli errors  $\sigma \in \{X, Y, Z\}$  with probability  $p$ :

$$\varepsilon(\rho) = (1-p)\rho + \frac{p}{3}(X\rho X + Y\rho Y + Z\rho Z). \tag{2.21}$$

This decomposition is not unique. The depolarizing channel is often written as a system having a probability  $p'$  of being replaced by a completely mixed state  $\mathbb{1}/2$ . In this case, the channel is written as:

$$\varepsilon(\rho) = (1-p')\rho + p'\frac{1}{2}\mathbb{1}, \tag{2.22}$$

with  $p' = \frac{4}{3}p$ . The depolarizing channel is a simple model that can reproduce the loss of the information in both the phase and the populations of the  $|0\rangle$  and  $|1\rangle$  states, effectively converting a qubit into a completely mixed state  $\mathbb{1}/2$ . This property of the depolarizing channel makes it a common choice to describe error sources in the study of QEC protocols, where it is also often referred to as symmetric depolarizing channel. It only consists of Clifford operations, making it convenient to implement efficient, classical computer simulations.

## 2. QUANTUM COMPUTING

---

Another interesting noise channel is amplitude damping, which reproduces the spontaneous decay of a two-level system from the excited state to the ground state. If we take the state  $|1\rangle$  as the excited state, the amplitude damping channel can be described by the following Kraus operators:

$$E_0 = |0\rangle\langle 0| + \sqrt{1-p}|1\rangle\langle 1| \quad (2.23)$$

$$E_1 = \sqrt{p}|0\rangle\langle 1|. \quad (2.24)$$

Amplitude damping is an example of a non-Clifford error channel, which makes its implementation in classical simulations more challenging.

The noise models studied in this chapter present a simple yet widely used framework to study the influence of noise on quantum computation. When studying particular implementations of quantum computation, the noise models can be adapted to reproduce the particular characteristics of a physical device. A better understanding of the noise processes can allow the use of different-error correcting techniques addressed to the noise sources that affect the device. Additionally, we can obtain more precise predictions of the performance of different protocols by using more accurate error models tailored to reproduce the behavior of the experiment. We study this type of problem in chapter 6, where we study a microscopically derived error model for ion traps and the feasibility of near term QEC implementations.

## Chapter 3

# Quantum error correction

### 3. QUANTUM ERROR CORRECTION

---

Preserving entanglement and quantum information from noise is one of the main challenges in the development of quantum computers. The main building block of QEC are QEC codes, which encode the information of a logical qubit in a set of physical qubits, so that errors on individual elements of the code can be corrected, preserving the encoded information. While similar codes have been developed for classical computers (e.g. repetition and parity-check codes), the nature of quantum mechanics presents some limitations that need to be overcome. The destructive nature of measurements, the continuous nature of errors, and the impossibility of copying states due to the no-cloning theorem are some of the key obstacles for QEC. Therefore, classical strategies for error correction, like the redundant storing of multiple copies of the information, are not directly applicable to the quantum problem.

The main idea behind QEC codes is to encode quantum information in a multi-qubit entangled state, effectively spreading the information over the different qubits in the code. Errors can be detected by realizing parity checks on multiple subsets of qubits. The information from these measurements, usually called the *syndrome*, can be used to find a correction for the errors in the code. The crucial property of these codes is that the syndrome does not give any information about the actual information encoded, but only about the location of the errors.

Following these basic principles, multiple QEC codes have been designed, with the corresponding quantum circuits to encode the information and the syndrome measurement [109–111, 113, 115, 208, 213, 214]. The information from the parity checks is transferred to additional *ancilla* qubits using entangling gates, where it can be measured without destroying the information stored in the *data* qubits (e.g. [215]). However, a naive implementation of these protocols can be problematic, as individual errors can propagate through the elements of the circuits and corrupt the information of multiple data qubits, ultimately leading to an uncorrectable error. For this reason, the design of *fault tolerant* protocols preventing uncontrolled propagation of errors through a circuit has become an essential requirement for near term implementations of QEC [128].

Small QEC codes can reduce the effective probability of an error corrupting the logical information encoded. However, to access the most powerful capabilities of quantum computation, this error probability needs to be diminished to negligible levels, so that increasingly long computations can be reliably executed. Scalable QEC codes are thus the next step on the road of QEC. Thanks to the threshold theorem, the probability of



---

an uncorrectable error can be reduced to arbitrarily low levels by increasing the size of a code, provided that the *physical error rate* (i.e. the rate at which errors occur on the physical qubits) on each qubit falls below a critical threshold and provided that fault-tolerant quantum circuit constructions are employed [100, 127–130, 216]. Topological QEC codes stand as the leading candidates for the scalability of QEC codes, with high error thresholds and the possibility of applying some logical gates in a transversal way. Additionally, the syndrome measurement requires only local operations between neighboring qubits, which simplifies the experimental implementation of these codes. Surface codes and color codes are currently the most prominent topological QEC codes, and some of their building blocks have already been implemented in state-of-the-art quantum devices with promising results (e.g. [89, 93, 147–157, 159–162, 164, 165]), with prominent results like fault tolerant state preparation and stabilizer readout [147, 170], repeated fault tolerant QEC cycles [171] or fault tolerant magic state preparation and injection [169].

There is, however, one more challenge in the scaling of QEC codes. The threshold of each code depends on our capability of correcting the errors appearing in the code. The theoretical optimal threshold can be studied using models from statistical mechanics, e.g. [217–220]. This threshold corresponds to the error rate at which it is no longer possible to distinguish if an error is equivalent to a logical operator or a trivial operation, in the thermodynamic limit. Finding the error class of a given error configuration, however, is an NP-hard problem [183, 184]. Therefore, we would need access to an arbitrarily extensive computational power. For any practical implementation of QEC, we need an efficient algorithm, able to interpret the syndrome and find a correction in polynomial time. These algorithms are called decoders, as they translate the information from the syndrome into the correction that needs to be applied. The challenge of finding computationally efficient decoders with high thresholds is one of the main problems addressed in this thesis, as we study two different decoders for color codes in the chapters 7 and 8.

The main result of the scaling of QEC codes is that the logical error rate can be reduced exponentially by increasing the size of the lattice. The condition to achieve this beneficial regime is that the physical error rate needs to fall under the threshold

### 3. QUANTUM ERROR CORRECTION

---

of the code<sup>1</sup>. As discussed, by finding better decoders we can potentially increase the practical threshold, getting closer to the limit defined by the optimal threshold. Thus, if the physical error rate of a quantum device is higher than the threshold, we need to find some alternative methods to reduce the effective error rate. The most obvious approach is to improve the experimental device, by using better components, controlling the sources of noise and improving the fidelity of the gates. However, by studying in detail the sources of noise, it is possible to develop error suppression techniques that take advantage of the particular properties of the errors to reduce further its effects on out encoded quantum state. Decoherence Free Subspaces (DFS), spin-echo pulses or refocusing techniques are some examples of these error suppression techniques. More information about the topic can be found in [101–106]. In chapter 6, we show that the use of error suppression techniques (refocusing in particular) is crucial to achieving the beneficial regime for QEC in near term ion trap devices, as it can reduce by several orders of magnitude the effects of some hazardous sources of noise.

In this chapter, we explain the main ideas, concepts and techniques of Quantum Error Correction (QEC) used and required for the research chapters. First, we introduce stabilizer codes, quantum circuits for stabilizer measurement and the basics of FT. Then, in Sec. 3.2 we discuss topological codes, with particular focus on color codes, the central family of QECC's used throughout the thesis. In Sec. 3.3, we study the problem of decoding and explore some of the leading decoding algorithms. Finally, we discuss a FT implementation of the minimal instance of a color code, the 7-qubit Steane code.

We used [100, 109, 115] as the primary source material during the writing of this chapter.

#### 3.1 Stabilizer codes

Stabilizer codes are a family of QEC codes that can encode one or more qubits into an entangled state of multiple physical qubits. By measuring certain operators, the stabilizers, it is possible to detect, identify and correct errors [211]. Stabilizer codes form a big family of codes that contains the most commonly used QEC codes.

---

<sup>1</sup>For this, one first needs to prove the existence of a threshold. Note that there are some useful codes, such as Bacon-Shor codes, which do not have a threshold [221].

The central idea of stabilizer codes is the use of the stabilizer operators to divide the Hilbert space of the system into orthogonal subspaces. Then, the system is prepared into one of these subspaces, called the code space, in which the logical information can be encoded. Typically, the code space is defined as the space for which all stabilizers have a +1 eigenvalue, for which we also need all stabilizers to commute between each other. Errors happening in the code can bring the state of the system outside of this code space. By measuring the stabilizer operators, we can obtain information about these errors, without obtaining any information about the encoded logical information. Furthermore, we protect the information from continuous errors, as the measurement of the stabilizers can collapse the state either back to the code space, or to a detectable error outside of the code space.

#### 3.1.1 Basic concepts

Let us begin by defining *stabilizer operators*. An operator  $S$  is said to stabilize state  $|\psi\rangle$  if it satisfies the identity  $S|\psi\rangle = |\psi\rangle$ , i.e.  $|\psi\rangle$  is a +1 eigenstate of  $S$ . An interesting property of stabilizer operators (stabilizers for short) is the fact that we can define subspaces of states stabilized by a stabilizer within a given Hilbert space. For instance, if we require a 2-qubit system to be stabilized by the  $Z_1Z_2$  operator (i.e. the Pauli  $Z$  operator applied to the qubits 1 and 2), this leaves only two orthogonal basis states, effectively reducing the dimension of the Hilbert space by half:

$$|0_L\rangle = |00\rangle, \quad |1_L\rangle = |11\rangle.$$

Thus, we can use stabilizers to reduce the size of the Hilbert space for a multi-qubit system down to a single effective (or *logical*) qubit. We call this simultaneous +1 eigenspace of all stabilizers the computational subspace or code space. A stabilizer code is defined by the computational subspace of a set of stabilizer operators in a multi-qubit system. As we will see in the following examples, stabilizer codes can detect errors by measuring the stabilizer operators, which will change their measured value if an error brings the state outside of the computational subspace [100, 115]. Finally, to define the logical operators that operate the qubits within the code space, the logical operators need to commute with all the stabilizers in the code.

### 3. QUANTUM ERROR CORRECTION

---

We define a parity check as the measurement of a stabilizer operator, since the stabilizers used in QEC can take two values:  $+1$  (even) or  $-1$  (odd). The measured value of a stabilizer is often referred to as its parity, and a  $-1$  result in this measurement is typically called an excitation.

The simplest example of stabilizer codes is the bit-flip repetition code, a 3-qubit code that allows the correction of a single bit-flip error. The stabilizers for the bit-flip repetition code are  $S_1 = Z_1Z_2$  and  $S_2 = Z_2Z_3$ . In the computational basis, the code space is then defined by the orthogonal basis  $\{|000\rangle, |111\rangle\}$ . These two states can encode a single logical qubit, with the logical zero as  $|0\rangle_L = |000\rangle$  and the logical one as  $|1\rangle_L = |111\rangle$ . Any superposition of these states is a valid logical state, as it remains within the code space, and measuring the stabilizers would not give us any information about the superposition. This property is crucial, as it allows us to obtain information about the errors without obtaining any information about the logical state of the encoded qubit. Logical Pauli  $X_L$ ,  $Y_L$  and  $Z_L$  can be applied by applying single-qubit Pauli operations on each of the individual qubits, e.g.  $X_L = X_1X_2X_3$ . Note that these logical operators commute with all stabilizers, and follow the expected commutation relation for the Pauli operators on a single qubit, i.e.  $[X_L, Y_L] = 2iZ_L$ .

A single bit-flip error  $X_i$  on any of the three qubits would take the state out of the code space. For example, a bit-flip on the first qubit could take an encoded state to the subspace for which  $S_1 = -1$  and  $S_2 = +1$ , with the new basis of the subspace being  $\{|100\rangle, |011\rangle\}$ . By measuring the stabilizers, we are able to detect this fault and correct the error. In the context of QEC, the result of measuring the stabilizers is called the *syndrome*. Since each possible single-qubit bit-flip error generates a different syndrome, we can make a table with all the possible outcomes of the stabilizer measurements (see Table 3.1). This table allows us to find the most probable error for the measured syndrome, assuming uncorrelated bit-flips happen with a small error rate  $p$ , and is called a lookup table (LUT). LUT's are the first example of a decoding algorithm, i.e. a method to obtain a recovery operation using the information from the syndrome.

This table will identify the most probable correction for each syndrome. However, any bit-flip error of weight 2 (affecting two different qubits) would lead to an incorrect interpretation of the syndrome, and the combination of the error and the correction would lead to the application of a logical  $X_L$  error, corrupting the encoded information.

Error	$S_1$	$S_2$
–	+1	+1
$X_1$	–1	+1
$X_2$	–1	–1
$X_3$	+1	–1

**Table 3.1: Lookup table for the 3-qubit repetition code.** This table relates the information from the syndrome (produced by measuring the stabilizers  $S_1 = Z_1Z_2$  and  $S_2 = Z_2Z_3$ ) to the most probable bit-flip error.

Another key limitation of the bit-flip repetition code is that it leaves the code unprotected from dephasing errors (Pauli  $Z$ ), since any single-qubit phase-flip error  $Z_i$  commutes with the stabilizers, hence being undetectable by the stabilizer measurement. Crucially, any  $Z_i$  error is equivalent to a logical  $Z_L$  error up to a combination of stabilizers, e.g.  $Z_1$ , since we can always write a state  $|\psi\rangle$  in the code space preceded by a product of stabilizers:

$$Z_1|\psi\rangle = Z_1S_2|\psi\rangle = Z_1Z_2Z_3|\psi\rangle = Z_L|\psi\rangle. \quad (3.1)$$

This exemplifies an interesting property of logical operators: we can rewrite them by multiplying them with any product of stabilizer operators. We can define the phase-flip repetition code by changing the stabilizers to  $S_1 = X_1X_2$  and  $S_2 = X_2X_3$ , which allows the code to correct one phase-flip ( $Z$  error) but leaves the code vulnerable to bit-flip errors. Despite its limitations, there is an intense research on the implementation of repetition codes, e.g. [150, 159, 174].

In order to protect a logical qubit from any Pauli error, the minimum code required is the 5-qubit code, defined by the following set of stabilizers:

$$S_1 = XZZXI$$

$$S_2 = IXZZX$$

$$S_3 = XIXZZ$$

$$S_4 = ZXIXZ$$

A lookup table (similar to Table 3.1) can be defined for the 5 qubit code, containing the correction to be applied for each of the  $2^4$  possible syndromes. Each of the 15

### 3. QUANTUM ERROR CORRECTION

---

non-trivial syndromes in the 5-qubit code corresponds to a single-qubit Pauli operation applied to one of the qubits in the code. Lookup tables are also the simplest type of decoders: algorithms that interpret the information from the syndrome to find the most probable correction that should be applied.

The repetition code and the 5 qubit code are two simple examples of stabilizer codes that illustrate some of the main features of this class of QEC codes. Typically, there are three variables of interest that can be used to indicate some of the characteristics of different QEC codes: the number of physical qubits  $n$ , the number of encoded logical qubits  $k$ , and the code distance  $d$  of the code.

To define the code distance  $d$  we first need to define the weight of an operator. The weight of a Pauli operator  $P = P_1 P_2 \dots P_n \in \mathbf{P}_n$  is the number of qubits on which it acts non-trivially, i.e. the number of terms in the tensor product which are not equal to the identity (e.g. the operator  $\mathbb{1}_1 X_2 Y_3 Z_4$  has weight 3). The distance  $d$  of a code is defined as the minimum weight of any logical operator (there are several logical operators, i.e.  $X_L, Z_L$  etc. ), or, in other words the smallest support size of the logical operators in the code.

A code of distance  $d$  ensures that all errors of weight  $\lfloor \frac{d-1}{2} \rfloor$  or smaller are correctable<sup>1</sup>. The value of these variables for a given code is typically written as  $[[n, k, d]]$ . For the 5-qubit code, we have  $[[5, 1, 3]]$ .

An important subclass of stabilizer codes we want to mention are CSS codes, named after the authors Robert Calderbank, Peter Shor and Andrew Steane, whose stabilizer generators consist of the tensor products of either  $X$  or  $Z$  operators [100, 213, 214]. In an over-simplified description, to construct a CSS code we begin with two classical parity check codes  $C_1$  and  $C_2$ . Then, we replace the parity checks in  $C_1$  with  $X$ -type stabilizers, and for  $C_2$  with  $Z$ -type stabilizers. From those stabilizers, a subset is chosen to generate a code, ensuring that all stabilizers commute with each other. A formal description of CSS codes can be found in [100]. CSS codes are a relevant class of codes for this thesis, as the codes we study (color codes) belong to this family. They have some interesting properties, e.g. bit and phase flip errors can be corrected separately.

With this introduction, we have seen the basic concepts of the theory of stabilizer codes. A more in-depth study of the properties of different QEC codes can be found

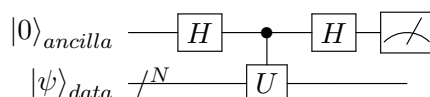
---

<sup>1</sup>The symbols  $\lfloor \cdot \rfloor$  represent the floor function.

in [100]. The next step in our introduction to stabilizer codes is the implementation of the stabilizer measurements as quantum circuits.

### 3.1.2 Circuits for stabilizer measurement

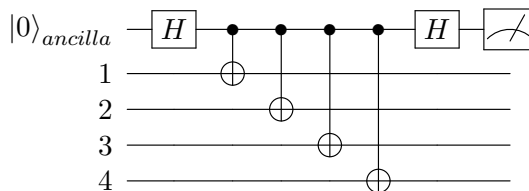
As we have seen in the previous section, stabilizer codes rely on measuring the stabilizer operators  $S_i$ . The result of these measurements, the syndrome, provides enough information to correct the errors in the code (assuming the total weight of the errors is smaller than the amount tolerated by the code). In general, we can measure any arbitrary operator  $U$  (assuming it is unitary, hermitian, and has eigenvalues  $\pm 1$ ) with the help of an additional qubit, called ancilla qubit, with the following quantum circuit [100]:



where the measurement is, unless stated otherwise, in the computational basis (or  $Z$  basis); and the ancilla is always prepared in the  $|0\rangle$  state. For the QEC codes used in this thesis, the stabilizer measurements that we will require consist of products of either Pauli  $Z$  operators or Pauli  $X$  operators. The circuits for these stabilizer measurements can be compiled using only Hadamard and CNOT gates [100]. This can be more easily understood if we write the unitary of the CNOT gate as

$$CNOT = |0\rangle_1 \langle 0|_1 \mathbb{1}_2 + |1\rangle_1 \langle 1|_1 X_2, \quad (3.2)$$

where qubit 1 is the control qubit, and qubit 2 is the target qubit. In this description, it is easy to see that we can measure a stabilizer  $S_X = X_1 X_2 X_3 X_4$  using an ancilla qubit  $a$  and the circuit



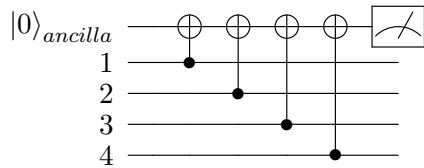
as the product of CNOT gates leads to the control- $S_X$  corresponding to the stabilizer:

### 3. QUANTUM ERROR CORRECTION

---

$$C_a S_X = |0\rangle_a \langle 0|_a \mathbb{1}_1 \mathbb{1}_2 \mathbb{1}_3 \mathbb{1}_4 + |1\rangle_a \langle 1|_a X_1 X_2 X_3 X_4. \quad (3.3)$$

Similarly, the stabilizer  $S_Z = Z_1 Z_2 Z_3 Z_4$  can be measured with the help of an ancilla qubit. The circuit corresponding to this measurement can be simplified to:



In both circuits, the result of the ancilla measurement corresponds to the outcome of the measurement of the stabilizers. These circuits present a simple way of implementing stabilizer readout protocols. Nevertheless, they are susceptible to dangerous error propagation events, as we will see in the next section.

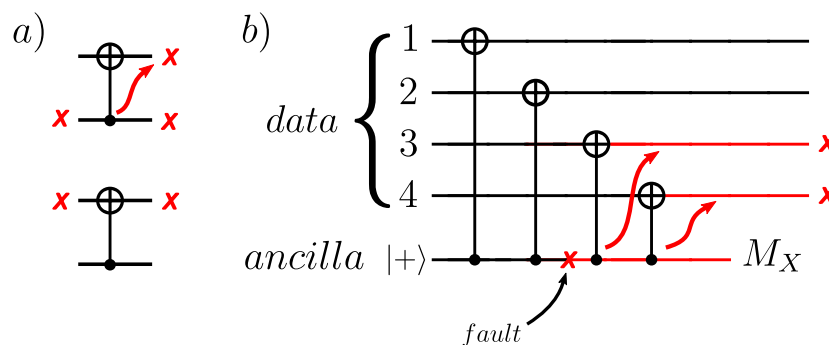
#### 3.1.3 Fault tolerance

Fault tolerance (FT) is a design methodology that allows QEC to remain effective in the presence of faults. The concept was first developed in classical computing [222], and its basic principle is to avoid errors cascading through the circuits used for gate operations and QEC. In a FT circuit element, a single error can cause **at most** one error in the output for each logical qubit block [115, 128]. With protocols able to correct several errors, this definition can be relaxed, so that a distance  $d$  code correcting  $t = \lfloor (d-1)/2 \rfloor$  errors requires that  $\leq t$  errors during an operation do not result in  $> t$  errors in the output to be considered FT [115].

In our description of stabilizer codes so far, we have implicitly assumed that errors happened before the stabilizer readout. However, in realistic implementations of QEC protocols, errors can occur at any moment during the realization of a circuit. Two-qubit entangling gates (like the CNOT gate) can propagate errors through the circuit (see Fig. 3.1). Potentially, this error propagation can turn a lower weight correctable error into a higher weight uncorrectable error, ultimately leading to a failure of the QEC protocol and the corruption of the encoded information. Uncontrolled error propagation through the circuits has a particularly damaging effect on small codes, since the low distance of the code means that a single fault propagating through a circuit can easily lead to a weight-two error, which could be uncorrectable for distance  $d = 3$  codes.



In Fig. 3.1 we show an example of how a single error occurring on the ancilla qubit can propagate to the data qubits leading to a weight-2 error. This vulnerability to error propagation events can have a very damaging effect on the performance of QEC codes.



**Figure 3.1: Error propagation through the syndrome measurement circuit.** **a)** bit-flip error propagation through a CNOT gate: an X error in the control qubit propagates to the target qubit, but it does not propagate from the target qubit to the control qubit. **b)** Standard stabilizer measurement of a stabilizer  $S = X_1X_2X_3X_4$ , where a single ancilla is used to measure the stabilizer. A single fault on the ancilla can propagate through the CNOTs, resulting in two X faults on the data qubits. This type of propagation is usually called a hook error.

To prevent the damaging effects of error propagation, there is a need to find fault tolerant procedures to operate on logical states. Stabilizer measurements and any kind of operation on the logical information should be realized fault tolerantly. The most straightforward way to apply operations fault-tolerantly to encoded qubits is the use of transversal operations. A logical gate is said to be transversal if it can be implemented as the parallel application of gates on a subset of qubits in a block of a quantum error-correcting code. Qubits from one block can only interact with corresponding qubits from another block or with an ancilla. Any transversal operation is automatically fault-tolerant. An example of transversal gates is the  $X_L$  (and  $Z_L$ ) operator in color codes and surface codes, which can be implemented by the parallel application of single qubit Pauli  $X$  ( $Z$ ) gates on the qubits belonging to the support of the logical operator (see Figs. 3.4, 3.7).

However, the possibility of implementing logical gates transversally on encoded qubits is limited on the upper end by the no-go theorem by Eastin and Knill [223], which states that if a quantum code is non-trivial (i.e. it can detect at least any error on a single qubit), then it does not have a transversally-realizable universal set of gates.

### 3. QUANTUM ERROR CORRECTION

---

One common method to implement fault-tolerantly the logical gates that cannot be implemented transversally is the use of magic states [224–227].

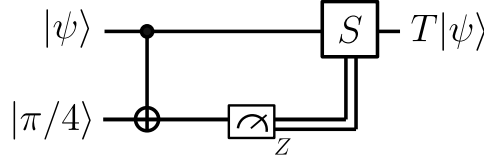
Magic states are special states that can be used as a resource for the implementation of logical gates fault-tolerantly. Magic states are defined as the pure states that can be obtained by applying any one-qubit Clifford operation to the states  $|M\rangle$  and  $|H\rangle$ :

$$|M\rangle\langle M| = \frac{1}{2} \left[ \mathbb{1} + \frac{1}{\sqrt{3}}(\sigma^x + \sigma^y + \sigma^z) \right], \quad (3.4)$$

$$|H\rangle\langle H| = \frac{1}{2} \left[ \mathbb{1} + \frac{1}{\sqrt{2}}(\sigma^x + \sigma^z) \right]. \quad (3.5)$$

These states are particularly useful for the application of non-Clifford gates, like the T gate, through gate teleportation [224]. The circuit in Fig. 3.2 shows an example of gate teleportation using an ancilla qubit prepared in the magic state  $|\pi/4\rangle$ :

$$|\pi/4\rangle_L = \frac{1}{\sqrt{2}}(|0\rangle_L + e^{+i\pi/4}|1\rangle_L). \quad (3.6)$$



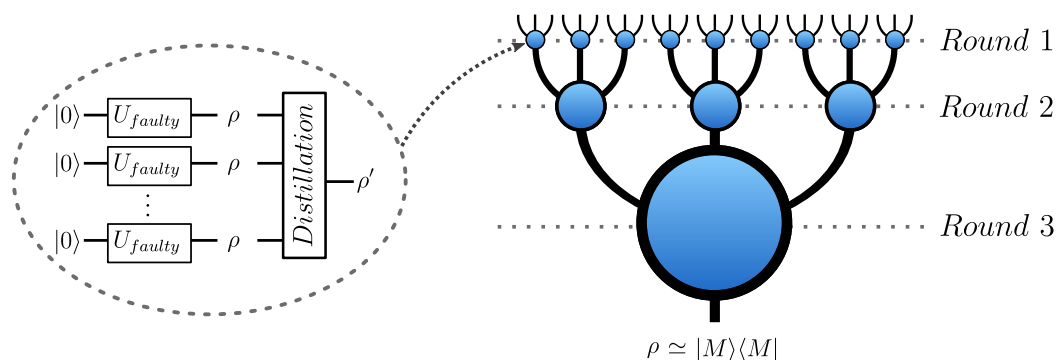
**Figure 3.2: Gate teleportation:** circuit for the gate teleportation of a T gate to a target qubit in an unknown state  $|\psi\rangle$  using an ancilla in the magic state  $|\pi/4\rangle$ .

$$|\psi\rangle_L |\pi/4\rangle_a = (\alpha |0\rangle_L + \beta |1\rangle_L) \otimes \frac{1}{\sqrt{2}}(|0\rangle_a + e^{+i\pi/4}|1\rangle_a). \quad (3.7)$$

By applying a *CNOT* gate, we create the following entangled state between the target qubit, previously in an unknown state  $|\psi\rangle_L = \alpha |0\rangle_L + e^{i\pi/4}\beta |1\rangle_L$ , and the ancilla qubit:

$$\begin{aligned} CNOT |\psi\rangle_L |\pi/4\rangle_a &= \frac{1}{\sqrt{2}} \left[ (\alpha |0\rangle_L + e^{i\pi/4}\beta |1\rangle_L) \otimes |0\rangle_a + (e^{i\pi/4}\alpha |0\rangle_L + \beta |1\rangle_L) \otimes |1\rangle_a \right] \\ &= \frac{1}{\sqrt{2}} (\alpha |0\rangle_L + e^{i\pi/4}\beta |1\rangle_L) \otimes |0\rangle_a + e^{i\pi/4} \frac{1}{\sqrt{2}} (\alpha |0\rangle_L + e^{-i\pi/4}\beta |1\rangle_L) \otimes |1\rangle_a \\ &= T_L |\psi\rangle_L \otimes 0_a + e^{i\pi/4} S_L^\dagger T_L |\psi\rangle_L \otimes |1\rangle_a. \end{aligned} \quad (3.8)$$

Then, by measuring the ancilla qubit in the Pauli  $Z$  basis, we collapse the state on the target qubit to either  $T_L |\psi\rangle_L$  in the case of a  $+1$  result, or to the state  $S_L^\dagger T_L |\psi\rangle_L$ , in which case we can apply an  $S$  gate to obtain the desired state  $T |\psi\rangle$ . This scheme can be applied fault tolerantly to a target encoded qubit if the QEC code allows the transversal application of the  $CNOT$  and  $S$  gates, provided that we have an encoded qubit in the magic state  $|\pi/4\rangle_L$  available. Using this method, we would need a source of magic states available to perform  $T$  gates fault tolerantly through a long quantum computation. Preparing encoded magic states with high fidelity is, however, a complicated task, as they cannot be prepared fault-tolerantly. To improve the fidelity of the produced magic states, we can use a protocol called magic state distillation [224], which takes many noisy magic states and distills a smaller set of less noisy magic states (see Fig. 3.3). To achieve the desired fidelity for a given quantum computation, we may require many rounds of distillation [13]. This increase in the qubit overhead is estimated to be one of the main bottlenecks for fault-tolerant implementations, which has made the study of magic state distillation protocols an active field of research [224–228].



**Figure 3.3: Magic state distillation.** Using non fault tolerant circuits  $U_{faulty}$ , we prepare multiple noisy copies  $\rho$  of the desired magic state  $|M\rangle$ . Then, a distillation circuit that uses only Clifford gates and Pauli measurements is used to obtain a magic state of a higher fidelity  $\rho'$ . This process of distillation can be repeated over many rounds to improve the fidelity of the output magic state.

The culmination of FT strategies is the idea of fault tolerant quantum computation, in which all operations are computed directly on encoded quantum states, without ever requiring the decoding of the state. Each qubit in the original quantum circuit would be replaced by an encoded block of qubits using QEC codes and FT circuit implementations. Operations between qubits would be realized via fault tolerant protocols,

### 3. QUANTUM ERROR CORRECTION

---

preventing error propagation between the encoded qubits involved in the operation. QEC would be realized periodically, to prevent the accumulation of errors [13, 109]. The idea of fault tolerant quantum computation underlies the entire construction of one critical result of QEC: the threshold theorem [100, 127, 129, 130]:

**Threshold theorem for quantum computation:** A quantum circuit containing  $p(n)$  gates may be simulated with probability of error at most  $\epsilon$  using  $\mathcal{O}(\text{poly}(\log p(n)/\epsilon)p(n))$  gates on hardware whose components fail with probability at most  $p$ , provided  $p$  is below some constant threshold,  $p < p_{th}$ , and given reasonable assumptions about the noise in the underlying hardware.

This result holds for a very general noise model, which includes, besides probabilistic errors, also decoherence, amplitude and phase damping, depolarization, and systematic inaccuracies in the gates. The main assumption on the noise model is locality (i.e. the noise process in different gates and qubits is independent), although the theorem can also be proven in the presence of exponentially decaying correlations both in time and space [129]. The key insight on this scaling is that the error rate can be suppressed exponentially with the size of the code. Therefore, provided the noise in individual gates is below a certain constant threshold and obeys physically reasonable assumptions, it is possible to reliably perform an arbitrarily long quantum computation, with only a small overhead in the size of the circuit necessary to ensure reliability [100, 127, 129, 130].

The development and implementation of all the protocols required for fault tolerant quantum computation is a formidable challenge that requires careful planification and examination of every step of the computation (see e.g. [155, 169, 225, 229]). Nevertheless, it stands as the central paradigm for realizing long quantum computations with high fidelity, using noisy quantum components. The work presented in this thesis aims to contribute to this ambitious project, moving one step forward towards fault tolerant quantum computation.

## 3.2 Topological codes

Topological QEC codes arrange qubits in lattices with different topologies, and encode the logical information in non-trivial cycles of the topology of the lattice [100, 109]. The logical operators are, thus, highly non-local, which makes them robust against local noise in the physical setup. Furthermore, syndrome extraction requires only local

operations between neighboring qubits, which simplifies the experimental implementation. The threshold theorem also applies to topological codes: the logical error rate can be suppressed exponentially with the code size, as long as the physical error rate is below a certain error threshold. Topological codes stand out as the QEC codes with some of the highest known thresholds [109, 131], which makes them attractive candidates for experimental realizations [147, 167–169, 171, 173, 230, 231].

The error threshold of a QEC code is defined as the error rate below which the logical error rate can be suppressed arbitrarily if the QEC protocol is scaled appropriately. The particular threshold on a QEC code depends on several factors. Firstly, it depends on the noise model, as the protocol has a different tolerance depending on the type of noise considered. Typically, the thresholds for topological codes are classified according to the following categories:

1. Code capacity: the measurement of the stabilizers is considered ideal and error-free, and the errors happen only before the measurement.
2. Phenomenological noise: we consider that the measurement of the stabilizers has a certain error probability, in addition to the errors before the QEC round. Typically, several rounds of stabilizer measurements are considered, since a single round of measurements cannot be trusted.
3. circuit-level noise: all elements of the circuits for the stabilizer measurement are considered faulty. The details on the error model and the protocol depend on the particular study, although depolarizing-type noise is typically considered.

Secondly, the threshold of a QEC code depends on our capacity to obtain a successful recovery operation from the information given by the syndrome measurement. Finding algorithms with a high success rate and which are efficient in terms of classical computational cost is the problem of decoding, which is discussed further in Sec. 3.3. The upper bound for the decoding problem corresponds to the optimal threshold, i.e. the threshold obtained if the decoding algorithm, given the information from the syndrome, applies the correction with the maximum probability of recovery. The optimal threshold can be computed for some models by mapping the problem to a classical statistical-mechanical model e.g. [217–220].

### 3. QUANTUM ERROR CORRECTION

---

The error threshold of a QEC code defines the error correction capabilities in the infinite-size limit. For finite-size lattices, however, it is often useful to define a different number: the pseudo-threshold. In this work, we define the pseudo-threshold of a code of distance  $d$  (or a QEC protocol) under a given noise to be the error rate  $p_c$  below which the logical error rate  $p_L$  is smaller than the physical error rate  $p$ . As opposed to the threshold, the pseudo-threshold of a QEC code depends on the size of the lattice, and defines the region of beneficial QEC, where the use of that QEC protocol provides an advantage ( $p_L < p$ ) over a non-encoded physical qubit.

Surface codes and color codes are the two leading approaches to topological QEC [109–113, 131, 132]. They share the main advantages of topological QEC, with similar error thresholds and local stabilizer measurements. In this thesis, we focus on the study of color codes, from the performance of the smallest instance of a color code (also called Steane code [113, 214]) in chapter 6, to the decoders needed for the effective scaling of the lattice in chapters 7 and 8. However, being the main alternative to color codes, it is important to understand the basic concepts of surface codes, and the advantages and disadvantages of each approach to topological QEC.

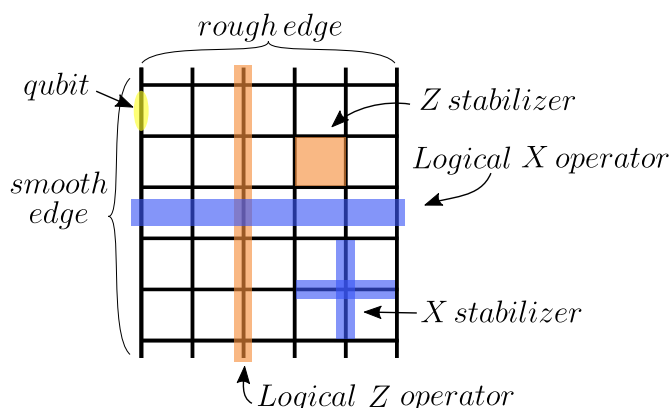
This section introduces the basic concepts of both surface codes and color codes. We explore the main characteristics that differentiate each other and the main challenges of decoding the syndrome in each code. In addition, we discuss some of the requirements to perform fault tolerant quantum computation with these codes.

#### 3.2.1 Surface code

Surface codes [110–112] are the “*textbook example*” of topological codes. They can be generated by arranging a square lattice, in which the qubits lie at the edges. The stabilizers of the surface code are different for the  $Z$  and  $X$  parity checks. Each  $X$  stabilizer is measured over the 4 qubit-edges touching a vertex, and each  $Z$  stabilizer is measured over the qubit-edges that form a square or *plaquette*. The number of logical operators depends on the topology of the lattice, and corresponds to non-trivial loops over the lattice [110, 232]. For instance, a sphere does not encode any logical qubit (as every loop can be reduced to a point), and a torus encodes two logical qubits. The support of the logical operators lies on non-trivial loops of the topology.

Since the physical implementation of lattices with periodic boundary conditions can be challenging in some platforms, it is interesting to study planar surface codes.

On lattices with open boundaries, we can distinguish two types of borders: rough and smooth. Logical  $X$  operators are generated by chains of  $X$  operations on qubits forming a chain from one smooth boundary to a different one, while logical  $Z$  operators link rough boundaries with  $Z$  operations. A more clear distinction between the types of borders can be seen in Fig. 3.4.



**Figure 3.4: Surface code lattice.** In surface codes, qubits lie at the edges of a square lattice. The  $Z$  stabilizers are measured over the plaquettes of the lattice, while the  $X$  stabilizers measure the qubits touching a vertex. The logical operators are constrained by the edges of the lattice. Logical  $Z$  operators connect rough edges, and logical  $X$  operators connect smooth edges.

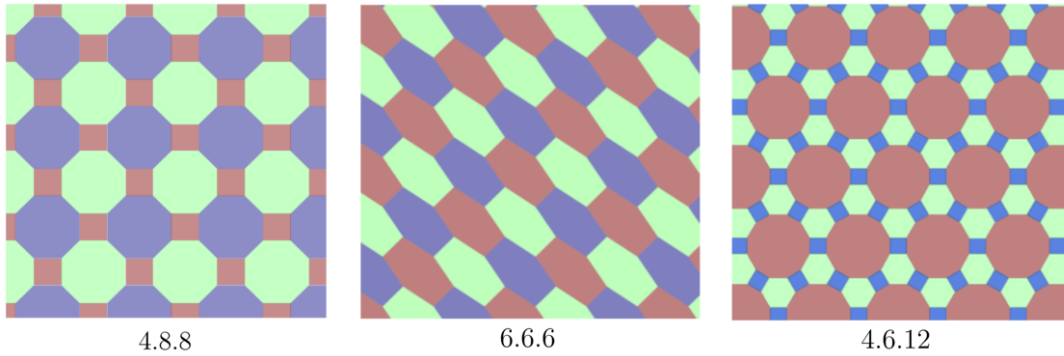
One key question on the application of surface codes for fault tolerant quantum computation is the realization of logical gates on the encoded qubits in a fault tolerant manner. In the ideal scenario, we would like to apply the logical gates *transversally*, i.e. by applying single-qubit gates to a subset of the qubits in the lattice. This possibility is restricted by the properties of the code. On surface codes, both logical  $X$  and logical  $Z$  can be applied transversally (by using the operators shown on Fig. 3.4), as well as the logical CNOT gate between two encoded qubits. However, neither the logical Hadamard, the logical phase-gate, or the T-gate can be applied transversally on surface codes. Fault tolerant implementations of those gates require more complex techniques, like lattice surgery, magic-state injection and magic-state distillation [224–226, 233], or the recent approach based on twist teleportation [234].

### 3. QUANTUM ERROR CORRECTION

---

#### 3.2.2 Color codes

Color codes are defined on two-dimensional lattices, where each qubit is connected to three stabilizers and three other qubits [113, 132]. On the lattice, qubits are represented by vertices, while the stabilizers are represented by three-coloured faces. The three colors are not a degree of freedom, but a property that emerges from the geometry of the codes that is called color for visualization purposes, and also gives name to this family of codes. There are three regular lattices that define color codes: the 6.6.6 lattice (hexagonal), the 4.8.8 lattice (squares and octagons) and the 4.6.12 lattice (see Fig. 3.5). Each face on the lattice represents both an  $X$  and a  $Z$  stabilizer. As a topological code, the number of logical operators depends on the topology of the lattice, and the logical operators form non-trivial loops. For closed lattices, the number of logical qubits as a function of the genus of the lattice is  $k = 4g$  [113, 235], which means that a toric color code lattice encodes 4 logical qubits (see Fig. 3.7). As with surface codes, it is possible to find planar lattices without periodic boundary conditions, with triangular color codes being a prominent example, encoding one logical qubit [113].

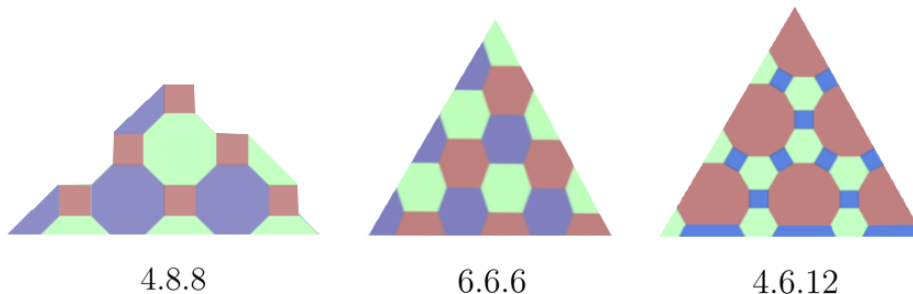


**Figure 3.5: Color code lattices.** Color codes can be arranged in three regular lattices. Qubits lie in the vertices, and the faces represent the stabilizer measurements for both  $X$  parity checks and  $Z$  parity checks. The logical operators depend on the particular topology of the lattice, e.g. toric code.

Color codes present several advantages over surface codes. Color codes present a better relationship between the distance of the code and the number of qubits needed. In particular, for triangular color codes of distance  $d$  (see Fig. 3.6), the number of data qubits required scales as  $n = \frac{1}{2}d^2 + d - \frac{1}{2}$  for the 4.8.8 lattice, which presents the highest asymptotic ratio of  $d^2$  to  $n$ , while the 6.6.6 lattice requires  $n = \frac{3}{4}d^2 + \frac{1}{4}$  and the



4.6.12 requires  $n = \frac{3}{2}d^2 - 3d + \frac{5}{2}$  [183]. Planar, square surface code lattices require on the order of  $n = d^2$  qubits, and the recently proposed triangular surface codes require  $n = \frac{3}{4}d^2 + \frac{1}{4}$  [236]. This can lead to a more efficient use of the resources for QEC, which can already be seen at small distances: at distance  $d = 5$ , color codes can be defined with only  $n = 17$  data qubits, while color codes require  $n = 25$  using the square lattice, and  $n = 19$  using the recently proposed triangular surface code. Nevertheless, the main advantage over surface codes is the possibility of the transversal application of the entire Clifford group when using the 4.8.8 color code lattice [113](see Fig. 3.7).

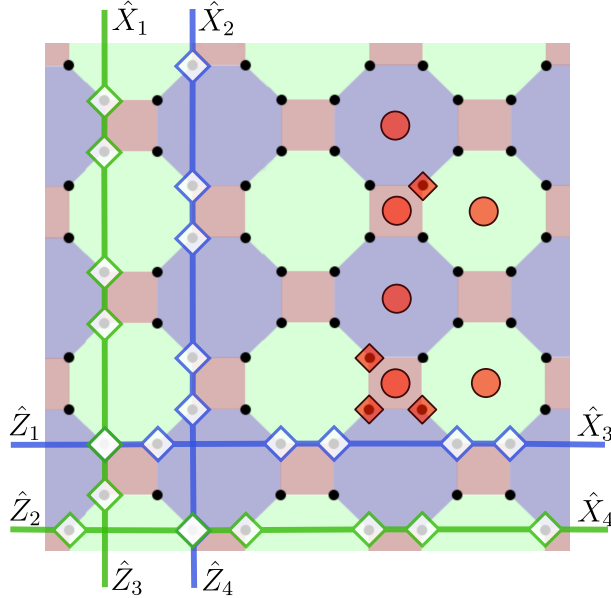


**Figure 3.6: Distance  $d = 7$  triangular color code lattices.** Triangular color code lattices encode a single logical qubit. The support of the logical  $X_L, Z_L$  operators is located on the qubits along one of the edges of the triangle.

Thus, the 4.8.8 color code lattice is optimal in the sense that the addition of one transversal non-Clifford gate (e.g. the T gate) would achieve a universal gate set, which would violate the Eastin and Knill no-go theorem [223]. This last gate is typically implemented by other means, like magic state distillation and injection [224–227], code switching [237], gauge fixing [238], code doubling or stacked codes [239, 240]. Since magic state distillation and magic state injection are generally the most expensive operations in fault tolerant QC, the possibility of applying more gates in a transversal way can present a significant advantage over surface codes [237].

On the other hand, the stabilizer operators required for color codes have a higher weight than the weight-4 stabilizers required for surface codes. This presents an additional challenge for a fault tolerant implementation, as higher weight stabilizers can lead to dangerous error propagation events through the lattice and require a more careful examination. As for the circuit-noise thresholds, currently the highest values reported lie between 0.07%-0.80%, depending on the particular model used [183, 237, 241]. In comparison, circuit-noise thresholds for surface codes have been reported between 0.14%-1%

### 3. QUANTUM ERROR CORRECTION



**Figure 3.7: Representation of the color code 4.8.8 lattice.** Qubits are represented by black vertices, and the stabilizers  $S_Z$  and  $S_X$  are represented by the colored plaquettes, which apply a parity check over the qubits on the vertices. The logical operators are strings of Pauli operators that extend over the torus in a non-trivial way. On the color code lattice on a torus, we have two independent logical operators for each non-trivial loop, i.e. four logical qubits. The support of the logical operators  $\hat{X}_i$  and  $\hat{Z}_i$  is represented by the blue and green lines, which represent the four non-trivial loops on the toric color code lattice. To illustrate the effect of errors in the lattice, we display an example of four physical bit-flip errors (qubits are marked with red diamonds), and the corresponding stabilizer excitations (stabilizers are marked with red circles).

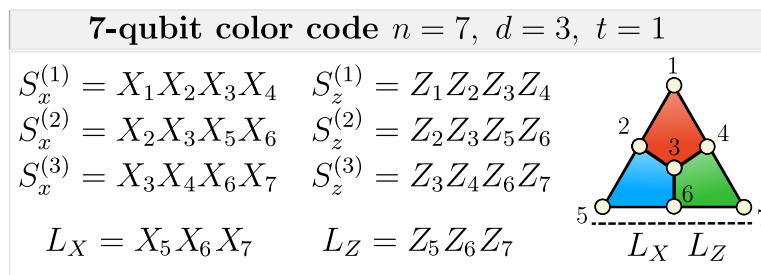
under similar restrictions [131, 183, 241, 242]<sup>1</sup>. Although color codes can also be defined in 3D [238], in this work we focus our discussion on 2D color code lattices.

#### 3.2.2.1 Steane code and flag-based readout techniques

The smallest implementation of the color code is the 7-qubit color code (Steane code) [113, 214]. This code allows the encoding of a single logical qubit in seven physical qubits, and has a logical distance  $d = 3$ , which allows the detection and correction of at least  $\lfloor \frac{d-1}{2} \rfloor = 1$  arbitrary phase or bit-flip fault on any of the physical qubits. The set of stabilizers and the logical operators are shown in Fig. 3.8. In addition, the code permits the transversal application of the entire Clifford group [113], which facilitates a

<sup>1</sup>Although a threshold of 1% was reported in [242], a follow-up work attempting to reproduce this value [241] failed to reproduce this result and reported 0.7%.

fault tolerant application of gates at the logical level. The small size of the code makes it an interesting milestone for near-term quantum devices.



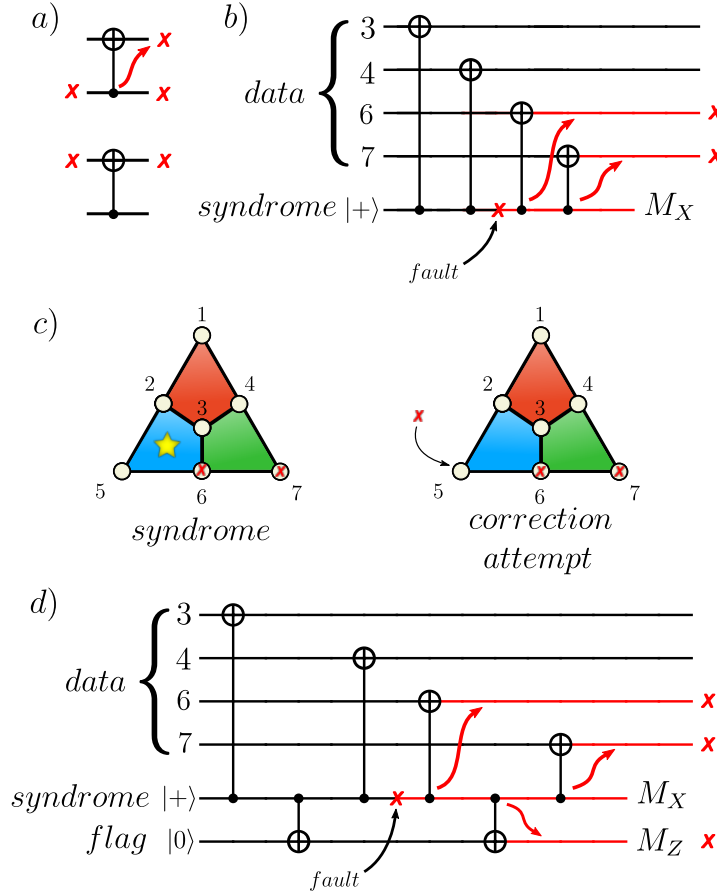
**Figure 3.8:** The distance  $d = 3$ , 7 qubit color-code (Steane code), encoding one logical qubit in  $n = 7$  physical qubits. The qubits are represented by the dots in the vertices, the logical operators  $L_X$  and  $L_Z$  act on the edge of the triangle, and the stabilizer checks  $S^{(i)}$  correspond to the plaquettes. The code can correct up to  $t = 1$  arbitrary fault on any of the seven physical data qubits.

As outlined in Sec. 3.1.3, a naive implementation of stabilizer measurements can allow a fault in the ancilla qubit to propagate to the data qubits, potentially compromising the encoded logical information. This type of error events where a single fault in an ancilla can propagate to several data qubits is called a hook error (see Fig. 3.9). To prevent this cascade effect during the stabilizer measurement, several FT schemes have been proposed [126, 215]. Due to present experimental capabilities, there has been intense recent activity developing resource-efficient FT schemes, such as FT readout techniques that exploit a single ancilla for certain types of microscopic noise [228, 241, 243, 244]. A notable example is the use of the so-called flag-based readout [114, 229, 245, 246], which substantially reduces the number of required ancillary qubits with respect to the initial FT schemes.

The flag-based readout scheme is a technique to detect and correct hook errors (see Fig. 3.9) by the use of an additional ancilla that will act as a flag to detect the hook errors. If an error is detected, a second measurement of the stabilizers will provide enough information to correct the faults and protect the logical information [245].

We can use the example in Fig. 3.9 to understand the intuition behind this scheme. During the measurement of the  $X$  stabilizer, a Pauli  $X$  error in the ancilla is not detected when measuring the ancilla in the  $X$  basis, but it can propagate through the CNOT's to the data qubits, leading to a potential weight-2 error that can be detected, but would be confused with a weight-1 error. This event breaks fault tolerance, as a

### 3. QUANTUM ERROR CORRECTION



**Figure 3.9: Error propagation through the syndrome measurement circuit.** **a)** bit-flip error propagation through a CNOT gate: an X error in the control qubit propagates to the target qubit, but it does not propagate from the target qubit to the control qubit. **b)** Standard stabilizer measurement of  $S_x^{(3)}$  (green), where a single ancilla is used to measure the stabilizer. A single fault on the ancilla can propagate through the CNOTs and result in two X faults on the data qubits (6 and 7). This is called a hook error. **c)** Syndrome readout: (left) after measuring all 6 stabilizers, only the blue stabilizer (marked with a star) will be excited by the two faults. (right) The correction corresponding to that syndrome would be to apply an X operation to qubit 5, which would complete the logical  $X_L = X_5 X_6 X_7$  operator and lead to a logical X error. **d)** Stabilizer readout with a flag qubit: hook errors can be detected by using an additional ancilla (flag qubit) in the stabilizer readout. If a flag qubit signals a fault, another round of syndrome readout is run, and the information obtained by the flag and the syndrome can be used to identify and correct the error [243].

single fault in the protocol can lead to a weight-two fault in the output, which, for a distance  $d = 3$  code like the Steane code, can lead to a logical error. The solution to this problem is the use of an additional *flag* ancilla. This additional qubit is coupled to the ancilla at critical points in the circuit, so that the potential hook error X in

the ancilla is also propagated to the flag qubit, which is measured in the  $Z$  basis. In this case, the event will be detected by the flag. However, this detection by itself does not give enough information to infer the exact fault that occurred. Thus, we realize a second stabilizer measurement that, together with the information of the flag, will provide enough information to correct the faults and restore the logical state.

The flagged measurement of the stabilizers allows a FT implementation of the Steane code with only  $7 + 2$  qubits, using two ancilla qubits to measure the stabilizers sequentially. However, a recent work [229] presented an alternative scheme that allows parallel flagged measurement of stabilizers. Using this scheme, it is possible to measure up to three different stabilizers simultaneously, utilizing the ancilla for each stabilizer measurement as a flag for the rest of the measurements. This scheme follows the same intuition as the single-flag scheme, where an error in one ancilla can propagate to another ancilla measuring the qubit in an orthogonal base, so that potential hook errors are detected and can be corrected using information from an additional syndrome readout.

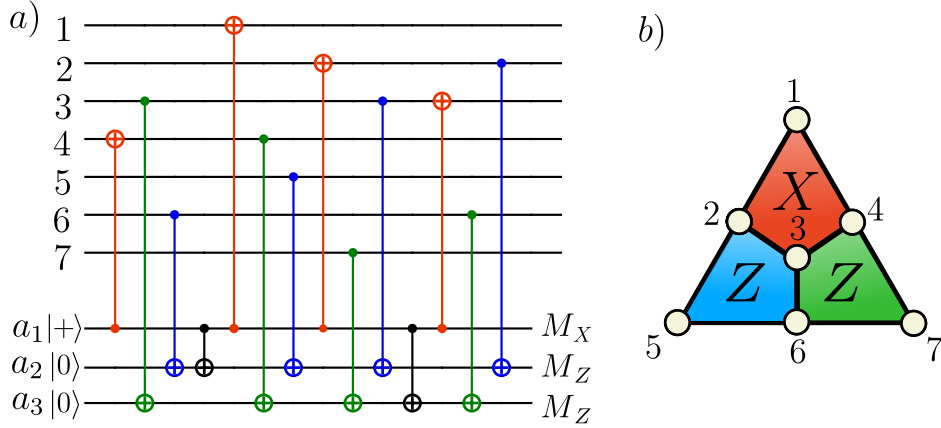
In the ion-trap platform that we consider in chapter 6, qubit measurement is the slowest operation as the qubits need to be re-cooled after each measurement operation. Since measurement operations can be done in parallel, this alternative scheme with a simultaneous measurement of 3 stabilizers can significantly reduce the duration of the QEC cycle. The circuits of the scheme can be seen in Fig. 3.10.

The QEC cycle for the Steane code with simultaneous readout of 3 stabilizers in parallel requires a total of  $7 + 3$  qubits. The 6 stabilizers are measured in two steps, measuring the stabilizers  $S_X^{(1)}$ ,  $S_Z^{(2)}$  and  $S_Z^{(3)}$  in parallel during the first step, and the stabilizers  $S_Z^{(1)}$ ,  $S_X^{(2)}$  and  $S_X^{(3)}$  in the second step. Both of these parallel measurements implement the flag scheme, where each ancilla is coupled to the others and acts as a flag for possible hook errors happening on the other ancillae. The details on this QEC protocol can be found in [229], and are explained further in the research chapter 6.

### 3.3 Decoders

The problem of decoding refers to the interpretation of the measured syndrome to obtain a recovery operation or correction. For small codes, like the Steane code, it is possible to compute the most probable error for each possible syndrome (assuming a

### 3. QUANTUM ERROR CORRECTION



**Figure 3.10: Parallel syndrome extraction**, as proposed by [229]. One can simultaneously measure 3 stabilizer checks in parallel in a fault tolerant way. **a)** Circuit for the simultaneous measurement of the stabilizers  $S_x^{(1)}$ ,  $S_z^{(2)}$  and  $S_z^{(3)}$  via ancillae  $a_1$ ,  $a_2$  and  $a_3$ , respectively. The gates used in the measurement of each stabilizer are highlighted in the color of the stabilizer they implement, and the two black CNOT gates between the ancilla qubits represent the crucial elements of this scheme, as they allow the dangerous hook errors to propagate to other ancillas, capable of detecting these events. **b)** Stabilizers measured with the circuits in a).

given error model), and find the best recovery operation for each case. This information can be used to write a lookup table (see e.g. 3.1), which can be consulted in  $\mathcal{O}(1)$  complexity to obtain an optimal correction (according to our error model). However, finding the lookup table is a complicated problem for larger codes, as the number of possible syndromes scales exponentially with the number of stabilizers. Finding scalable and efficient decoders for topological QEC codes is one fundamental problem for the scalability of QEC protocols.

This section discusses some of the most prominent decoding algorithms in the literature, both for surface codes and color codes. The two main magnitudes that characterize different decoding algorithms are the efficiency of the decoder (the scaling of the computational complexity) and the threshold of the decoder. The figure of merit to evaluate the thresholds is the optimal decoder, that finds the recovery operation with the maximum success probability for each syndrome. An important aspect to take into account when evaluating decoders is the error model for which they are designed, i.e. code capacity, phenomenological noise, or circuit-level noise (see Sec. 3.2).

The most relevant decoding algorithm for surface codes is the minimum weight perfect matching (MWPM) [111, 185, 186, 247], which stems from one key property of

excitations in the surface code. Let us assume a toric lattice with periodic boundary conditions, and focus only on bit-flip errors, as the argument would be analogous for phase-flip errors. Each qubit is connected to only two  $Z$  stabilizers and only two  $X$  stabilizers. Therefore, an bit-flip error on a qubit will create two excitations of the nearby  $Z$  stabilizers. If there is a second error next to it, the stabilizer shared by both errors will now have even parity, and only two excitations will remain. For longer error chains, this rule remains valid, and there will be only two excitations on the ends of the chain. Since excitations of stabilizers are created in pairs connected by chains of errors, we can find a correction by finding a pairing of the excitations measured in the syndrome. Furthermore, if we assume that the probability of an error is small, we can find the most probable error by finding a pairing for which the total length of the chains is minimized. Fortunately, there is an efficient classical algorithm that solves this problem in polynomial time, and gives the name to the decoder: Minimum Weight Perfect Matching (MWPM). For lattices with open boundaries, the algorithm can be applied by including the possibility of chains with an ending on the boundary. The runtime complexity of MWPM as a function of the number of qubits  $N$  is  $\mathcal{O}(N^4)$  for a straightforward implementation of the original Blossom algorithm [185, 247], which can be optimized to  $\mathcal{O}(N^{2.5})$  using recent advancements [248, 249]. One important detail to note is that, although MWPM can find the most probable error chain, this solution is different from the optimal decoder, which would find the correction with the highest probability of recovery. For this, the optimal decoder needs to consider the degeneracies in the solution that stem from the application of stabilizers, since operations that differ in the application of a product of stabilizers are equivalent in stabilizer codes (also referred to as *equivalent up to stabilizers*). In general, this is computationally expensive and requires a prohibitive amount of computational resources.

MWPM is the most widely used decoder for the surface code, and it is of particular relevance for this thesis, since it is the reference to which other decoders can be compared, as it combines a high threshold with polynomial efficiency. Furthermore, the MWPM is used as a subroutine by some color code decoders [196, 241] that first map the color code into two surface codes [192], and then applies MWPM on each surface code before proceeding further with the algorithm. If we include errors in the measurements, the decoder can be adapted by adding a third dimension for the measurement

### 3. QUANTUM ERROR CORRECTION

---

round. Thus, it can be applied successfully for code capacity, phenomenological noise, and circuit-level noise [183, 186].

Decoding color codes presents a more challenging problem. While in the surface code excitations appear always in pairs at the end of chains of errors, the dynamics of excitations in the color code are more complicated (see Fig. 3.7). A single-qubit error creates three excitations on the neighboring stabilizers. Chains of excitations can generate pairs of excitations of the same color. However, the chains of excitations can also branch into networks of errors, generating more complex excitations patterns that can become harder to decode. As a result, the decoder algorithms become more complex than for surface codes. A variety of color code decoders have been proposed over the last years, like ideas relating a mapping of the color code to surface codes [191–194, 196], renormalization group decoders [189], neural networks [197, 201, 204], and many others, e.g. [190, 193, 195, 196, 250, 251]. Currently, the best threshold for code capacity noise has been achieved using the restriction decoder [196], which achieves a 10.2% threshold by mapping color codes to two different copies of surface codes. This threshold is to be compared with the optimal value for the code capacity of 10.9%, that can be obtained by mapping color codes to a 2D random-bond Ising model with three-body interactions [218–220]. In chapter 7 we study some adaptations of the so-called Union Find decoder [193], while in chapter 8 we study a modification to a decoder based on the rescaling of the lattice [189]. The topic of decoders for color codes will be explored more in-depth during those chapters.



## Chapter 4

# Simulation of quantum circuits

## 4. SIMULATION OF QUANTUM CIRCUITS

---

In previous chapters (2 and 3) we have seen how the effect of different error channels can be modeled using Kraus maps and open quantum system dynamics; and how we can use QECCs to detect and correct errors in a quantum code. However, to evaluate the performance of a quantum protocol under the influence of noise, the capability of simulating quantum systems becomes a fundamental tool. By using simulations, we can predict the performance of QECCs, study the failure scenarios and evaluate the usefulness of different protocols (e.g. [169, 171, 197, 252–254]).

Accurate and fast algorithms for the simulation of quantum systems are thus an indispensable resource in the study of QEC. One of the main challenges of simulating quantum systems lies in the scaling of the Hilbert space in which states are represented. Since our goal in this thesis is to study the performance of QEC protocols, we restrict ourselves to systems of qubits. If we want to represent an arbitrary wavefunction of a system of  $n$  qubits, this can mean storing a vector with  $\mathcal{O}(2^n)$  complex numbers. An arbitrary unitary matrix applied to this system would also scale exponentially in size with  $\mathcal{O}(2^{2n})$  entries. Simulations that compute *trajectories* (i.e. evolution of a single pure state) for arbitrary wavefunctions in this fashion are commonly called state-vector simulations. Furthermore, to fully represent the dynamics of a QEC protocol, we might be interested in finding the resulting density matrix of the system after applying a given error channel. Again, the size of an arbitrary density matrix for such a system would scale with  $\mathcal{O}(2^{2n})$ . Due to this exponential scaling, the computational cost of working with such matrices can become prohibitive for even small systems with less than a few tens of qubits.

However, some studies point out that it is not necessary to keep track of the entire density matrix of a system in order to reproduce the dynamics of a realistic system [255, 256]. This opens the possibility for classical algorithms to be developed and used for simulating quantum systems. Among these algorithms, we find simulations using matrix-product-states (MPS) [256, 257], quantum Monte Carlo (QMC) [258–262], path-integral Monte Carlo (PIMC) [263, 264], simulated quantum annealing (SQA) [265, 266], stabilizer simulation [211, 252, 267–269], density matrix renormalization group (DMRG) [270–276], Pauli propagation [181, 277], as well as optimized implementations to reproduce wavefunctions and density matrices e.g. [253, 278]. The development of classical algorithms to reproduce quantum systems is an active field of research, with multiple platforms and packages that implement different algorithms [279–282]

---

covering a broad range of fields, e.g. quantum computation, QEC, quantum annealing, condensed matter or quantum chemistry. Each algorithm offers some advantages and disadvantages over the rest, with trade-offs in e.g. efficiency, fidelity or universality. For the remainder of the thesis, we focus on applications in quantum computation and the simulation of quantum circuits for QEC.

Regarding the simulation of quantum circuits, there are some subsets of quantum processes that can be efficiently simulated with classical computers. From the Gottesman-Knill theorem [100, 212], we know that quantum circuits can be simulated efficiently if they consist only of Clifford gates and state preparation and measurement in the computational basis. Although this set of operations does not allow for any arbitrary operation, the gates required for most QEC protocols are contained within this set. This means that it is possible to simulate efficiently any ideal QEC protocol that fulfills this condition. The main restriction of this theorem will affect, then, the error channels, that will require a Clifford approximation in order to simulate them efficiently (e.g. [182, 252, 268, 269]). Depending on the particular error model, this restriction can have a significant impact on the precision of the simulations, as we will see in chapter 6 with the study of crosstalk errors, a particular type of error channel that applies coherent entangling rotations to pairs of qubits.

The next obvious question concerns the algorithm capable of the efficient simulation of Clifford circuits. Stabilizer simulation stands as a well-known algorithm that represents states by using a representation based on the stabilizer formalism [100, 211, 267]. Together with some properties of the Pauli group of operators, this compact representation of the states allows the implementation of a quantum simulation with polynomial scaling. Nevertheless, this efficiency can only be achieved when restricting the set of operations to Clifford operations and measurements in the Pauli basis<sup>1</sup>. Therefore, when comparing stabilizer simulations with state-vector simulations (that reproduce the entire wavefunction of the system), there is a trade-off between computational complexity and accuracy of the simulations, since non-Clifford operations will be approximated.

In this chapter, we describe the two main simulation algorithms that we use throughout the thesis: state-vector simulation and stabilizer simulation. We discuss the working principle, and the advantages and disadvantages of each method. This chapter aims

---

<sup>1</sup>Recent works are studying how circuits with a limited number of non-Clifford operations can be simulated within the stabilizer formalism [182].

## 4. SIMULATION OF QUANTUM CIRCUITS

---

to introduce the reader to the basic concepts needed to understand the algorithms and their limitations. Finally, we describe some of the alternative techniques available for the simulation of quantum systems.

### 4.1 State-vector simulation

State-vector simulations reproduce the evolution of pure state trajectories  $|\psi(t)\rangle$  by storing all the amplitudes  $c_n$  of the state in a given basis in a vector, e.g.

$$|\psi\rangle = \sum_n^N c_n |n\rangle = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}. \quad (4.1)$$

Unitary evolution can be computed directly by realizing the product of the unitary matrix and the vector of the state

$$(U|\psi\rangle)_i = \sum_k U_{i,k} c_k. \quad (4.2)$$

Measurement of an operator  $M$  can be simulated by computing first the probability of each outcome, and then using a random number to simulate the random outcome  $m$  of the measurement. If we have the projectors  $M_m$  for each of the outcomes, we can compute the probability of each outcome from the state in the standard way:

$$p(m) = \langle\psi| M_m^\dagger M_m |\psi\rangle. \quad (4.3)$$

As we have seen, state-vector simulations are, at its core, a direct numerical implementation of the simulation of wavefunctions. If we work with stochastic noise described by a Kraus map, they allow the simulation of individual evolutions, from which we can obtain the statistics by repeating multiple iterations using standard Monte Carlo techniques [262]. Thus, state-vector simulations provide a complete tool to study quantum circuits and obtain precise estimates for any arbitrary evolution. This includes not only Clifford operations, but also any arbitrary operation applied on any number of qubits.

On the other hand, there is a price for the power of state-vector simulations to simulate any arbitrary quantum evolution, which is the high computational cost. As

discussed in the introduction, the size of the vector representing an arbitrary  $n$  qubit state scales exponentially as  $\mathcal{O}(2^n)$ , and the worst-case scenario for a unitary matrix affecting these states scales with  $\mathcal{O}(2^{2n})$ . Although the implementation of the algorithm can be optimized using e.g. sparse matrices to represent the states [281], the computational cost is the main limiting factor of this algorithm. Simulation of long circuits or large number of qubits can become a very costly computational task.

Therefore, motivated by the need for fast algorithms to evaluate noisy quantum circuits, we want to study the use of alternative simulation methods capable of efficiently simulating quantum circuits. This search leads us to stabilizer simulation, an algorithm that operates in polynomial time at the cost of restricting the available operations.

## 4.2 Stabilizer simulation

While the state-vector simulations described in the previous section provide a reliable method to simulate any arbitrary quantum system, we have seen that the algorithm becomes very inefficient with the scaling of the quantum system. The computational cost of the algorithm increases exponentially, making computations increasingly slow as we try to simulate more qubits. This result is expected, as an arbitrary quantum evolution cannot be efficiently simulated with a classical computer.

In this section, we explore how the stabilizer formalism leads to an efficient way of representing and evolving quantum states, as long as all the operations applied in the circuit are constrained to Clifford gates, measurements in the computational basis, and state preparation in the same basis. We explain how we represent a quantum state, and how each operation can be applied in stabilizer simulations. The goal of this section is to introduce the reader to the basic concepts behind stabilizer simulation, since this algorithm is used for the simulations in 6. A more detailed explanation can be found in [100].

The working principle behind stabilizer simulation lies in the stabilizer formalism. The core idea is that we can represent a state  $|\psi\rangle$  by the set of stabilizer operators  $S_i$  (see Sec. 3.1.1) that stabilize that state. This, by itself, does not necessarily give us any computational advantage. The key idea that leads to the efficiency of this algorithm is the use of group theory, in particular, with the use of the Pauli group  $P_n$  for  $n$  qubits (see Eq. 2.9). By including the factors  $\pm 1$  and  $\pm i$  to the set of Pauli matrices, we

#### 4. SIMULATION OF QUANTUM CIRCUITS

---

ensure that the group is closed under multiplication. We can then generate the general Pauli group by the  $n$ -fold tensor product of Pauli matrices on the different qubits and the multiplicative factors  $\pm 1$  and  $\pm i$ . By restricting our set of operations to Clifford gates and measurements in the computational basis, the stabilizers that describe the states in our simulations will remain in this Pauli group. This allows a very compact representation of the stabilizers, that can be operated with a low computational cost.

In order to simulate the evolution of a given state, we need to compute how the different stabilizer operators are transformed. Using the properties of stabilizers and unitary operators, we can easily identify the transformation of the stabilizer operators after a unitary evolution  $U$ :

$$U|\psi\rangle = US|\psi\rangle = USU^\dagger U|\psi\rangle = S'|\psi'\rangle. \quad (4.4)$$

Thus, we can compute the evolution of a state by updating the stabilizers following the rule  $S' = USU^\dagger$ . For instance, we can prepare the state  $|01\rangle$  by its stabilizers  $S_1 = Z_1$  and  $S_2 = -Z_2$ . Then, we can apply a Hadamard gate  $H_1$  on the first qubit by changing the stabilizers as  $S'_1 = H_1 Z_1 H_1^\dagger = X_1$  and  $S_2 = -H_1 Z_2 H_1^\dagger = -Z_2$ . To illustrate the algorithm, let us use an example with a two-qubit state. We can describe the state  $|00\rangle$  with the stabilizers  $S_1 = Z_1$  and  $S_2 = Z_2$ . In the following table, we show how the state evolves under some common Clifford gates both in the bra-ket representation and the stabilizer representation:

Operation:	-	$H_1$	$C_1NOT_2$	$X_1$
State:	$ 00\rangle$	$\frac{ 00\rangle+ 10\rangle}{\sqrt{2}}$	$\frac{ 00\rangle+ 11\rangle}{\sqrt{2}}$	$\frac{ 10\rangle+ 01\rangle}{\sqrt{2}}$
$S_1$	$Z_1$	$X_1$	$X_1X_2$	$X_1X_2$
$S_2$	$Z_2$	$Z_1$	$Z_1Z_2$	$-Z_1Z_2$

The last tool needed in stabilizer simulation is the measurement in the computational basis. In order to measure an operator  $M$ , we need to consider the following two cases:

1.  $M$  commutes with all the stabilizers  $S_i$ . In this case, the measurement result is deterministic, since either  $M$  or  $-M$  is a stabilizer for the state. The result of the measurement will be  $+1$  or  $-1$ , respectively. The state will not change, and the stabilizers will remain invariant after the measurement.

2.  $M$  anti-commutes with one or more of the stabilizers  $S_i$ . In this case, the measurement result is assigned at random, and both outcomes  $m = \pm 1$  occur with probability  $\frac{1}{2}$ . The stabilizer that anti-commuted with  $M$  is then replaced by  $mM$ , where  $m$  is the measurement outcome. If more than one of the stabilizers anti-commuted with  $M$ , we can reduce it to the case in which only one stabilizer (say,  $S_1$ ) anti-commutes with  $M$  by replacing the anti-commuting stabilizers  $S_i$  with the stabilizer  $S_1 S_i$ , which now commutes with  $M$ .

We can illustrate this process using the same two-qubit state  $|00\rangle$  as example, with stabilizers  $S_1 = Z_1$  and  $S_2 = Z_2$ . If we want to measure any of the qubits in the computational basis, the corresponding measurement operator is  $M_i$ , which commutes with both stabilizers and yields  $+1$ , as expected. A more interesting example is the measurement of a Bell state, like  $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$  with stabilizers  $S_1 = X_1 X_2$  and  $S_2 = Z_1 Z_2$ . If we measure the first qubit in the computational basis, the measurement operator  $M = Z_1$  anti-commutes with the stabilizer  $S_1$  and commutes with stabilizer  $S_2$ . Following the rule, the result  $m = \pm 1$  of the measurement will be assigned at random, and stabilizer  $S_1$  will be replaced by the measurement operator. Thus, the set of stabilizers will now be  $S_1 = mZ_1$ ,  $S_2 = Z_1 Z_2$ . We can also write the second stabilizer as the product of both of them, leaving the set as  $S_1 = mZ_1$ ,  $S_2 = mZ_2$ . This coincides with the result in bra-ket notation, which would be  $|00\rangle$  for the  $+1$  measurement, and  $|11\rangle$  for the  $-1$  measurement.

The main limitation of stabilizer simulation is the restriction of the allowed operations to the Clifford group and the measurements in the computational basis. Standard noise models, like the depolarizing or the dephasing channels, can be simulated exactly. Stabilizer simulations become a fantastic tool in that case, thanks to their low computational cost. However, error models including more complex operations, like coherent rotations or amplitude damping, need to be approximated. Consequently, the choice between stabilizer simulations and state-vector simulations has to be done by taking into account not only the computational complexity of the task, but also the particular nature of the model being simulated. This question is explored in more depth in chapter 6, where we analyze the differences between coherent and incoherent modeling of errors for a particularly challenging error channel.

### 4.3 Other simulation techniques

We have seen two alternative methods to simulate quantum circuits. With the stabilizer simulation, we can get quick estimates from a simulation, while with the state-vector simulations we can achieve a higher level of precision, at the cost of computational time. We use both of these methods in the simulations in chapter 6. There are, however, other algorithms that allow either a higher precision, or a faster performance in the simulation of quantum systems. By studying these methods we can better understand the problem of simulating a quantum circuit. In this section, we will briefly discuss two of these alternative algorithms focusing on the simulation of noisy quantum circuits: density matrix simulation and Pauli propagation.

A density matrix describes the statistics of the final state after a noisy quantum circuit. Thus, the most straightforward and intuitive method to obtain accurate results on the performance of a quantum circuit is to simulate the evolution of the entire density matrix directly. Since all the statistics of the different stochastic processes happening on the noise are included in this density matrix, a single run of this algorithm on a noisy quantum circuit is enough to obtain complete information on the final state after a quantum circuit. Using this method, we avoid the need for a Monte Carlo simulation, where we need to simulate multiple runs of the circuit to obtain precise statistics on the results, usually requiring even millions of runs for a single point. Recent works, e.g. [197, 253], have used this method to obtain estimates of the performance of different QEC protocols.

The cost of density matrix simulations, however, is often higher than the potential benefit of requiring only a single run to obtain all the results, due to the double exponential scaling of the density matrix with the number of qubits  $\mathcal{O}(2^{2n})$ . This means that, despite being a desirable algorithm for systems with a small number of qubits, the computational cost of a density matrix simulation becomes prohibitive as we increase the number of qubits. This problem is aggravated by the nature of the QEC circuits that we study in this thesis, since they require the measurement and reset of ancilla qubits. A density matrix simulation of these circuits would require an additional qubit in the register every time an ancilla qubit is measured and reset, effectively increasing the scaling of the density matrix even further with the length of the circuit. Therefore,



density matrix simulations were discarded as an option to evaluate the circuits studied in this thesis.

The second algorithm that we discuss in this section is Pauli propagation [181, 277]. The goal of this algorithm is not to simulate the evolution of a given state after a quantum circuit. Instead, the focus is on simulating the effective noise operator that would be applied on a state in addition to the ideal circuit. We can compute this effective operator by propagating the noise operator through the unitaries of the circuit. This process can be easily understood with an example. Imagine a circuit composed of a series of 2 ideal unitaries  $U_C = U_2U_1$ . If we assume that a noise operator  $U_N$  is applied between  $U_1$  and  $U_2$ , the total unitary of the noisy circuit can be hard to analyze. However, we can apply the following transformation:

$$U_2U_NU_1 = (U_2U_NU_2^\dagger)U_2U_1 = U'_N U_C, \quad (4.5)$$

which would leave the total evolution as the ideal circuit unitary  $U_C$ , followed by the effective noise channel  $U'_N = U_2U_NU_2^\dagger$ . In this form, it is easier to analyze the effect of the noise on the final state, as the action of the ideal circuit  $U_C$  on a given state (usually the most computationally expensive operation) would only need to be computed once, and we can apply the effective noise directly to the ideal final state and check if the noise channel corresponds to a logical error or a correctable error.

If we restrict the error channel to operations within the Clifford group, we can use the techniques from stabilizer simulations to propagate the effective noise unitary through the circuit using the simple rules of Clifford operations. In addition, the noise channel obtained after each simulation is independent of the initial state, as no state is assumed during this operation. The main highlight of this algorithm lies on its high performance, as it allows for even faster simulations, since we only need to keep track of one operator on each simulation (in stabilizer simulations, we need to track the evolution one stabilizer operator for each qubit in the register). On the other hand, the disadvantage of this algorithm lies on the restrictions on the noise channels that can be simulated. Since the algorithm is independent of the initial state, simulating accurately error channels that depend on the state of the qubits (e.g. amplitude damping) becomes a more challenging task.

## 4. SIMULATION OF QUANTUM CIRCUITS

---

In this chapter we have seen some of the main algorithms commonly used to simulate noisy quantum circuits and study their performance. Depending on the complexity of the noise model, the desired accuracy, and the computational power available, there is a trade-off between the different methods. This challenge is explored more deeply in chapter 6, where we use and compare state-vector and stabilizer simulations to estimate the performance of a QEC protocol on a realistic ion trap device. In the next chapter, we study these ion trap devices and how they can be applied for quantum computation.

## Chapter 5

# Ion traps

## 5. ION TRAPS

---

This chapter, introduces the main concepts required to understand how ion traps can be used for quantum computation, and we review some of the recent advances in ion trap technology. The chapter aims to give the reader an overview of the state of the art of ion trap systems, and how they stand as potential platforms for a viable implementation of quantum computers. We introduce the leading approaches to ion traps and the operations included in the ion trap toolkit. These concepts will be relevant for chapter 6, where we realize a feasibility study of QEC on near-term ion traps.

For more in detail reviews on the topic, an interested reader can find several review papers [283–290] summarizing both the state of the art of current experiments and explaining the physics of ion trap systems. We used those review papers as the main source material for the writing of this chapter. In this work, we focus on qubit-oriented implementations, although other approaches have also been explored, like the use of ion traps to realize quantum annealers [291–293].

### 5.1 Ion traps for quantum computation

Ion traps, as a physical system for the realization of quantum computation, encode quantum information in the electronic states of ions in e.g. Paul traps or Penning traps. It is possible to interact, manipulate and measure these states using electromagnetic fields. This approach to quantum computation was first proposed by Cirac and Zoller [294], and since then many groups all over the world are showing successful experiments in the field. Trapped-ion experiments have already shown high fidelity single-qubit gates, two-qubit gates and measurements, as well as long coherence times [90–92, 139–145, 295]. Thus, ion traps stand as one of the most promising platforms for the implementation of large scale quantum computation, with multiple experiments and proofs of principle already published, e.g. [27, 34, 147–156, 169, 171, 296]. As of today, ion traps are one of the platforms that fulfill the five criteria established by DiVincenzo [87] to assess the viability of a physical implementation of a quantum computer:

1. A scalable physical system with well characterized qubits.
2. The ability to initialize the state of the qubits to a simple, well defined and determined state.
3. Long relevant decoherence times, much longer than the gate operation time.

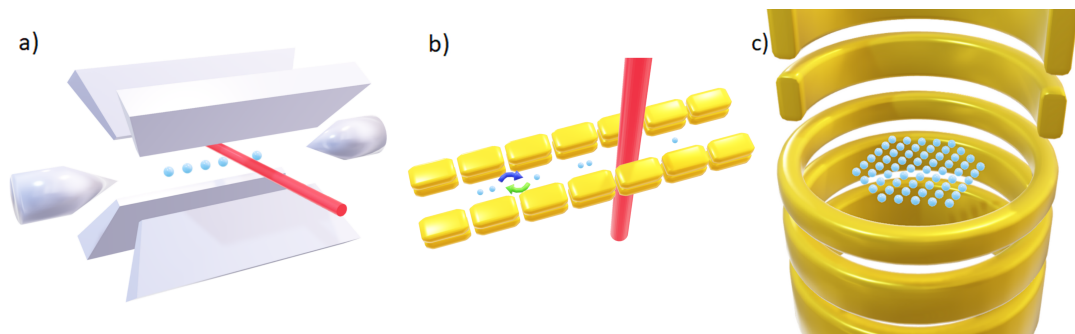
4. A “universal” set of quantum gates.
5. A qubit-specific measurement capability.

Over the following sections, we will see how each of these points is addressed in ion traps and the main approaches for the implementation and scalability of ion trap devices.

## 5.2 Scalability approaches

The scalability of the platform is one of the critical features required for the implementation of a large-scale quantum computer. To be precise, a quantum computer is considered *scalable* if the number of basic computational elements can be increased on-demand without suffering a loss in performance and with a reasonable increase in cost, energy usage, or footprint. In this respect, classical computers have achieved scalability, which can be seen in how the empirical Moore’s law was followed and the number of transistors on a single chip has doubled every 18 months.

Currently, there are three leading approaches for the scalability of ion traps: static string ion traps, shuttling and Penning traps (Fig. 5.1) [88, 149, 288, 297–299]. Here we summarize the main aspects of each approach:



**Figure 5.1: Artistic representation of the different scalability approaches.** **a)** Static string ion trap: ions are trapped in a single linear string of ions that can be operated using addressed laser beams. Entangling operations between pairs of ions are available via coupling with the center of mass (COM) modes. **b)** Shuttling: an array of traps can be used to trap multiple ions that can be moved between the different traps. Ions in the same trap form a Coulomb crystal in which entangling operations can be realized. **c)** Penning trap: ions can be trapped in a 2D array, allowing a larger amount of ions to be trapped in the same trap.

## 5. ION TRAPS

---

1. **Static string ion trap.** In this approach, ions are trapped in a single, linear Paul trap, using an oscillating electric field to confine the ions in a cylindrical pseudopotential. An additional static field traps the ions in the axial dimension. Gates can be applied using addressed laser beams that can be focused on individual ions. In this trap, ions are arranged in a single long line, and the electronic levels can be coupled to the levels of oscillation of the center of mass (COM). This allows the direct implementation of entangling gates between any two ions in the trap [149, 285, 300–302]. This approach can be scaled by connecting several traps using e.g. optical coupling [303, 304].
2. **Shuttling-based.** In this approach, ions are trapped in an array of Paul traps. Entangling operations can be realized between ions localized in the same interaction zone in the trap. After applying the desired sequence of operations, these ions can be split and moved between the different sections of the trap via shuttling, so that entangling operations between different subsets of ions can be realized [149, 151, 171, 299, 305–310]. To scale the system, there are several ideas, e.g. having a long linear array with several interaction zones, or connecting multiple traps in an array [311]. However, entangling operations between qubits in traps localized at large distances can become increasingly costly, as the ions would need to be shuttled to the same trap in order to apply an entangling operation.
3. **Penning traps.** In this approach, ions are trapped by static electric and magnetic fields in a small volume of space. Typically, this volume is constrained to a planar surface, allowing 2D arrays of ions arranged in the same trap, which presents a big advantage for scaling the system. Gates and measurements can be realized using laser beams addressing single sites or multiple ions [146, 312–316]. Multiple Penning traps can also be used, shuttling ions between the different traps [297, 317, 318]. These traps have also been studied for the use of trapped electrons instead of ions, potentially offering faster operations [319–322]. Penning traps have been used very successfully for quantum simulation of, e.g. 2D spin models and quantum annealing [316, 323, 324].

The scalability of ion-trap devices is an active field of research, and multiple other approaches are being studied [298]. Among the various lines of research, we find:

- The possibility of trapping ions in a planar arrangement [325].
- The use of electrons for trapped-ion quantum computing [326].
- The use of photons to entangle qubits in distant traps [303, 304, 327–332].
- The use of shuttling or quantum teleportation between different traps [311, 318, 333].
- The use of Rydberg ions [334, 335].

For the rest of the thesis, we focus on the static string ion trap, as this is the approach for which we focus our feasibility study in Chapter 6, where we describe the work done for the publication [1].

### 5.3 Qubits in the ion trap

The qubits of the ion trap are encoded in the electronic states of the ions in the trap, with two energy levels representing the states  $|0\rangle$  and  $|1\rangle$ . This pair of states can come from any two long-lived or meta-stable levels. Depending on the energy difference between these two states, there are four different types of qubits, each with its advantages and drawbacks [283]:

1. Zeeman qubits, where the pair of states lie in the same electronic orbital and hyperfine level, and are split by an external magnetic field, with transition frequency on the order of megahertz. Although qubit lifetimes are extremely long, Zeeman qubits have a high sensitivity to fluctuations in the magnetic fields. Single and two-qubit gates are typically performed using two-photon stimulated Raman transitions, as operations using direct RF drive is difficult to spatially focus, leading to potential crosstalk between neighboring qubits.
2. Hyperfine qubits, with an energy difference on the order of gigahertz. The ground-state hyperfine levels are generated by the interaction of a non-zero nuclear spin with the valence electron. These states are some of the most long-lived states available, with spontaneous emission-limited lifetimes close to the age of the universe.

## 5. ION TRAPS

---

3. Fine structure qubits, which use a pair of states in the D manifold  $D_{3/2}$  and  $D_{5/2}$ . The transition frequency for these qubits is on the order of terahertz, and the lifetimes are on the order of seconds, mostly due to leakage rather than a decay to the other qubit state.
4. Optical qubits, with the use of a state in the ground-state manifold and a state in a metastable D level. As the name suggests, these qubits have optical transitions on the order of hundreds of terahertz, typically in the visible to near-IR region of the spectrum. These qubits require a high level of control over the lasers used in the gates, and can achieve lifetimes on the order of seconds. However, the transition frequency is an advantage for the scalability of near term experiments due to the high development of components and integrated technology made for the red and IR range.

Apart from the electronic levels, the ions in the ion trap also have an energy level associated with the ion chain's different modes of motion. These energy levels can be used to cool the ion chain by using the sideband frequencies.

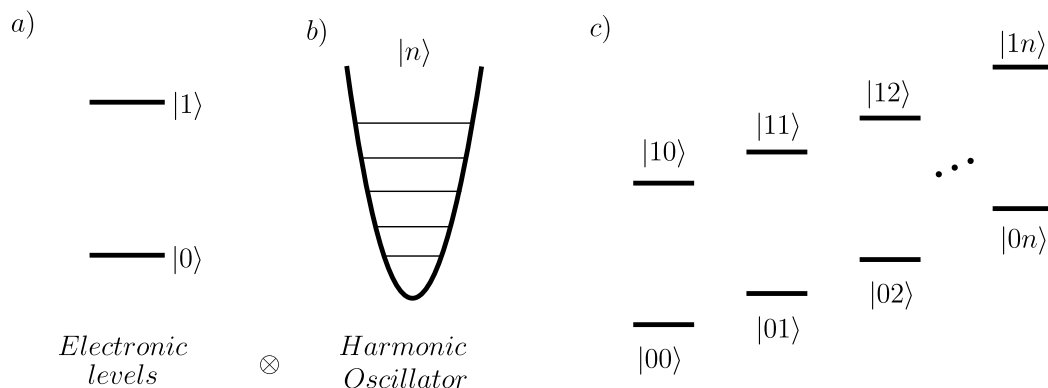
In this work, we focus on optical qubits, although most of the techniques and results also apply to the other types of qubit levels.

### 5.4 Qubit control

A fundamental part of quantum computing is the capability to control the state of a qubit. This includes the possibility of preparing the qubits in the desired state, realizing single-qubit operations, entangling operations between multiple qubits, and measuring qubits on a well-defined basis. As seen in the previous chapters, for universal quantum computation we need access to a two-qubit entangling gate and the capability of arbitrary single-qubit rotations, which can be approximated with the use of the Hadamard gate and the T gate.

This section gives an overview of the ion trap toolkit, including state preparation and measurement, and the implementation of single and two-qubit gates. For this, we first need a basic understanding of the energy landscape. The state of an ion in the ion trap can be described as the tensor product of the electronic state, in which we encode





**Figure 5.2: Energy levels of the ion trap.** The state of an ion in the ion trap can be written as a tensor product of **a)** the electronic levels, which typically encode the qubit information, and **b)** the energy levels of the harmonic oscillator, corresponding to the oscillation of the ion chain within the ion trap. **c)** The energy level scheme of a single trapped ion.

the qubits ( $|0\rangle$  and  $|1\rangle$ ), and the vibrational state of the COM axial mode, which can be described with a harmonic oscillator [286, 336, 337] (see Fig. 5.2).

Addressed laser beams are used to interact with the ions in the trap. To illustrate the interaction of the state with the laser, we can use the Hamiltonian of a two-level system interacting with a quantized harmonic oscillator through laser light. For a single trapped ion interacting with near-resonant laser light, and after applying the rotating wave approximation assuming that the laser detuning and Rabi frequency are much smaller than optical frequencies, as well as the Lamb-Dicke approximation for cold tightly bound ion strings, we obtain [286, 336]:

$$H = \hbar\Omega\{\sigma_+e^{-i(\Delta t-\phi)} + \sigma_-e^{i(\Delta t-\phi)} + i\eta(\sigma_+e^{-i(\Delta t-\phi)} - \sigma_-e^{i(\Delta t-\phi)})(ae^{-i\omega_t t} + a^\dagger e^{i\omega_t t})\}. \quad (5.1)$$

In this equation,  $a^\dagger$  and  $a$  are the creation and destruction operators for the vibrational modes,  $\sigma_\pm$  are the electronic raising and lowering operators.  $\Omega$  characterizes the strength of the laser field in terms of the Rabi frequency and  $\omega_t$  is the trap frequency.  $\phi$  is the phase of the field with respect to the atomic polarization,  $\Delta$  is the laser-atom detuning, and  $\eta$  is the Lamb-Dicke parameter.

## 5. ION TRAPS

---

There are three cases of particular interest with respect to the values of the laser detuning  $\Delta$ :  $\Delta = 0$  and  $\Delta = \pm\omega_t$ . After applying a second rotating wave approximation, assuming that only one transition is relevant at a time, and discarding time-dependent terms, we arrive at the following regimes:

1.  $\Delta = 0$  carrier transition, in which the qubit states are changed:

$$H^{carrier} = \hbar\Omega(\sigma_+e^{i\phi} + \sigma_-e^{-i\phi}). \quad (5.2)$$

2.  $\Delta = \omega_t$  blue sideband transition, in which we simultaneously excite the qubit state and create a phonon:

$$H^+ = i\hbar\Omega\eta(\sigma_+a^\dagger e^{i\phi} - \sigma_-ae^{-i\phi}). \quad (5.3)$$

3.  $\Delta = -\omega_t$  red sideband transition, in which we simultaneously excite the qubit state and destroy a phonon:

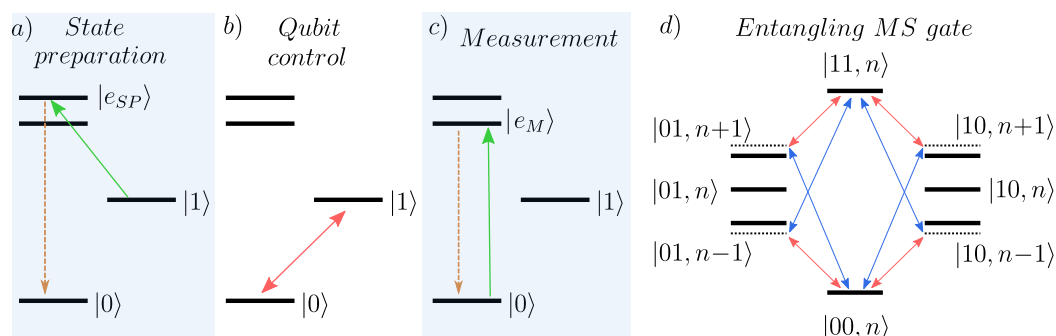
$$H^- = i\hbar\Omega\eta(\sigma_-a^\dagger e^{-i\phi} + \sigma_+ae^{i\phi}). \quad (5.4)$$

With this brief insight about the interactions between laser and ions, we can proceed to an overview of the main operations in the ion trap (Fig. 5.3).

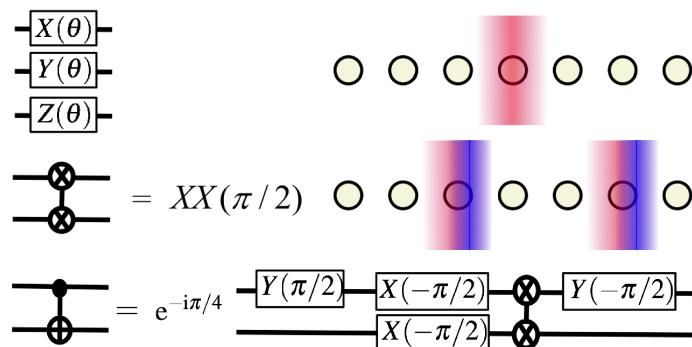
First, we can prepare states in  $|0\rangle$  with the help of a metastable excited state  $|e_{SP}\rangle$ . By driving the transition between the  $|1\rangle$  state and the metastable  $|e_{SP}\rangle$  state, we can prepare the state in  $|0\rangle$ , as the population in  $|e_{SP}\rangle$  quickly decays to the desired state.

Similarly, we can drive a transition between the  $|0\rangle$  state and a metastable state  $|e_M\rangle$  to measure the state of the qubit. As the decay from  $|e_M\rangle$  to the  $|0\rangle$  state emits photons, driving the transition will allow us to detect these photons, determining the result of the measurement. In contrast, the driving would not affect the state of an ion in  $|1\rangle$ , leaving the ion in a so-called *dark state*.

Single-qubit rotations can be implemented as well through addressed laser beams. Any linear combination of Pauli  $X$  and  $Y$  rotations can be achieved by driving Rabi oscillations between the two-qubit levels with the right phase. Pauli  $Z$  rotations can be achieved in several ways. A straightforward method to effectively implement a  $Z$  rotation of angle  $\phi$  is to shift all following rotations on that qubit by an angle  $-\phi$ .



**Figure 5.3: Operations in the ion trap.** **a)** State preparation can be achieved by driving the transition between the  $|1\rangle$  state and a metastable state  $|e_{SP}\rangle$ , which quickly decays to the  $|0\rangle$  state. **b)** Single-qubit operations can be implemented using addressed laser beams, that can drive Rabi oscillations between  $|0\rangle$  and  $|1\rangle$ . **c)** A qubit can be measured by driving the transition from the  $|0\rangle$  to a metastable state  $|e_M\rangle$ . The fast decay of  $|e_M\rangle$  scatters photons that can be captured with a detector. **d)** The entangling two-qubit Mølmer-Sørensen (MS) gate can be implemented using bichromatic laser beams slightly detuned from the blue and red sidebands [210, 338, 339]. The MS gate can also realize multi-qubit entangling operations between subsets of ions in the same ion chain.



**Figure 5.4: Universal Gate set for the static ion-string approach:** By using addressed laser beams, it is possible to apply arbitrary single-qubit rotations, and entangling two-qubit Mølmer-Sørensen (MS) gates [210, 338, 339] between any pair of ions.

Alternatively,  $Z$  rotations can be decomposed into  $X$  and  $Y$  rotations. Additionally, a far detuned laser beam can shift the energy difference between the two states  $|0\rangle$  and  $|1\rangle$  by  $\Delta E = \Omega^2/\Delta$  through an AC-Stark shift effect.

Finally, two-qubit entangling operations can be achieved by driving the two target ions with bichromatic laser light slightly detuned from the blue and red sidebands (see Fig. 5.3). In this way, it is possible to apply entangling operations between any pair of

## 5. ION TRAPS

---

ions located on the same ion string [210, 338, 339], which equips the static string ion trap with complete qubit connectivity by exploiting the common vibrational mode as a quantum data bus. Although it is possible to realize multi-qubit entangling operations involving any subset of ions in the trap, this operation is potentially problematic in the context of FT QEC, as a single fault in one of the ions could propagate to the rest of the ions involved in the gate. Thus, for the rest of this thesis, we focus only on the two-qubit MS gate.

This set of gates allow the implementation of the universal set of unitary operations  $\{H, T, CNOT\}$  [340] by using combinations of single-qubit rotations and the two-qubit entangling MS gate (Fig. 5.4). Furthermore, for the static-string approach with single-ion laser addressing, the system has full two-qubit connectivity with this gate set. This allows for a better connected system of qubits, which benefits the performance, reducing the number of gates required to reproduce a general circuit [302].

## Part II

# Research results



## Chapter 6

# Feasibility study of QEC with trapped ions

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS

---

In this chapter, we describe our work on the numerical analysis on the performance of near-term ion traps for QEC, published in [1]. We realize a feasibility study of the implementation of recent flag-based protocols for QEC in a linear ion trap, with a particular focus on the effects and measures against crosstalk errors.

The focus of this work is centered around the simulation of realistic error models, derived from microscopic models and first principles. We put special attention to the study of crosstalk errors, an error source that affects neighboring ions during the application of quantum gates. We study the impact of this error source, and the efficacy of suppression techniques based on refocusing pulse sequences. This study aims to find the regime for which the application of QEC is beneficial for the preservation of the logical information of a qubit, establishing target values for experimental implementations and identifying the most relevant problems that should be addressed.

We begin in Sec. 6.1 with an overview of the current context, including previous works and more recent developments in the field. We also specify the particular details of the FT protocol used in our simulations and the software platform used to perform them. Then, Sec. 6.2 describes the different error models that we use to describe the physical phenomena that affect the qubits on ion trap devices. This section includes our different models for crosstalk and our modeling for the imperfections on leakage repumping sequences (a method to bring back to the code space qubits that have leaked to states outside of the computational space). The damaging effects of crosstalk are analyzed in Sec. 6.3, where we also study the difference between the coherent and incoherent descriptions of crosstalk noise using numerical simulations. In Sec. 6.4, we describe the refocusing pulse schemes used to suppress crosstalk, and study analytically the potential reduction of noise that can be achieved. In Sec. 6.5, we show our numerical results of the feasibility study of FTQEC with ion traps, where we use extensive numerical simulations to reproduce the effect of our realistic error models and the capabilities of the FTQEC protocol to recover the initial quantum information. Finally, the chapter concludes with a summary of our work and an outlook for future extensions and applications of the work.

On the material presented in this section, I built on error models presented in previous works [149, 177, 178], contributing to refining the models and realizing the analysis of the effects of imperfections on the leakage repumping pulse sequence. For the numerical simulations, I built the code for the simulations using a starting baseline



written by Ciarán Ryan-Anderson, from which I revised and tested the implementation of the quantum circuits for the FTQEC protocols. I modified the code that simulates the error models adding new code for the simulation of the different models of crosstalk, leakage and leakage repumping. Using that code, I ran the different simulations and subsequent numerical analysis from which we obtained the results shown in this thesis and published in [1]. Additionally, I realized the analytical study of the crosstalk suppression using the refocusing pulse sequence.

## 6.1 Background

The work presented in this chapter builds on previous works in the field [149, 176–178], and is oriented towards the existing challenges of near-term quantum devices and the development and implementation of small scale QEC protocols, addressing different fronts. Overall, this feasibility study contributes to the development of the field by presenting a thorough analysis of flag-based color-code QEC performance in ion traps using the static single-string approach (see chapter 5.2). We use detailed and realistic error models that take into account various physical noise sources, which we believe is crucial to provide a realistic estimate of the expected performance of near-term trapped-ion experiments.

On one side, we study the error-correcting capabilities of a recently proposed QEC protocol [229], in which the 7-qubit color code [113], also known as the Steane code [214], is used in combination with a flag-based readout technique to implement the stabilizer measurement circuits fault tolerantly while minimizing the use of resources. The details of this protocol were described in the earlier section 3.2.2.1. Flag-based readout techniques [114, 229, 245, 246] are part of an intense recent activity aiming for the development of resource-efficient FT schemes and exploit present experimental capabilities [126, 215, 225, 241, 243, 244]. The implementation of the Steane code, the smallest instance of color codes, with the capability of a transversal implementation of the entire group of Clifford gates [113]. This code has proven to be a big focal point of recent research activity, as shown by recent breakthrough experiments in the realization of FTQEC [169, 171].

Secondly, we study the protocol from the point of view of ion traps, using realistic error models and a gate set corresponding to the single-string linear trap approach.

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS

---

This choice is motivated by recently demonstrated experimental capabilities [300, 301], which include the possibility to perform high-fidelity entangling gates addressed on specific subsets of ions. This leads to effective all-to-all connectivity for two-qubit entangling gates, as mentioned in chapter 5. Focusing on this platform, we use a realistic description of the noise based on a microscopic model of the different physical processes present in ion trap experiments, that builds on previous works [149, 176–178]. In this study, we place a particular focus on the effects of crosstalk, a noise source with damaging effects reported and studied by recent research [101, 107, 139, 179, 306, 341–344]. In ion traps, crosstalk noise occurs during the application of quantum gates, and stems from unwanted interactions between the target ions and the neighboring ions. In the context of FTQC, crosstalk noise can potentially break the FT character of FT circuit constructions, as they lead to undesired dynamics that violate either (or both) of two principles: spatial locality (i.e. gates should only affect the target qubits) and independence of local operations (i.e. the effect of a gate should be independent of other gates being applied to the system) [101, 107, 139, 341–343, 345]. This feasibility study, thus, provides a valuable contribution to the progress of QEC, assessing the viability of a particular FTQEC protocol on near-term ion trap devices. We use exhaustive modeling of the different error sources, which we believe is critical to provide realistic estimations of the expected performance.

Additionally, we study the use of refocusing pulse sequences to suppress the damaging effects of crosstalk gates on two-qubit entangling gates. In recent work, an alternative technique to minimize the effects of crosstalk has been proposed, based on a smart rearrangement of the qubits in the QEC circuit in order to minimize the most dangerous crosstalk interactions in the implementation of QECCs in the compass-code family of codes [344]. In general, multiple techniques have been proposed in the literature to mitigate errors at different levels, like the optimization of the gate implementation [101–106, 346, 347], the modification of the QEC circuit and codes [114, 229, 245, 246] and other advanced protocols [107, 108].

In our numerical simulations, we use the open-source software PECOS [181, 280], developed by Ciarán Ryan-Anderson. We implement incoherent simulations using the stabilizer simulator included in PECOS, and coherent simulations using the state-vector simulator from ProjectQ [281, 348] as a backend to PECOS. A more detailed description of the differences between both simulation procedures can be found in chapter 4. We

simulate the FTQEC protocol for parallel, flagged stabilizer readout presented by [229] and described in the background chapter 3.2.2.1.

## 6.2 Error model

In this section, we explain the complete error model used in our study. We assume a single ion species, trapped in a linear Paul trap, and operated by addressed laser beams. In particular, we focus on the  $^{40}\text{Ca}^+$  ion as an optical qubit, building on previous models [178]. The  $|0\rangle$  state is represented by the ground state  $|S_{1/2}, m_j = -1/2\rangle$  and the  $|1\rangle$  state, by the state  $|D_{5/2}, m_j = -1/2\rangle$ .

We consider the universal gate set described in chapter 5.4, and shown in figure 5.4. The set of operations allowed by the static-string ion trap device include single and two-qubit gates, as well as state preparation and measurements.

Understanding the set of operations used in the quantum circuit is crucial for developing the realistic error model used in the simulations. There are several noise sources that can affect the information stored in the qubits in the ion trap. We make a systematic analysis by subdividing these sources into 3 categories:

1. **Idle errors:** this category collects the different processes that affect the qubits over time when no gates are being applied. In particular, this includes dephasing, amplitude damping and leakage.
2. **Gate errors:** these errors originate from imperfections in the gates applied to the qubits. In this category, we include the following sources: single-qubit gates, two-qubit entangling gates, state preparation, and measurements.
3. **Crosstalk:** when applying the gates, a fraction of the laser light from the gate can affect neighboring ions, originating undesired noise on the qubits. This can happen in every gate, but the effects have a higher impact in the application of the MS gate, as we discuss in Sec. 6.2.3.

In the following sections, we describe briefly, for each error source, the physical background, and the model that we use in the simulations. Table 6.1 shows a summary of the anticipated values for the duration and fidelity of the gates and operations on trapped ions at the time of the publication of our work [1].

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS

---

Operation	Anticipated duration	Anticipated Infidelity
Two-qubit MS gate	$15\mu s$	$2 \cdot 10^{-4}$
One-qubit gate	$1\mu s$	$1 \cdot 10^{-5}$
Measurement	$30\mu s$	$1 \cdot 10^{-4}$
Qubit reset	$10\mu s$	$1 \cdot 10^{-4}$
Re-cooling	$100\mu s$	$\bar{n} < 0.1$
Leakage repumping	$20\mu s$	$2 \cdot 10^{-4}$

**Table 6.1: Extended trapped-ion QEC toolbox.** Description of expected near-term experimental trapped-ion capabilities for a QCCD approach to FT QEC (Values taken from Ref. [178]). These values correspond to a particular ion trap setup. Currently, better values for particular gates have already been achieved in dedicated experiments. For a more detailed review of the current best fidelities and gate durations, the reader can consult Table 1 in [283].

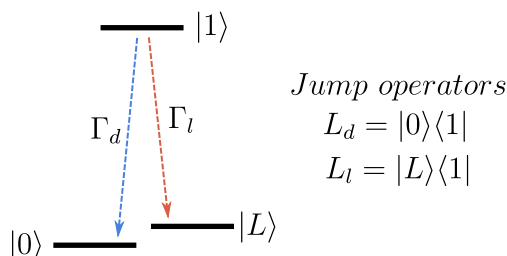
### 6.2.1 Idle errors

In this category, we group the three main processes that lead to noise on the idle ions in the ion trap: dephasing, amplitude damping and leakage. This type of noise affects the state of the qubits during the time in which no gate is being applied to them. We consider trapped-ion qubits encoded in an optical qubit transition (e.g.  $^{40}\text{Ca}^+$ , see chapter 5.4). To simulate them, we take into account the duration of the gates, shown in Table 6.1.

Dephasing and amplitude damping are two standard error channels described in the literature (see chapter 2.3). Dephasing is a process that causes decoherence due to, e.g. ambient fluctuating magnetic fields or laser frequency drifts, and in this work we model it as temporally and spatially uncorrelated. Amplitude damping is caused by the spontaneous decay from the metastable state  $|1\rangle$  to the ground state  $|0\rangle$ .

The process of leakage is similar to amplitude damping, and occurs when the state  $|1\rangle$  decays to the Zeeman sublevel  $|L\rangle = |S_{1/2}, m_j = +1/2\rangle$ , which lies outside of the qubit subspace, as opposed to the ground state  $|0\rangle = |S_{1/2}, m_j = -1/2\rangle$  (see Fig. 6.1). For the experimental system we study, the decay time  $T_1 = 1/\Gamma$  is around 1.1 seconds, and the fraction of leakage with respect to amplitude damping is  $\Gamma_l = \frac{4}{9}\Gamma_d$  [178], with  $\Gamma = \Gamma_d + \Gamma_l$ . To include leakage and amplitude-damping in the simulations, it is possible to fully simulate the process using Kraus operators (see chapter 2.3) and using an auxiliary classical bit to represent if a qubit lies on the computational subspace or in the leaked state  $|L\rangle$ . This approach has been described in previous works [178],

where the authors also described an implementation in terms of circuits identities to simulate the process. This modeling, however, requires the use of coherent simulations (i.e. state-vector simulations, see chapter 4). To simulate leakage and amplitude damping efficiently in stabilizer simulations, we need to find a way to approximate the channel using only Clifford operations. In our simulations, we approximate both channels using a Clifford approximation that I derived, coded and tested, building on previous models based on the approach presented in Ref. [268]. To understand it, let us first describe the exact dynamics of the noise.



**Figure 6.1: Spontaneous decay in leakage and amplitude damping.** The processes of leakage and amplitude damping represent the dynamics of the spontaneous decay from the state  $|1\rangle$  to the states  $|0\rangle$  and  $|L\rangle$ . This evolution can be described with the Lindblad jump operators  $L_d$  and  $L_l$  (see Eq. 6.1). The rates of decay for the experimental system we study follow a ratio of  $\Gamma_l/\Gamma_d = \frac{4}{9}$  [178].

Leakage and amplitude damping can be described with a Lindblad master equation with jump operators  $L_d = |0\rangle\langle 1|$  and  $L_l = |L\rangle\langle 1|$  and decay rates  $\Gamma_d$  and  $\Gamma_l$  (where the subindex  $d$  refers to the amplitude damping decay and the subindex  $l$  to the leakage decay):

$$\dot{\rho} = \sum_i \gamma_i (L_i \rho L_i^\dagger - \frac{1}{2} \{L_i^\dagger L_i, \rho\}) \quad (6.1)$$

$$= + \Gamma_d (|0\rangle\langle 1| \rho |1\rangle\langle 0| - \frac{1}{2} \{|1\rangle\langle 1|, \rho\}) \quad (6.2)$$

$$+ \Gamma_l (|L\rangle\langle 1| \rho |1\rangle\langle L| - \frac{1}{2} \{|1\rangle\langle 1|, \rho\}).$$

When integrated over a given time  $t$ , the dynamics of this process can be described exactly with a quantum channel with the following Kraus operators:

$$\varepsilon_{d,l}(\rho) = \begin{cases} K_0 = |0\rangle\langle 0| + |L\rangle\langle L| + \sqrt{1-p_l-p_d}|1\rangle\langle 1| \\ K_1 = \sqrt{p_d}|0\rangle\langle 1| \\ K_2 = \sqrt{p_l}|L\rangle\langle 1| \end{cases}, \quad (6.3)$$

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS

---

where  $p_l = \frac{\Gamma_l}{\Gamma}(1 - e^{-\Gamma t})$ ,  $p_d = \frac{\Gamma_d}{\Gamma}(1 - e^{-\Gamma t})$ . We can see how this channel would be problematic in stabilizer simulations, as the operator  $K_0$  cannot be applied using Clifford operations. Thus, we need to find an approximation that only requires the use of unitary gates within the Clifford group and measurement-induced translations [268]. For this, we split the operator  $K_0$  into three operators  $K'_0$ ,  $K'_1$  and  $K'_2$ , leading to the following Kraus map for the approximated leakage and amplitude damping channel  $\varepsilon'_{d,l}$ :

$$\varepsilon'_{d,l}(\rho) = \begin{cases} K'_0 & = \sqrt{1 - p_l - p_d} \mathbb{1} \\ K'_1 & = \sqrt{p_l + p_d} |0\rangle \langle 0| \\ K'_2 & = \sqrt{p_l + p_d} |L\rangle \langle L| \\ K'_3 & = \sqrt{p_d} |0\rangle \langle 1| \\ K'_4 & = \sqrt{p_l} |L\rangle \langle 1|. \end{cases} \quad (6.4)$$

This approximation can be efficiently implemented in stabilizer simulations, and works in the following steps. First, we check that the state is not in  $|0\rangle$  or  $|L\rangle$  already<sup>1</sup>. We found this check to be particularly relevant, as a qubit in the simulation could otherwise undergo a non-physical transition from  $|0\rangle$  to  $|L\rangle$ . This problem does not appear when considering amplitude damping only. Then, we use a random number to sample the probability distribution of a decay event with  $p = 1 - \exp(-\Gamma\Delta t)$ , where  $\Delta t$  denotes the duration of the time step. If a decay event is observed, then a second random number determines if the qubit decays to the leaked state  $|L\rangle$  or to the ground state  $|0\rangle$ <sup>2</sup>.

To evaluate the validity of the approximation, we employ the Hilbert distance, defined for two states  $\rho_A$  and  $\rho_B$  as

$$D_H(\rho_A, \rho_B) = \text{Tr}((\rho_A - \rho_B)^\dagger(\rho_A - \rho_B)). \quad (6.5)$$

With this measure, we checked that for any initial state  $\rho_0 = |\psi\rangle \langle \psi|$ , with  $|\psi\rangle$  being a state in the computational basis, the following condition was fulfilled:

$$D_H(\rho_0, \varepsilon_{d,l}(\rho_0)) \leq D_H(\rho_0, \varepsilon'_{d,l}(\rho_0)). \quad (6.6)$$

---

<sup>1</sup>That is, considering the probability of the qubit being in one of those states, a test that we can only do without collapsing the wavefunction in a simulation.

<sup>2</sup>According to the electronic branching ratio  $\Gamma_l = \frac{4}{9}\Gamma_d$ , that determines the proportion of leakage and amplitude damping [178].

This ensures that our model represents an upper bound to the error induced to the system, and thus remains conservative. A more detailed analysis for the amplitude damping channel without leakage was presented in [268].

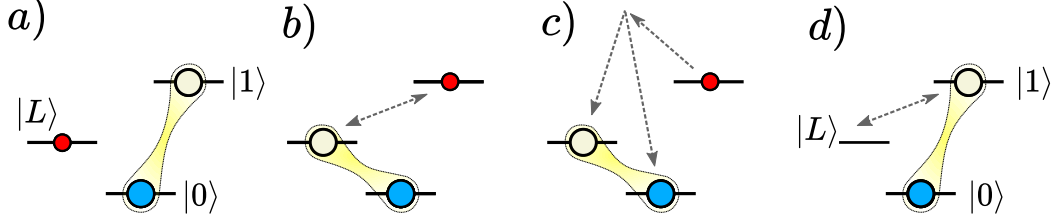
The simulation of leaked qubits involves an additional aspect with respect to other noise sources: we need to define how a qubit in the leaked state evolves when quantum gates are applied, and how this affects the rest of the qubits involved in the gates. In our simulations, we follow a similar procedure as in Ref. [349]. If a qubit falls into the leaked state, it is projected to the  $|0\rangle$  state, and a classical variable is used to record that the qubit has leaked. If leaked qubits are exposed to laser fields that aim at realizing single-qubit unitaries, they do not apply any evolution to the state, as the laser fields are off-resonant and do not bring the state back to the computational qubit subspace. If a leaked qubit would take part in an MS gate, then the lasers are still near-resonant with the sidebands of the unleaked qubit. However, the unleaked qubit will evolve in a closed trajectory in phase space equivalent, to leading order, to the identity operator. Furthermore, when a non-leaked qubit takes part in an MS entangling gate with a leaked qubit, no additional noise is applied to the non-leaked qubit, as discussed in [178]. Measurement and state preparation operations reset leaked qubits to the  $|0\rangle$  state since both operations are followed or realized by optical pumping into  $|0\rangle$ . Thus, during measurement and state preparation, the value of the classical variable used to track leaked qubits is modified to indicate that qubits undergoing these operations are no longer leaked.

However, it might be desirable to bring leaked qubits back to the computational subspace without having to measure them. Although it is possible to detect the presence of leakage using quantum non-demolition measurements<sup>1</sup> [350, 351], it is desirable to use a correction protocol that will bring a leaked qubit back to the computational subspace, but that would leave the state of the qubit unaffected otherwise. The repumping sequence described in [178] (see Fig. 6.2) fulfills this purpose, and it can reset a leaked state reset to  $|0\rangle$  while leaving computational states largely unaffected. This is achieved by first spectroscopically decoupling (*hiding*) the population in the excited state by

---

<sup>1</sup>We can use an ancilla qubit to detect leakage in a qubit, as the MS gate acts as identity when one of the qubits is leaked. Thus, we can design a protocol to use this property to detect leakage without disturbing the state of the qubit [350, 351].

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS



**Figure 6.2: Leakage repumping sequence:** **a)** initial state, where the initial populations are represented by the coloured circles. **b)** We apply a  $\pi$  pulse between the states  $|1\rangle$  and  $|L\rangle$ , swapping the populations. **c)** We apply a pulse between the  $|1\rangle$  state and a metastable state, that will rapidly decay to either  $|L\rangle$  or  $|0\rangle$ , with ratios that we assume to be equal for this simplified model. **d)** A final  $\pi$  pulse swaps the population in  $|L\rangle$  to  $|1\rangle$ .

swapping it to the leaked state. Then, we apply a pulse that would bring any population initially in the leaked state (now occupying the excited state) to a metastable state that will rapidly decay to either the leaked state or the ground state. Finally, the last pulse will bring the population in the leaked state, where the initial population in  $|1\rangle$  was hidden, back to the excited state. A schematic representation of the process can be seen in Fig. 6.2.

However, the leakage repumping protocol requires the use of a pulse sequence that can be faulty and lead to additional errors. In our work [1], we develop a single model that simulates as a single noise channel the different errors that can occur during the leakage repumping sequence. In the ideal scenario, the total effect of this pulse sequence is equivalent to an identity operator if the qubit was not leaked, and both the population and coherences remain protected. If the qubit had leaked, it would be reinitialized to either the ground or the excited state. In that case, the next round of QEC is expected to correct that error. Let us follow the evolution of the initial populations  $\rho_1$ ,  $\rho_0$  and  $\rho_L$  through the ideal pulse sequence in Fig. 6.2:

$$\rho_{11}^{(a)} = \rho_1 \rightarrow \rho_{11}^{(b)} = \rho_L \rightarrow \rho_{11}^{(c)} = 0 \rightarrow \rho_{11}^{(d)} = \rho_1 + \rho_L/2 \quad (6.7)$$

$$\rho_{LL}^{(a)} = \rho_L \rightarrow \rho_{LL}^{(b)} = \rho_1 \rightarrow \rho_{LL}^{(c)} = \rho_1 + \rho_L/2 \rightarrow \rho_{LL}^{(d)} = 0 \quad (6.8)$$

$$\rho_{00}^{(a)} = \rho_0 \rightarrow \rho_{00}^{(b)} = \rho_0 \rightarrow \rho_{00}^{(c)} = \rho_0 + \rho_L/2 \rightarrow \rho_{00}^{(d)} = \rho_0 + \rho_L/2 \quad (6.9)$$



In practice, the leakage repumping sequence relies on single-qubit pulses that are susceptible to errors. We can estimate the errors that can happen during the process and the final result on the qubit by checking the effects of single errors in the pulses. For this simple model, we consider that each of the single-qubit gates involved in the sequence can fail with probability  $p$ , effectively leaving the populations unchanged. Assuming  $p \ll 1$ ,  $1 - p \simeq 1$  and considering only terms up to  $\mathcal{O}(p)$ , the populations after each pulse are:

$$\rho_{11}^{(b)} = \rho_L + p\rho_1 \rightarrow \rho_{11}^{(c)} = p\rho_L \rightarrow \rho_{11}^{(d)} = (1 + \frac{p}{2})\rho_1 + (p + \frac{1}{2})\rho_L \quad (6.10)$$

$$\rho_{LL}^{(b)} = \rho_1 + p\rho_L \rightarrow \rho_{LL}^{(c)} = \frac{2+p}{2}\rho_1 + \frac{1+2p}{2}\rho_L \rightarrow \rho_{LL}^{(d)} = p(\rho_1 + \frac{3}{2}\rho_L) \quad (6.11)$$

$$\rho_{00}^{(b)} = \rho_0 \rightarrow \rho_{00}^{(c)} = \rho_0 + \frac{pL}{2} + \frac{p}{2}\rho_1 \rightarrow \rho_{11}^{(d)} = \rho_0 + \frac{pL}{2} + \frac{p}{2}\rho_1 \quad (6.12)$$

From these populations, the final  $\rho_{LL}^{(d)}$  corresponds to leakage errors. In  $\rho_{11}^{(d)}$ , the term in  $\frac{p}{2}\rho_1$  corresponds to dephasing, as that population is the result of the incoherent decay after pulse  $c$ . Finally, the term  $\frac{p}{2}\rho_1$  in  $\rho_{00}^{(d)}$  has the same effect as an amplitude damping decay. We can summarize the possible outcomes in a single error channel for the entire sequence that can be easily implemented in the numerical simulations:

1. If the qubit was leaked: we have a probability of twice the single gate error probability  $3p/2$  to bring the qubit back to the leaked state.
2. If the qubit was not leaked: then, with probability  $p$  it can end in the leaked state, with  $p/2$  it can suffer amplitude damping and with  $p/2$  it can suffer dephasing.

### 6.2.2 Operational noise

Under this category, we study the effects of imperfections in the operations applied to the ions. Most error models on this topic have been widely studied in the literature and are well known by the scientific community, and have been described in chapter 2. In addition to the noise channels described in this section, we consider crosstalk noise in all operations, but these effects are studied in the following section.

1. **Single-qubit operations.** Single-qubit gates are implemented by addressed laser beams. The main noise sources are the fluctuations in the laser beam intensity and phase. The fluctuations in the laser beam intensity result in an over or

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS

---

under rotation in the axis of rotation of the gate ( $X, Y, Z$ ). The fluctuations on the phase of the gate affect the  $X$  and  $Y$  gates, as it changes the axis of rotation to a combination of both.

In our simulations, we model the errors in all single-qubit gates as a symmetric depolarizing channel, which is the most widely used model for this type of gates (see chapter 2.3):

$$\rho = (1 - p)\rho + p\frac{1}{3}(X\rho X + Z\rho Z + Y\rho Y). \quad (6.13)$$

2. **Two-qubit gates.** The entangling, two-qubit MS gate is implemented by addressed laser beams on the two qubits involved. The MS scheme creates a state-dependent force by exploiting the laser-ion interaction in the regime of resolved phonon sidebands. During this process, the qubit-phonon dynamics can affect the result of the gate [347]. Therefore, the motional state of the ion string is an additional source of error. This can lead to motional errors, like a residual phonon-spin entanglement and fluctuations of the Rabi frequencies due to Debye-Waller (thermal) effects. Other sources of error can affect the qubits, like dephasing and fluctuations of the laser intensity and phase.

The model used in our simulations is represented by the map  $D_{MS}(\rho)$ , which was obtained using a structured microscopic error channel that differs from the standard depolarizing noise. This model incorporates the effects from different physical sources and imperfections in the gate, including off-resonant carriers, intensity and phase fluctuations in the lasers, or Debye-Waller factors and residual qubit-phonon entanglement for the additional vibrational modes [1, 178]. The model, developed in [178], was then tailored with parameters matching the ion-trap experiment and used to obtain the error model for the noisy MS gate. The map  $D_{MS}(\rho)$  admits a Kraus decomposition (see chapter 2.3) after retaining only the most-likely operators that stem from the process-tomography numerical reconstruction of the microscopic quantum channel.

$$\rho \rightarrow D(\rho) = \sum_n K_n \rho K_n^\dagger. \quad (6.14)$$

To simulate the noisy MS gate, we apply the ideal unitary, followed by the error channel described by the Kraus operators  $K_i$  obtained from the model:

$$D_{MS}(\rho) = \begin{cases} K_0 = \sqrt{(1-p)} & \mathbb{1}_4 \\ K_1 = \sqrt{p \cdot 0.80} & X_1 X_2 \\ K_2 = \sqrt{p \cdot 0.05} & Y_1 \\ K_3 = \sqrt{p \cdot 0.05} & Y_2 \\ K_4 = \sqrt{p \cdot 0.05} & X_1 Z_2 \\ K_5 = \sqrt{p \cdot 0.05} & Z_1 X_2 \end{cases} . \quad (6.15)$$

3. **State preparation and measurement errors.** Within the ion-trap toolbox, the state of a qubit can be prepared or reset to the  $|0\rangle$  state at any point during the computation, and can be measured in the computational basis (see chapter 5). We model the noise channel of these operations by incoherently applying a Pauli  $X$  error after state preparation and before the measurement gates. We refine the model by including imperfections in the state preparation process that can lead to a leakage error.

### 6.2.3 Crosstalk

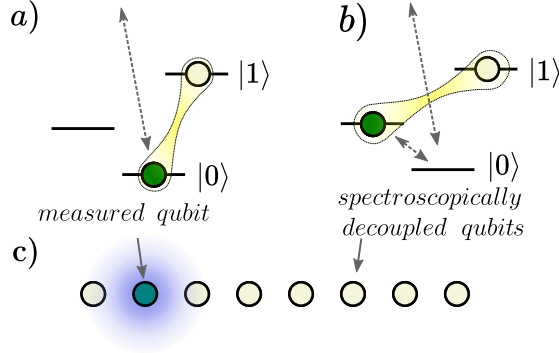
Crosstalk errors correspond to noise processes caused by a gate that affects qubit that were not intentionally involved in the gate operation. In ion traps, where gates are applied by addressed optics and laser beams, crosstalk originates from laser light oriented to the target qubit affecting its neighbors in the trap. This happens due to the finite size of the laser spot and the close position of the ions in the trap, and affects both single and two-qubit gates.

In single-qubit gates, this noise can be significantly suppressed by techniques such as composite sequences [352] or dynamical decoupling [102], and is thus neglected in our numerical simulations [1].

During the qubit measurement, crosstalk can also occur due to the scattering of photons from the measured ions, which can interact with the rest of the spectator ions in the trap. This unwanted process can be prevented by applying spectroscopic decoupling pulses to nearby spectator ions [1, 147], such that the scattered photons become off-resonant (see Fig. 6.3).

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS

---



**Figure 6.3: Qubit measurement and spectroscopic decoupling.** **a)** During the measurement, a laser drives a transition from the  $|0\rangle$  state to a metastable state with a fast decay. If the  $|0\rangle$  state is measured, the ion scatters photons in all directions, potentially affecting the neighboring ions. **b)** To avoid this crosstalk noise, spectroscopic decoupling pulses can shift the population in the  $|0\rangle$  state to a different energy level without a resonant transition at the frequency of the scattered photons. The original state can be recovered with another pulse sequence, preserving the original superposition. **c)** Sketch of the ion string.

Since the effects of crosstalk on the single-qubit gates and measurement operations are well under control in the experimental devices and can be effectively neglected, in this work we focus on the effects of crosstalk that stem from the two-qubit entangling MS gate, and from here onwards we consider crosstalk exclusively on two-qubit gates.

On two-qubit gates crosstalk plays a much more critical role, as it involves a multi-qubit operation that can generate entanglement between unwanted pairs of qubits. This type of noise can be very harmful to the performance of the ion trap, and potentially break fault tolerance [101, 107, 139, 342, 343, 345]. In our work, we present two different models for crosstalk errors: the *entangling model* and the *Stark-shift model*.

In the entangling crosstalk model, we assume that the leading order effect of crosstalk is a partial application of the MS gate rotation on every pair of gate and neighbor ions, which stems from the fact that the neighboring ions would receive a fraction of the laser amplitude, thus reducing the frequency of the rotation applied by the gate. This imperfect focusing of the laser beams affects the neighboring ions due to two main sources. First, the beams will couple to the spectator ions with a residual relative Rabi frequency. Second, this imperfect gate introduces a collective spin operator that involves both the target and the neighboring ions with residual light coupling. The resulting effect is the introduction of a spin-spin interaction involving the neighboring

ions. For an MS gate rotation of an angle  $\theta$  (e.g.  $\theta = \pi/2$  for a fully-entangling MS gate), the leading order effect can be modelled as an  $X_g X_n$  rotation of angle  $\theta_c$  between all pairs of target qubits  $g$  and neighbor qubits  $n$ . The rotation angle will be a fraction  $\epsilon_{CT}$  of the original MS gate rotation  $\theta$ , and depends on the residual intensity of light  $I_n$  on the spectator ions, which will be a fraction of the intensity of the gate in the target ions  $I_g$ :

$$\theta_c = \epsilon_{CT}\theta \simeq \theta \cdot \sqrt{\frac{I_n}{\langle I_g \rangle}}. \quad (6.16)$$

The details of the microscopic derivation of this model can be found in [1]. The resulting noise model for an MS gate of angle  $\theta$  is a unitary evolution that can be written as:

$$E(\theta)_{\text{crosstalk}} = \prod_{\substack{g \in G, \\ N \in \text{neig}(G)}} \exp\left(-i \epsilon_{CT} \frac{\theta}{2} X_g X_N\right). \quad (6.17)$$

where  $G = \{i, j\}$  is the set of gate qubits the ideal MS gate acts on,  $\text{neig}(G)$  is the set of qubits that are neighbors to  $i$  and/or  $j$  but do not include  $i$  or  $j$ , and  $\epsilon_{CT} = \Omega_n/\Omega$  depends on the fraction of the light intensity illuminating the spectator qubits with respect to the intensity addressed to the active qubits. In particular,  $\Omega$  is the Rabi frequency for the two-ion MS gate and  $\Omega_n$  is the relative Rabi frequency of the residual light on the neighboring ions. We neglect any effects of crosstalk beyond nearest neighbors, as we assume a sufficiently narrow Gaussian profile of the laser light, with an exponentially suppressed electric field away from the ions.

For a single pair of a gate qubit  $g$  and a neighboring qubit  $n$ , the crosstalk error channel is  $\rho \rightarrow D_{2q}^{\text{ct}}(\rho)$ , where  $D_{2q}^{\text{ct}}(\rho)$  is a coherent evolution that can be written as

$$D_{2q}^{\text{ct}}(\rho) = \cos^2\left(\frac{\theta\epsilon_{CT}}{2}\right) \rho + \sin^2\left(\frac{\theta\epsilon_{CT}}{2}\right) X_g X_n \rho X_g X_n + i\frac{1}{2} \sin(\theta\epsilon_{CT}) [\rho, X_g X_n]. \quad (6.18)$$

In order to implement the crosstalk error channel in stabilizer simulations, we need to find an approximation using only Clifford operations. To this end, we use the Pauli Twirling Approximation (PTA) method, a simple yet effective technique to obtain an approximation that can be efficiently simulated within stabilizer simulations [353, 354]. To obtain the PTA of a given process  $\Lambda(\rho)$ , we need to apply a *twirling* transformation.

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS

---

The twirling of a process over a set of operators  $\mathcal{A} = \{A_k\}_{k=1}^K$  is defined by the following operation

$$\tilde{\Lambda}(\rho) = \frac{1}{K} \sum_{A \in \mathcal{A}} A^\dagger \Lambda(A\rho A^\dagger) A, \quad (6.19)$$

with  $\tilde{\Lambda}(\rho)$  being called the twirled process. When twirling over the  $n$ -qubit Pauli basis  $\mathcal{A}_n = \{\mathbb{1}, X, Y, Z\}^{\otimes n}$ , the twirled process  $\tilde{\Lambda}(\rho)$  is diagonal in the Pauli basis,

$$\tilde{\Lambda}(\rho) = \sum_{A \in \mathcal{A}_n} p_A A \rho A^\dagger, \quad (6.20)$$

where the  $p_A$  are probabilities that can be directly measured experimentally without complete process tomography, which makes it practical for efficient model construction [353, 355]. PTA has proven to be a reliable method to simulate error channels in noisy, low-distance QEC circuits [354, 356], which we further test in Sec. 6.3 through numerical simulations.

When applying the twirling transformation of Eq. 6.19 to the coherent crosstalk model in Eq. 6.18 over the  $n$ -qubit Pauli basis  $\mathcal{A}_n = \{\mathbb{1}, X, Y, Z\}^{\otimes n}$ , the coherent term gets cancelled, and we obtain the PTA of the crosstalk channel:

$$\tilde{D}_{2q}^{\text{ct}}(\rho) = \begin{cases} K_{c,0} = \sqrt{1-p_c} \mathbb{1}, \\ K_{c,1} = \sqrt{p_c} X_g X_n. \end{cases} \quad (6.21)$$

From this equation, we can define the probability of a single crosstalk event as  $p_c = \sin^2(\epsilon_{CT} \theta/2)$ , an event that leads to a correlated 2-qubit bit-flip error on a pair of active-neighbor qubits. The approximated channel described by Eq. 6.21 defines a stochastic version of the crosstalk channel that we can use in stabilizer simulations.

For an ion located in between two gate ions, there will be twice the amount of residual light on it. This increases the probability of crosstalk effectively by a factor of 4, as it doubles the angle of the rotation [1].

Crosstalk errors of this type are an inherently coherent source of noise, as it applies an entangling rotation to every pair  $g$ - $n$ . Therefore, we model it as a coherent rotation in the state-vector simulations that simulate the complete wavefunction of the system using Eq. 6.18. This operation, however, is not allowed in stabilizer simulations, which requires us to use the alternative description shown in Eq. 6.21. This has some important implications for the performance of QEC, as coherent crosstalk can interfere

coherently and build up, leading to an increased effective error rate. The study of the difference between the coherent and incoherent descriptions is one of the focal points of this work, and is explored in the subsequent sections of this thesis.

In addition to the entangling crosstalk model, we consider an alternative implementation of the addressed entangling MS gate motivated by the work from experimental groups [357]. The main idea is to apply a power imbalance between the intensities of the bichromatic light fields that couple the red and blue phonon sidebands (see Fig. 5.3). The difference in intensities leads to an ac-Stark shift that differs for active and spectator ions. With adequate tuning, the gate qubits will experience a two-photon resonance and evolve following the ideal MS gate dynamics, while the spectator ions will be effectively exposed to a weak and far off-resonant light. This setting, thus, avoids the unwanted entangling interactions between spectator and active ions observed in our first crosstalk model.

In this *Stark-shift* crosstalk model, the leading-order effect of the residual crosstalk is an ac-Stark shift on the spectator ions, resulting in small unwanted but systematic  $Z$ -rotations, which can be described as

$$E_{\text{crosstalk}}^{(2)} = \prod_{n \in \text{neig}(G)} \exp\left(-i \mu_n \frac{\pi}{4} Z_n\right), \quad (6.22)$$

where we use the same notations as in Eq. (6.17) and introduce  $\mu_n$  as the residual ac-Stark shift from the off-resonant sidebands that act on the spectator ions due to crosstalk. The microscopic derivation of this model can be found in [1].

For the stabilizer simulations, we apply the PTA on Eq. 6.22, which leads to a single-qubit incoherent dephasing channel. The effects on each of the neighboring spectator qubits using this approximation is described by the Kraus operators

$$K_0 = \sqrt{1 - p_c} \mathbb{1}_n, \quad K_1 = \sqrt{p_c} Z_n, \quad (6.23)$$

with  $p_c = \sin^2(\mu_n \pi/4)$ .

The effects of this crosstalk are less harmful to the computation, as they do not create entanglement between unintended pairs of qubits. In addition, knowledge about the exact intensity of this noise channel can allow for a direct compensation using single-qubit gates. In Sec. 6.5, we study the effects of uncompensated Stark-shift crosstalk on

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS

---

the performance of QEC using numerical simulations.

### 6.3 Adversity of crosstalk on QEC

#### 6.3.1 Coherent vs. incoherent crosstalk

In this section, we study the impact of the entangling crosstalk noise on the performance of the Steane code, using the FT parallel flagged readout scheme [229] described in 3.2.2.1. For these simulations, we for the moment neglect all other noise sources and focus on the effects of crosstalk, which avoids unnecessary complications for the study of the difference between the coherent and incoherent descriptions of crosstalk.

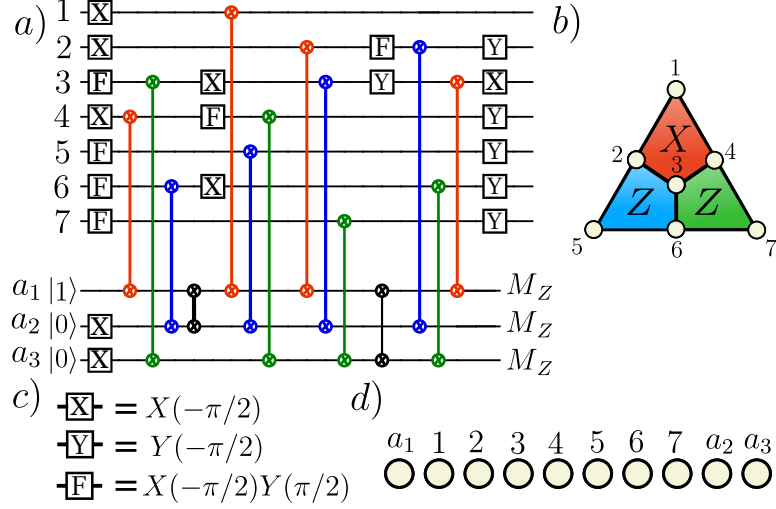
We consider a QEC protocol consisting of individual rounds of state preparation, stabilizer readout, recovery operation, and final state measurement. A round begins with a FT state preparation using a single ancilla [178], followed by a round of FT stabilizer measurement using the previously described flag scheme (see Fig. 6.4). At this moment, if an error is detected, we realize a second round of stabilizer measurement without flags. After the stabilizer measurements, we apply a recovery operation based on the information obtained from the stabilizer measurements, using a lookup table<sup>1</sup>. Finally, we measure all the qubits to check if a logical error has happened, applying a classical round of error correction if needed. A sketch of this protocol can be seen in Fig. 6.5, which shows the circuits with CNOTs for a more intuitive representation. In our simulations, we compile the circuits into MS gates, as shown in Fig. 6.4. Notably, this protocol has been implemented in recent trapped ion experiments [171], which highlights the relevance of this study.

We simulate multiple iterations of this experiment to estimate the logical error rate of the circuit for each set of error parameters. To estimate the error rates corresponding to coherent crosstalk errors, we perform state-vector simulations of the QEC protocol, using the crosstalk descriptions given in Eq. (6.17). Then, to compare with the incoherent version of crosstalk, we run simulations using the same protocol and parameters with stabilizer simulations, using the incoherent crosstalk model described in Eq. (6.21). Each run of the QEC protocol results in either a success or failure event. The estimate for the logical error rate is obtained as the average over many Monte Carlo samples. To

---

<sup>1</sup>A precomputed table where we find the best operation according to the syndrome and our knowledge on the noise model, see e.g. Fig. 3.1, Ch. 3.3.



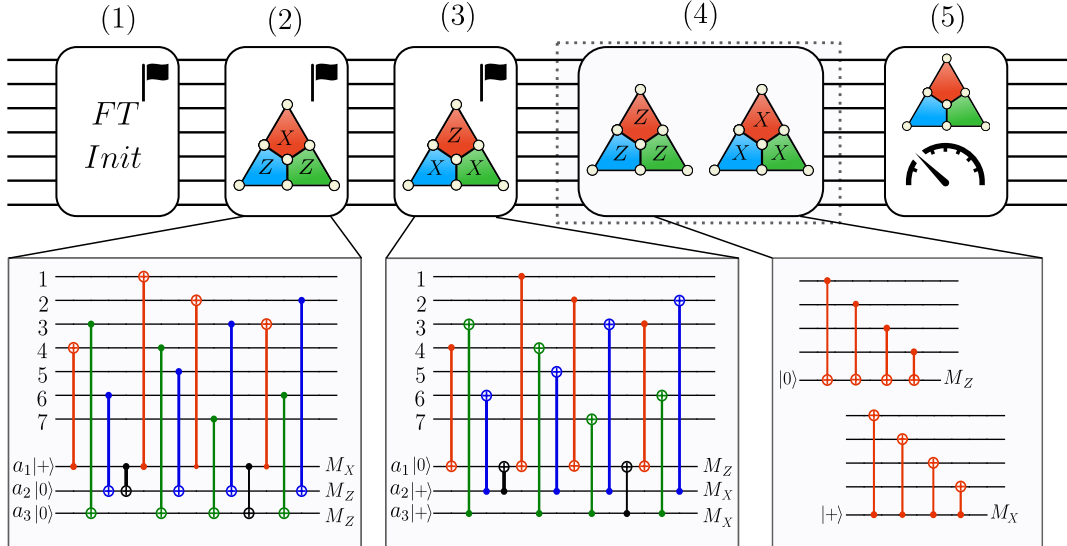


**Figure 6.4: Circuits for parallel syndrome extraction:** a) Circuit for the parallel and FT measurement of the stabilizers  $S_x^{(1)}$ ,  $S_z^{(2)}$  and  $S_z^{(3)}$  via ancillae  $a_1$ ,  $a_2$  and  $a_3$ , respectively, as shown in b). The circuit, presented in [229] with CNOTs, has been compiled into MS gates with the addition of single-qubit gates, shown in c). A similar circuit can be used to measure fault tolerantly the stabilizers  $S_z^{(1)}$ ,  $S_x^{(2)}$  and  $S_x^{(3)}$ . Using those two circuits, it is possible to measure all 6 stabilizers with three ancillae in only two sets of operations. d) Sketch of the order of the qubits in the ion string.

obtain the error bar of each estimate, we use the error from the binomial distribution  $err = \sqrt{p_{log}(1-p_{log})/n}$ . The dependence of the error on the logical error rate  $p_{log}$  implies that a larger number of samples is required for the low  $p_{log}$  regime. Therefore, we use a different number of samples for each point in the figures shown in this chapter, ranging from  $10^5$  to  $10^7$  samples per point, until the error bars are smaller than the markers used in the plots.

The error model from the simulations of Fig. 6.6 contains only crosstalk events. For the coherent simulations, this implies the application of additional coherent rotations after every MS gate. Since these events are systematic errors, the only stochastic process in this simulation is the measuring process, where the wave function is partially collapsed. The total number of measurement events for the worst case scenario is only 12 (measuring the 6 stabilisers twice), so it is possible to simulate individually each measurement outcome path and compute analytically from the wave function the resulting logical error probability. To compute the total logical error probability, we can sum the result of each *path*, weighted by the probability of that measurement

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS



**Figure 6.5: QEC protocol:** the QEC round used to estimate the logical error probability works as follows: first, we initialize the state using a FT protocol **(1)** using a single ancilla as a flag [178]. If the flag is raised, the state is discarded, and we restart the protocol. Then, we begin the syndrome extraction protocol: **(2)** parallel flagged syndrome extraction of the stabilizers  $S_x^{(1)}$ ,  $S_z^{(2)}$  and  $S_z^{(3)}$  (circuit redrawn from Ref. [229]). If no flag is raised, we proceed to **(3)** and measure the stabilizers  $S_z^{(1)}$ ,  $S_x^{(2)}$  and  $S_x^{(3)}$  using parallel flagged readout. If any flag was raised during **(2)** or **(3)**, we realize a second full round of stabilizer readout without flags. Using three ancillae, we can measure simultaneously groups of three stabilizers. Finally, **(5)** the state is measured. We can apply the recovery operation corresponding to the syndrome and check if the protocol succeeded or failed. In the figure, the circuits are shown using CNOT gates. For the simulations, we compiled the circuits into MS gates to adapt them for the trapped-ion universal gate set. An example is shown in Fig. 6.4 for the circuit in **(2)**.

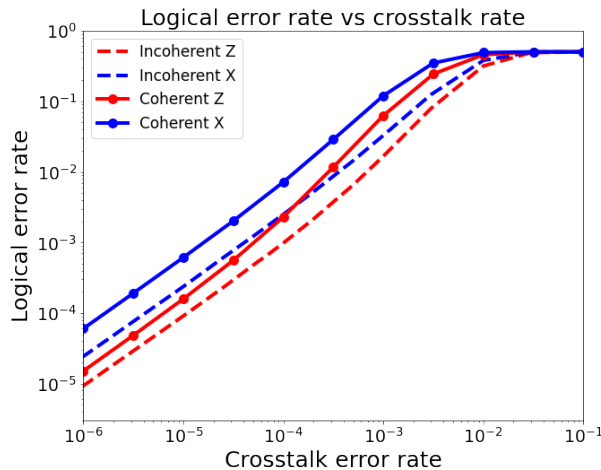
combination:

$$p_{log} = \sum_j \prod_i^{2^{12}} m_i^{(j)} p_L(\{m_i\}_j) \quad (6.24)$$

Here,  $m_i$  is the outcome of a measurement in the  $i^{th}$  measurement,  $\{m_i\}_j$  is a combination of measurement outcomes (or measurement path) and  $p_L(\{m_i\}_j)$  is the logical error probability for the resulting state after path  $j$ . Following this alternative, exhaustive sampling of all measurement outcomes, we can accurately estimate the logical error probability for a given crosstalk value using only  $2^{12}$  simulations, independently of the crosstalk error amplitude.

The simulations show that the logical error rates for coherent crosstalk have the

same order of magnitude but are larger than the logical error rates obtained from incoherent simulations (see Fig. 6.6). To obtain an estimation of the different scaling in the low crosstalk regime (in particular, for  $p_c < 10^{-4}$ ), we use a linear fit, which shows a linear scaling of the logical error rate with the crosstalk error rate. Interestingly, we find that the logical error rate for the coherent version of crosstalk is a factor of  $3.0 \pm 0.3$  larger than its incoherent counterpart. This analysis indicates that the incoherent approximation underestimates the expected logical error rates, which leads to an overestimation of the QEC capabilities.



**Figure 6.6: Logical error rates for coherent and incoherent crosstalk errors.** Here, we estimate the logical error rate using a limited error model that includes crosstalk errors only and set all other error sources to zero. We measure the logical- $X$  (logical- $Z$ ) error rate by simulating a QEC round on the state  $|+\rangle_L$  ( $|0\rangle_L$ ) (colours blue and red in the figure, respectively). We compare the results of simulating the crosstalk errors coherently (using state vector simulations and the crosstalk model on Eq. 6.17, solid lines) or incoherently (using Pauli propagation simulations and the crosstalk error model on Eq. 6.21, dashed lines). The results show that the logical error rates for coherent simulations are worse than the incoherent version by a factor of about 3, but this difference remains approximately constant as crosstalk increases. The error bars for the points are smaller than the size of the marker.

### 6.3.2 Impact of entangling crosstalk noise on QEC performance

Once we have studied the isolated effects of crosstalk and compared its coherent and incoherent approximations, we can analyze the impact of crosstalk on the performance of QEC when combined with the rest of the noise sources described in the previous section. We analyze the performance of the QEC protocol under the full error model

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS

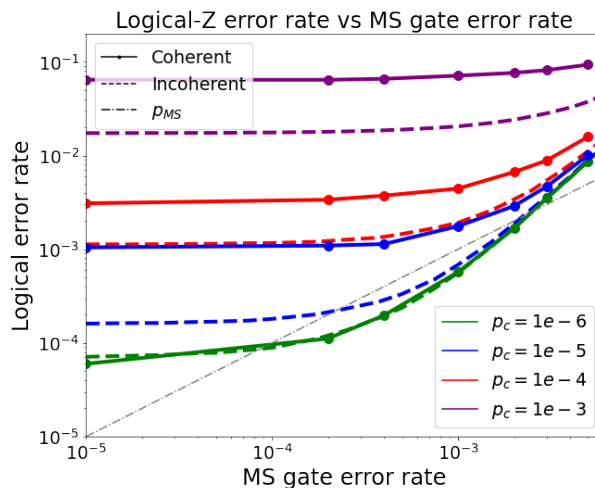
---

for different levels of crosstalk, using as control parameter the physical error rate on the entangling MS gate, which is the leading error rate in experimental implementations. The logical error rate is estimated by preparing the logical  $|+\rangle_L = (|0\rangle_L + |1\rangle_L)/\sqrt{2}$  state, as the performance of the QEC cycle in the absence of crosstalk is worse for the  $|+\rangle_L$  state than for the  $|0\rangle_L$  state. Although crosstalk errors in particular have a higher impact on the  $|0\rangle_L$  state, environmental noise has a higher impact on the logical  $|+\rangle_L$ , as this state is more susceptible to dephasing. Therefore, by measuring this state our simulations remain conservative in this respect when considering the goal of sustaining long-lived logical qubits.

In Figure 6.7 we show our results for the performance of QEC under our comprehensive noise model for near-term, static-string ion trap devices (see section 6.2), using both coherent and incoherent noise models. Although the main control parameters are the crosstalk and MS gate error rates, the simulation includes other sources of errors with the following rates: single-qubit errors ( $p_s = 10^{-5}$ ), state preparation and measurement errors ( $p_{sp} = p_m = 10^{-4}$ ), uncorrelated dephasing with coherence time  $T_2 = 2.2s$ , amplitude damping and leakage, with a lifetime  $T_1 = 1.1s$  of the metastable qubit state  $|1\rangle$  (see Sec. 6.2.1). The floor levels observed on the logical error rate for low values of the MS gate error rate are determined by the fixed values of these additional error sources. These values of the parameters are used in the remaining simulations in this chapter.

The MS gate error rate  $p_{MS}$  is used as the main control parameter because the MS gate fidelity is currently the bottleneck in experimental realizations. We consider the QEC protocol to be beneficial if the logical error rate  $p_L < p_{MS}$ . Thus, we define the pseudo-threshold as the highest value of  $p_{MS}$  for which we find a beneficial QEC regime.

From the results in Fig. 6.7, we find that a pseudo-threshold can be obtained for an MS gate error rate below  $p_{MS} \lesssim 2 \cdot 10^{-3}$ , as long as the crosstalk error rate is controlled below  $p_c \simeq 10^{-6}$ . This MS gate error rate is in reach for current technologies, as recent works have shown with demonstrations of FT applications of the Steane code [169, 171] in the following year after the publication of this work. A key observation in our results is the fact that a pseudo-threshold for beneficial QEC can no longer be found for crosstalk values above  $p_c \simeq 10^{-5}$ , which can be challenging using current



**Figure 6.7: Pseudo-threshold as a function of the entangling-gate error rate for specific crosstalk errors:** We show the results, on a log-log scale, for preparing and sustaining logical  $|+\rangle_L$  state. Note that for low MS gate errors, the curves flatten. This is because, for each line, only the MS gate errors vary and other error rates such as crosstalk, idle, single-qubit gate, initialization, and measurement errors are held fixed. From the results of this plot, we see that if the crosstalk error rate is at  $10^{-6}$  or better, then a pseudo-threshold exists. Furthermore, a pseudo-threshold of approximately  $2.7 \times 10^{-3}$  can be obtained. In addition to the MS gate errors, the simulation includes other sources of errors with the following rates: single-qubit errors ( $p_s = 10^{-5}$ ), state preparation and measurement errors ( $p_{sp} = p_m = 10^{-4}$ ), uncorrelated dephasing with coherence time  $T_2 = 2.2s$ , amplitude damping and leakage, with a lifetime  $T_1 = 1.1s$  of the metastable qubit state  $|1\rangle$ . These parameters are used in the remaining simulations in the paper. The details on these error models are described in Section 6.2. The floor levels of the logical error rate observed for low MS gate error rates stem from the damaging effects of these additional error sources. The error bars for the points are smaller than the size of the marker.

optical addressing techniques. The shuttling-based approach (see chapter 5.2) avoids this problem by placing the target ions in separated ion-crystals, where entangling operations can be applied without affecting neighboring ions. In turn, this approach is more limited by idle errors, as the ions spend more idle time in between operations, since they need to be moved around to be operated. Recent work [171] implemented the QEC protocol discussed in this chapter successfully using the shuttling-based approach.

This difficulty motivates the use of error suppression techniques to reduce the levels of crosstalk. In Sec. 6.3.3, we study the use of the alternative implementation of the MS gate (using power imbalance to change the nature of crosstalk, as described in Eq. (6.22) and [1]), and the impact of the different resulting Stark-shift crosstalk on the performance of QEC. In Sec. 6.4, we study the use of refocusing pulse sequences to

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS

---

suppress the levels of crosstalk by several orders of magnitude and allow the scheme to achieve a beneficial QEC regime.

Another interesting remark is the fact that the regime of beneficial QEC vanishes when the MS gate error rate is reduced further (e.g. below  $10^{-4}$ ). In that limit, the other noise sources that are kept constant in this single-parameter model would be the dominant factors preventing the demonstration of quantum logic advantage, which leads to the plateau-like levelling of the logical error rate. Similar behavior has been observed in previous works, e.g. [344]. The choice of the metric to evaluate the benefits of QEC is an open question that makes it difficult to define a single threshold value for multi-parameter noise models, and different metrics to quantify the performance of QEC have also been studied in the literature [149, 178]. Some examples of different metrics are the use of the memory time as control variable (which serves to study how the fidelity can be preserved over time, taking into account the duration of the gates and some waiting time) or the use of a noise parameter  $\lambda$  that adjusts the rates of the different noise sources proportionally (e.g. the single-qubit gate error rate as  $p_s = \lambda/10$ , and the two-qubit gate error rate as  $p_{MS} = \lambda$ ).

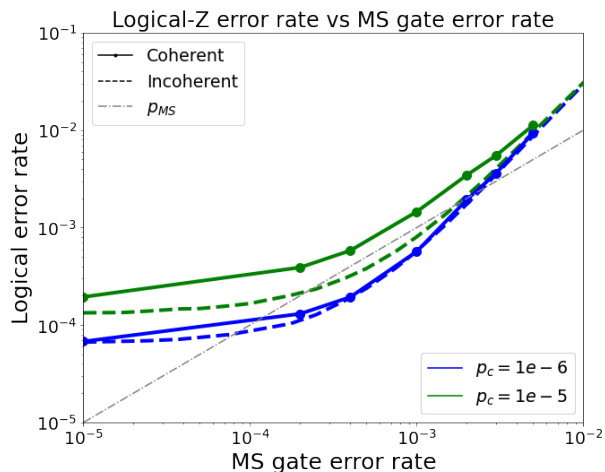
### 6.3.3 Impact of Stark-shift crosstalk noise on QEC performance

In this section, we study the effects of crosstalk using the Stark-shift crosstalk model explained in Sec. 6.2, corresponding to the application of a power imbalance between the intensities of the red and blue sideband transitions to suppress the entangling interactions of crosstalk.

As in the previous case, we simulate the performance of the same QEC protocol with crosstalk errors, both as coherent  $Z$  rotations or as incoherent phase errors on neighboring ions. Fig. 6.8 shows the results of our simulations, where we see that a pseudo-threshold for beneficial QEC can be found for crosstalk values on the order of  $10^{-5}$  for coherent crosstalk, and on the order of  $10^{-6}$  for incoherent crosstalk.

Although we do not study the compensation of these new crosstalk rotations in this work, it could be possible to compensate it directly using single-qubit gates, which can be performed with high fidelity [358]. The reason is that the systematic  $Z$  rotations originated from this crosstalk model do not create any entanglement. The crucial question for this approach lies in the balance between the fidelity of the single-qubit gates and the magnitude of the crosstalk errors, since a direct compensation of crosstalk

## 6.4 Active suppression of crosstalk with refocusing pulse sequences



**Figure 6.8: Pseudo-threshold for power-imbalanced suppression of crosstalk:** A log-log graph comparison of the logical error rates for different values of crosstalk, using the alternative crosstalk model with ac-Stark shifts on the neighboring ions. Here we determine the pseudo-threshold for the Steane code using parallel measurements of stabilizers as we vary MS gate and crosstalk error rates. We show the results for preparing a logical  $|+\rangle$  state. The additional noise sources modeled in the simulation using the same parameters as described in Fig. 6.7. The error bars for the points are smaller than the size of the marker.

in this way might have a more damaging effect than crosstalk itself, depending on the relative magnitude of both error sources.

A more relevant approach to compensate this type of crosstalk requires the precise measurement of its magnitude. Once it has been evaluated, it is possible to compensate for these rotations by adjusting the phases (rotation and entangling axes) of subsequent gate operations.

## 6.4 Active suppression of crosstalk with refocusing pulse sequences

As shown in the previous section, the impact of crosstalk on the performance of the QEC can be, by itself, sufficient to negate the benefits of the use of the QEC code. Crosstalk values achievable by realistic experimental conditions lie outside of the regime where a pseudo-threshold exists, and thus outside of the beneficial QEC region.

However, it is also possible to lower the effective value of the crosstalk rate by applying techniques at different levels of computation (e.g. by optimizing the ordering

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS

---

of the ions in the trap to minimize the most dangerous crosstalk interactions [344]). In particular, we focus on a gate-level technique to suppress crosstalk: refocusing pulse sequences [20, 108, 359]. This technique will allow the control of crosstalk down to the levels of fidelity of the well-controlled single-qubit gates.

### 6.4.1 Analytical suppression of residual noise after refocusing

This section begins with an analytical study of the refocusing scheme and its ideal behavior. After that, we analyze how the presence of errors in the gates used in the refocusing pulse sequence affect the protocol's crosstalk suppression capabilities. Finally, we support these results with numerical simulations, showing the performance of the flag-based QEC protocol including crosstalk errors and the use of the refocusing scheme. With this numerical analysis, we aim to provide valuable insight into the expected performance of future experiments on the path towards the realization of a functional logical qubit with beneficial QEC.

The idea behind the refocusing pulse sequence technique is to take advantage of the coherent nature of crosstalk to cancel the unwanted rotation, in an approach similar to the pulse sequences used in [20, 359] to remove unwanted couplings. When we apply a MS gate of angle  $\phi$  on the qubits, there will be an  $X_G X_N$  rotation of a fraction of that angle  $\epsilon_{CT}\phi$  applied to every pair of gate and neighbor due to the crosstalk effect. Since it is a coherent rotation, we can use the following property to our advantage:

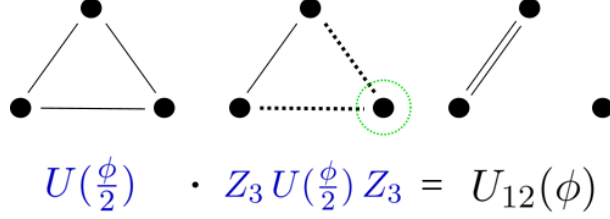
$$Z_2 e^{i\phi X_1 X_2} Z_2 = e^{-i\phi X_1 X_2}. \quad (6.25)$$

If we want to apply an MS gate of angle  $\phi$ , we subdivide the gate in two half rotations  $U(\frac{\phi}{2})$ . Before and after applying the second half-MS gate, we apply a  $Z$  rotation on each one of the neighboring ions. This changes the sign of all the unwanted crosstalk rotations without altering the interaction between the two gate ions, and these unwanted rotations will then cancel with the previous crosstalk rotation. We can see a simplified sketch of this interaction in Fig. 6.9.

In the ideal case, this simple sequence cancels all the crosstalk interactions perfectly. Nevertheless, the experimental realization of this scheme involves the use of gates that are intrinsically faulty. Therefore, we need to realize a closer analysis to ob-



## 6.4 Active suppression of crosstalk with refocusing pulse sequences



**Figure 6.9: Graphical example of the behavior of the refocusing scheme.** We want to apply the rotation  $U_{12}(\phi)$ , which for our case will be the MS gate. By subdividing the gate in two half rotations  $U(\frac{\phi}{2})$ , and applying a  $Z_3$  rotation on the unwanted qubit 3, we can cancel the rotations applied between qubit 3 and its neighbors. The end result is our desired gate. Similar graph representations have been used previously in e.g. [360, 361].

tain the effective error channel for the implementation of a MS gate using the refocusing sequence.

To estimate the remaining uncompensated errors in the refocusing scheme, we consider a simple noise model, in which the single-qubit refocusing pulses suffer stochastic amplitude or phase fluctuations. These fluctuations produce stochastic rotations in any the three Pauli axes, which we can model using standard depolarising noise (see Eq. 2.21). These dynamics of the imperfections in the  $\pi$ -pulses used for the refocusing can be written as a master equation:

$$\dot{\rho} \simeq \sum_{i=1}^3 \frac{1}{3} \langle \epsilon_Z^2 \rangle \left( \sigma_n^{(i)} \rho \sigma_n^{(i)\dagger} - \frac{1}{2} \left\{ \sigma_n^{(i)\dagger} \sigma_n^{(i)}, \rho \right\} \right) \quad (6.26)$$

where  $\epsilon_Z$  is a stochastic variable with probability distribution  $p(\epsilon_Z)$ ,  $\sigma_n^{(i)}$  are the Pauli matrices applied on qubit  $n$ , and  $\rho$  denotes the density operator of the system. This can be associated with a single-qubit error rate  $p = \langle \epsilon_Z \rangle^2 = s_Z^2$ , assuming that the distribution of the errors in the single-qubit  $Z$  gates follows a Gaussian distribution with standard deviation  $s_Z$ , and there are no systematic errors

$$\int d\epsilon_Z p(\epsilon_Z) \epsilon_Z = 0. \quad (6.27)$$

The resulting map of each noisy  $Z$  gate is then

$$\epsilon_Z(\rho) = (1 - p)Z\rho Z + p\frac{1}{3}(X\rho X + Y\rho Y + \mathbb{1}\rho\mathbb{1}). \quad (6.28)$$

As for the crosstalk, we consider the following notation for the crosstalk unitary

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS

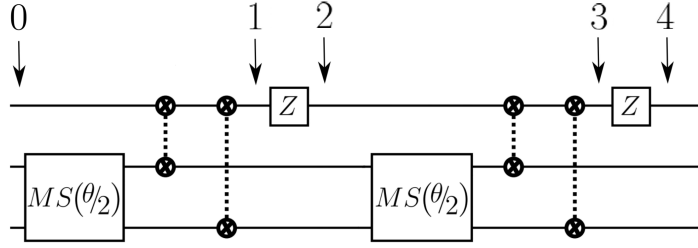
---

after one half MS gate and after a full MS gate to simplify the calculations:

$$U_{CT/2} = e^{-i\frac{\theta}{2}\epsilon_{CT}X_GX_N} \quad (6.29)$$

$$U_{CT} = e^{-i\theta\epsilon_{CT}X_GX_N} \quad (6.30)$$

To analyze the evolution of the system, we can then compute the Kraus decomposition of the noise channel by integrating the dynamics of the system, and averaging over the stochastic noise. We consider the state at different steps of the refocusing procedure: 0) initial state, 1) state after the first half-MS gate, 2) state after the first refocusing pulse, 3) state after the second half-MS gate, 4) final state after the last refocusing pulse (see Fig. 6.10).



**Figure 6.10: Schematic view of the refocusing pulse sequence.** The time steps one to four represent the different points used as reference during the calculations in the main text for  $\rho_1$ ,  $\rho_2$ ,  $\rho_3$  and  $\rho_4$ .

The state of the system at each point in the circuit can be described by applying each noisy circuit element successively:

$$\rho_1 = U_{CT/2}\rho_0U_{CT/2}^\dagger \quad (6.31)$$

$$\rho_2 = (1-p)Z\rho_1Z + p\frac{1}{3}(X\rho_1X + Y\rho_1Y + \mathbb{1}\rho_1\mathbb{1}) \quad (6.32)$$

$$\rho_3 = U_{CT/2}\rho_2U_{CT/2}^\dagger \quad (6.33)$$

$$\rho_4 = (1-p)Z\rho_3Z + p\frac{1}{3}(X\rho_3X + Y\rho_3Y + \mathbb{1}\rho_3\mathbb{1}). \quad (6.34)$$

The final state  $\rho_4$  can be obtained as a function of the initial state  $\rho_0$  by solving the system of equations. From that expression, we can obtain the Kraus operators that represent the evolution of the noisy refocusing pulse sequence. Then, we can obtain the leading order evolution of the system in terms of the single qubit error rate  $p$ :

## 6.4 Active suppression of crosstalk with refocusing pulse sequences

---

$\mathcal{O}(1)$  :

$$\sqrt{(1-p)^2} ZU_{CT/2} ZU_{CT/2} = \sqrt{(1-p)^2} \mathbf{1} \quad (6.35)$$

$\mathcal{O}(p)$  :

$$\sqrt{p(1-p)/3} ZU_{CT/2} \mathbf{1}U_{CT/2} = \sqrt{p(1-p)/3} ZU_{CT} \quad (6.36)$$

$$\sqrt{p(1-p)/3} ZU_{CT/2} XU_{CT/2} = \sqrt{p(1-p)/3} iYU_{CT} \quad (6.37)$$

$$\sqrt{p(1-p)/3} ZU_{CT/2} YU_{CT/2} = \sqrt{p(1-p)/3} (-i)X \quad (6.38)$$

$$\sqrt{p(1-p)/3} \mathbf{1}U_{CT/2} ZU_{CT/2} = \sqrt{p(1-p)/3} Z \quad (6.39)$$

$$\sqrt{p(1-p)/3} XU_{CT/2} ZU_{CT/2} = \sqrt{p(1-p)/3} (-i)Y \quad (6.40)$$

$$\sqrt{p(1-p)/3} YU_{CT/2} ZU_{CT/2} = \sqrt{p(1-p)/3} iX \quad (6.41)$$

$\mathcal{O}(p^2)$  :

$$\sqrt{p^2/9} \mathbf{1}U_{CT/2} \mathbf{1}U_{CT/2} = \sqrt{p^2/9} U_{CT} \quad (6.42)$$

$$\sqrt{p^2/9} \mathbf{1}U_{CT/2} XU_{CT/2} = \sqrt{p^2/9} XU_{CT} \quad (6.43)$$

$$\sqrt{p^2/9} \mathbf{1}U_{CT/2} YU_{CT/2} = \sqrt{p^2/9} YU_{CT} \quad (6.44)$$

$$\sqrt{p^2/9} XU_{CT/2} \mathbf{1}U_{CT/2} = \sqrt{p^2/9} XU_{CT} \quad (6.45)$$

$$\sqrt{p^2/9} XU_{CT/2} XU_{CT/2} = \sqrt{p^2/9} U_{CT} \quad (6.46)$$

$$\sqrt{p^2/9} XU_{CT/2} YU_{CT/2} = \sqrt{p^2/9} Z \quad (6.47)$$

$$\sqrt{p^2/9} YU_{CT/2} \mathbf{1}U_{CT/2} = \sqrt{p^2/9} YU_{CT} \quad (6.48)$$

$$\sqrt{p^2/9} YU_{CT/2} XU_{CT/2} = \sqrt{p^2/9} ZU_{CT} \quad (6.49)$$

$$\sqrt{p^2/9} YU_{CT/2} YU_{CT/2} = \sqrt{p^2/9} \mathbf{1}. \quad (6.50)$$

We find that the resulting dynamics can be described by incoherent dynamics, for which the order  $\mathcal{O}(p)$  Kraus operators of the residual errors after the refocusing pulse

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS

---

sequence are

$$\begin{aligned}
 K_0 &= \sqrt{1 - 2p} \mathbf{1}_n, & K_3 &= \sqrt{2\frac{p}{3}} X_n, \\
 K_1 &= \sqrt{\frac{p}{3}} Z_n U_{CT}, & K_4 &= \sqrt{\frac{p}{3}} Z_n, \\
 K_2 &= \sqrt{\frac{p}{3}} Y_n U_{CT} & K_5 &= \sqrt{\frac{p}{3}} Y_n.
 \end{aligned} \tag{6.51}$$

Here, the operators  $K_3$  to  $K_5$  correspond to standard depolarising error channel. The operators  $K_1$  and  $K_2$  correspond to two-body jump operators describing residual *incoherent* crosstalk between gate and spectator ions. Crucially, the error rate of these residual errors,  $p_{CT}^{residual} = p/3$ , is strongly suppressed by the single-qubit error rate  $p$ .

From this analysis, we can conclude that the application of the refocusing scheme is expected to suppress the effects of the crosstalk noise by a factor that depends on the single-qubit error rate. Thus, the application of the protocol provides a powerful tool to reach crosstalk error rates with a negligible influence on the QEC performance of the logical qubit when compared with other present noise sources. To support this claim, we realize numerical simulations of the QEC protocol, implementing the refocusing pulse sequence for every MS gate. The results of this analysis are shown in the following section.

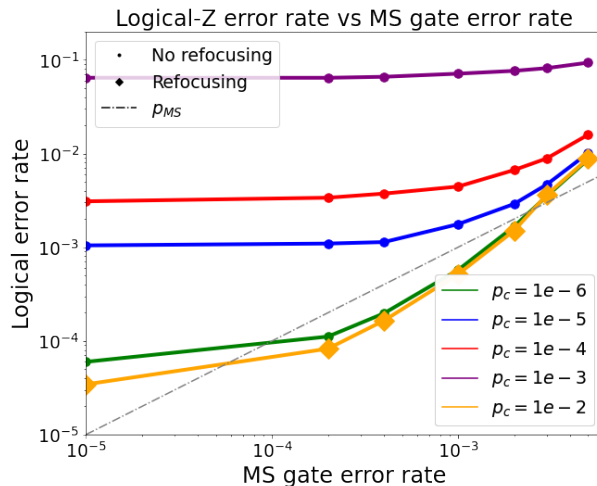
### 6.5 Numerical analysis of the performance of QEC with active crosstalk suppression

In order to test the potential for crosstalk suppression provided by the refocusing scheme in a practical setting, we implement the refocusing pulse sequence in every MS gate realized during the flag-based, color code QEC protocol studied during this chapter, proposed by [229], and described in the preceding sections (see Fig. 6.5).

As discussed in Sec. 6.2, we use an extensive error model that aims to describe realistically all the error sources present in a near-term, static-string ion trap. For the simulations shown in Fig. 6.11, we use state-vector simulations and we focus on the coherent modeling of the entangling crosstalk error model. In the figure, we overlap the results from the previous simulations of coherent, non-suppressed crosstalk errors for several crosstalk rates from  $p_c = 10^{-6}$  to  $p_c = 10^{-2}$ ; and the logical error rate

## 6.5 Numerical analysis of the performance of QEC with active crosstalk suppression

---



**Figure 6.11: Pseudo-threshold for active suppression of crosstalk:** Here we determine the pseudo-threshold for the Steane code using parallel measurements of stabilizers and introducing the refocusing scheme in the simulations. We overlap the results of the previous simulations, where we do not apply any crosstalk suppression technique. From the results on this plot, we see that at least four orders of magnitude reduction in the effect of crosstalk. The additional noise sources modeled in the simulations use the same parameters as described in Fig. 6.7. The error bars for the points are smaller than the size of the marker.

obtained with active crosstalk suppression using the refocusing pulse sequence and a crosstalk error rate of  $p_c = 10^{-2}$ . Our results show how the refocusing pulse sequence can significantly suppress the damaging crosstalk effects, achieving a reduction of the crosstalk effects by at least four orders of magnitude. This result is consistent with the results from the analytical study shown in the previous section. With the use of the refocusing sequence, crosstalk levels are suppressed down to a level at which we find a clear window of beneficial QEC, with a significant margin for MS gate fidelities of  $10^{-3}$ , which are within reach of current ion trap experiments. Therefore, this analysis shows a clear path towards the realization of a functional logical qubit in the regime of beneficial QEC for the static-string ion-trap approach.

Notably, recent experiments posterior to the publication of this work have successfully shown the implementation of the Steane code in ion traps [169, 171]. In [171], the authors show the successful realization of repeated rounds of QEC in a ten qubit shuttling-based ion trap device. They implement the parallel flag-based readout studied in this work [229], realizing the procedure repeatedly to maintain coherence. With the current performance of their device, they nearly achieve the beneficial QEC regime.

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS

---

They support the experiment with numerical simulations, from which they estimate that the pseudo-threshold can be within reach if the two-qubit gate error rate is reduced by approximately a factor of 3, and the dephasing and leakage rates are substantially reduced. On the other hand, in [169] the authors show the first FT implementation of a universal set of gates on encoded qubits. This includes the transversal application of single and two-qubit gates (CNOT), but also the implementation of the non-Clifford T gate, for which they implement a FT protocol for magic state preparation and injection. In their experiment, they use the same static-string ion trap approach and the parallel flag-based readout [229] studied in this chapter. Remarkably, the authors report the success of the FT protocols, as they observe an improvement in performance of the encoded qubits with FT protocols, despite the increase in gate-count and complexity required to ensure fault tolerance. As in the previous study, they support their work with numerical simulations, with which they estimate closely the performance of their experimental results. Both of these studies highlight the utility of numerical simulations as a tool to support the experimental efforts.

### 6.6 Summary and outlook

In this chapter of the thesis, we have presented the results of the feasibility study of QEC in ion traps published in Ref. [1]. In this work, we analyze the performance of a recently proposed flag-based stabilizer readout scheme [229], by simulating its application on a near-term, single-string ion trap device equipped with a gate set including recently demonstrated experimental capabilities [300, 301]. We use an exhaustive noise model that aims to reproduce the realistic behavior of the quantum device, based on microscopic modeling of the different error sources, that builds over previous works focused on a shuttling-based approach to ion traps [149, 178]. During this study, we put a particular emphasis on the study of crosstalk noise, a notably damaging error source that has been reported and studied by recent works (e.g. [101, 107, 139, 179, 306, 341–344]) that has the potential to break the FT properties of quantum circuit constructions. Beyond simulating its impact on the performance of QEC, we study the use of a crosstalk-suppression protocol and prove, both analytically and numerically, its efficacy as a tool to reduce crosstalk noise down to the level where its effects can be neglected when compared with other error sources present in the device.

The results from our study yield a very positive message, as they show that the use of the refocusing scheme can allow near-term experimental devices to reach a beneficial QEC regime with this flag-based QEC protocol. In fact, recent studies posterior to the publication of our work have successfully implemented in ion traps the simultaneous flag-based stabilizer readout proposed in [229]. In [169], they used the static-string ion trap approach to realize the first demonstration of a FT implementation of a universal set of gates on encoded qubits, including two-qubit operations between encoded qubits and the FT implementation of a non-Clifford gate using magic state injection. Using FT protocols (including the FT QEC readout scheme studied in this work), they show an improved performance of the encoded qubits with respect to a non-FT implementation. In [171], they implemented repeated rounds of FT QEC in a shuttling-based ion trap device, with a performance close to the beneficial QEC regime. Both works rely on numerical simulations to support their experimental results, which highlights the relevance of simulations in the development of QEC, both to predict the behavior of a protocol, and to estimate the best route to improve the performance of the experiments.

With regards to the Stark-shift crosstalk model (Sec. 6.3.3), we find that this error source has a reduced impact on the performance of QEC. Our results indicate that a pseudo-threshold can be found for the anticipated experimental values, even without the use of further suppression techniques.

Looking forward, the realization of further feasibility studies is an active field of research, that must attempt to stand always several steps beyond the current experimental capabilities, identifying potential problems and interesting targets for the physical realization. Thus, it would be interesting to expand the study on several fronts: the study of different codes and protocols (e.g. higher distance color codes and surface codes, as well as different approaches to the FT implementation of quantum circuits), alternative physical platforms (e.g. shuttling-based ion traps, superconducting qubits or Rydberg atoms), more refined error models accounting for the particularities of each platform, and the development of the numerical techniques utilized in the simulation of quantum circuits.

The improvement of the numerical techniques, particularly the incoherent simulations, can also be a relevant line of research to explore. Incoherent simulations are a computational tool that allows the execution of quick estimations of the performance of a circuit, with a computational cost being several orders of magnitude faster than

## 6. FEASIBILITY STUDY OF QEC WITH TRAPPED IONS

---

the state vector simulations. Although not as general as the state vector simulations (which are not restricted to Clifford operations), they can provide invaluable help in the development of the field, as shown e.g. in [169]. As the number of qubits of the experimental platforms increases, the use of incoherent simulations such as stabilizer simulations or Pauli propagation [277] can become crucial in the development of quantum computation and QEC. Furthermore, recent works are studying the possibility of realizing stabilizer simulations of circuits with a few non-Clifford gates [182], which could significantly improve the range of application of stabilizer simulations.



## Chapter 7

# Union-find decoder for color codes

## 7. UNION-FIND DECODER FOR COLOR CODES

---

In this chapter, we explore an alternative decoding algorithm, based on the work presented originally by Delfosse and Nickerson [193]. First, in section 7.1.1 we summarize the main steps of the original algorithm on its application to the surface code. Then, in section 7.2 we present our adaptation of the algorithm for color codes, and in section 7.3 we analyze its performance.

On the work presented in this section, I developed the algorithm for the adapted version of the decoder described in 7.2, as well as the numerical implementation of the algorithm. I realized the numerical analysis to estimate the threshold of the algorithm, as well as the analysis of the percolation of the clusters (Sec. 7.3). I also wrote the code for my implementation of the original decoder, that I used to benchmark its performance in Sec. 7.1.2.

### 7.1 Background

The union-find decoding algorithm was first presented in [193]. This algorithm appeared as a highly efficient decoding algorithm valid both for surface codes and color codes. With an almost-linear  $\mathcal{O}(N\alpha(N))$  complexity<sup>1</sup> in the number of qubits  $N$ , this algorithm stands out as one of the fastest decoders for topological codes, improving over the polynomial order (between  $\mathcal{O}(N^{2.5})$  and  $\mathcal{O}(N^7)$ , depending on the implementation) of the Minimum Weight Perfect Matching (MWPM) algorithm [185, 247–249].

The basis of the almost-linear efficiency of the union-find decoder’s algorithm lies in the possibility of finding and merging disjoint sets in order  $\mathcal{O}(N\alpha(N))$  [362]. Since the  $\alpha(N)$ , the inverse of Ackermann’s function, has a extremely slow growth in computational complexity, the algorithm is usually labelled as *almost-linear*. The second building block for the algorithm is the peeling decoder [194], which solves the reduced problem of decoding the surface code for the erasure channel. The peeling decoder assumes knowledge of the positions where an error might have happened, and produces a maximum likelihood correction in linear time  $\mathcal{O}(N)$ .

The working principle of the union-find decoder relies on a property of the toric surface code: for a given region of the lattice with an odd number of stabilizer excitations, there exists at least one chain of errors that exits the region. The strategy of

---

<sup>1</sup> $\alpha(N)$  is the inverse of the Ackermann’s function, a very slowly growing function that is smaller than 3 for all practical purposes [193, 362]

the algorithm is to define regions of qubits with potential errors around each excitation in the lattice, and iteratively expand these regions, also called clusters, by including the neighboring qubits and stabilizers. This process eventually leads to clusters merging together. If the number of excitations of the resulting cluster is odd, we need to keep expanding it, as we know that there is at least one chain of errors that exits the cluster. If the number of excitations within a cluster is even, then we know that a correction can be found wholly within the cluster, and we stop expanding it. Once the number of excitations within all clusters is even, we can use the peeling decoder to find a correction, as the clusters define the positions where errors might have happened.

This property of surface codes, however, does not apply to color codes. An error in a single qubit generates three excitations, and chains of errors can generate complex patterns of excitations in the color code (an example can be found in Fig. 3.7). The solution proposed by the authors in [193] relies on projecting the color code into three surface codes [192]. In this chapter, we present an alternative version of the union-find decoder that relies on the particular properties of the 3-colored excitations of the color code. Clusters grow around excitations until they reach an even parity in all three colors, ensuring that it is possible to find a valid correction within the clusters.

The code capacity threshold (see Sec. 3.2) for the Union-find decoder on surface codes is 9.9% [193]. This threshold proves slightly lower than the 10.3% achieved with the standard MWPM algorithm. There is, therefore, a trade off between high threshold and low computational complexity when comparing these algorithms. The same trend remains true for phenomenological noise, as the Union-find decoder reaches 2.6%, as opposed to the 2.9% found using MWPM.

On color codes, the threshold of the Union-find algorithm as described in [193] is 8.4%, which is higher than the 7.8% obtained by [189] using the rescaling decoder studied in chapter 8. However, in a more recent publication it has been shown that a threshold of 10.2% can be achieved efficiently in color codes using the Restriction decoder [196]. This algorithm decodes the color code by mapping it to two surface codes (rather than the three copies required for the previous algorithms) and applying MWPM on each of them. The bottleneck of the efficiency of the Restriction decoder is the MWPM algorithm. This can be improved by substituting the MWPM step with the Union Find algorithm, which gives the Restriction decoder the almost-linear computational complexity, but reduces the threshold to 9.8%.

## 7. UNION-FIND DECODER FOR COLOR CODES

---

In this chapter, we present an alternative decoder for color codes, based on the ideas of the union-find decoder. Our decoder achieves a threshold of 6% for code capacity noise. Furthermore, we analyze the behavior of cluster growth near the threshold by studying the percolation of clusters through the lattice, which provides a deeper understanding of the decoder.

### 7.1.1 Union find decoder for surface codes

In this section, we go through the details on the implementation of the Union Find decoder, as described in [193]. We explain how each of the steps works and how they are implemented to achieve the almost linear efficiency. Our variation of the decoder builds on these main steps, and is explained in section 7.2. Although the original algorithm was described for an error channel consisting on erasure and Pauli errors, we focus only on Pauli errors and neglect erasure during the rest of the chapter.

In section 7.1.1.1 we explain the process of cluster growth and the details on how to merge the clusters efficiently, with an almost-linear computational complexity. Then, in section 7.1.1.2 we explain the peeling decoder algorithm, that finds a valid correction within a cluster by creating a spanning tree and *cutting* the leaves of the tree.

#### 7.1.1.1 Growing and merging clusters

In this algorithm, we consider a node to be either a stabilizer or a qubit. In the surface codes, these are represented as vertices and edges of the square lattice, respectively (we consider only the  $S_X$  stabilizers). For simplicity, we consider only phase flip errors  $Z$ , which trigger the  $S_X$  stabilizers. The application of the algorithm for the  $S_Z$  stabilizers, located in the square faces, is analogous. The connection between a stabilizer and a neighboring qubit is called a half edge (see Fig. 7.1).

A cluster is defined as a set of connected nodes, and its parity corresponds to the binary sum of the parity of the stabilizers belonging to the cluster. The parity of a cluster takes the values 0 or 1, which we refer to as even or odd, respectively. The boundary of the cluster is the set of nodes that belong to the cluster, but that have at least one incident half edge that connects to a node that does not belong to the cluster. During a step of cluster growth, we add to the cluster the nodes that have a half edge that connects to the boundary of the cluster. We also refer to this step as growing the cluster by half an edge.

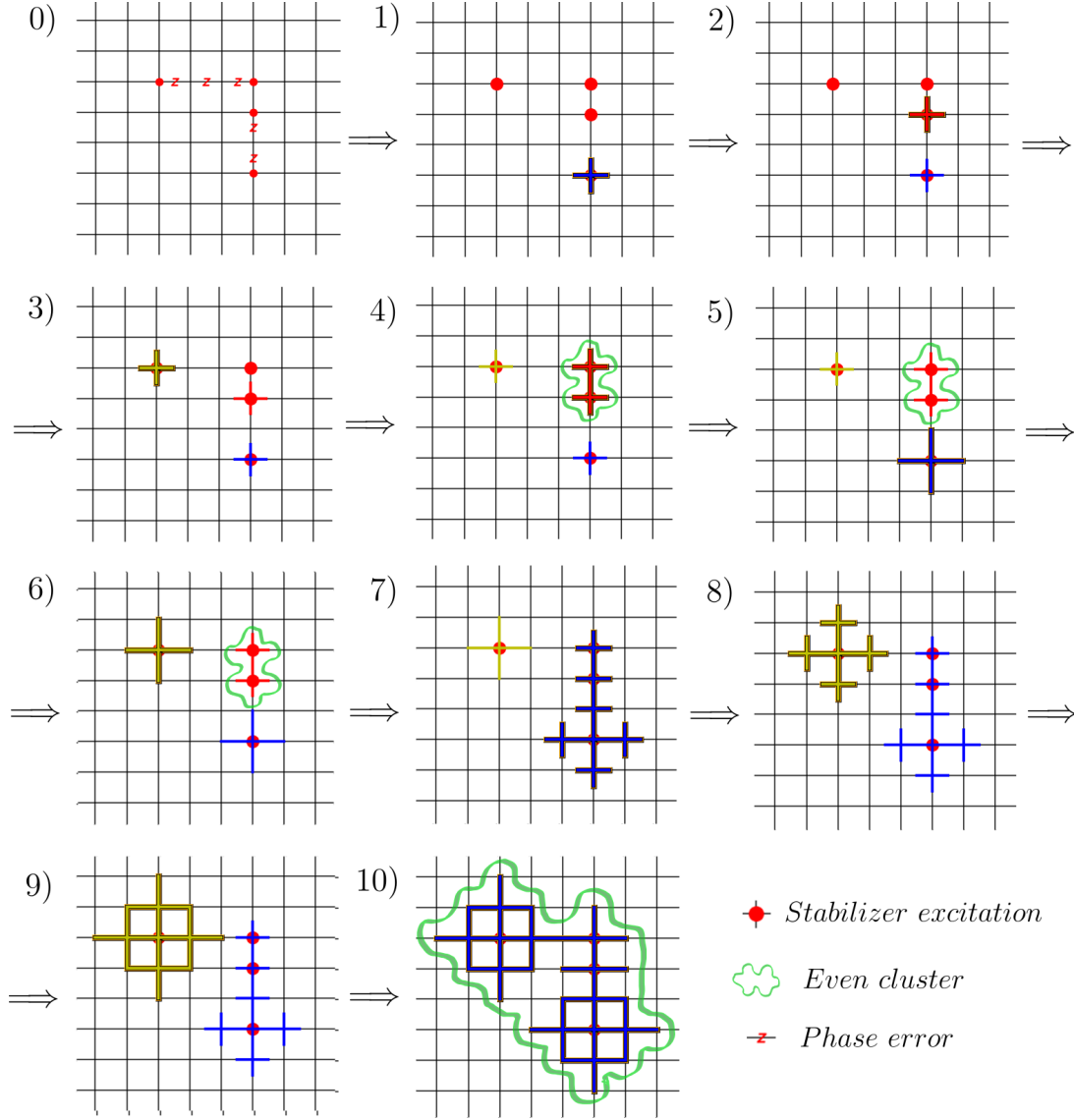
After a step of cluster growth, two clusters  $C_i, C_j$ , can become connected, i.e. there is at least one node that belongs to both clusters. When this happens, we define a new cluster  $\tilde{C}$  corresponding to the set of nodes that belong to any of the two clusters or to both. This new cluster  $\tilde{C}$  replaces the previous clusters  $C_i$  and  $C_j$ , which are considered to be merged into  $\tilde{C}$  (see Fig. 7.1).

The process of cluster growth is the main step of the Union-find decoder. The goal of this step is to find a minimal set of qubits within which a correction can be found. The algorithm finds this set of qubits by growing clusters around the excitations in the syndrome. Clusters with odd parity will grow and merge with others until the parity within each cluster is even. In the toric surface code, this condition ensures that a valid correction can be found within the clusters, since stabilizer excitations are always generated in pairs. This means that if a cluster has an odd number of excitations, there exists a chain of errors leading outside the cluster. If a cluster has an even number of excitations, it is possible to find a configuration of errors located on qubits belonging to the cluster that generates the pattern of excitations found in the stabilizers belonging to the cluster. Although the basic idea of this algorithm is simple, realizing the growth and merging of the clusters in an efficient manner requires a well thought organization of the data structures. In this section we explain the details of this organization, and other important steps to maximize the performance of the decoder, as exposed originally in [193].

The algorithm starts by creating a cluster on each excited stabilizer, each cluster containing a single node, and storing all odd clusters in a queue. As all of them contain exactly one excitation, all the clusters start with an odd parity. Then, the algorithm will systematically grow and merge the odd clusters until all of them have an even parity. In order to find if two clusters have merged after a step of cluster growth, we define the *fusion list*, a list that contains the nodes that were added to the cluster during the last step of cluster growth. We use the fusion list as an intermediate step, first finding all the nodes that will be added to the cluster and adding them to this list, and then checking if any of these nodes already belongs to a different cluster, in which case both clusters will be merged. This process can be summarized in the following steps, that are applied repeatedly until all clusters have an even parity:

1. Pick the odd-cluster  $C$  with the smallest boundary.

## 7. UNION-FIND DECODER FOR COLOR CODES



**Figure 7.1:** Example of cluster growth in the surface code lattice. On each cluster growth step, one cluster with odd parity will increase its size by half an edge around the boundary. The cluster that grows on each step is underlined with a thicker line. Clusters with even parity are marked with a green line. At step 0, we show the original error configuration: phase errors  $Z$  on five qubits, that generate four stabilizer excitations, marked in red.

2. Find the nodes that connect to the boundary of  $C$  and add them to the fusion list.
3. For each node in the fusion list, we find if the node is shared with a different

cluster, in which case we merge both clusters. Otherwise, we add the node to the cluster, and update the boundary of  $C$ .

4. After updating the cluster  $C$  and its boundary, if the parity is still odd we update the queue with odd clusters.
5. Stop if there are no remaining clusters with odd parity.

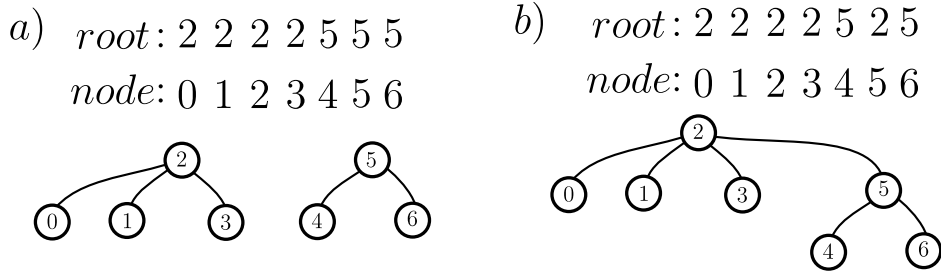
Every time we grow an odd-cluster of boundary size  $n_b$ , there is typically only one chain of errors going out of the cluster. This means that we are adding approximately  $n_b - 1$  incorrect links to the cluster. Thus, growing first the smallest clusters will reduce the amount of incorrect links that we add to the cluster. To ensure that the algorithm always picks the odd-cluster with the smallest boundary in an efficient way, the list of odd clusters is stored as a priority queue [363]. A priority queue is a special type of data structure in which each element is associated with a priority value (in this case, the priority value is related to the size of the cluster). Elements stored in this special "queue" can be recovered on the basis of their priority, so that higher priority elements are obtained first. Depending on the implementation, inserting and removing elements from a priority queue can be done in  $\mathcal{O}(n)$  steps with the number of elements  $n$  in the queue. With this data structure, we can efficiently find the next cluster to grow without affecting the scaling of the algorithm's computational complexity.

The key problem of growing the clusters is to find an efficient way to check if a new node added to a cluster was already part of a different cluster. This problem corresponds to the management of a disjoint-set data structure. An efficient solution to this problem is to use an union-find data structure [362], from which this decoder takes the name. The main working principle for this algorithm is to store the nodes and clusters as a tree structure. A tree is a data structure that stores nodes in a hierarchy. It consists of a central node or *root*, structural nodes or *branches*, and subnodes or *leaves*. It can be defined recursively: a tree consists of a root, and zero or more subtrees, such that there is an edge from the root of the tree to the root of each subtree. Nodes different from the root with zero subtrees are called leaves, else they are called branches. The key feature of this data structure is that each node in the tree stores a reference to the root of its subtree, so that finding the root of the tree can be done by following the connections from each subtree to its root (see Figs 7.2, 7.3).

## 7. UNION-FIND DECODER FOR COLOR CODES

---

Every cluster is defined by a root node, and each cluster is a tree of nodes with that root node as the root of the tree. Therefore, a cluster will be formed by a group of nodes with a common root that identifies the cluster. Using this data structure, we can merge two clusters by pointing the root of the smallest cluster to the root of the largest, which only takes one operation and is therefore  $\mathcal{O}(1)$  (Fig. 7.2). Finding the cluster to which a node belongs can be done by following the tree to the root. For this process to be fast, we need the trees to be shallow, and the nodes to be connected as directly to the root as possible. As we merge clusters, the cluster tree becomes deeper, as we do not update every branch of the tree in the merging step. However, each time we follow the path from a node to the root of the cluster, we can update all of the nodes in the path and link them directly to the root of the cluster, reducing the time of future searches in the tree. We can see an example of this process in Fig. 7.3. The analysis of the efficiency of this algorithm leads to the scaling  $\mathcal{O}(N\alpha(N))$ , where  $\alpha(N)$  is a functional inverse of the Ackermann's function [362].



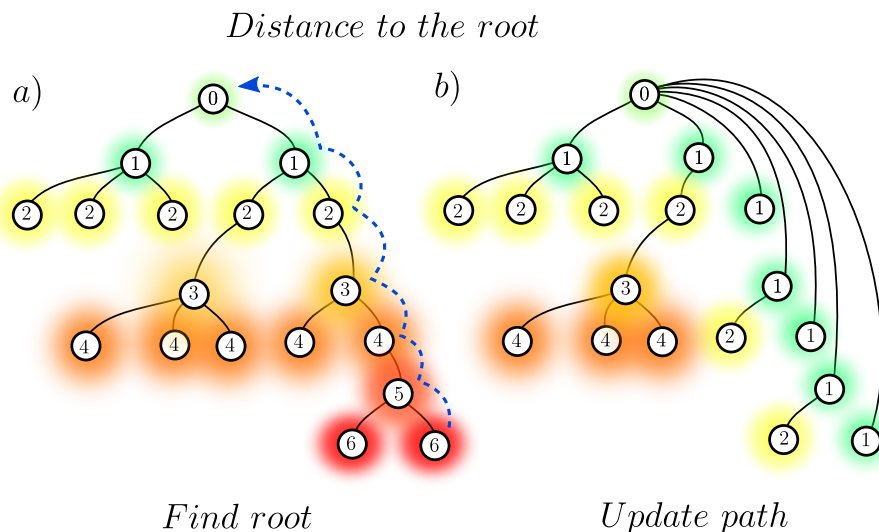
**Figure 7.2: Tree structure of the clusters:** in the algorithm, clusters of nodes are stored as trees, where each cluster is identified by a root node. To find the root of a given node, it is sufficient to follow the tree. Typically, nodes will be directly linked to the root of the cluster, and this process will be  $\mathcal{O}(1)$  in complexity. To merge two clusters, it is sufficient to change the root of the smaller cluster to the root of the larger cluster. **a)** example of two clusters and the tree structure. For each node, we store its root. **b)** After merging both clusters, the only change needed is to update the root of the smaller cluster.

Once all clusters have even parity, a solution can be found by placing corrections only on qubits within the clusters. In the next section, we describe the last step of the algorithm, the peeling decoder, that finds a correction for each cluster in  $\mathcal{O}(N)$  time.

### 7.1.1.2 Peeling decoder

The peeling decoder was originally presented in [194] as a decoder for the erasure channel. The decoder assumes that the possible positions of the errors are known,





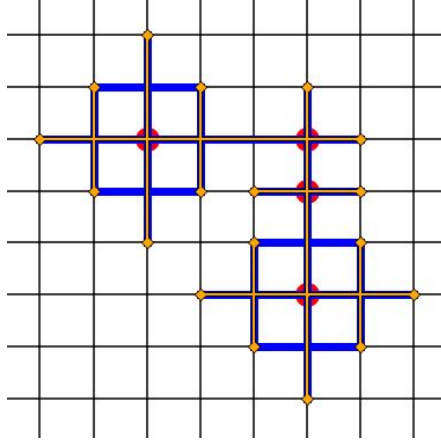
**Figure 7.3: Find root algorithm.** In this figure, we show an example of the search of the root of a given node, and the update of the tree structure afterwards. Each node is labeled with the distance to the root of the tree, which represents how many steps are needed to find the root of the tree by following the root of each node. **a)** Original tree of the cluster, where we try to find the root of the node in the bottom right. This process requires 6 steps, following the path marked with the dashed blue line. **b)** After finding the root, all nodes belonging to the path are updated. This reduces significantly the cost of future searches on the tree.

and finds a correction within the set of erased qubits. This scenario is equivalent to the state after the cluster growth. As described in the previous section, a solution for the decoder problem can be found within the cluster sets. Therefore, we can apply the peeling decoder considering the qubits within the clusters as erased qubits, thus finding a correction for the decoding problem.

The algorithm of the peeling decoder consists of two main steps. First, we create a spanning tree for each cluster (i.e. an alternative graph that includes all nodes but without cycles, see Fig. 7.4). For the spanning tree, we consider the stabilizers as nodes, and the qubits as edges. The *leaves* of the spanning tree are the stabilizers that are connected to the tree by only one edge (qubit), which is also considered a part of the leaf. Second, the *leaves* of the spanning tree are removed systematically until the tree is empty. In this pruning process, we apply correction to the qubit-leaf if there is an excitation in the stabilizer at the end of the leaf. This correction changes the parity of the two stabilizers connected to that qubit, which effectively displaces the excitation along the *branch* until it meets a different excitation, leaving a path of corrections

## 7. UNION-FIND DECODER FOR COLOR CODES

---



**Figure 7.4: Spanning tree.** In this figure, we show an example of a spanning tree (orange) for a given cluster (blue). The leaves of the tree (marked with squares) are the starting point of the next step of the peeling decoder, where nodes of the cluster are being removed by following the branches from the leaves to the root, adding corrections if the end of the leaf has an excitation. In this process, the excitations are connected by chains of corrections that follow the branches of the spanning tree.

connecting both of them.

The spanning tree of a graph can be found in linear time, and the pruning step only requires one operation for each node in the tree. Thus, the complexity of the peeling algorithm is  $\mathcal{O}(N)$ . The spanning tree can be found by running a search algorithm on each cluster, including edges to the tree as the algorithm explores the cluster but preventing the formation of cycles by avoiding edges to nodes already in the cluster. In our implementation, we apply a BFS (breath first search) algorithm to explore the tree and generate the spanning tree.

The solution found by the peeling decoder is not unique, and depends on the generation of the spanning tree. Nevertheless, all possible corrections are equivalent up to stabilizers, provided that the original cluster was smaller than the lattice size and cannot contain a logical operator entirely within the cluster. This means that the success or failure of the union-find decoder depends mostly on its capability to find a set of clusters that contain the correction without growing any cluster beyond the lattice size. When this happens, the peeling decoder can easily introduce recovery operations that lead to a logical error when combined with the error, as the result of the possible corrections now can be equivalent up to stabilizers either to the application of a logical operator or to the identity. This would depend on the generation of the spanning tree,

which is a step that does not take into consideration the position of the excitations and would thus lead to a high failure rate. In the following section, we study the percolation threshold for the clusters depending on the physical error rate, and how it can be related to the critical error probability for the threshold for QEC.

### 7.1.2 Results and analysis

In this section, we show the results from our implementation of the union-find algorithm on the surface code. We evaluate the performance of the algorithm by estimating numerically the code capacity threshold, using Monte Carlo simulations to estimate the logical error rate as a function of the physical error rate and the size of the lattice. Furthermore, we discuss the limiting factors of the performance of the decoder and study the percolation threshold for the growing clusters in the algorithm.

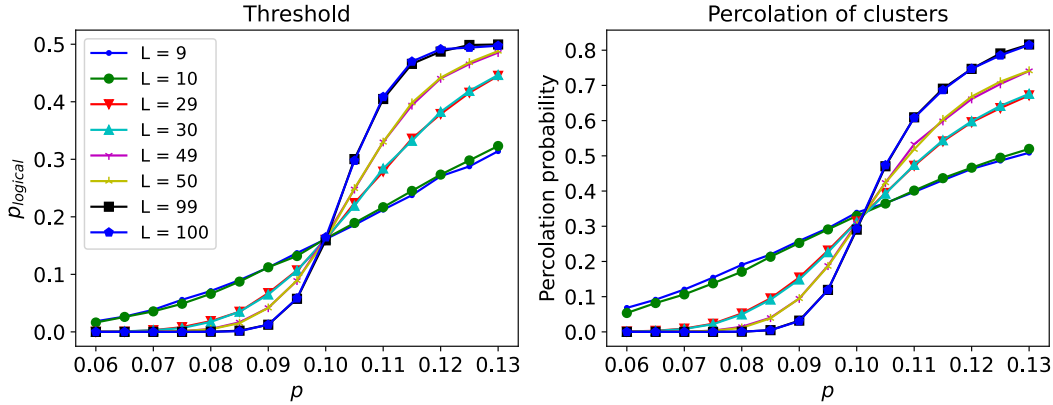
We estimate the logical error probability  $p_{log}$  for different lattice sizes of distance  $L$  and physical error rates  $p$ . We obtain this estimate by generating multiple samples in which we generate an error configuration  $\{e_i\}$  by applying a phase error on each qubit with probability  $p$ , we measure the syndrome with ideal measurements and we apply the decoder to obtain a recovery operation  $\{r_i\}$ . We then check if the combination of the error and the recovery operation is equivalent up to a product of stabilizers to the identity operation (success) or to the logical  $Z_L$  operator of one of the encoded qubits, which corresponds to a failure on that encoded qubit. Since there are two encoded qubits, we consider the logical error rate of the code to be the average of their logical error rate. We estimate the error for each point  $\sigma_p$  using the error of a binomial distribution  $\sigma_p = \sqrt{\frac{p_{log}(1-p_{log})}{N}}$ , where  $p_{log}$  is our estimate for the logical error rate and  $N$  is the number of samples for each point.

The threshold  $p_c$  corresponds to the physical error rate such that, if  $p < p_c$ , then we can arbitrarily reduce the logical error rate by increasing the size of the lattice. We can estimate this threshold as the crossing point of the logical error rate  $p_{log}(p, L)$  as a function of the physical error rate  $p$  for different lattice sizes  $L$ , i.e.  $p_{log}(p_c, L_A) = p_{log}(p_c, L_B) \forall (L_A, L_B)$  with  $L_A \neq L_B$ . To estimate this point from the results of the simulations, we use a linear interpolation to estimate the logical error rate between our samples, and we compute and estimate of the crossing point  $\tilde{p}_c(L_A, L_B)$  for each pair of lattice sizes  $L_A$  and  $L_B$ :  $p_{log}(\tilde{p}_c(L_A, L_B), L_A) = p_{log}(\tilde{p}_c(L_A, L_B), L_B)$ . Finally, we estimate the threshold  $p_c$  as the average of all crossing points between all pairs of lattice

## 7. UNION-FIND DECODER FOR COLOR CODES

sizes  $p_c \simeq \sum_{i=1}^{N_L-1} \sum_{j=i}^{N_L} \tilde{p}_c(L_i, L_j)$ , where  $N_L$  is the number of lattice sizes considered, and  $L_i$  is each of the lattice sizes we sampled.

Using our implementation of the union-find decoder and considering only phase errors (as phase and bit flip errors can be corrected independently in the surface code), we obtain a code capacity threshold of 10.0%. This result is slightly higher to the 9.9% obtained in the original paper [193]. Although the difference is within the error bars of our result, this small difference could be explained by potential differences in the details of the implementation of the cluster growth algorithm, and the generation of the spanning tree. The results of our simulations are shown on Fig. 7.5.



**Figure 7.5: Percolation and logical error thresholds on the surface code with the Union-find algorithm.** On the left, we show the percentage of clusters that percolate through the lattice (i.e. a logical operator can exist within the cluster) as a function of the physical error rate  $p$ . On the right, we show the logical error rate after decoding as a function of the physical error rate. Since there are two logical operators (vertical and horizontal) on the toric lattice, we consider the presence of any logical operator as a logical error. We find that the thresholds for percolation and logical failure are both  $\simeq 10.0\%$ , which might indicate a close relation between both properties of the decoder. Each point was obtained using  $2 \cdot 10^4$  Monte Carlo samples, and the error bars are smaller than the marker size.

The union-find decoder consists on two main steps: the cluster growth and the peeling decoder. As discussed in the previous section, the possible solutions given by the peeling decoder are always equivalent up to stabilizers once the clusters are fixed, as long as no cluster is larger than the size of the system. In other words, the possible corrections after the cluster growth step are equivalent if no cluster can *percolate* through the lattice, in which case a logical operator can be applied entirely within a cluster. We say that a cluster can percolate through the lattice if it is possible

to find a path of connected qubits that extends from one side of the lattice to the opposite side, with all qubits in the path belonging to the same cluster. If the clusters do not percolate and form isolated *islands* within the code, any correction will always correspond to closed loops within the clusters. This means that the cluster growth step is a critical part of the algorithm, and an over-growth of the clusters can lead to a reduction in the threshold.

In order to check this hypothesis, we estimate numerically the percolation threshold for the clusters grown by the union-find algorithm. We define the percolation threshold  $p_c^{perc}$  as the physical error rate for which, if  $p < p_c^{perc}$ , then the probability that a cluster will percolate  $p^{perc}(p, L)$  through a lattice of distance  $L$  tends to zero as we increase the size of the lattice  $\lim_{L \rightarrow \infty} p^{perc}(p < p_c^{perc}, L) = 0$ . Using Monte Carlo simulations, we estimate the probability that a cluster percolates through the lattice as a function of the physical error rate  $p^{perc}(p, L)$ . We consider that a cluster percolates through the lattice if it is possible to find a path from one side of the lattice to the opposite side contained entirely within the cluster. We apply a standard Depth First Search algorithm to find the existence of such a percolating path [363]. For the toric code, we can check if a cluster percolates both in the vertical and horizontal directions, which leads to two estimates of the percolation probability. Due to the symmetry of the lattice, both percolation rates should represent the same magnitude, and we estimate the percolation rate of the lattice as the average of the percolation rate in the horizontal and vertical directions. We estimate these percolation rates  $p^{perc}(p, L)$  by averaging over multiple error configurations. We estimate the errors on these estimates as the error for a binomial distribution  $\sigma^{perc} = \sqrt{\frac{p^{perc}(1-p^{perc})}{N}}$ , where  $N$  is the number of samples.

To estimate the percolation threshold, we estimate the crossing point between the  $p^{perc}(p, L)$  curves for different lattice sizes, using the same method described for the code capacity threshold, and considering a linear interpolation between the sampled points. In our simulations, we find that the percolation threshold for the clusters generated by the union-find algorithm is 10.0%. This is the same result that we obtained for the code-capacity threshold, which supports our hypothesis that the presence of percolating clusters is the limiting factor in the performance of the decoder. Once the clusters percolate through the lattice, the peeling decoder is unable to distinguish between different corrections, and the failure probability becomes increasingly high.

## 7. UNION-FIND DECODER FOR COLOR CODES

---

In the MWPM algorithm this problem is avoided, as all possible connections between the stabilizer excitations are always considered, at the price of a higher computational complexity and therefore a slower performance.

We have studied the application of the union-find decoder on the surface code, and we have seen how the percolation of the clusters can limit the performance of the algorithm. In [193, 196], the authors propose two methods for applying the union-find decoder on color codes by mapping the color code lattice into multiple surface-like codes [192] and applying the union-find algorithm on those additional codes. These lattices, also called the *restricted lattices*, are obtained from the color codes by removing the stabilizers corresponding to one of the colors. Errors in the original lattice generate pairs of excitations in the restricted lattices, which allows surface-code decoders to find a matching for those excitations. The best results are obtained using the algorithm described in [196], which, in a simplified explanation: i) generates two restricted lattices that map the information from the color code syndrome, ii) applies the union-find algorithm on each restricted lattice to obtain a correction and iii) combines the corrections from both restricted lattices using a local rule to find the final recovery operation. Using this approach, the authors obtained a threshold of 9.8%. In the next section, we present an alternative adaptation of the union find algorithm for color codes that avoids the need of the restricted lattices, working directly on the original color code lattice, and study the difference in performance with respect to the algorithms presented in [193, 196].

### 7.2 Extension to color codes

In this section, we present an efficient color code decoder based on the ideas from [193]. The computational complexity of the decoder is almost linear, and it is designed for depolarizing noise in any of the three color code lattices. In particular, we present the algorithm on the 6.6.6 (hexagonal) and 4.8.8 (square-octagon) lattices, but the algorithm is easily extensible to the 4.6.12 lattice.

The decoder's algorithm works by growing clusters of *dangerous* positions around the stabilizer excitations. In this process, we limit the set of positions in which the correction will be localised. Finally, with a peeling decoder, we can find a correction

within each cluster. The working principle of the algorithm is analogous to the Union-find decoder for surface codes. However, a new definition of parity and adaptations to the cluster growth and the peeling decoder need to be made for the decoder to work in color codes. As an alternative solution, in [193] and the more recent [196] the authors apply the decoder to the color codes only by first mapping color codes into several copies of surface codes [192].

The main steps of the algorithm are:

1. Initialize clusters around each excitation in the syndrome. Create a list with the odd-clusters.
2. While there are odd-clusters remaining:
  - (a) Choose and grow the smallest odd cluster.
  - (b) Merge connected clusters and update the parity.
3. Apply the peeling decoder on all the remaining clusters:
  - (a) For each cluster, find a spanning tree within the set of qubits.
  - (b) Start peeling the tree by *cutting* qubits on the leaves of the tree. Apply a correction if there is an excited stabilizer.

These main steps are the same as in the original decoder for the surface code. However, some adjustments need to be done for the decoder to work on color codes.

In Sec. 7.2.1 we explain the details on the cluster growth in color codes, the condition for an even parity and the merging of clusters. In Sec. 7.2.2 we explain how to apply a modified peeling decoder for the clusters to find the final correction. Finally, in Sec. 7.3 we show the performance of the decoder and analyze the limiting factors that affect the threshold. We also compare the performance with the alternative applications of the union find decoder on color codes described in [193, 196].

### 7.2.1 Growing and merging clusters

The union find algorithm for color codes is a generalization of the decoder for surface codes. The main working principle behind the algorithm is that, given a cluster on the lattice, if the parity of the excitations within the cluster is odd, that means that at

## 7. UNION-FIND DECODER FOR COLOR CODES

---

least one chain of errors ends outside of the cluster. Therefore, if we initialize a small cluster around each excitation on the code, we can grow and merge the clusters. As long as a cluster has an odd parity inside the cluster, there is a chain of errors that ends outside of the cluster. Therefore, we should keep increasing its size until all the remaining clusters have even parity. At the end, we know that the correction can be within the clusters. In this section, we focus on the details and changes to the algorithm necessary for the particularities of the color code lattice.

### **Parity:**

First, the condition for a cluster to be considered even/odd needs to be changed, as there are error patterns that generate an odd number of excitations. This means that counting the number of excitations is not enough to determine if a correction can be found within a cluster. Instead, we need a more precise definition for the parity of a cluster. In the color code, stabilizers have a *color* assigned. This can be stored as three binary numbers  $\{R, G, B\}$  for each cluster. The parity of each one of these numbers corresponds to the number of excitations of each color, modulo 2. From these three numbers, we can define a condition for a cluster to be considered even/odd. In the color code, excitations are created either by pairs of the same color, or by triads of red, blue and green. Therefore, we will consider a cluster to be *even* when its parity is either  $\{R, G, B\} = \{0, 0, 0\}$  or  $\{R, G, B\} = \{1, 1, 1\}$ , since that will mean that it is possible to find a correction that will satisfy the syndrome condition contained entirely within the cluster.

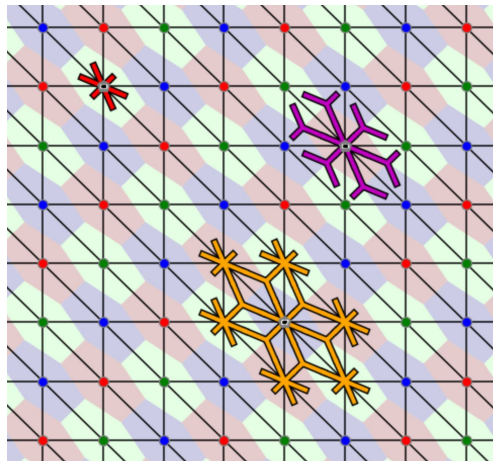
### **Growing clusters:**

The second change we need to include is on the algorithm to grow the half edges. In the surface code, half edges were grown from every stabilizer in the cluster. Edges that were already half grown, would be completed as a full edge in the next growing step, including the next stabilizer in the cluster. This was possible since every qubit is connected to only two stabilizers, and therefore forms an edge in the graph. In the color code, the situation is different, as qubits are no longer edges, but triangles, since qubits in the color code are connected to three different stabilizers, one of each color.

To grow the boundary of a cluster in the color code, we consider two types of boundary: a stabilizer node and a qubit node. Instead of keeping a list of the edges



and half edges of the cluster's boundary, the boundary is defined as a collection of stabilizers and qubits. In the growth step, the stabilizers in the boundary will grow the cluster by including the neighboring qubits. In a similar way, the qubits in the boundary will grow the cluster by including the neighboring stabilizers. Notice that this growth scheme is equivalent to growing half an edge in the surface code, since the connection between qubits and stabilizers in the surface code corresponds to half edges in the lattice. On figure 7.6 we can see an example of the different steps of cluster growth on the color code.



**Figure 7.6: Example of cluster growth on color codes.** In this figure, we can see different steps of the cluster growth, from a single stabilizer growing to the neighboring qubits (top, red), to growing the next step towards the neighboring stabilizers (right, purple), to growing a step further to the next qubits (bottom, orange).

### 7.2.2 Peeling decoder on color codes

In this section, we present an alternative version of the peeling decoder on color codes, adapted for the particular properties of the color code lattice. As we have seen in the previous section, the resultant graph from growing the clusters in the color code has some important differences with respect to the surface code. The most relevant difference regarding the behavior of the peeling decoder stems from the fact that the connectivity of the qubits in the color code is three. This change will affect the peeling algorithm, as the process of *cutting a leaf* and applying a correction to a qubit will change the parity of three stabilizers instead of simply displacing the excitation along

## 7. UNION-FIND DECODER FOR COLOR CODES

---

the branch of the spanning tree.

### **Finding the spanning tree:**

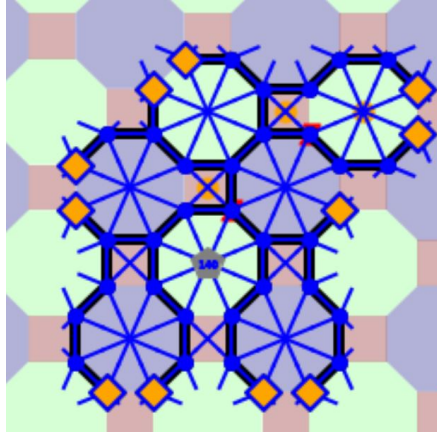
In the surface code, the spanning tree was formed by a set of stabilizers connected by the qubit-edges. The leaves of the spanning tree are the qubit-edges with only one connection to the rest of the tree. In the color code, the spanning tree is formed only by qubits, and the stabilizers are not part of the tree. The edges of this tree correspond to the edges of the stabilizers in the standard representation of a color code (with stabilizers drawn as coloured plaquettes and qubits in the vertices of the stabilizers). An alternative visualization for the connectivity of the qubits is to represent the code in the reciprocal lattice, where qubits correspond to triangles and the stabilizers are the vertices. In the reciprocal lattice, the spanning tree connects the centers of the triangle-qubits that belong to the tree. The neighborhood of a qubit corresponds to the qubits that share an edge. In Fig. 7.7 we show an example of the spanning tree of a cluster on the 4.8.8 lattice.

Once we understand the connectivity of the qubits on the spanning tree, the generation of the spanning tree is analogous to the algorithm for the surface code. We run a search algorithm (BFS) starting on any qubit belonging to the cluster. The search algorithm explores the cluster and adds the qubits to the spanning tree. However, it only adds an edge to the tree if the qubit was unexplored. This condition avoids the formation of cycles and allows the algorithm to add all the qubits of the cluster to the tree, generating a spanning tree for the cluster. In addition, it stores a list of the qubits with only one connection to the tree. These qubits are the leaves of the tree, from which the peeling decoder starts the decoding process. Since each qubit on the cluster is only explored once, the algorithm runs in  $\mathcal{O}(N)$  and does not increase the complexity of the decoder.

### **Peeling the tree:**

In the surface code, when peeling the tree we look at the stabilizer at the end of the qubit-leaf. If it was excited, then we change the value of the stabilizer at the base and apply a correction. In the color code, the spanning tree does not have any connection to the stabilizers. In addition, a correction on a qubit affects the sign of three different stabilizers.

The peeling process on the color code clusters is applied in the following steps:



**Figure 7.7: Spanning tree of qubits within a cluster.** A spanning tree can be created by linking all qubits in a cluster and avoiding loops. In the figure, the spanning tree is defined by the thick black lines, and the thin blue lines represent the connectivity of qubits and stabilizers from previous steps. The tree can be *peeled* by following the leaves, marked with orange diamonds.

1. First, we pick a qubit (the leaf) connected to only one other qubit in the spanning tree.
2. The leaf shares two stabilizers with the qubit of the spanning tree to which it is connected. If the remaining stabilizer of the leaf has odd parity, a correction is applied on the qubit-leaf. Notice that the parity of the other two stabilizers will be addressed on future pruning steps.
3. The leaf is removed from the tree.
4. The first two steps are repeated until there is only one qubit remaining on the tree.
5. There should be either 3 or 0 excitations remaining in the cluster. A correction is applied on the last qubit of the tree if there are exactly 3 excitations around the qubit, removing them and finishing the correction of the cluster.

### 7.2.3 Implementation details

Now that we discussed the main differences, we can proceed with a step-by-step explanation of the decoder.

## 7. UNION-FIND DECODER FOR COLOR CODES

---

1. First, we initialize the main data structures. That is, we create a cluster for each excitation in the stabilizers, and we initialize the forest data structure. The forest is an array that contains, for each stabilizer in the code, which node is the root of the cluster it belongs to. It is a quick way to find if two nodes belong to the same cluster or not, and the base for the union-find algorithm for disjoint sets [362]. For nodes that do not belong to any cluster, the initial value is set at -1. We initialize a similar data structure for qubits. The last data structure to initialize corresponds to the odd clusters. This list is a priority queue that contains both the root of the clusters and the size of their boundary. The priority queue is an efficient way to find the smallest cluster [363], which is necessary to start the cluster growth process.
2. The next step is to start growing the clusters. In a while loop, we will be choosing the smallest odd cluster in the queue and growing the boundary. We create an empty *fusion* list. for each stabilizer in the boundary of the cluster, we add the neighboring qubits; and for each qubit in the boundary, we add the neighboring stabilizers, as explained in the previous section (Fig. 7.6). Before adding the new qubits and stabilizers to the cluster, they are added to a fusion list. Once all boundary nodes have been grown, we loop over the fusion list to check if the new nodes are part of a different cluster, in which case we have to merge the clusters.
3. The process of merging the clusters is very simple. We simply combine the sets containing the qubits, stabilizers and boundaries, and update the parity of the cluster. The last step is to update the tree, where each node contains information about its root cluster. Instead of updating the root of each node in the small cluster to point at the new root, we only need to point the root of the small cluster towards to new root.
4. After growing the cluster, and after the possible merging events, we check its parity. If the parity is odd (which for the color code, as explained, means that there is at least a chain of errors that leads outside of the cluster), then we add this cluster back to the odd cluster queue, in which its priority to be grown again will depend of the size of its boundary. If there are odd clusters remaining, we will keep growing them until all of the clusters have even parity.

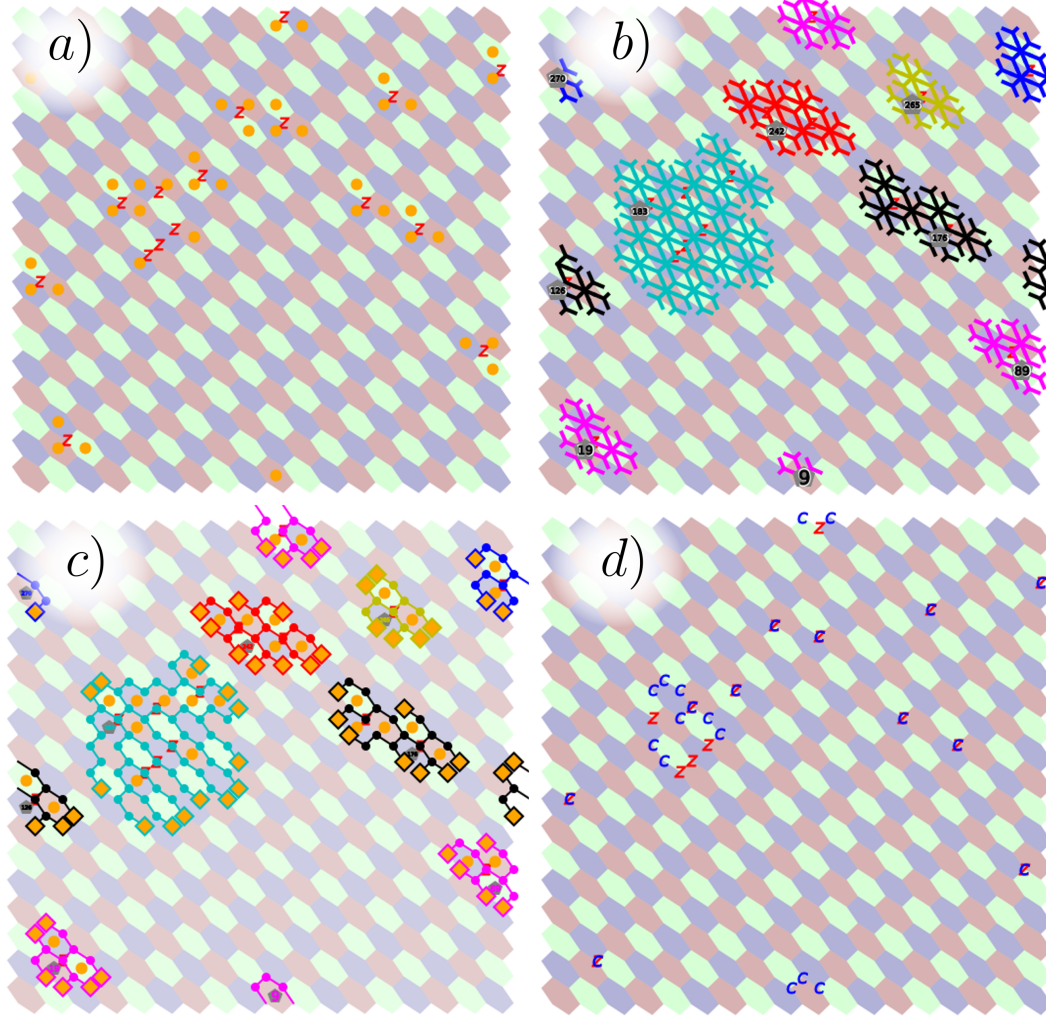
5. Once all clusters have even parity, we know that a correction can be found entirely within each cluster. The next step is to apply the peeling decoder, as described in Sec. 7.2.2. Therefore, for each cluster, we need to find a spanning tree that links all qubits in the cluster without any loops. We can do this by running a simple search algorithm, like Breadth First Search (BFS). Starting in the root of the tree, we grow a tree by expanding the network through neighboring qubits. This allows us to find a spanning tree that will contain all qubits in the cluster. In addition, this growth process will provide us with a list of the *leaves* of the tree, from which we will start pruning in the next step. An example spanning tree can be seen in Fig. 7.7.
6. Finally, the last step of the decoder is to *peel* the trees to find the correction. In a similar spirit as the peeling decoder for the surface code, we look at the value of the stabilizer at the end of the leaf. If the stabilizer has an odd parity, we add a correction to the qubit and update the surrounding stabilizers before cutting the qubit-leaf. By applying this iterative process we can find a correction for the syndrome within the cluster. Although the parity of the stabilizers in the cluster changes as we cut the tree and add corrections, the last update that we do on each stabilizer is done by the last leaf that points at that stabilizer. This means that corrections will be applied to ensure that the parity of the stabilizer will end as even. After peeling all the clusters in the code, the decoding process is complete.

An example of the full process can be seen in Fig. 7.8.

### 7.3 Performance of the decoder

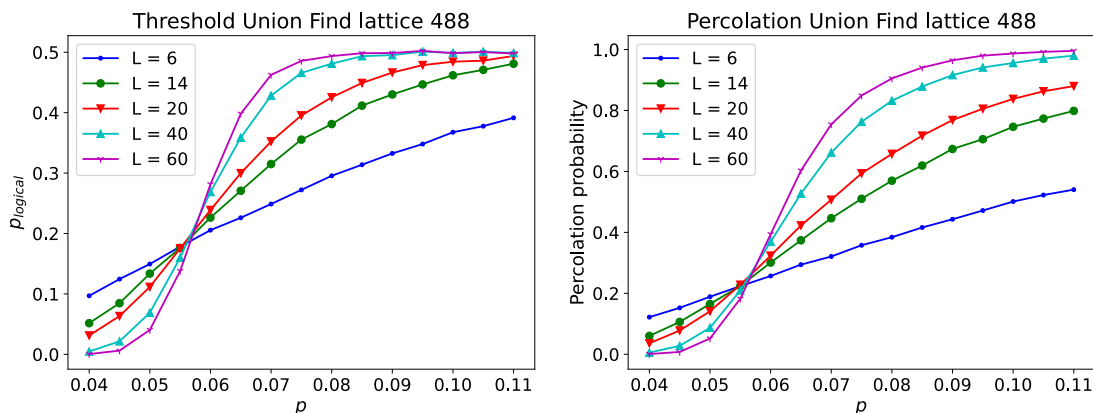
In this section, we present the numerical results of the performance of the decoder on color codes, using our variation of the original algorithm. We compute an estimation of the threshold of the decoder using Monte Carlo simulations, as in Sec. 7.1.2. The results for the color code lattices 4.8.8 and 6.6.6 are shown in Figures 7.9 and 7.10, respectively. On the 4.8.8 lattice, we obtain a threshold around 5.6%, while for the 6.6.6 lattice our estimate is 7.0% for code capacity noise.

## 7. UNION-FIND DECODER FOR COLOR CODES



**Figure 7.8: Full execution of the Union Find decoder.** **a)** Initial error (marked with a red  $Z$  on the qubits) and syndrome (orange dots over the excited stabilizers). **b)** Grown clusters around the excited stabilizers. The original root of the cluster is marked in grey. Different clusters are drawn in different colors. **c)** Spanning trees of the clusters, generated by running a BFS algorithm. This process removes the loops in the clusters from the previous step. The leaves of the trees are marked with orange diamonds. **d)** Correction from the peeling decoder. The qubits with a correction are marked with a blue  $C$ , and the original errors with a red  $Z$ . In this case, the original error was corrected up to stabilizers.

The thresholds obtained with our variation of the algorithm proposed in [193] are significantly lower than the threshold obtained using the original version. To understand this discrepancy, it is useful to analyse the behavior that leads to the lower threshold on our variation with respect to the approach presented in [193, 196]. In par-



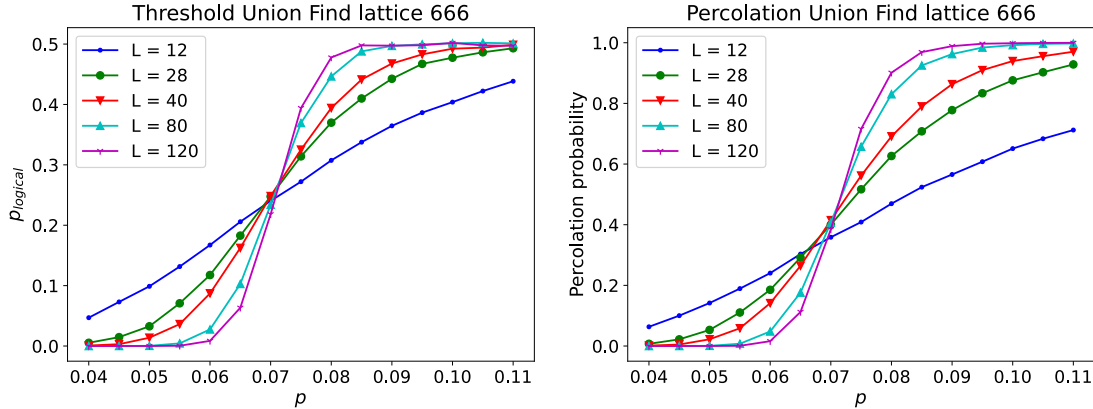
**Figure 7.9: Percolation and logical error thresholds on the color code 4.8.8 lattice with the Union-find algorithm.** On the left, we show the logical error rate after decoding as a function of the physical error rate  $p$ . Since there are two logical operators (vertical and horizontal) on the toric lattice, we consider the presence of any logical operator as a logical error. On the right, we show the percentage of clusters that percolate through the lattice (i.e. a logical operator can exist within the cluster) as a function of the physical error rate. We find that the thresholds for percolation and logical failure are both  $\simeq 5.6\%$ , which might indicate a close relation between both properties of the decoder. Each point was obtained using  $2 \cdot 10^4$  Monte Carlo samples, and the error bars are smaller than the marker size.

ticular, the under-performance of the decoder can be understood by the rapid growth of the clusters. The high connectivity of the qubits and stabilizers in color codes leads to a quick growth in the boundary of the clusters after very few growth-steps. This, in turn, leads ultimately to the percolation of the cluster over the lattice, i.e. the cluster grows larger than the side of the lattice, and it becomes possible to apply a logical operator entirely within the cluster.

To support this hypothesis, we estimated the percolation threshold of the clusters in the color code with our union-find algorithm. Using Monte Carlo simulations, we estimated the percentage of percolating clusters as a function of the single qubit error probability. We consider that a cluster percolates if it is possible to find a path from one side of the lattice to the opposite one that lies entirely within the cluster. We find that the percolation threshold coincides with the threshold for code capacity both in the 6.6.6 lattice and in the 4.8.8 lattice.

In Fig. 7.11, we show the results of the computational time required for the decoder for different lattice sizes  $L$ , averaged for the different error rates  $p$  sampled in Fig. 7.10. Note that the number of qubits scales as  $L^2$  with the code distance in the toric code.

## 7. UNION-FIND DECODER FOR COLOR CODES



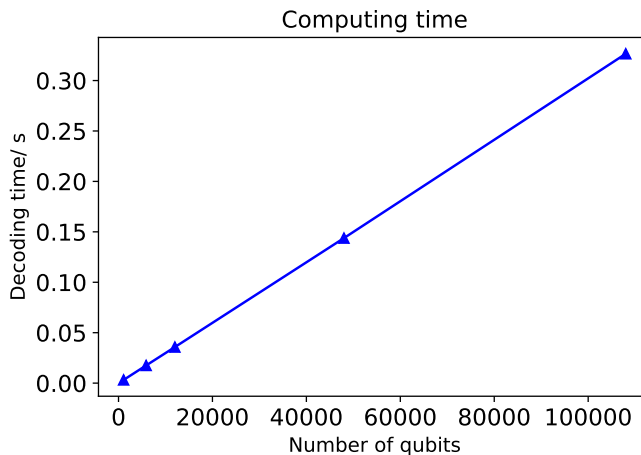
**Figure 7.10: Percolation and logical error thresholds on the color code 6.6.6 lattice with the Union-find algorithm.** On the left, we show the logical error rate after decoding as a function of the physical error rate  $p$ . Since there are two logical operators (vertical and horizontal) on the toric lattice, we consider the presence of any logical operator as a logical error. On the right, we show the percentage of clusters that percolate through the lattice (i.e. a logical operator can exist within the cluster) as a function of the physical error rate. We find that the thresholds for percolation and logical failure are both  $\simeq 7.0\%$ , which might indicate a close relation between both properties of the decoder. Each point was obtained using  $2 \cdot 10^4$  Monte Carlo samples, and the error bars are smaller than the marker size.

We run these simulations on an Intel Core i7-8650U CPU processor with 1.90GHz. The results are consistent with the predicted  $\mathcal{O}(N\alpha(N))$  scaling, comparable to the most efficient non-parallelized decoders present in the literature [193, 196], which present the same scaling.

The adaptation of the union-find decoder proposed in the original paper [193] projects the color code onto three restricted lattices [192], and then applies the union-find decoder on each of the resulting surface codes. Since the qubits and stabilizers on the surface code have a much smaller connectivity, the percolation threshold is larger and stops becoming the bottleneck of the decoder. Using this mapping, they obtain a threshold of 8.4% on the 6.6.6 lattice. This decoder, however, failed to produce a solution for all error cases, and the threshold is still below the one for surface codes.

Interestingly, in [196] the authors presented an improved adaptation of the decoder for color codes. The main difference from the previous work lies in the use of the restricted lattices. Instead of applying the decoder on the three restricted lattices, the authors designed an algorithm that achieves a correction using only two of them, as outlined in Sec. 7.1.2. Using this variation, the authors found a threshold of 9.8%,





**Figure 7.11: Scaling of the computing time.** In the figure, we show the computational time required by the decoder to produce a correction for different lattice sizes, averaged over the sampled error rates in Fig. 7.10. The results show a nearly linear scaling, which is consistent with the theoretical *almost linear* scaling  $\mathcal{O}(N\alpha(N))$ . Each point was obtained from  $3 \cdot 10^5$  samples.

which can be improved to 10.2% by replacing the union-find step on the surface codes with the standard MPWM decoder.

## 7.4 Summary and outlook

In this chapter, we have explained our work on the study of the union-find decoder. We have presented an original, alternative adaptation of the algorithm that applies the basic principles of the decoder on the particularities of the color code lattices. We build on the work from [193] and extend it by an analysis of the percolation threshold of the clusters grown in the algorithm. We found that the percolation threshold of the clusters generated during the decoder process is related to the threshold obtained from the decoder which, we hypothesize, explains the limits on the performance of the algorithm.

The algorithm presented is highly efficient, with an almost-linear computational complexity. Furthermore, the decoder could be adapted to include additional sources of error, like qubit losses [364], thanks to the flexibility of the cluster growth. However, the threshold of the decoder can prove a limiting factor, as it lies below the threshold found in other competing decoders [193, 196]. In particular, the so called restriction decoder

## 7. UNION-FIND DECODER FOR COLOR CODES

---

described on [196] stands as the best alternative, as it outperforms other competing color code decoders both in threshold and efficiency. In addition, this decoder can also potentially be adapted for phenomenological noise, erasure, and qubit losses. This effort has already shown promising results in more recent works, e.g. [237].

From this work, we want to highlight the fact that the color code lattice still presents some properties worth studying. The analysis of the failure cases of this decoder can bring additional insight about the behavior of decoding algorithms and the limitations of different decoding techniques.

## Chapter 8

# Rescaling decoder

## 8. RESCALING DECODER

---

In this chapter, we describe our work on the renormalization group (RG) decoder for the 4.8.8, 2D color code lattice with periodic boundary conditions, which resulted in the publication [2]. We develop a decoding algorithm based on the ideas of rescaling decoders proposed in [187–189], and obtain a code capacity threshold of 6.0%.

We begin Sec. 8.1 by contextualizing briefly the project, explaining and clarifying the fundamental concepts required to understand the decoding problem, as well as the motivation behind the work and an intuition on the renormalization group (RG) decoding paradigm. Then, in Sec. 8.2, we describe in detail the different steps and subroutines involved in the decoding algorithm. In Sec. 8.3, we show the results of our numerical simulations, from which we obtain the code capacity threshold of the decoder using finite-size scaling analysis. The chapter concludes with a short summary and an outlook for future extensions of the work.

All the work in this section stems from code that I personally wrote, including the decoder algorithm, as well as the numerical simulations and analysis presented in this chapter. I also derived the mathematical formulations required for the application of the decoder to the 4.8.8 lattice, explained in sections 8.2.4 and 8.2.5.

### 8.1 Background

The work presented in this chapter is motivated by the long-term goals of FTQEC, and the use of large-scale topological QEC codes that allow the detection and correction of errors during storage and processing of quantum information [100, 109]. If we use FT quantum circuit constructions [127, 130], the threshold theorem ensures that the logical error rate can be arbitrarily suppressed by increasing the size of the code, as long as the physical error rates fall below a given critical error threshold [127, 129, 130]. The three main factors that influence the value of this error threshold are the choice of QEC code, the noise model, and the decoder algorithm used, which determines our capability to correctly interpret the syndrome (i.e. the information obtained from the parity checks realized during the QEC protocol) to infer a recovery operation with a high success probability.

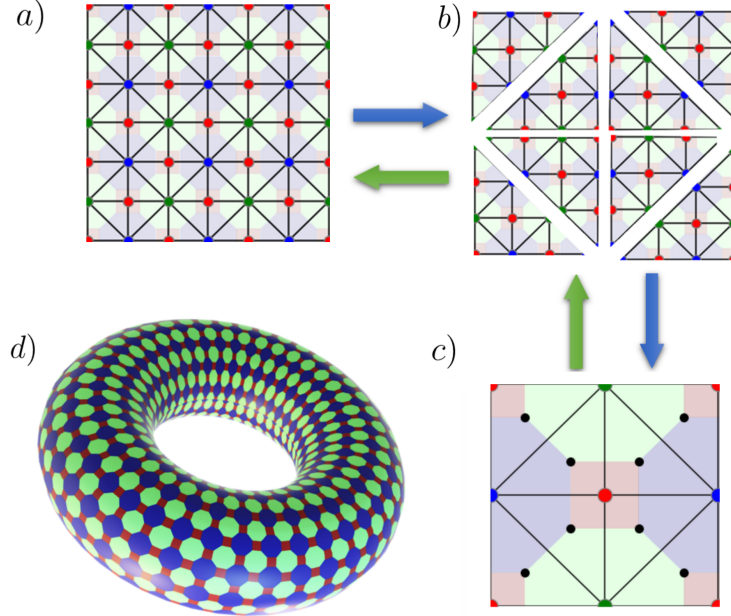
In this chapter, our focus lies on the development of an efficient decoder algorithm for color codes, and the 4.8.8 color code lattice in particular. As seen in the background (chapter 3), color codes are topological QEC codes that encode the logical information

into topological properties of the system, and use parity checks that involve low-weight local measurements on qubits located in neighboring locations on the lattice [113, 132]. To execute operations on the encoded logical qubits, the logical gates must operate fault tolerantly, i.e. the circuits that implement the operations should act in such a way that they cannot inadvertently spread errors beyond the scope of what the QEC code is able to correct. The most straightforward way to avoid this spreading is by acting separately on the data qubits within the code block, which is known as a transversal implementation of a logical gate. However, the possibility of implementing logical gates transversally is limited in first instance by the choice of QEC code, and ultimately by the no-go theorem by Eastin and Knill [223], which dictates that it is impossible to implement a universal gate set fault tolerantly. The non-transversal implementation of logical gates in a fault tolerant manner is a resource intensive procedure, and it is typically approached by the use of magic state distillation and injection [224–227, 237]. The 4.8.8 color code lattice is remarkable because it allows for the transversal implementation of the entire Clifford group [113], which distinguishes it from surface codes or color codes on hexagonal lattices. It is therefore optimal in the sense that adding any other (non-Clifford) gate would render the gate set universal and hence violate the no-go theorem.

To operate large distance codes, it is crucial to have an efficient decoding algorithm with a high success rate of the recovery operations proposed by the decoder. This rate is reflected on the threshold value, for which the upper bounds can be obtained (for some error models) through mapping the QEC problem onto a classical statistical-mechanical model [111, 218, 220]. This accuracy of decoding has to be balanced with the computational time required to run the decoding algorithm. Although for a quantum memory it is potentially fine to keep a record of the measured error syndromes and compute the recovery operation in a classical postprocessing stage, the situation is different once we start to perform logical quantum computations. For this case, the intermediate state of the computation depends on the decoder outcome, and the quantum computation may have to wait for the decoding algorithm to finish, which slows down the quantum computation and allows for new errors accumulate during that time.

The development of decoders for color codes is an active field of research, and several decoding algorithms have been proposed for different noise models [189–197, 201, 250, 251]. In this work, we concern ourselves with code capacity noise, i.e. data

## 8. RESCALING DECODER



**Figure 8.1: Sketch of a rescaling step in the 4.8.8 color code lattice.** Qubits are represented by triangular faces, and the coloured vertices represent the stabilizers. The initial lattice **a)** can be split into multiple cells **b)**, which can be decoded locally. Each triangular cell can then be mapped into a single effective qubit in a rescaled lattice **c)**. This process of rescaling can be repeated until the final lattice is small enough to apply a brute force decoding. Corrections on the smaller lattices can then be backpropagated to the original lattice, finding the final correction. **d)** In this work we consider a 4.8.8 color code lattice, where each qubit is involved in one 4-qubit stabilizer associated to a square plaquette and in two 8-qubit stabilizers on octagonal plaquettes, with periodic boundary conditions. The lattices considered host codes with parameters  $[[n, k, d]] = [[8 \cdot 9^m, 4, 2 \cdot 3^m]]$ , where  $n$  is the number of qubits in the lattice,  $k$  the number of logical qubits,  $d$  is the distance of the code, and the integer  $m$  denotes the number of rescaling steps. The blue arrows represent the initial order of operations, in which the lattice is rescaled until the smallest system size is reached. Green arrows represent the backpropagation process, from the smallest rescaled lattice back to the original code.

qubit noise, for which the threshold's upper bound is known to be 10.9% [218, 220]. Currently, the decoder with the best known performance for code capacity noise in terms of the threshold is the restriction decoder, which achieves 10.2% when using MWPM as a subroutine [196], and has the runtime complexity of MWPM ( $\mathcal{O}(N^4)$  for a straightforward implementation of the original Blossom algorithm [185, 247], which can be optimized to  $\mathcal{O}(N^{2.5})$  using recent advancements [248, 249]). The restriction decoder can also be used with a union find decoder as subroutine [193], which achieves

an estimated 9.8% threshold [196] with a worst case complexity of decoding that scales as  $\mathcal{O}(N\alpha(N))$ . In chapter 7, we explore a variation of the union-find approach.

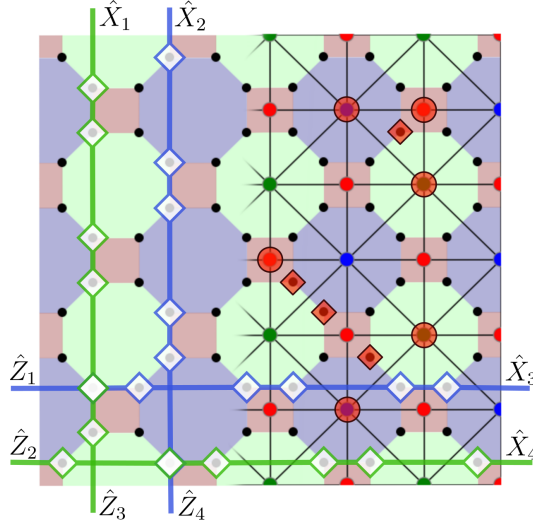
In this chapter, we consider a decoding paradigm based on local decoding and iterative rescaling of the lattice known as renormalization-group (RG) decoding. This paradigm was initially introduced in [188] for surface codes, and extended to hexagonal color code lattice (6.6.6) in [189]. These algorithms have a runtime complexity of  $\mathcal{O}(N \log(N))$ , as the rescaling of the lattice can be done in linear time  $\mathcal{O}(N)$ , since it is based on local operations, and the number of iterations of the rescaling process grows as  $\mathcal{O}(\log(N))$ . The local nature of the rescaling algorithm means it can be parallelized, achieving an overall scaling of  $\mathcal{O}(\log(N))$ , which holds the potential for drastic improvements in decoding runtime.

As introduced in Ch. 3, color codes are stabilizer codes [113, 132, 365] defined on face-three-colorable trivalent graphs, (see e.g. Fig. 8.2 for the case of the 4.8.8 lattice coloring, as well as some examples on different error cases). Qubits are identified with the vertices and each face  $i$  of the graph defines two stabilizer generators  $S_X^{(i)}$  and  $S_Z^{(i)}$  involving all vertices in the boundary of the respective face [113, 365]. In this chapter, we consider as well an alternative representation of color codes using the reciprocal lattice, in which qubits are represented by triangles, and stabilizer operators are represented by the vertices (see Fig. 8.2). These stabilizers involve purely  $X$  or  $Z$  Pauli operators, which makes color codes part of the Calderbank-Shor-Steane (CSS) code family [213, 214]. Moreover, we consider periodic boundary conditions, which implies that two independent qubits can be encoded in each of the two independent non-trivial loops of the resulting torus (see Fig. 8.2), leading to a total of four logical qubits. As a part of the CSS stabilizer codes, it is possible to detect and correct phase-flip and bit-flip errors separately by using the syndrome from the  $X$  and  $Z$  stabilizers, respectively. Throughout this work we focus on independent bit- and phase-flip noise. One of the symmetries of the color code is a self-duality under exchanging  $X$  and  $Z$  stabilizers, which in particular implies that we can focus purely on bit-flip errors to obtain the code capacity threshold.

### 8.1.1 Introduction to the rescaling decoding approach

In this section, we introduce a qualitative description of the decoder algorithm, with the purpose of providing a general picture of the decoder and the basic underlying

## 8. RESCALING DECODER



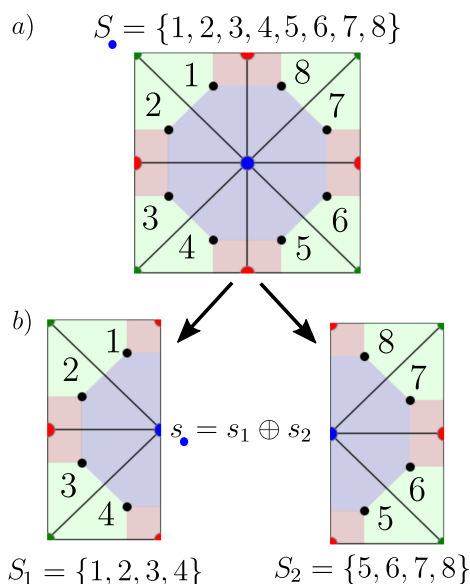
**Figure 8.2: Representation of the color code 4.8.8 lattice.** On the primal lattice (left), qubits are represented by black vertices, and the stabilizers  $S_Z$  and  $S_X$  are represented by the colored plaquettes, which apply a parity check over the qubits on the vertices. In the reciprocal lattice (right), the qubits are represented by triangles, and stabilizers are represented by colored vertices, which apply a parity check over the qubits for which the stabilizer is a vertex. The logical operators are strings of Pauli operators that extend over the torus in a non-trivial way. On the color code lattice on a torus, we have two independent logical operators for each non-trivial loop, i.e. four logical qubits. The support of the logical operators  $\hat{X}_i$  and  $\hat{Z}_i$  is represented by the blue and green lines, which represent the 4 non-trivial loops on the toric color code lattice. To illustrate the effect of errors in the lattice, we display an example of four physical bit-flip errors (qubits are marked with red diamonds), and the corresponding stabilizer excitations (stabilizers are marked with red circles).

principles. We study the details of each of the particular decoding steps in section 8.2.

The main step of the decoder is to split the code lattice into small cells. These cells can be decoded locally and the result can then be combined into the global decoding decision. A key problem to this decoding approach is that, when trying to divide the lattice into cells, one has to necessarily cut through some stabilizers, which are then shared between different cells. The proposed solution to that problem is to split these stabilizers into two, which we refer to as *half stabilizers* from here onwards. This splitting of the stabilizers opens the possibility to decode each cell individually by using the local syndrome from the half stabilizers that apply to the cell (Fig. 8.3). Thus, the way in which each stabilizer is split between the cells determines the correction applied to the cells. After applying a local decoder on each cell, this cell can be treated as



an individual effective qubit (two when using a square cell) of a now rescaled color code, where we have rescaled the lattice to a smaller version of itself. This is possible because the syndrome on the boundaries and the bulk of the cell have been addressed (i.e. the combination of the error and the proposed correction would recover the even parity of those stabilizers). The process of splitting and rescaling the lattice can then be repeated, ultimately leading to a lattice small enough so that a brute force decoding can be applied. This rescaling process is illustrated in Fig. 8.1(a)-(c) for the 2D color code on the square-octagon lattice.



**Figure 8.3: Example of the splitting of a stabilizer.** **a)** The original stabilizer  $S$ , represented by the blue vertex, realizes a parity check on the qubits 1 to 8, both in the  $Z$  and the  $X$  basis. Qubits are represented by black dots inside the triangles. **b)** We can split the stabilizer into the half stabilizers  $S_1$  and  $S_2$ . Each half stabilizer represents the parity of its set of qubits  $S_1 = \{1, 2, 3, 4\}$ ,  $S_2 = \{5, 6, 7, 8\}$ . The binary sum  $\oplus$  of the parities of both half stabilizers thus fulfills the condition  $s = s_1 \oplus s_2$ . We call the choice of parity assignments of half stabilizers for a given stabilizer a *splitting*. For notation, we use uppercase letters to refer to the stabilizer operators, and lowercase for their parity value, which is represented as a 0 for even parity and 1 for odd parity.

### 8.1.2 Motivation and previous work

The study of the RG decoding approach presented in this work is motivated by a prior study of the behavior of the algorithm in the hexagonal lattice, following the

## 8. RESCALING DECODER

---

work by Sarvepalli and Raussendorf [189], where they achieved a threshold of 7.8% for code capacity noise. We reproduced the decoder for the hexagonal code with our own code, and tested the relevance of each of the decoding steps. We implemented some of the variants of the decoding process proposed in [189], including the use of triangular vs. square cells (see Fig. 8.4), and the two different variants that the authors present for the splitting and rescaling algorithms, which affect how the decoder algorithm splits the stabilizers that are shared between different cells.

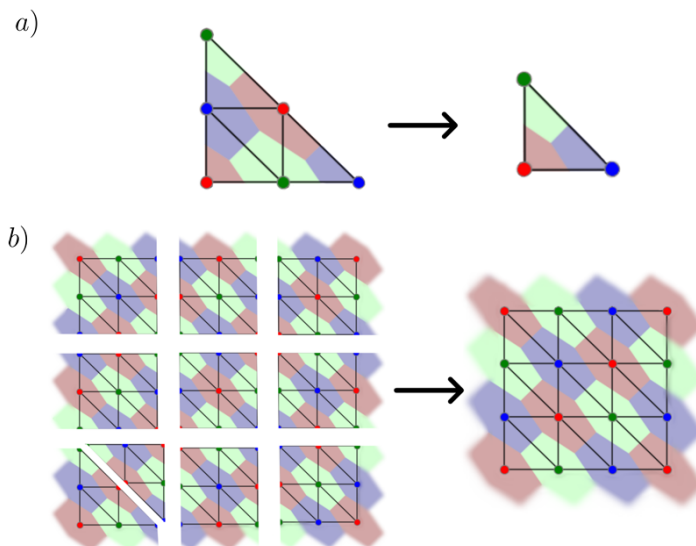
The first variant finds a splitting configuration (i.e. a choice of how to split each stabilizer in the boundaries between cells) through a *hard decision* process, in which the choice for each splitting is updated to the one that leads to the most probable correction given the current choice of the neighboring splittings. The second *soft decision* variant computes an estimate of the probability of each possible splitting choice, and then uses those probabilities to further refine the estimates. We obtained better results using this second approach (which is explained in detail in the following section), both in terms of the convergence of the splitting updates to a solution, and the resulting threshold of the decoder. Additionally, we experimented with different methods to update the splitting configuration: updating the splittings in parallel, sequentially, and randomly selecting the next splitting to update. We found the best performance in the former, and thus we implemented that method in the 4.8.8 decoder.

Although we did not realize an exhaustive study on the impact of the different variations on the decoder due to time constraints, the insight obtained from the performance of the decoder and the close examination of problematic error patterns was invaluable for the development of the RG decoder for the 4.8.8 lattice.

### 8.2 The decoder algorithm for the 4.8.8 color code lattice

In this section, we present the decoding algorithm step by step, explaining the details behind each of the different phases of the algorithm. The input of the decoding algorithm is the measured syndrome and a prior estimate of the qubit error rate, while the output is a recovery operation for each of the qubits in the lattice:

(1.) We estimate the error probability of the qubits using the information from the syndrome (Fig. 8.5) in two steps. The first step consists on the application of a belief propagation algorithm, through which stabilizers and qubits share information



**Figure 8.4: Sketch of a rescaling step in the 6.6.6 rescaling decoder.** **a)** A triangular cell of 4 qubits can be rescaled to an effective qubit. **b)** Two triangular cells can be combined to form a square cell, which can be rescaled to two effective qubits. With each rescaling step, the number of effective qubits is reduced by a factor of four. The square cell in the lower left corner has been split into the initial triangular cell for visualization purposes. The figure shows a cut of a larger color code lattice.

(see Sec. 8.2.2). After that, we use the parity of the stabilizers located in the corners of the cells to update the estimates of the error probabilities of the qubits around them (see Sec. 8.2.3).

(2.) We subdivide the code into cells. The stabilizers  $S$  between two adjacent cells are split, while preserving the parity of the sum:  $s_a \oplus s_b = s$ . There are two possible ways of splitting a stabilizer, as seen in Fig. 8.6.

(3.) We find a configuration for the choices of splittings (Fig. 8.3) and assign a probability for each splitting choice (details in Sec. 8.2.4):

(3.a) We compute the initial splitting probabilities using the probabilities of the qubits involved.

(3.b) We update the probabilities of the different splitting choices (see Fig. 8.7) using local information from the neighboring cells, as described in Sec. 8.2.4. The update is applied simultaneously on all splittings of the lattice. Several global update steps

## 8. RESCALING DECODER

---

### Algorithm 1: Rescaling decoder

---

**Data:** Measured syndrome

**Result:** Recovery operation

**while** *Lattice larger than minimum size* **do**

    Estimate the error probability of each qubit (1.)

    Split the code into cells (2.)

    Split the stabilizer values between the cells (3.)

    Apply a local decoder on each cell (4.)

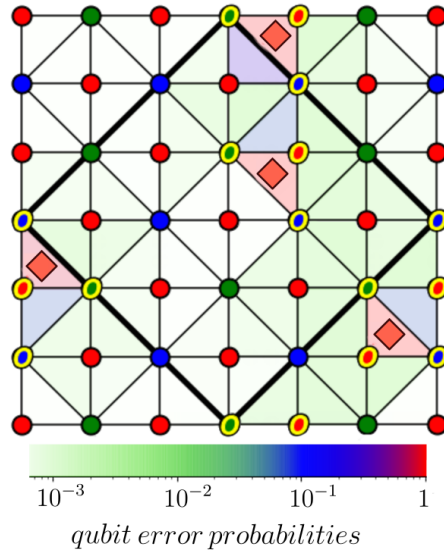
    Rescale the cells to effective qubits (5.)

**end**

Apply a brute force decoder on the final lattice (6.)

Back-propagate the errors to the original lattice (7.)

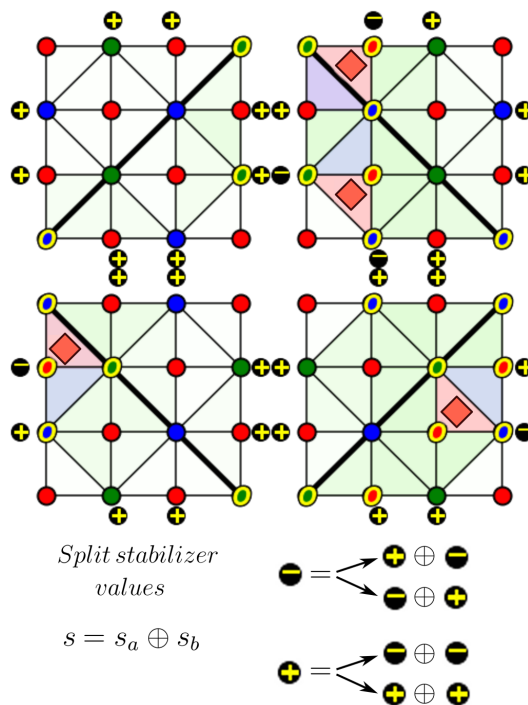
---



**Figure 8.5: Example of updated qubit error probabilities after belief propagation.** We show a 72 qubit color code with periodic boundary conditions, where qubits are represented by the coloured triangles, and stabilizers are represented by the coloured circles in red, blue and green. The non-trivial stabilizers are represented with a yellow border, and the physical errors are represented by red diamonds. During belief propagation, stabilizers and qubits share information locally, which leads to a refined estimate of the qubit error probabilities. In this example, the qubit error probabilities are represented by the color of the triangles (the qubits), ranging from white to red, as indicated in the color bar. The thicker solid black line has been drawn as a guide to the eye when comparing the figure with Fig. 8.6, after dividing the lattice into square cells.

are used until convergence is reached for the split choice.

## 8.2 The decoder algorithm for the 4.8.8 color code lattice

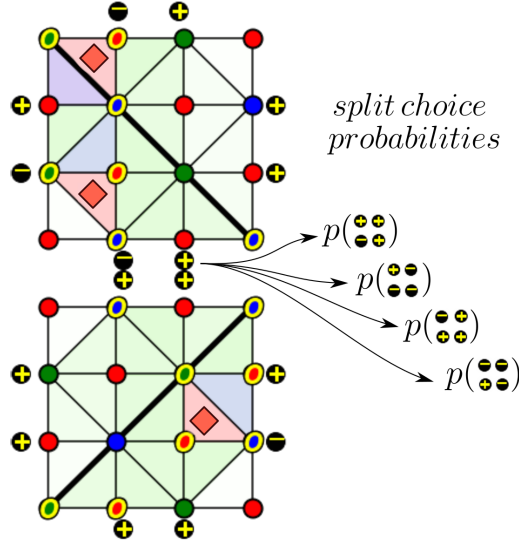


**Figure 8.6: Splitting of the lattice into cells.** We show the same code as in Fig. 8.5 after splitting the lattice into 4 different cells. The stabilizers at the boundary between cells need to be split into half stabilizers, as in Fig. 8.3. The parity of each half stabilizer  $s_a$  and  $s_b$  is represented as a 0 for even parity and a 1 for odd parity, and the binary sum  $\oplus$  of the parity of both half stabilizers needs to equal the parity  $s$  of the original stabilizer  $S$ . For each stabilizer, there are two alternative splittings into half stabilizers. The black solid line in the cells has been drawn to distinguish the two effective qubits that result from the rescaling of the cells. The physical errors in this particular example have been marked with red diamonds, and the excited stabilizers are marked with a yellow border.

(4.) After fixing the half stabilizers, each cell has a local syndrome. Using a lookup table (a pre-computed list containing all possible errors compatible with the syndrome), we can decode the cells and find a correction (Fig. 8.8). At this step, we ignore the syndrome of the corners, as their parity will become the syndrome of the rescaled lattice. The parity of those stabilizers will thus be addressed in the decoding process applied to the subsequent rescaled lattices.

(5.) The cells can now be rescaled to effective qubits (Fig. 8.9). We can compute the error probability of the rescaled qubits as the probability of applying a logical operator in the cell (see section 8.2.5).

## 8. RESCALING DECODER



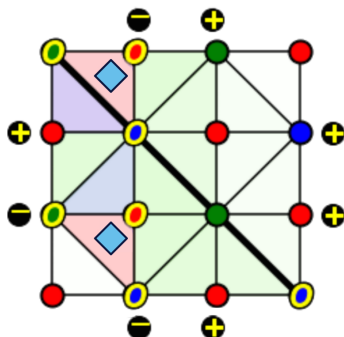
**Figure 8.7: Sketch of the splitting of two stabilizers.** A cell shares two stabilizers with each of the neighboring cells. There are four different ways to split a pair of stabilizers in half stabilizers. During the splitting updates, we compute an estimate of the probability of each of the four splitting choices. The signs shown in the dark circles correspond to the parity of the half stabilizers. The two cells shown in the figure are part of the lattice shown in Fig. 8.5.

(6.) We create a new code by rescaling each cell on the lattice to effective qubits. If the code is small enough, we can decode the lattice by finding the most probable error with a lookup table (Sec. 8.2.6). Otherwise, we repeat steps 1-5 for the new code.

(7.) Once we have the corrections on the smallest lattices, we can back-propagate them to the original lattice to obtain the final recovery operation.

After applying the recovery operation, we can check in our simulations if the combination of the error and our correction corresponds to the application of a logical operator on any of the four encoded qubits in the lattice, which would represent a logical error. This can be done efficiently by checking the parity of the qubits along the support of each of the four logical operators (see Fig. 8.2), i.e. for instance the parity of the number of bit-flips along the qubits in the support of  $\hat{Z}_i$  determines if a logical  $\hat{X}_i$  operator was applied.

It is important to notice that the smallest lattice size and the size of the unit cell



**Figure 8.8: Local decoding of cells.** A single cell can be decoded using the syndrome from the half stabilizers in the boundary and the parity of the bulk stabilizers. To find the correction, a brute force decoder is applied, which finds the most probable correction (the qubits in this suggested correction are marked by blue diamonds) using the estimates of the error probabilities of each qubit. The parity of the corner stabilizers is ignored during the decoding of the cell, but the parity of these stabilizers is updated depending on the corrections applied. In this particular example, the parity of the stabilizer in the upper left corner would be changed by the proposed correction, which will be taken into account in the rescaled lattice. The signs in the dark circles represent the parity of the half stabilizers in this cell.

determine the code size (total number of qubits) for which the decoder can be applied. In this work, we consider a minimum lattice size of 8 qubits. Thus, the lattice sizes for which the decoder can be applied depend on the number of rescaling steps  $m$  as  $[[n, k, d]] = [[8 \cdot 9^m, 4, 2 \cdot 3^m]]$ , where  $n$  is the number of qubits in the lattice,  $k$  the number of logical qubits and  $d$  is the distance of the code. The number of qubits in the lattice begins with 8 qubits as the minimum lattice size (see Fig. 8.1), and each rescaling step introduces a factor of 9 in the number of qubits (see Fig. 8.9). Similarly, the code distance  $d$  of the smallest lattice size is 2, and each rescaling step increases the logical distance by a factor of 3 (see logical operators in Fig. 8.2).

### 8.2.1 Minimal cell

In order to choose an appropriate cell, three conditions need to be fulfilled (cf. [189]):

1. A logical operator can be defined for the cell.
2. There exists a valid correction for every possible syndrome.
3. The cell can map the entire code to a smaller version of itself.

## 8. RESCALING DECODER

---

A valid cell for which these conditions are fulfilled can be found by choosing a triangular cell for which the three corners are of different colors.

The minimal cell that fulfills these conditions in the 4.8.8 lattice is the 9-qubit cell represented in Fig. 8.9. The cell shares two split stabilizers with each neighboring cell, and an additional stabilizer is contained entirely within the cell. The syndrome for this cell contains seven stabilizer measurements (one red stabilizer inside the cell, and six half stabilizers on the boundaries), and four different corrections are possible for each possible syndrome due to the logical operator and the stabilizer contained within the cell. The 4.8.8 lattice can be mapped to a smaller version of itself using this cell, reducing the number of qubits by a factor of 9 with each step. In Fig. 8.1, we show a single step of splitting the lattice and rescaling the cells.

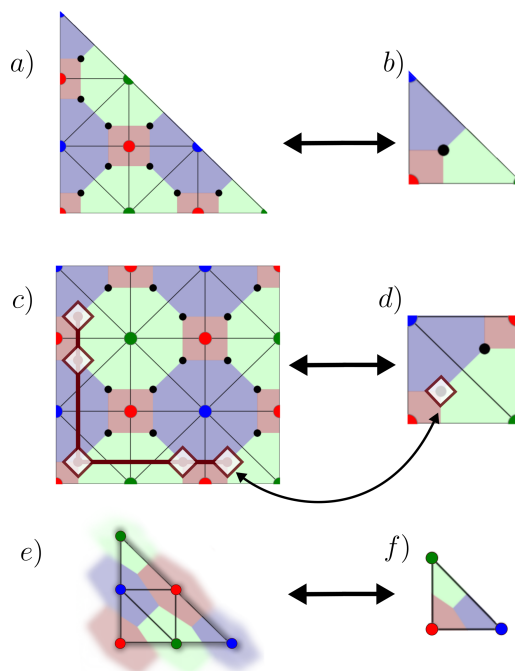
To simplify some of the decoding steps, two triangular cells can be combined into a square cell, consisting of two effective qubits (as shown in Fig. 8.8). This square cells now shares a boundary with four neighboring cells, with a total of 8 half stabilizers located in the boundaries and four stabilizers contained entirely within the cell. The use of square cells reduces the number of stabilizer splittings needed in each decoding step. This approach also leads to a better performance in the hexagonal lattice [189]. In this work, we study the decoder considering the use of square cells.

The use of square cells also opens the possibility to consider the error probabilities of the two effective qubits corresponding to the cell as joint probabilities, preserving any possible correlations between them, instead of considering both effective qubits as completely independent. Although we compute these joint probabilities during the rescaling step (see section 8.2.5), there is further potential to explore in this variation, as we neglect these correlations during the BP propagation algorithm.

### 8.2.2 Belief propagation

Belief propagation (BP) is a general method used to compute or approximate marginal probabilities from a multivariate probability distribution based on Bayesian inference, also called the sum-product message passing algorithm [366–368]. In the context of decoding, BP is typically formulated in terms of bipartite graphs, where the vertices on one side represent the (qu)bits, and the vertices on the other side represent the parity checks (see Fig. 8.10). In the classical setting, it can be used successfully to decode some low-density parity check (LDPC) codes, provided that the number of loops in the





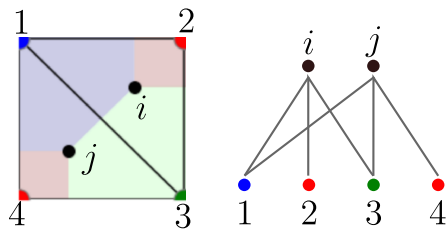
**Figure 8.9: Rescaling cells in the 4.8.8 lattice.** **a)** Minimal cell for the rescaling decoder in the 4.8.8 lattice. This 9-qubit cell can be mapped to a single effective qubit **b)** during the rescaling of the lattice. **c)** A square cell can be used during the decoder process to reduce the number of stabilizer splittings. This cell is mapped to two effective qubits, as shown in **d)**. An error on the rescaled qubit can be back-propagated (curved double-arrow) to the original lattice by applying the effective logical operator of the cell: the qubits marked in **c)** with white squares represent the logical operator of the effective qubit marked in **d)**. By applying this logical operator, the parity of the corner stabilizers corresponding to the effective qubits is changed, while preserving the parity of the rest of the stabilizers in the cell. **e)** For comparison, we show the minimal cell for the rescaling decoder in the hexagonal lattice [189], with only four qubits. This cell can be rescaled to an effective qubit **f)**.

underlying graph is not too large [367]. However, BP is known to fail as a decoder for topological QEC codes, for which the bipartite graph contains a high density of loops. There are two main failure scenarios: either the BP decoder does not converge at all, or the solution is inconclusive or wrong. This is attributed to the degenerate nature of topological codes [369], where a syndrome can be corrected using many different but equivalent recovery operations (e.g. by adding a product of stabilizer operators, which do not affect the logical information encoded in the computational subspace).

In the context of the rescaling decoder, the BP algorithm is introduced as an inter-

## 8. RESCALING DECODER

---



**Figure 8.10: Sketch of a bipartite graph in the belief propagation algorithm.**

In the left, we show a small fraction of a color code, with two qubits and four stabilizers. In the right, we show the corresponding bipartite graph, where each qubit is connected to the stabilizers that have support on that qubit. Even in this small graph, a loop can be found by following the path 1,  $i$ , 3,  $j$ , which leads back to 1.

mediate step to address the limitations of the splitting algorithm. The goal of this step is not to find a correction, but to compute a new estimate of the qubit error probabilities using the information from the outcome of the surrounding stabilizer measurements and the prior information about the error rates of the neighboring qubits. During the algorithm, the error probability  $p_i$  of each qubit is updated using the information from the neighboring qubits and the parity of the surrounding stabilizers. This local update is repeated over several rounds or iterations, that effectively spread the information of each qubit and stabilizer (likelihood of having an error, and parity, respectively) over the neighboring region of the lattice. After BP, the rescaling algorithm has access to an improved estimate of the error probability of each qubit, which helps the following steps of the decoder to distinguish between different error cases.

In the rescaling decoder, the problems with convergence of BP are avoided, as a convergence of the algorithm is not required for BP to be useful. With each iteration of BP, each qubit will gain information about an increasingly larger region of the lattice, improving the estimate of the error rate. However, the number of rounds of BP is restricted to 3 during the decoding process to prevent the negative effects of loops in the graph. We found in our simulations that the performance of the decoder was reduced when using a different number of rounds, but the study of the use and impact of BP for the decoding process was outside of the scope of our work.

The BP algorithm is based on the exchange of information between qubits and stabilizers. Qubits and stabilizers are called nodes and are arranged in a lattice, so that qubits are only connected to the neighboring stabilizers and vice-versa (see Fig. 8.10). The information is shared in packages called messages, that contain information about

---

## 8.2 The decoder algorithm for the 4.8.8 color code lattice

---

the likelihood of errors in different positions. The steps of the algorithm are shown in Algorithm 2. The equations used for the messages are derived below.

---

**Algorithm 2:** Belief propagation

---

**Data:** Measured syndrome  $\{s_j\}$  and initial error probabilities  $\{p_i\}$

**Result:** Updated error probabilities  $\{p_i^{updated}\}$

Each qubit sends an initial message to the neighboring stabilizers  $M_{q \rightarrow s}^{(0)}$  (Eq. 8.9)

**for** Number of rounds ( $N = 3$ ) **do**

    Each stabilizer sends a message to the neighboring qubits  $M_{s \rightarrow q}$  (Eq. 8.10)

    Each qubit sends a message to the neighboring stabilizers  $M_{q \rightarrow s}$  (Eq. 8.11)

**end**

Update the estimates of the qubit error probabilities  $\{p_i^{updated}\}$  (Eq. 8.12)

---

The main building block of belief propagation is Bayes theorem, which is applied to update the probability of an error in a qubit with the information from the syndrome:

$$p(q_i = 1|\{s\}) = \frac{p(q_i = 1)}{p(\{s\})} \cdot p(\{s\}|q_i = 1) \quad (8.1)$$

$$p(q_i = 0|\{s\}) = \frac{p(q_i = 0)}{p(\{s\})} \cdot p(\{s\}|q_i = 0) \quad (8.2)$$

where  $p(q_i = 1|\{s\})$  is the probability of an error on qubit  $q_i$  given the syndrome  $\{s\}$ ,  $p(q_i = 1)$  is the prior, or the previous information on the qubit error rate,  $p(\{s\}|q_i)$  is the probability of the given syndrome assuming an error on qubit  $q_i$  (usually called the likelihood ratio) and  $p(\{s\})$  is the probability of the syndrome event. This last term is effectively an unknown normalization factor, which would be hard to compute in general. However, we can cancel that term if we compute the quotient of both quantities. Thus, we can write:

$$\begin{aligned} \frac{p(q_i = 0|\{s\})}{p(q_i = 1|\{s\})} &= \frac{p(q_i = 0)}{p(q_i = 1)} \cdot \frac{p(\{s\}|q_i = 0)}{p(\{s\}|q_i = 1)} \\ &= \frac{1 - p_i}{p_i} \prod_j \frac{p(s_j|q_i = 0)}{p(s_j|q_i = 1)}, \end{aligned} \quad (8.3)$$

where the product in  $j$  compiles information for all the parity  $s_j$  of the stabilizers affecting the qubit. This quotient will correspond to the information sent from each

## 8. RESCALING DECODER

---

stabilizer  $j$  to the qubit, and can be written, depending on the parity of the stabilizer, as

$$M_{s=0 \rightarrow q} = \frac{p(s=0|q_i=0)}{p(s=0|q_i=1)} = \frac{p(\text{even})}{p(\text{odd})}, \quad (8.4)$$

$$M_{s=1 \rightarrow q} = \frac{p(s=1|q_i=0)}{p(s=1|q_i=1)} = \frac{p(\text{odd})}{p(\text{even})}. \quad (8.5)$$

Here,  $p(\text{even})$  and  $p(\text{odd})$  correspond to the probability of an even/odd number of error events happening on the remaining qubits involved in the parity check. Therefore, we can compute the messages from the stabilizers by computing the probability of an even/odd number of error events happening on the remaining qubits. This probability can be written using the following general formula to find the probability of an even/odd number of error events, given the probabilities  $p_i$  of each individual event:

$$p(\text{even}|\{p_i\}) = \frac{1}{2} + \frac{1}{2} \prod_i (1 - 2p_i), \quad (8.6)$$

$$p(\text{odd}|\{p_i\}) = \frac{1}{2} - \frac{1}{2} \prod_i (1 - 2p_i). \quad (8.7)$$

Then, the messages from the qubits to the stabilizers can be updated using the information from the stabilizers. The messages to the stabilizers in the next cycle are:

$$M_{q \rightarrow s_i} = \frac{1-p}{p} \prod_{\substack{j \in N(q) \\ j \neq i}} M_{s_j \rightarrow q}, \quad (8.8)$$

where  $N(q)$  refers to the three stabilizers in the neighborhood of qubit  $q$ . In this way, the information from the qubits and stabilizer spreads through the code, leading to an improved estimation of the error probabilities for the qubits.

Nevertheless, in a practical numerical simulation, the direct use of Eqs. (8.4), (8.5) and (8.8) can lead to numerical problems, when the values of the error probabilities of the qubits are close to zero. In the RG decoder, this scenario can occur frequently. For example, the rescaling of a region where no stabilizer excitations were detected would lead to a much smaller error rate in the effective rescaled qubits, since applying a correction in the rescaled qubit would imply a larger number of corrections being applied in the former lattice.

---

## 8.2 The decoder algorithm for the 4.8.8 color code lattice

Due to this problem, it is useful to work instead with the log-likelihood ratio. With that formulation, the initial message from qubits to stabilizers is

$$M_{q \rightarrow s}^{(0)} = \log \frac{1 - p_q}{p_q}. \quad (8.9)$$

The equation for the messages from stabilizers to qubits can be written as:

$$M_{s_i \rightarrow q_j} = (1 - 2 \cdot s_i) 2 \tanh^{-1} \left[ \prod_{\substack{k \in N(s_i) \\ k \neq j}} \tanh(M_{q_k \rightarrow s_i}/2) \right] \quad (8.10)$$

where  $s_i$  represents the parity of the stabilizer measurement. Similarly, the equation for messages from qubits to stabilizers can be simplified as:

$$M_{q_i \rightarrow s_j} = \sum_{\substack{s_k \in N(q_i) \\ k \neq j}} M_{s_k \rightarrow q_i} + \log \frac{1 - p_i}{p_i}. \quad (8.11)$$

After the last iteration of message passing, we can obtain the updated estimate of the error probability. The equation for the final estimate of the error probability can be written as:

$$p_i^{updated} = \left[ 1 + \exp \left( \log \frac{1 - p_i}{p_i} + \sum_j M_{s_j \rightarrow q_i} \right) \right]^{-1}. \quad (8.12)$$

### 8.2.3 Corner updates

During the decoding process, the splitting updates and the cell decoder ignore the syndrome of the stabilizers located in the corners of the cells. This can be problematic for some error cases, as the decoder would not be able to distinguish cases that differ only on the syndrome of the corner stabilizers. For this reason, it is beneficial to use the information from the syndrome in the corners of the cells to modify the error probability of the qubits near the corners. In the context of the decoder, this step is applied after BP, using the resulting estimates of the qubit's error probability.

Given the parity  $s_i$  of a given corner stabilizer, we need the parity of the qubits in the cell plus the parity of the qubits outside to be equal to  $s_i$ . Thus, for a given qubit

## 8. RESCALING DECODER

---

$j$  in the cell with a prior estimate  $p_j$ , we can compute the updated error probability of an error on qubit  $j$  given the parity of the corner stabilizer  $p(j|s_i, p_e)$  as:

$$p(j|s_i = 0) = \frac{p_j \cdot p_e(\text{odd})}{p_j \cdot p_e(\text{odd}) + (1-p_j)p_e(\text{even})}, \quad (8.13)$$

$$p(j|s_i = 1) = \frac{p_j \cdot p_e(\text{even})}{p_j \cdot p_e(\text{even}) + (1-p_j)p_e(\text{odd})}, \quad (8.14)$$

where  $p_e$  corresponds to the probability of the qubits outside the cell to have a total parity that is even/odd. We can use the Eqs. (8.6) and (8.7) to compute the probability of  $n$  qubits having a total even or odd parity.

During our study of the algorithm in the hexagonal lattice, we found that the omission of this step had a significant negative impact in the performance of the decoder. However, we only experimented briefly with the use of corner updates before the splitting updates. After this step, no direct communication is established between cells localized diagonally opposite to a corner stabilizer. It would be interesting to study how additional modified corner updates realized during the splitting algorithm affect the overall decoder performance.

### 8.2.4 Splitting the stabilizers

The basic cells on the 4.8.8 lattice share two splittings with each neighboring cell (see Fig. 8.9). This leads to a new phenomenon compared to the more straightforward case of the 6.6.6 lattice, as the probability of splitting of each stabilizer now depends on the splitting configuration of the other splitting shared with that cell. Therefore, we need to consider the joint probabilities, which simultaneously consider the probability of a splitting configuration of the two stabilizers  $s_1$  and  $s_2$  shared between the cells.

This complication in the way we store and compute the probabilities of splitting configurations is one of the main challenges in the application of the RG decoder to the 4.8.8 lattice, and it affects most of the other steps of the algorithm, as splittings need to be considered in pairs taking into account the joint probabilities:

1. The first estimate for the joint probabilities can be obtained from the probabilities of the qubits involved in each stabilizer having an even/odd number of errors. This does not consider the rest of the splittings in the cell, only the probabilities of the qubits involved in the splitting.

## 8.2 The decoder algorithm for the 4.8.8 color code lattice

---

2. During the splitting updates, we update the joint probabilities for the splitting configurations. Each step updates our estimate for the probabilities of each splitting using the information from the two cells that share the stabilizers.
3. In the rescaling step, the probability of error in the rescaled qubit involves the probabilities of the different splitting configurations. This means that the error probability of the rescaled qubit takes into account the uncertainty in the splitting choice.

Next, let us study the mathematical details used to update the splitting probabilities. The first equation to consider corresponds to the probability of a given error configuration. Assuming our estimate of the error probabilities of each individual qubit, we can compute the probability of an error configuration  $C = \{e_0, e_1, \dots, e_{n_q-1}\}$  (where  $e_i = 0, 1$  represents the absence or presence of an error on qubit  $i$ , and  $n_q$  is the number of qubits in a cell) by adding a factor of  $p_i$  for each qubit with an error, and a factor of  $1 - p_i$  for each qubit without an error,

$$p(C) = \prod_{i=0}^{n_q-1} p_i^{e_i} (1 - p_i)^{1-e_i}, \quad (8.15)$$

where we assumed no correlation between the error probability  $p_i$  of each qubit. During the rescaling process, the use of square cells with two qubits will give us access to the joint probabilities of each qubit pair that belongs to the same cell. We can use this additional information about the correlations between different qubits to modify this equation as:

$$p(C) = \prod_{i=0}^{n_q/2-1} p(e_i, e_{i+1}), \quad (8.16)$$

where  $p(e_i, e_{i+1})$  corresponds to the element of the joint probabilities corresponding to our prior knowledge from the error probabilities of qubits  $i$  and  $i + 1$ .

Using the probability of a given error configuration, we can compute the probability of all error configurations that are compatible with a given syndrome as the sum of  $p(C)$  over all configurations  $C$  compatible with the syndrome  $\{s\} = \{s_0, s_1, \dots, s_{n_s-1}\}$  ( $s_i = 0, 1$  represents the even or odd parity of the stabilizer or half stabilizer  $i$ ):

$$p(e|\{s\}) = \sum_C p(C). \quad (8.17)$$

## 8. RESCALING DECODER

---

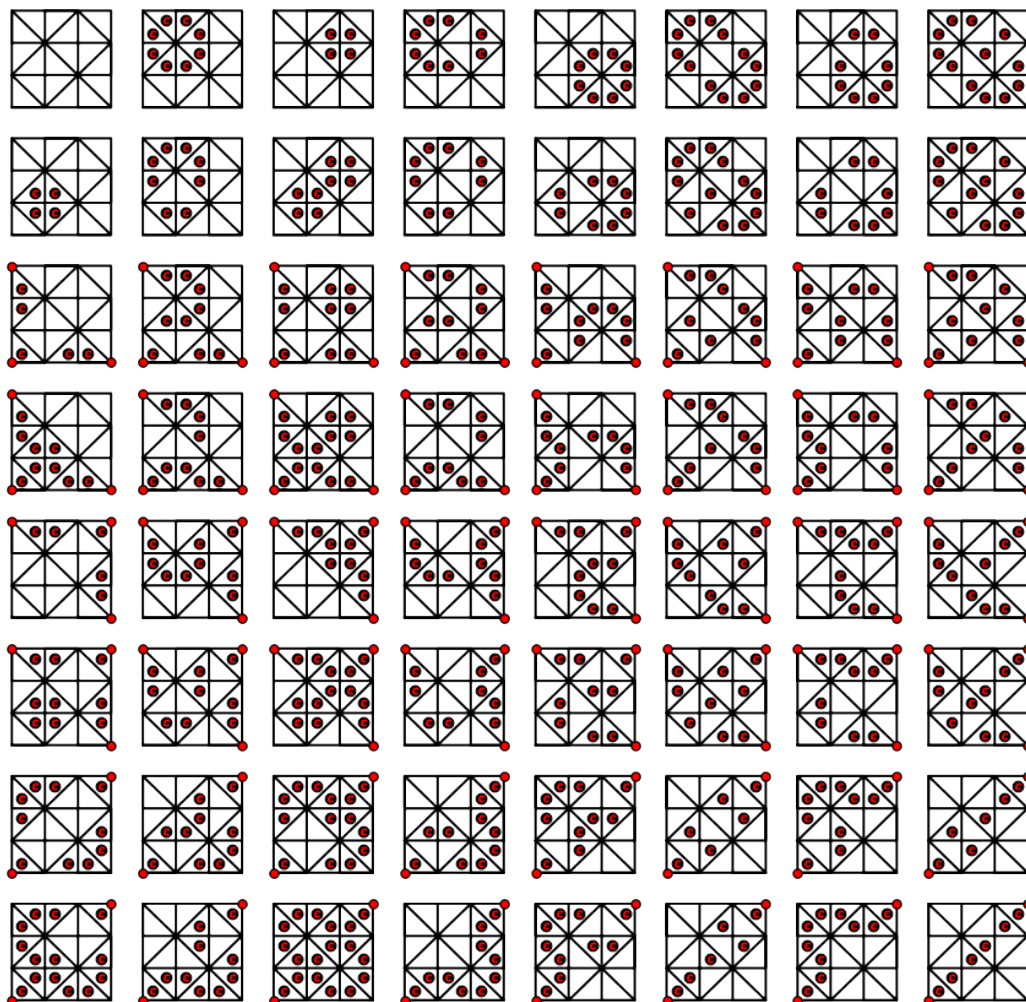
For the 18-qubit square cell that we use for the 4.8.8 color code, each syndrome includes eight half stabilizers (two on each side of the square) and four additional stabilizer measurements corresponding to the bulk stabilizers inside the cell. Here, as stated earlier, the parity of the corners is ignored, as the parity of the corner stabilizers will be solved in the following rescaling steps. On the square cell, there are two logical operators that can be defined, one for each of the effective qubits to which the cell is rescaled. This means  $2^2$  possible classes of configurations compatible with any syndrome in the cell. In addition to that, all possible product combinations with the four bulk stabilizers lead to equivalent error configurations, adding up to a total of  $2^{2+4} = 64$  possible configurations (see example in Fig. 8.11) over which is to be summed according to Eq. (8.17).

While using all 64 configurations would lead to more accurate estimations, this also involves a high constant-factor overhead in the computational time required by the decoder. Thus, to improve the performance of the algorithm with regard to computing time, we approximate Eq. (8.17) by considering only one configuration, using the cell's lookup table (in the example of Fig. 8.11, that would mean the first configuration in the figure). With this approximation, we effectively reduce the computational overhead by a factor of 64 while keeping one of the terms of highest weight in Eq. (8.17). Although this restriction ignores alternative error configurations which could be more probable due to the uneven error probabilities of the qubits, in our simulations we found no significant change in the correction performance when using system sizes larger than 72 qubits, while the computational time required by the decoder dropped by an order of magnitude. Therefore, we used this approximation for the simulations shown in Sec. 8.3.

Once we have seen how to compute the probability of all possible configurations compatible with a given syndrome (or one of the most likely configurations if we use only the choice from the lookup table), we can define the probability of a given half splitting  $s_i^l$  given a certain fixed syndrome on the rest of splittings (where the superscript  $l$  corresponds to the cell on the left, see Fig. 8.12). This probability corresponds to the fraction of possible configurations compatible with the splitting choice, compared with the total probability of the possible configurations for all splitting choices,

$$p(s_i^l | \{s\}) = \frac{p(e|s_i, \{s\})}{p(e|s_i = 0, \{s\}) + p(e|s_i = 1, \{s\})}. \quad (8.18)$$

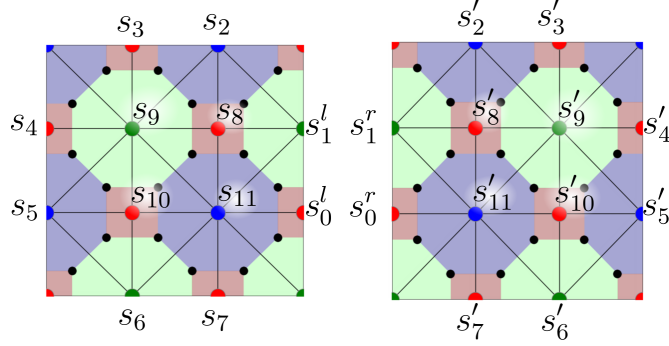




**Figure 8.11: Example of the different corrections compatible with a given syndrome in the square cell.** The figure shows all the possible error configurations that are compatible with the trivial syndrome in the square lattice. The configurations are obtained by computing all the possible combinations of the two logical operators and the four bulk stabilizers, for a total of  $2^{2+4} = 64$  configurations. To facilitate the visualization, we represent the qubits with a correction with a red  $C$  within a black circle. Since the cell decoder does not take into account the parity of the stabilizers in the corners, we can see how different corrections can affect the parity of the stabilizers in the corner. These changes in the parity of the stabilizers in the corners are propagated to the rescaled lattice, where it is taken into account.

Seeing that every cell shares two stabilizers with each of its neighbors, the splitting choice for these two stabilizers is not independent. Therefore, we need to consider the joint probabilities, where we consider each of the combined splitting choices for the two

## 8. RESCALING DECODER



**Figure 8.12: Notation of stabilizer splitting.** Example of a pair of splittings between two neighboring cells in the 4.8.8 lattice. The splittings 0 and 1 are being updated, and the super index indicates if we refer to the left or right half stabilizer. The parity of the (half) stabilizers on the left cell are represented as  $s_i$ , while the parity of the (half) stabilizers on the right cell are notated with  $s'_i$ .

splittings shared between two cells. Therefore, the probability of a splitting choice for the stabilizers  $s_0^l$  and  $s_1^l$  given a fixed choice on the rest of splittings (which we write as  $\{s\}$  for simplicity) can be written as

$$p(s_0^l, s_1^l | \{s\}) = \frac{p(e | s_0^l, s_1^l, \{s\})}{\sum_{i,j=0}^1 p(e | i, j, \{s\})}. \quad (8.19)$$

Note that this expression is equivalent for all splitting pairs, and we only wrote it explicitly for the first two splittings to simplify the notation. Since the rest of the splitting choices are not fixed, we can compute the estimate for the half splitting within a cell by combining the information from all splitting choices. For this, we can use the information of the joint probabilities  $p(s_k, s_{k+1})$ , which corresponds to our current estimate of the probability of the half splitting  $s_k, s_{k+1}$  (e.g. the probability of the splitting choice could be  $p(s_2 = 1, s_3 = 1)$ ). The estimate for the probability of a half splitting configuration within the cell can then be computed as the sum of Eq. (8.19) for each splitting configuration of the other splittings involved in the cell

$$p(s_0^l, s_1^l) = \sum_{\{s\}_k} p(s_0^l, s_1^l | \{s\}_k) \prod_{k=1}^3 p(s_{2k}, s_{2k+1}). \quad (8.20)$$

Finally, we want to find the probability of a given splitting choice. This necessarily involves the configurations in both the left and right cells, and the value of the parity

## 8.2 The decoder algorithm for the 4.8.8 color code lattice

---

$v_i$  of the stabilizers that we are measuring. Combining the information from both cells and ensuring the consistency condition  $s_i^l \oplus s_i^r = v_i$ , we can obtain the next estimate for the probability of a given splitting choice as

$$p_{split}(s_0^l, s_1^l) = \frac{p(s_0^l, s_1^l)p'(s_0^r = s_0^l \oplus v_0, s_1^r = s_1^l \oplus v_1)}{\sum_{i,j=0}^1 p(i, j)p'(i \oplus v_0, j \oplus v_1)}, \quad (8.21)$$

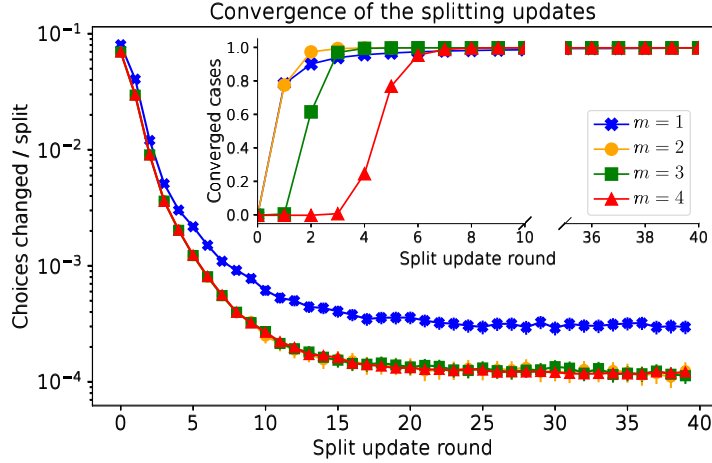
where  $\oplus$  represents a binary sum, and  $\bar{v}_i = 1 \oplus v_i$ . Using these equations, we can update our estimate for each of the splitting choices.

During the splitting algorithm, we apply global updates to the probabilities of the splitting choices by simultaneously updating the splitting probabilities of all splitting pairs in the code. All of the updates for the estimates of the splitting probabilities depend on the estimates from the previous step, which are not overwritten until all new estimates have been computed. This also means that the splitting algorithm can be easily parallelizable, since each splitting pair can be updated independently.

Although the number of global updates required for convergence can vary between runs, we empirically find that the average number of updates does not scale significantly with the lattice size, with less than 15 update rounds required. We test the convergence by measuring the average number of changes in the splitting choice per splitting and round. We find that this ratio does not increase with the system size (Fig. 8.13). Furthermore, we estimate the fraction of cases that have converged after  $n$  split update rounds. For a given error case, we define the number of rounds until convergence as the last update round with less than  $3m$  changes in the split choice (with  $m$  being the number of rescaling steps required for that lattice size). Notice that this criterion is particularly strict since the number of splittings grows exponentially with  $m$  as  $N_{splittings} = 8 \cdot 9^{m-1}$ , but the number of splittings that we allow to remain active only grows linearly with  $m$ . Nevertheless, we found that, for the system sizes that we worked with, the time required for the majority of the system to converge according to this criteria grows slowly with  $m$ .

To study the impact of the convergence of the splitting updates on the performance of the decoder we gathered statistics separately for the success and failure cases. However, we did not find evidence of a significant difference in the convergence speed with the curves for success and failure, with both curves almost overlapping for system sizes larger than  $m = 1$ . It remains to be studied how the number of splitting update rounds

## 8. RESCALING DECODER



**Figure 8.13: Convergence of the splitting updates.** We show the statistics of the performance of the splitting updates. In the main plot, we show the average fraction of splittings that have changed the split choice on each split update round. In the inset plot, we show the cumulative fraction of cases that have converged by each split update round. We define a case to have converged at round  $r$  if that was the last round with more than  $3m$  changes in the splitting choice. For both plots, the lines connecting numerical data points have been drawn as a guide to the eye.

affects the performance of the decoder and whether the convergence of the splitting updates leads or not to a better overall solution.

### 8.2.5 Rescaling of the cells

Once we choose a splitting configuration, a correction can be found within each of the cells in the lattice. Using the syndrome from the half stabilizers in the boundary and the parity of the stabilizers within the cell, as well as the estimates for the error probabilities of the qubits, the most probable correction for a cell can be found with the help of a lookup table. The next step in the decoder is to rescale the cells, mapping each square cell to a pair of qubits. The error probability of each qubit on the rescaled lattice also needs to undergo a rescaling process. In this section, we discuss the details on how to compute the resulting error probability for the rescaled qubits.

The main idea to understand in the rescaling of the qubit error probability is the fact that an error on a qubit in the rescaled lattice corresponds to the application of a logical operator on the qubit of the original lattice (see figure 8.9). If the original correction in the cell is  $C$ , we can write a first estimate of the error probability of the

## 8.2 The decoder algorithm for the 4.8.8 color code lattice

---

rescaled qubit given the splitting choice  $\sigma$  as:

$$p(L|\sigma) = \frac{\sum_{\{S_b\}} p(C + L + \{S_b\})}{\sum_{\{S_b\}} p(C + \{S_b\}) + p(C + L + \{S_b\})}, \quad (8.22)$$

where the sum over  $\{S_b\}$  represents all possible combinations of the bulk stabilizers,  $L$  is the logical operator and  $p(E)$  is the probability of a given error configuration  $E$ .

For this estimate, we assumed that the splitting choice from the previous step is correct. However, from the previous splitting step, we know that we have an uncertainty in the splitting choice. In addition, we have an estimate of the probability of each splitting choice. Thus, we can include this information about the other splitting choices, weighted by the probability of each splitting choice, in the equation for the probability of an error in the rescaled qubit.

In order to include the information from alternative splitting configurations, we first need to understand what the probability of an alternative splitting means, and how to relate it to the correction  $C$  that we applied on the cell during the previous step. In particular, we need to find  $p(L|\tilde{\sigma}_k)$  for the alternative splittings  $\tilde{\sigma}_k$ .

For the decoder of the 6.6.6 lattice [189], the key idea to find this expression is that, by applying the stabilizer of a given splitting, we can effectively change the choice of that splitting, as there is an odd number of qubits from that stabilizer on each cell. By applying the half stabilizer corresponding to the splitting, we could relate the different corrections that correspond to each splitting choice, thus finding an expression for  $p(L|\tilde{\sigma}_k)$ .

In contrast, on the 4.8.8 lattice, the support of the half stabilizers on each cell has an even number of qubits. This means that applying the stabilizer does not change the value of the splitting choice for the splitting corresponding to that stabilizer. However, for the 4.8.8 color code, splittings come in pairs, as each cell shares two stabilizers. By applying the stabilizer of one of these two splittings, we can effectively change between the two choices of the neighbor splitting, and thus relate the corrections corresponding to both splitting choices. An example of this equivalence is shown in Fig. 8.14.

Following this rule, we can systematically find the probability  $p(L|\sigma_k)$  for each of the  $2^8$  possible splitting configurations within a given cell with these steps:

1. Find the difference in splitting choice between the reference splitting  $\sigma_0$  (the one with the maximum probability, chosen to find the correction) and the alternative

## 8. RESCALING DECODER

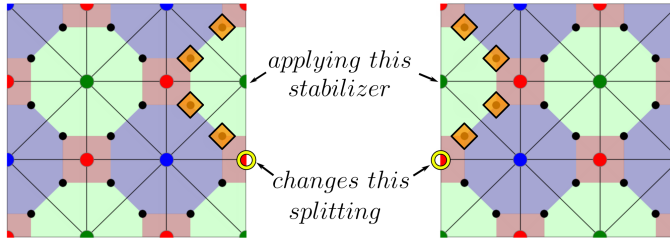
---

splitting  $\sigma_k$ :  $\Delta_k = (\sigma_0 - \sigma_k) \pmod 2$ .

2. For each splitting in  $\Delta_k$ , add the half stabilizer of its neighbor splitting. We call this product of half stabilizers  $\delta_k$ .
3. Compute the conditional probability  $p(L|\sigma_k)$  by adding to each configuration in Eq. (8.22) the product of half stabilizers  $\delta_k$ .
4. The final probability of error in the rescaled qubit can be computed as the sum of each of the conditional probabilities, weighted by our estimate of the probability of each splitting choice,

$$\begin{aligned}
 p(L) &= \sum_k p(L|\sigma_k) p(\sigma_k) \\
 &= \sum_k \frac{p(\sigma_k) \sum_{\{S_b\}} p(C + L + \{S_b\} + \delta_k)}{\sum_{\{S_b\}} p(C + \{S_b\} + \delta_k) + p(C + L + \{S_b\} + \delta_k)}.
 \end{aligned} \tag{8.23}$$

Here, the sum over  $k$  selects the different combinations of splitting choices, and the sum over  $\{S_b\}$  runs over all possible combinations of products of the bulk stabilizers.



**Figure 8.14: Equivalence between applying a stabilizer and changing the split choice.** Each cell shares two stabilizers with each of the neighboring cells, which need to be split. These two splittings form a pair of splittings. By applying the half stabilizer of one of the stabilizers, the resulting correction corresponds to a change in the splitting choice of the other splitting in the pair. In the figure, we can see one example in which, by applying the half stabilizer of one of the splittings, we change the parity of the neighbor splitting and obtain a valid correction for the new syndrome. The qubits affected by applying the stabilizers are marked with orange diamonds, and the stabilizer for which the splitting choice is changed are marked in yellow.

Finally, there is one more factor that we need to take into account in the rescaling of the cells. Since each cell corresponds to two different qubits, the error probabilities of

---

## 8.2 The decoder algorithm for the 4.8.8 color code lattice

the two qubits in the cell are not independent. Thus, we can compute the probabilities of having an error ( $L_i$ ) on each of the two logical qubits on the cell, leading to the joint probabilities for the four possible cases after the rescaling (error on the first qubit, error on the second qubit, error on both or error on none):

$$\tilde{C}_{\{S_b\},k} = C + \{S_b\} + \delta_k, \quad (8.24)$$

$$p(\mathbb{1}_0, \mathbb{1}_1) = \sum_k \frac{\sum_{\{S_b\}} p(\tilde{C}_{\{S_b\},k})}{D_k} p(\sigma_k), \quad (8.25)$$

$$p(L_0, \mathbb{1}_1) = \sum_k \frac{\sum_{\{S_b\}} p(L_0 + \tilde{C}_{\{S_b\},k})}{D_k} p(\sigma_k), \quad (8.26)$$

$$p(\mathbb{1}_0, L_1) = \sum_k \frac{\sum_{\{S_b\}} p(L_1 + \tilde{C}_{\{S_b\},k})}{D_k} p(\sigma_k), \quad (8.27)$$

$$p(L_0, L_1) = \sum_k \frac{\sum_{\{S_b\}} p(L_0 + L_1 + \tilde{C}_{\{S_b\},k})}{D_k} p(\sigma_k), \quad (8.28)$$

$$D_k = \sum_{\{S_b\}} \sum_{l_0, l_1=0}^1 p(L_0^{l_0} + L_1^{l_1} + \tilde{C}_{\{S_b\},k}). \quad (8.29)$$

Using these equations, we can compute the joint error probabilities for the rescaled qubits in tuples. We can then use these probabilities directly, or obtain the error probabilities of the individual qubits by marginalizing the second qubit from the probability distribution. In our simulations, we keep the joint probabilities of the two qubits and use them for the final decoding step, but we did not adapt the BP decoder to take into account the correlations encoded in the joint probabilities of the qubits.

### 8.2.6 Decoding of the final lattice

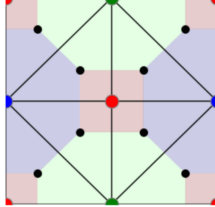
After a sufficient number of rescaling steps, we can reach a lattice size that is small enough to apply a brute force decoding. This means that, with the help of a lookup table, we compute the probability of each error configuration compatible with the syndrome, and choose the most probable error configuration as the recovery operation.

In our simulations, we consider a final lattice size of 8 qubits and distance  $d = 2$ , as shown in Fig. 8.15. While, in normal conditions, a distance two code should only be able to detect the presence of an error, our decoder has the advantage of having an

## 8. RESCALING DECODER

---

estimate for the error rate of each qubit in the lattice, which can allow the brute force decoder to distinguish between otherwise equivalent error patterns.



**Figure 8.15: Smallest lattice size for a 4.8.8 color code with periodic boundary conditions.** After the last rescaling step, we obtain a lattice with 8 qubits, six independent stabilizers (three  $X$  and three  $Z$  stabilizers) and distance  $d = 2$ . Although a code of this distance should only be able to detect errors, we can use it to obtain a correction, since the error rates of the different qubits involved are not homogeneous.

A larger lattice size can be chosen as the final lattice. However, the computational complexity of the brute-force decoder scales exponentially with the system size. Therefore, an alternative decoder could be advised when working with larger final lattices, a combined approach that could be studied in further research.

### 8.3 Results

We estimate the code capacity threshold of the decoder using Monte Carlo simulations. We generate random distributions of errors for different physical error rates and evaluate the logical error rate after decoding on each of the four logical qubits. We consider an error model with independent bit and phase flip errors, as detailed in Sec. 8.1. The dependence of the logical error rate as a function of the bit-flip error rate is presented in Fig. 8.16, showing the average of the error rate of the four logical qubits. To obtain the error bar of each estimate, we use the error from the binomial distribution  $err = \sqrt{p_{log}(1 - p_{log})/n}$ . The dependence of the error on the logical error rate  $p_{log}$  implies that a larger number of samples is required for the low  $p_{log}$  regime. Additionally, the computational cost of the simulations scales with the system size, which limits the number of samples we could obtain with the available resources. Therefore, we use a different number of samples for each point in the figure, ranging from  $10^6$  samples per point for the smaller system size to  $10^3$  samples per point for the larger system size  $m = 5$ .

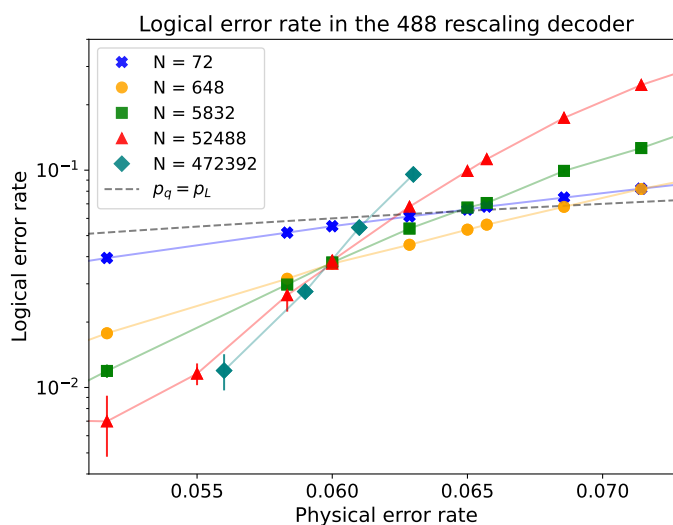


For each system size, the point at which the logical error rate equals the physical error rate is called the pseudo-threshold  $t(L)$  [370]. Using a simple linear interpolation between the two closest points to the crossing with the  $p_{\text{physical}} = p_{\text{logical}}$  line, we obtain an estimate for the pseudothreshold for each system size.

The threshold of the decoder corresponds to the limit  $\lim_{L \rightarrow \infty} t(L)$  of the pseudothresholds when we have an infinite size lattice. Using the estimates from the pseudothresholds, we find the infinite size limit by fitting the pseudothresholds to the following ansatz [371]:

$$t(L) = aL^{-\frac{1}{\nu}} + t_{\infty}, \quad (8.30)$$

where  $t(L)$  is the pseudothreshold at system size of code distance  $L$ , and the unknown parameters are:  $t_{\infty}$ , the threshold in the infinite limit;  $\nu$ , the scaling exponent and the coefficient  $a$ . Using a non-linear least squares fit <sup>1</sup> we obtain a threshold  $t_{\infty} \simeq 6.0\%$  and a scaling exponent  $\nu \simeq 1.6$ . The results from our simulations are shown in Fig. 8.17.

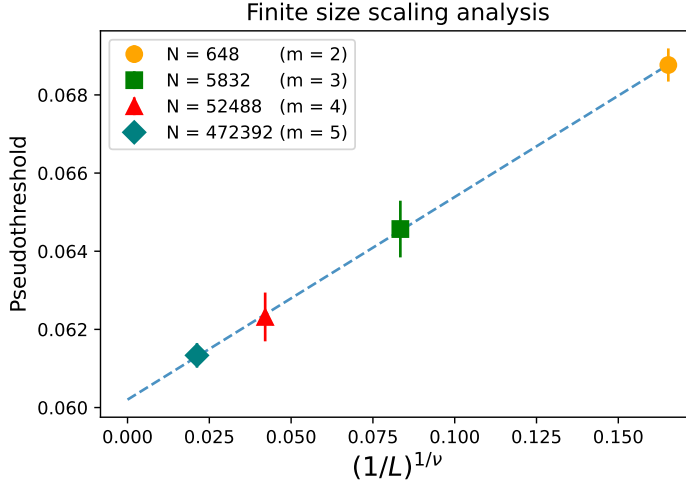


**Figure 8.16: Logical performance.** We plot the average logical error rate vs. the physical bit-flip (phase-flip) error rate for increasing size of the code lattice. Below  $p = 6.0\%$ , the logical error rate decreases with increasing system size, see main text and Fig. 8.17.

Note that the results for system size  $m = 1$  are particularly influenced by the finite-size effects, and we neglected the contribution from the level-1 pseudothreshold from

<sup>1</sup>We use the subroutine provided by the *scipy* python package, with source code available at <https://github.com/scipy/scipy/blob/v1.7.1/scipy/optimize/minpack.py>.

## 8. RESCALING DECODER



**Figure 8.17: Code capacity noise error threshold for the 4.8.8 color code decoder.** By fitting the pseudothresholds to a finite-size scaling ansatz of Eq. (8.30), we estimate the threshold of the decoder as 6.0% for code capacity noise (independent phase and bit flips errors with ideal syndrome measurement).

$m = 1$  when computing the threshold.

### 8.4 Summary and outlook

This chapter of the thesis has presented the results that we achieved in [2], as well as the analytical derivations that we used to develop the algorithm. In this work, we have presented an efficient decoder based on the local decoding and rescaling of the lattice that can be applied to the 4.8.8 color code lattice. This lattice is particularly interesting for FTQC, as it allows the transversal application of the complete Clifford group. The computational complexity of the decoder is  $\mathcal{O}(N \log(N))$ , and can be parallelized straightforwardly to achieve a remarkable  $\mathcal{O}(\log(N))$  scaling with the number of qubits  $N$ . We obtained a code capacity threshold of 6.0%, which lies below the 9.9 – 10.3% obtained by other recent decoders, like [193, 196].

Beyond the results and methods presented here, there are multiple open fronts for further research. First, a detailed study of the impact of the different steps of the decoder can lead to an improvement in the decoder performance. Of particular interest would be the study of alternative methods to communicate between cells, combining the splitting updates with message passing algorithms sharing information about the

syndrome in the corners. An interesting extension could be to extend the decoder ideas to work with open boundary conditions, which would make it easier to implement for most physical quantum computing platforms. This should be possible by simply adapting the rules for the local decoding around the boundaries, but further research needs to be made in order to confirm this hypothesis. Additionally, replacing the final brute-force decoding with another efficient decoder (e.g. the restriction decoder [196]) could reasonably provide new interesting results, and would allow additional flexibility in the choice of lattice sizes.

However, perhaps the most interesting continuation would be to extend the decoder for more realistic noise models, like phenomenological noise or even circuit noise. This has been achieved in surface code RG decoders [187] by, in an oversimplified way, using three-dimensional cells that extend over different QEC rounds. This approach could be adapted as well for color codes, although we can foresee some challenges in the implementation on the 4.8.8 lattice, as the minimal cell would likely be of sufficient size to significantly increase the computational cost of decoding, introducing a large constant overhead on the local decoding of cells, which would potentially require a combination with neural network techniques. Alternatively, the ideas from the RG surface code decoding from [187] could be combined with the algorithm presented in [192, 196] to project the color code into several copies of surface codes, which could then be decoded using the RG decoding algorithm for surface codes.

Once the step from code capacity noise to circuit-level noise has been achieved, the capability of the decoder to handle the error rates of individual qubits could conceivably open the possibility of introducing complex noise patterns (e.g. models learned directly from experimental observations [179]) capable of achieving higher prediction capabilities.

## 8. RESCALING DECODER

---

## Chapter 9

# Conclusions and outlook

## 9. CONCLUSIONS AND OUTLOOK

---

Quantum technologies are reaching the point where experimental implementations start to approach the theoretical requirements for practical applications. During the writing of this thesis, some important landmarks have already been achieved. We have seen the first demonstrations of quantum supremacy (or quantum advantage) [95–97, 372], or the demonstration of FT protocols with encoded qubits [167, 169, 171]. The field of quantum computation and quantum information has attracted large investments from governments and privates, and we are experiencing a rising number of research groups, from universities to big companies, as well as several new start-ups [71, 86]. Multiple potential applications for quantum computation are being proposed, e.g. [18, 19, 27, 54, 64–71], as well as a plethora of quantum algorithms [373–376]. Significant progress has been achieved on the research of different platforms for quantum technologies [18, 19, 88, 93, 94, 283, 377–380]. Industries are finding applications for quantum processors to optimize their operations, in fields that cover from chemistry and robotics, to finance and logistics [68–78] among many others. The success of recent experiments (e.g. [89, 95–97, 169, 171, 306, 381]), as well as the promising theoretical applications (e.g. [8, 14, 18, 19, 32, 79, 84, 85, 382–385]), have made quantum computing a very exciting field to be working in, with plenty of opportunities for new applications and research.

In the near term, we can expect multiple demonstrations of applications and improvements on NISQ devices [65, 125, 386], with growing numbers of qubits and improved fidelity on the gates and coherence times. On these near term implementations, the resolution of problems using quantum optimization (e.g. quantum adiabatic computing, quantum annealing, etc.) [79, 80, 82–85, 292], hybrid algorithms [59–61, 64] or quantum simulations of physical systems [14, 15, 21, 22, 25, 27, 28, 32, 36–38] stand out as some of the most promising realizations that we can expect to see over the following years [374–376]. Some experiments have already shown proof of principle demonstrations of algorithms like Shor’s factorization [387], Grover’s search [388], quantum annealing [389] or HHL algorithm to solve systems of linear equations [390, 391].

In the medium term, we can expect further demonstrations of fault tolerant quantum computation, as the progress in different platforms gets them closer and closer to meeting the requirements of FTQEC [88–93]. Important milestones have already been achieved, with experiments showing gate fidelities near or within the threshold of topological QEC, like ion trap [90–92, 140, 143, 392] or superconducting qubit

---

[89, 89, 134, 393] experiments. Furthermore, many groups have also demonstrated some of the building blocks of QEC, from FT state preparation and stabilizer readout [147, 170], to repeated FT QEC cycles [171] as well as FT magic state preparation and injection [169]. With these experiments, we were able to witness a hallmark feature of FT circuit design: an improvement of the performance of encoded qubits, despite the increased gate count and complexity of the underlying circuits required by the FT implementations of encoding and manipulating logical qubits. The following important steps to be achieved include the demonstration of repeated error correction and FT gate operations for larger-distance logical qubits, and the increase in the number of logical qubits, both of which require the challenging task of increasing the number of qubits while maintaining the high fidelities of operations. In parallel to the experimental development, the theoretical research must contribute to developing better FT protocols and techniques to reduce the effects of noise and optimize the use of resources (e.g. [163, 227, 237]). This theoretical effort requires classical tools that can help us predict and understand the behavior of the different approaches and identify the bottlenecks that limit the progress.

In regard to the long term goals of quantum computation, the applications with higher potential involve the implementation of FTQC, which can allow the use of quantum algorithms that require deep circuits and a large number of logical qubits, e.g. [100, 109, 131, 175, 394]. With an increase in the ratio of physical qubits per logical qubit, the fidelity on the logical gates can be significantly improved, exploiting the potential of the threshold theorem to achieve long quantum computations with many logical qubits [127, 129, 130]. Topological QEC codes such as the surface code [110–112] and color codes [113, 132], encode the logical information into topological properties of the system, and allow the realization of operations between encoded qubits. They stand out as the QEC codes with some of the highest known thresholds [109, 131]. Currently, the best candidates for the realization of large scale FTQC are superconducting qubits, trapped ions and photon-based QC [93, 302, 395–398]. The modular design of superconducting qubits in 2D arrays makes them a natural platform for scaling to a large number of qubits. Furthermore, they present some of the fastest gates, with gate times in the order of nanoseconds. However, they face several challenges, like the implementation of quick and reliable long distance interactions, and the difficulty in the production of identical qubits [93, 378, 379, 395]. Trapped ions, on the other

## 9. CONCLUSIONS AND OUTLOOK

---

side, overcome this problem by using perfectly identical ions, with the highest gate fidelities to date and long coherence times [90, 140, 149, 306, 399]. For ion traps, the challenge lies on the scaling, as controlling an increasing number of qubits in the same Paul trap can become difficult. Multiple approaches to the scalability of ion traps have been proposed, e.g. long ion chains with all-to-all connectivity, 2D arrays of ion traps with shuttling, communication between different traps, 2D Penning traps or the use of trapped electrons [149, 283, 297, 299, 308, 321, 322, 326, 400–403, 405]. Photonic devices use photons as information carriers, which are clean decoherence-free quantum systems with high fidelity single qubit gates, and easily scalable. The main challenges are photon losses, imperfect detection and generation of photons, and the development of high fidelity, deterministic two-qubit gates [380]. This platform presents an outstanding advantage in applications for quantum communication and quantum cryptography, and experiments of quantum advantage have already been proven with photonic devices [96, 97]. The research on the theoretical side can prove crucial in the road towards large scale FTQC, having the potential to reduce the resources required for FTQEC. The design of improved FT protocols for magic state distillation, which is one of the most significant bottlenecks of FTQC [169, 175, 224–227, 237, 382], or even the design of QEC codes and protocols that avoid the need for magic states (e.g. 3D color codes or stacked codes [238–240]), are some examples of theoretical challenges that can push forward the realization of large scale FTQC.

The research presented in this thesis contributes towards the existing challenges in FTQEC, both on the near and long term challenges. The realization of feasibility analysis plays a fundamental role in the development and evaluation of FT protocols (e.g. [171]), characterizing the bottlenecks and potential improvements. Furthermore, it can direct the experimental efforts towards the most damaging error sources, optimizing the use of resources and improving our understanding of the problems and challenges. With this purpose, in chapter 6 we present a feasibility study of the implementation of FTQEC protocols on near term ion trap devices. In this project, we have shown the existence of a beneficial regime for the application of QEC, and identified target values for the fidelities required on the experimental devices. Additionally, we have studied the impact of the different error sources that affect ion trap devices, using microscopic models and a realistic description of the noise processes present in ion trap experiments.



---

This modelling allows us to present realistic expectations for near term ion trap experiments. One aspect of particular relevance is the effect of crosstalk, a noise source that stems from unwanted interactions between the target ions and the neighboring ions during the application of quantum gates. The damaging effects of crosstalk have been reported by recent research, with potential measures to suppress it being a relevant topic within the field [101, 107, 139, 179, 306, 341–344]. In our work, we have shown how using refocusing pulse sequences can be successfully used to reduce the effects of crosstalk, placing the region of beneficial QEC within reach of near term experiments. Furthermore, we combine different numerical simulation techniques to study the effects of the coherent nature of crosstalk, showing the relevance of state vector simulations vs stabilizer simulations, which are often overlooked in the literature due to the higher computational cost. Overall, the work presents a comprehensive overview of the state and capabilities of near term ion traps, contributing to the development of the field and the implementation of FTQEC protocols.

Concerning future improvements, our techniques can be extended to account for most of the relevant existing challenges in the development of quantum computing devices and FT protocols. The noise models can be expanded to account for other sources of noise, like correlated noise and non-markovian noise, and include error models for other devices and platforms. For example, shuttling-based ion traps or superconducting qubits could be very interesting candidates. Furthermore, replacing the noise models with experimentally measured noise channels can lead to higher accuracy of the simulations and an improved understanding of the sources of errors that affect the devices. In another line, the simulations could be used to compare the efficacy of different FT protocols for different platforms, allowing the development of tailored protocols that address the particular characteristics of the noise on a given device. Finally, further study can be done in evaluating the precision of the different simulation techniques. As we have shown in our work, the approximations realized during stabilizer simulations can underestimate dangerous coherent sources of noise. However, the more precise state-vector simulations require a longer computational time, limiting the speed at which different approaches can be tested. An intermediate solution could be the use of an improved simulation algorithm, capable of estimating possible coherent amplification and adjusting the simulation accordingly to allow stabilizer simulations to predict more accurately the behavior of noise, while remaining a scalable method for a larger

## 9. CONCLUSIONS AND OUTLOOK

---

number of qubits. These techniques could notably speed up the pace of research, while state vector simulations could be reserved to confirm the final results.

The second line of work presented in this thesis is motivated by the long term goals of FTQEC. One of the challenges of scaling QEC codes lies in the capability of classical processing to interpret the syndrome measurements, which encode information about the location of the errors in the code. To operate on topological codes, particularly codes of larger distance, it is crucial to have an efficient decoding algorithm that provides a recovery operation with a high success rate. This rate is reflected in the threshold value, and it can, for some noise models, be benchmarked against known upper bounds on decoding performance obtained through mapping the quantum error correction problem onto a classical statistical-mechanical model [111, 218, 220]. However, this accuracy of the recovery operation needs to be balanced with the computational time required by the decoding algorithm. While for a quantum memory it is potentially fine to keep a backlog of measured error syndromes and compute the correction later during the classical post-processing phase, this is not the case anymore once we start to perform logical quantum computations. Here, the intermediate state of the computation will depend on the decoder outcome, such that the quantum computation may have to wait for the decoding algorithm to finish, time during which new errors can accumulate. The development of decoders for color codes has been and still is an active field of research [189–197, 201, 250, 251]. With this purpose, we study two efficient decoding algorithms for color codes in chapters 7 and 8. Color codes are particularly relevant for the future implementation of FTQC, as they present some of the highest qubit to code distance ratios and, in the case of the 4.8.8 color code, the capability of transversal application of the whole Clifford group [113]. Chapter 7 presents a Union-Find based decoder, inspired by recent works [193], with a computational complexity that scales almost linearly with the number of qubits  $N$ . This places the decoder among the most efficient decoders in the literature. On the other hand, chapter 8 presents a decoder based on rescaling for the square-octagon 4.8.8 color code lattice, with a computational complexity of  $\mathcal{O}(N \log(N))$ , that can be parallelized to achieve an  $\mathcal{O}(\log(N))$  scaling with the number of qubits  $N$ . Previous works on rescaling decoders for topological codes were focused on surface codes [187, 188] and the hexagonal color code lattice [189]. Our work provides a valuable extension of this decoder’s approach to the more relevant 4.8.8 color code lattice, capable of applying the complete set of Clifford gates

---

transversally [113, 238]. The application of the rescaling decoder to this lattice involves additional challenges due to the properties of the lattice, as shown in Chapter 8 and the published work [2].

The work on the decoders offers multiple possibilities for further research. The threshold found in our union-find decoder shown in chapter 7 reaches only 5.6% for the 4.8.8 and 7.0% for the hexagonal color code lattice, while the original work in [193] achieves a threshold around 8.6%. Our study on the percolation of the growing clusters suggests that the limiting factor of the decoder lies on the quick growth of the clusters, that can easily merge covering the entire lattice with a single cluster before the algorithm converges. A possible next step would be modifying the growth algorithm to expand the clusters preferentially in some directions, directing the growth towards the most probable candidates and reducing the percolation probability, at the cost of an increase in the computational complexity. An exciting extension of the ideas of the decoder would be using a noise model including qubit loss. Recent research [364, 406] proposed a method to modify the color code lattice and the stabilizers in the absence of lost qubits, in which stabilizers are cut and extended over the losses to preserve the logical operators and the color code properties. A union-find decoder could easily be extended for this scenario, as the growth of the clusters can be adapted for the modified shape of the stabilizers in non-regular lattices. Similarly, the decoder algorithm can be extended for the more experimentally relevant phenomenological and circuit noise models, introducing multiple rounds of stabilizer readout. However, addressing first the problem of the excessive growth of the clusters might be advisable, since this approach could also lead to a lower threshold than other existing decoders in the literature, like the restriction decoder [196].

Regarding the work on the rescaling decoder, there are multiple open fronts for further research. Modifying the decoder to account for open boundary conditions could improve considerably the applicability of the decoder, since the experimental implementation of periodic boundary conditions can be challenging for most physical platforms. As in the previous decoder, the extension to phenomenological noise and circuit noise models would be arguably the most interesting target. Any realistic quantum device is likely to experience faulty measurements and more complex noise patterns, making circuit noise models the most relevant for future applications. This could be achieved by using three-dimensional cells, an approach that was applied successfully in surface

## 9. CONCLUSIONS AND OUTLOOK

---

codes [187]. In this aspect, this decoder has the additional potential to introduce interesting noise models within the local cell decoder, where local parameters of the particular device (e.g. correlations between qubits) can be introduced into the algorithm, modifying the probabilities of different error patterns accordingly. Recent works have studied how these complex error models could be learned from experimental observations [179], which could then be used to model the cell decoder more accurately. Alternatively, the cell decoder could be replaced by a neural network, that could be trained to introduce precise information from the particular experimental device. Adapting the rescaling ideas to codes that allow the implementation of a universal set of transversal gates, which can substantially reduce the resources required for FT [238–240], would be an interesting extension. As for the computational complexity, the  $\mathcal{O}(N \log(N))$  scaling of the decoding time with the number of qubits  $N$  can be potentially improved to  $\mathcal{O}(\log(N))$  with parallelization, which would significantly improve the maximum lattice size to which the decoder could be applied and could lead to an advantage over other decoders in the literature. One of the decoder’s challenges lies in the lack of flexibility with respect to the system size, since the decoder is restricted to a number of qubits that can be rescaled exactly to the expected minimum lattice size. This issue can be overcome by replacing the brute force decoder applied to the smallest system size (for which the computational complexity scales exponentially) with an alternative, more efficient decoder (e.g. [196]). The information about the estimates of the error probabilities of the effective qubits resulting from the rescaling steps can be used by decoders based on MWPM, which can introduce these probabilities as weights for the matching algorithm. The study of these combined approaches could lead to interesting algorithms that could benefit from the particular advantages of the different decoders. Finally, further research on the impact of each of the steps of the decoder could lead to a better understanding of the overall algorithm. With our current research, we can only speculate about the factors that limit the threshold below the results obtained by alternative approaches found in the literature. An improved understanding of the multiple algorithms used in this complex decoder could inspire the development of new decoding algorithms and unveil interesting properties of the different color code lattices.

# References

- [1] P. Parrado-Rodríguez, C. Ryan-Anderson, A. Bermudez, and M. Müller, “Crosstalk suppression for fault-tolerant quantum error correction with trapped ions,” *Quantum* **5** (2021) 487. 9, 63, 72, 73, 75, 80, 82, 83, 85, 86, 87, 93, 102
- [2] P. Parrado-Rodríguez, M. Rispler, and M. Müller, “Rescaling decoder for 2D topological quantum color codes on 4.8.8 lattices,” (2021) , [arXiv:2112.09584](#). 9, 132, 162, 171
- [3] M. R. Williams, *A History of Computing Technology, 2nd Edition*. IEEE Computer Society Press, Washington, DC, USA, 2nd ed., 1997. 2
- [4] B. J. Copeland, “The Modern History of Computing,” in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, ed. Metaphysics Research Lab, Stanford University, Winter 2020 ed., 2020. 2
- [5] D. Swade and C. Babbage, *Difference Engine: Charles Babbage and the Quest to Build the First Computer*. Viking Penguin, 2001.
- [6] M. Godfrey, “First draft report on the EDVAC by John von Neumann,” *IEEE Annals of the History of Computing* **15** (1993) 27. 2
- [7] E. Winsberg, “Computer Simulations in Science,” in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, ed. Metaphysics Research Lab, Stanford University, Winter 2019 ed., 2019. 2
- [8] R. P. Feynman, “Simulating physics with computers,” in *Feynman and computation*, p. 133. CRC Press, 2018. 2, 3, 166
- [9] C. C. Cocks, “A note on non-secret encryption,” 1973. 3

## REFERENCES

---

- [10] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Commun. ACM* **21** no. 2, (1978) 120. 3
- [11] N. Ferguson and B. Schneier, *Practical Cryptography*. John Wiley and Sons, Inc., USA, 2003. 3
- [12] P. W. Shor, “Algorithms for quantum computation: discrete logarithms and factoring,” in *Proceedings 35th annual symposium on foundations of computer science*, p. 124, Ieee. 1994. 3
- [13] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, “Surface codes: Towards practical large-scale quantum computation,” *Physical Review A* **86** no. 3, (2012) 032324. 3, 6, 35, 36
- [14] D. S. Abrams and S. Lloyd, “Simulation of many-body fermi systems on a universal quantum computer,” *Physical Review Letters* **79** no. 13, (1997) 2586. 3, 166
- [15] G. D. las Cuevas, W. Dür, M. V. den Nest, and M. A. Martin-Delgado, “Quantum algorithms for classical lattice models,” *New Journal of Physics* **13** no. 9, (2011) 093021. 166
- [16] I. M. Georgescu, S. Ashhab, and F. Nori, “Quantum simulation,” *Rev. Mod. Phys.* **86** (2014) 153.
- [17] T. Schaetz, C. R. Monroe, and T. Esslinger, “Focus on quantum simulation,” *New Journal of Physics* **15** no. 8, (2013) 085009.
- [18] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, “Quantum computational chemistry,” *Rev. Mod. Phys.* **92** (2020) 015003. 3, 4, 166
- [19] I. Kassal, J. D. Whitfield, A. Perdomo-Ortiz, M.-H. Yung, and A. Aspuru-Guzik, “Simulating chemistry using quantum computers,” *Annual review of physical chemistry* **62** (2011) 185. 3, 4, 166

- 
- [20] D. Hayes, S. T. Flammia, and M. J. Biercuk, “Programmable quantum simulation by dynamic hamiltonian engineering,” *New Journal of Physics* **16** no. 8, (2014) 083027. 96
- [21] L. Bassman, M. Urbanek, M. Metcalf, J. Carter, A. F. Kemper, and W. de Jong, “Simulating quantum materials with digital quantum computers,” (2021) , [arXiv:2101.08836](https://arxiv.org/abs/2101.08836). 166
- [22] D.-W. Zhang, Y.-Q. Zhu, Y. X. Zhao, H. Yan, and S.-L. Zhu, “Topological quantum matter with cold atoms,” *Advances in Physics* **67** no. 4, (2018) 253. 166
- [23] A. Trabesinger, “Quantum simulation,” *Nature Physics* **8** (2012) 263.
- [24] R. Blatt and C. F. Roos, “Quantum simulations with trapped ions,” *Nature Physics* **8** no. 4, (2012) 277.
- [25] A. Houck, H. Türeci, and J. Koch, “On-chip quantum simulation with superconducting circuits,” *Nature Physics* **8** no. 4, (2012) 292. 166
- [26] A. Aspuru-Guzik and P. Walther, “Photonic quantum simulators,” *Nature Physics* **8** (2012) 285.
- [27] C. Schneider, D. Porras, and T. Schaetz, “Experimental quantum simulations of many-body physics with trapped ions,” *Reports on progress in physics. Physical Society (Great Britain)* **75** (2012) 024401. 60, 166
- [28] A. Friedenauer, H. Schmitz, J. Glueckert, D. Porras, and T. Schätz, “Simulating a quantum magnet with trapped ions,” *Nature Physics* **4** (2008) 757. 3, 166
- [29] D. Porras and J. I. Cirac, “Effective quantum spin systems with trapped ions,” *Phys. Rev. Lett.* **92** (2004) 207901.
- [30] I. Bloch, J. Dalibard, and S. Nascimbene, “Quantum simulations with ultracold quantum gases,” *Nature Physics* **8** (2012) 267.
- [31] D. A. Lidar and O. Biham, “Simulating ising spin glasses on a quantum computer,” *Phys. Rev. E* **56** (1997) 3661.

## REFERENCES

---

- [32] R. Harris, Y. Sato, A. J. Berkley, M. Reis, F. Altomare, M. H. Amin, K. Boothby, P. Bunyk, C. Deng, C. Enderud, S. Huang, E. Hoskinson, M. W. Johnson, E. Ladizinsky, N. Ladizinsky, T. Lanting, R. Li, T. Medina, R. Molavi, R. Neufeld, T. Oh, I. Pavlov, I. Perminov, G. Poulin-Lamarre, C. Rich, A. Smirnov, L. Swenson, N. Tsai, M. Volkmann, J. Whittaker, and J. Yao, “Phase transitions in a programmable quantum spin glass simulator,” *Science* **361** no. 6398, (2018) 162. 166
- [33] C. Navarrete-Benlloch, I. de Vega, D. Porras, and J. I. Cirac, “Simulating quantum-optical phenomena with cold atoms in optical lattices,” *New Journal of Physics* **13** no. 2, (2011) 023024. 3
- [34] J. Barreiro, M. Müller, P. Schindler, D. Nigg, T. Monz, M. Chwalla, M. Hennrich, C. F. Roos, P. Zoller, and R. Blatt, “An open-system quantum simulator with trapped ions,” *Nature* **470** (2011) 486. 3, 60
- [35] B. P. Lanyon, C. Hempel, D. Nigg, M. Müller, R. Gerritsma, F. Zähringer, P. Schindler, J. T. Barreiro, M. Rambach, G. Kirchmair, M. Hennrich, P. Zoller, R. Blatt, and C. F. Roos, “Universal digital quantum simulation with trapped ions,” *Science* **334** no. 6052, (2011) 57.
- [36] G. S. Paraoanu, “Recent progress in quantum simulation using superconducting circuits,” *Journal of Low Temperature Physics* **175** no. 5-6, (2014) 633. 166
- [37] C. Robens, S. Brakhane, D. Meschede, and A. Alberti, “Quantum walks with neutral atoms: Quantum interference effects of one and two particles,” *Laser Spectroscopy* (2016) 1.
- [38] C. Gross and I. Bloch, “Quantum simulations with ultracold atoms in optical lattices,” *Science* **357** no. 6355, (2017) 995. 166
- [39] R. Islam, E. Edwards, K. Kim, S. Korenblit, C. Noh, H. Carmichael, G.-D. Lin, L.-M. Duan, C.-C. Joseph Wang, J. Freericks, and et al., “Onset of a quantum phase transition with a trapped ion quantum simulator,” *Nature Communications* **2** no. 1, (2011) 377. 3



- 
- [40] S. Pirandola, “End-to-end capacities of a quantum communication network,” *Communications Physics* **2** (2019) 51. 3
- [41] Y. Zhao, *Quantum cryptography in real-life applications: Assumptions and security*. PhD thesis, University of Toronto, 2009.
- [42] C. Portmann and R. Renner, “Security in quantum cryptography,” (2021) , [arXiv:2102.00021](https://arxiv.org/abs/2102.00021).
- [43] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, J. L. Pereira, M. Razavi, J. S. Shaari, M. Tomamichel, V. C. Usenko, G. Vallone, P. Villoresi, and P. Wallden, “Advances in quantum cryptography,” *Adv. Opt. Photon.* **12** no. 4, (2020) 1012. 5
- [44] E. Diamanti, H.-K. Lo, B. Qi, and Z. Yuan, “Practical challenges in quantum key distribution,” *npj Quantum Information* **2** no. 1, (2016) 16025. 3
- [45] N. Shettell, W. J. Munro, D. Markham, and K. Nemoto, “Practical limits of error correction for quantum metrology,” *New Journal of Physics* **23** no. 4, (2021) 043038. 3
- [46] E. Polino, M. Valeri, N. Spagnolo, and F. Sciarrino, “Photonic quantum metrology,” *AVS Quantum Science* **2** no. 2, (2020) 024703.
- [47] T. Eberle, S. Steinlechner, J. Bauchrowitz, V. Händchen, H. Vahlbruch, M. Mehmet, H. Müller-Ebhardt, and R. Schnabel, “Quantum enhancement of the zero-area sagnac interferometer topology for gravitational wave detection,” *Phys. Rev. Lett.* **104** (2010) 251102.
- [48] S. L. Danilishin and F. Y. Khalili, “Quantum measurement theory in gravitational-wave detectors,” *Living Reviews in Relativity* **15** no. 1, (2012) 5.
- [49] M. A. Taylor and W. P. Bowen, “Quantum metrology and its application in biology,” *Physics Reports* **615** (2016) 1.
- [50] S. Fernández-Lorenzo and D. Porras, “Quantum sensing close to a dissipative phase transition: Symmetry breaking and criticality as metrological resources,” *Phys. Rev. A* **96** (2017) 013817.

## REFERENCES

---

- [51] D. S. Simon, G. Jaeger, and A. V. Sergienko, “Quantum metrology,” in *Quantum Metrology, Imaging, and Communication*, p. 91. Springer, 2017.
- [52] V. Giovannetti, S. Lloyd, and L. Maccone, “Advances in quantum metrology,” *Nature Photonics* **5** no. 4, (2011) 222. 3
- [53] H.-Y. Huang, M. Broughton, J. Cotler, S. Chen, J. Li, M. Mohseni, H. Neven, R. Babbush, R. Kueng, J. Preskill, and J. R. McClean, “Quantum advantage in learning from experiments,” (2021) , [arXiv:2112.00778](#). 3
- [54] X. Gao, Z. Zhang, and L. Duan, “An efficient quantum algorithm for generative machine learning,” (2017) , [arXiv:1711.02038](#). 4, 166
- [55] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, “Power of data in quantum machine learning,” *Nature Communications* **12** no. 1, (2021) 2631.
- [56] W. Guan, G. Perdue, A. Pesah, M. Schuld, K. Terashi, S. Vallecorsa, and J.-R. Vlimant, “Quantum machine learning in high energy physics,” *Machine Learning: Science and Technology* **2** no. 1, (2021) 011003.
- [57] M. Schuld, I. Sinayskiy, and F. Petruccione, “An introduction to quantum machine learning,” *Contemporary Physics* **56** no. 2, (2015) 172.
- [58] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, “Quantum machine learning,” *Nature* **549** no. 7671, (2017) 195. 3
- [59] A. Ajagekar, T. Humble, and F. You, “Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems,” *Computers & Chemical Engineering* **132** (2020) 106630. 3, 166
- [60] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, “Continuous-variable quantum neural networks,” *Physical Review Research* **1** no. 3, (2019) 033063.
- [61] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, “Hybrid quantum-classical algorithms and quantum error mitigation,” *Journal of the Physical Society of Japan* **90** no. 3, (2021) 032001. 166

- 
- [62] J. Liu, K. H. Lim, K. L. Wood, W. Huang, C. Guo, and H.-L. Huang, “Hybrid quantum-classical convolutional neural networks,” *Science China Physics, Mechanics & Astronomy* **64** no. 9, (2021) 290311.
- [63] N. Mathur, J. Landman, Y. Y. Li, M. Strahm, S. Kazdaghli, A. Prakash, and I. Kerenidis, “Medical image classification via quantum neural networks,” (2021) , [arXiv:2109.01831](https://arxiv.org/abs/2109.01831).
- [64] S. Fernández-Lorenzo, D. Porras, and J. J. García-Ripoll, “Hybrid quantum–classical optimization with cardinality constraints and applications to finance,” *Quantum Science and Technology* **6** no. 3, (2021) 034010. 3, 4, 166
- [65] J. Preskill, “Quantum Computing in the NISQ era and beyond,” *Quantum* **2** (2018) 79. 6, 166
- [66] A. Ajagekar and F. You, “Quantum computing for energy systems optimization: Challenges and opportunities,” *Energy* **179** (2019) 76.
- [67] M. Willsch, D. Willsch, F. Jin, H. De Raedt, and K. Michielsen, “Benchmarking the quantum approximate optimization algorithm,” *Quantum Information Processing* **19** no. 7, (2020) 197.
- [68] R. Orús, S. Mugel, and E. Lizaso, “Quantum computing for finance: Overview and prospects,” *Reviews in Physics* **4** (2019) 100028. 6, 166
- [69] A. Bouland, W. van Dam, H. Joorati, I. Kerenidis, and A. Prakash, “Prospects and challenges of quantum finance,” (2020) , [arXiv:2011.06492](https://arxiv.org/abs/2011.06492).
- [70] A. Luckow, J. Klepsch, and J. Pichlmeier, “Quantum computing: Towards industry reference problems,” *Digitale Welt* **5** no. 2, (2021) 38.
- [71] A. Perdomo-Ortiz, A. Feldman, A. Ozaeta, S. V. Isakov, Z. Zhu, B. O’Gorman, H. G. Katzgraber, A. Diedrich, H. Neven, J. de Kleer, and et al., “Readiness of quantum optimization machines for industrial applications,” *Physical Review Applied* **12** no. 1, (2019) 014004. 4, 166
- [72] M. Muradi and R. Wanka, “Sample-based motion planning for multi-robot systems,” in *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, p. 130, IEEE. 2020.

## REFERENCES

---

- [73] M. Streif, S. Yarkoni, A. Skolik, F. Neukart, and M. Leib, “Beating classical heuristics for the binary paint shop problem with the quantum approximate optimization algorithm,” *Physical Review A* **104** no. 1, (2021) 012403.
- [74] A. Mehta, M. Muradi, and S. Woldetsadick, “Quantum annealing based optimization of robotic movement in manufacturing,” in *International Workshop on Quantum Technology and Optimization Problems*, p. 136, Springer. 2019.
- [75] C. Roch and S. Langer, “The capacitated vehicle routing problem,” *Digitale Welt* **3** no. 2, (2019) 30.
- [76] V. E. Elfving, M. Millaruelo, J. A. Gámez, and C. Gogolin, “Simulating quantum chemistry in the seniority-zero space on qubit-based quantum computers,” *Physical Review A* **103** no. 3, (2021) 032605.
- [77] D. J. Moylett, N. Linden, and A. Montanaro, “Quantum speedup of the traveling-salesman problem for bounded-degree graphs,” *Physical Review A* **95** no. 3, (2017) 032323.
- [78] A. Crispin and A. Syrichas, “Quantum annealing algorithm for vehicle scheduling,” in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, p. 3523. 2013. 4, 6, 166
- [79] T. Albash and D. A. Lidar, “Adiabatic quantum computation,” *Rev. Mod. Phys.* **90** (2018) 015002. 4, 166
- [80] E. Farhi, J. Goldstone, and S. Gutmann, “A quantum approximate optimization algorithm,” (2014) , [arXiv:1411.4028](https://arxiv.org/abs/1411.4028). 4, 166
- [81] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, “A variational eigenvalue solver on a photonic quantum processor,” *Nature Communications* **5** no. 1, (2014) 4213.
- [82] P. Díez-Valle, D. Porras, and J. J. García-Ripoll, “Quantum variational optimization: The role of entanglement and problem hardness,” *Physical Review A* **104** no. 6, (2021) 062426. 166

- 
- [83] D. Amaro, C. Modica, M. Rosenkranz, M. Fiorentini, M. Benedetti, and M. Lubasch, “Filtering variational quantum algorithms for combinatorial optimization,” (2021), [arXiv:2106.10055](https://arxiv.org/abs/2106.10055).
- [84] A. Lucas, “Ising formulations of many np problems,” *Frontiers in Physics* **2** (2014) 5. 166
- [85] S. Hadfield, “Quantum algorithms for scientific computing and approximate optimization,” (2018), [arXiv:1805.03265](https://arxiv.org/abs/1805.03265). 4, 166
- [86] “Quantum computing report.”  
<https://quantumcomputingreport.com/players/>. Accessed: 2021-12-12. 4, 166
- [87] D. P. DiVincenzo, “The physical implementation of quantum computation,” *Fortschritte der Physik* **48** no. 9-11, (2000) 771. 4, 60
- [88] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O’Brien, “Quantum computers,” *Nature* **464** no. 7285, (2010) 45. 4, 61, 166
- [89] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, *et al.*, “Superconducting quantum circuits at the surface code threshold for fault tolerance,” *Nature* **508** no. 7497, (2014) 500. 25, 166, 167
- [90] C. Ballance, T. Harty, N. Linke, M. Sepiol, and D. Lucas, “High-fidelity quantum logic gates using trapped-ion hyperfine qubits,” *Physical Review Letters* **117** no. 6, (2016) 060504. 6, 60, 166, 168
- [91] Y. Wang, S. Crain, C. Fang, B. Zhang, S. Huang, Q. Liang, P. H. Leung, K. R. Brown, and J. Kim, “High-fidelity two-qubit gates using a microelectromechanical-system-based beam steering system for individual qubit addressing,” *Phys. Rev. Lett.* **125** (2020) 150505.
- [92] P. H. Leung, K. A. Landsman, C. Figgatt, N. M. Linke, C. Monroe, and K. R. Brown, “Robust 2-qubit gates in a linear ion crystal using a frequency-modulated driving force,” *Phys. Rev. Lett.* **120** (2018) 020501. 6, 60, 166

## REFERENCES

---

- [93] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, “Superconducting qubits: Current state of play,” *Annual Review of Condensed Matter Physics* **11** no. 1, (2020) 369. 6, 25, 166, 167
- [94] T. M. Graham, M. Kwon, B. Grinkemeyer, Z. Marra, X. Jiang, M. T. Lichtman, Y. Sun, M. Ebert, and M. Saffman, “Rydberg-mediated entanglement in a two-dimensional neutral atom qubit array,” *Phys. Rev. Lett.* **123** (2019) 230501. 4, 166
- [95] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, and et al., “Quantum supremacy using a programmable superconducting processor,” *Nature* **574** no. 7779, (2019) 505. 4, 166
- [96] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, and et al., “Quantum computational advantage using photons,” *Science* **370** no. 6523, (2020) 1460. 4, 168
- [97] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, M. Gong, C. Guo, C. Guo, S. Guo, L. Han, L. Hong, H.-L. Huang, Y.-H. Huo, L. Li, N. Li, S. Li, Y. Li, F. Liang, C. Lin, J. Lin, H. Qian, D. Qiao, H. Rong, H. Su, L. Sun, L. Wang, S. Wang, D. Wu, Y. Xu, K. Yan, W. Yang, Y. Yang, Y. Ye, J. Yin, C. Ying, J. Yu, C. Zha, C. Zhang, H. Zhang, K. Zhang, Y. Zhang, H. Zhao, Y. Zhao, L. Zhou, Q. Zhu, C.-Y. Lu, C.-Z. Peng, X. Zhu, and J.-W. Pan, “Strong quantum computational advantage using a superconducting quantum processor,” *Phys. Rev. Lett.* **127** (2021) 180501. 4, 166, 168
- [98] C. G. Almudever, L. Lao, X. Fu, N. Khammassi, I. Ashraf, D. Iorga, S. Varsamopoulos, C. Eichler, A. Wallraff, L. Geck, A. Kruth, J. Knoch, H. Bluhm, and K. Bertels, “The engineering challenges in quantum computing,” in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, p. 836. 2017. 4

- 
- [99] L. E. Ratchiff, S. Mohr, G. Huhs, T. Deutsch, M. Masella, and L. Genovese, “Challenges in large scale quantum mechanical calculations,” *WIREs Computational Molecular Science* **7** no. 1, (2017) e1290.
- [100] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000. 4, 5, 14, 16, 17, 18, 19, 25, 26, 27, 30, 31, 36, 51, 53, 132, 167
- [101] L. M. K. Vandersypen and I. L. Chuang, “Nmr techniques for quantum control and computation,” *Reviews of Modern Physics* **76** no. 4, (2005) 1037. 4, 5, 26, 74, 84, 102, 169
- [102] L. Viola, E. Knill, and S. Lloyd, “Dynamical decoupling of open quantum systems,” *Physical Review Letters* **82** no. 12, (1999) 2417. 83
- [103] M. H. Goerz, K. B. Whaley, and C. P. Koch, “Hybrid optimization schemes for quantum control,” *EPJ Quantum Technology* **2** (2015) 21.
- [104] H. Mabuchi and N. Khaneja, “Principles and applications of control in quantum systems,” *International Journal of Robust and Nonlinear Control* **15** (2005) 647.
- [105] R. Chakrabarti and H. Rabitz, “Quantum control landscapes,” *International Reviews in Physical Chemistry* **26** no. 4, (2007) 671.
- [106] H. M. Wiseman and G. J. Milburn, *Quantum Measurement and Control*. Cambridge University Press, 2009. 5, 26, 74
- [107] P. Murali, D. C. McKay, M. Martonosi, and A. Javadi-Abhari, “Software mitigation of crosstalk on noisy intermediate-scale quantum computers,” *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems* (2020) 1001. 74, 84, 102, 169
- [108] D. A. Lidar, *Review of Decoherence-Free Subspaces, Noiseless Subsystems, and Dynamical Decoupling*, p. 295. John Wiley & Sons, Inc., 2014. 4, 74, 96

## REFERENCES

---

- [109] B. M. Terhal, “Quantum error correction for quantum memories,” *Reviews of Modern Physics* **87** no. 2, (2015) 307. 5, 7, 24, 26, 36, 37, 38, 132, 167
- [110] A. Kitaev, “Fault-tolerant quantum computation by anyons,” *Annals of Physics* **303** no. 1, (2003) 2. 5, 38, 167
- [111] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, “Topological quantum memory,” *J. Math. Phys.* **43** no. 9, (2002) 4452. 5, 24, 46, 133, 170
- [112] A. Y. Kitaev, *Quantum Error Correction with Imperfect Gates*, p. 181. Springer US, Boston, MA, 1997. 38, 167
- [113] H. Bombin and M. A. Martin-Delgado, “Topological quantum distillation,” *Phys. Rev. Lett.* **97** no. 18, (2006) 180501. 5, 8, 24, 38, 40, 41, 42, 73, 133, 135, 167, 170, 171
- [114] D. A. Lidar and T. A. Brun, *Quantum error correction*. Cambridge University Press, 2013. 5, 7, 43, 73, 74
- [115] S. J. Devitt, W. J. Munro, and K. Nemoto, “Quantum error correction for beginners,” *Reports on Progress in Physics* **76** no. 7, (2013) 076001. 5, 24, 26, 27, 32
- [116] M. Grassl, T. Beth, and T. Pellizzari, “Codes for the quantum erasure channel,” *Phys. Rev. A* **56** (1997) 33. 5
- [117] T. C. Ralph, A. J. F. Hayes, and A. Gilchrist, “Loss-tolerant optical qubits,” *Phys. Rev. Lett.* **95** (2005) 100501.
- [118] M. Mirrahimi, Z. Leghtas, V. V. Albert, S. Touzard, R. J. Schoelkopf, L. Jiang, and M. H. Devoret, “Dynamically protected cat-qubits: a new paradigm for universal quantum computation,” *New Journal of Physics* **16** no. 4, (2014) 045014. 5
- [119] S. Lloyd and J.-J. E. Slotine, “Analog quantum error correction,” *Physical Review Letters* **80** no. 18, (1998) 4088. 5
- [120] S. L. Braunstein, “Error correction for continuous quantum variables,” in *Quantum Information with Continuous Variables*, p. 19. Springer, 1998.



- 
- [121] D. Gottesman, A. Kitaev, and J. Preskill, “Encoding a qubit in an oscillator,” *Physical Review A* **64** no. 1, (2001) 012310. 5
- [122] M. Grassl, L. Kong, Z. Wei, Z.-Q. Yin, and B. Zeng, “Quantum error-correcting codes for qudit amplitude damping,” *IEEE Transactions on Information Theory* **64** no. 6, (2018) 4674. 5
- [123] M. Hillery, V. Bužek, and A. Berthiaume, “Quantum secret sharing,” *Physical Review A* **59** no. 3, (1999) 1829. 5
- [124] D. Gottesman, “Theory of fault-tolerant quantum computation,” *Phys. Rev. A* **57** (1998) 127. 5
- [125] E. Knill, “Quantum computing with realistically noisy devices,” *Nature* **434** (2005) 39. Article. 166
- [126] D. P. DiVincenzo and P. W. Shor, “Fault-tolerant error correction with efficient quantum codes,” *Phys. Rev. Lett.* **77** (1996) 3260, [arXiv:quant-ph/9605031](#). 43, 73
- [127] P. W. Shor, “Fault-tolerant quantum computation,” in *Proceedings of 37th Conference on Foundations of Computer Science*, p. 56. IEEE, 1996. 5, 7, 25, 36, 132, 167
- [128] J. Preskill, *Fault-Tolerant Quantum Computation*, p. 213. World Scientific, 1996. 5, 24, 32
- [129] D. Aharonov and M. Ben-Or, “Fault-tolerant quantum computation with constant error rate,” *SIAM Journal on Computing* **38** no. 4, (2008) 1207. 5, 7, 36, 132, 167
- [130] J. Preskill, “Reliable quantum computers,” *Proc. R. Soc. Lond. A.* **454** no. 1969, (1998) 385. 5, 7, 25, 36, 132, 167
- [131] R. Raussendorf and J. Harrington, “Fault-tolerant quantum computation with high threshold in two dimensions,” *Physical Review Letters* **98** no. 19, (2007) 190504. 5, 7, 37, 38, 42, 167

## REFERENCES

---

- [132] H. Bombin and M. A. Martin-Delgado, “Topological computation without braiding,” *Phys. Rev. Lett.* **98** no. 16, (2007) 160502. 5, 38, 40, 133, 135, 167
- [133] A. Acín, I. Bloch, H. Buhrman, T. Calarco, C. Eichler, J. Eisert, D. Esteve, N. Gisin, S. J. Glaser, F. Jelezko, and et al., “The quantum technologies roadmap: a european community view,” *New Journal of Physics* **20** no. 8, (2018) 080201. 6
- [134] M. A. Rol, F. Battistel, F. K. Malinowski, C. C. Bultink, B. M. Tarasinski, R. Vollmer, N. Haider, N. Muthusubramanian, A. Bruno, B. M. Terhal, and L. DiCarlo, “Fast, high-fidelity conditional-phase gate exploiting leakage interference in weakly anharmonic superconducting qubits,” *Phys. Rev. Lett.* **123** (2019) 120502. 6, 167
- [135] V. Negîrneac, H. Ali, N. Muthusubramanian, F. Battistel, R. Sagastizabal, M. S. Moreira, J. F. Marques, W. J. Vlothuizen, M. Beekman, C. Zachariadis, N. Haider, A. Bruno, and L. DiCarlo, “High-fidelity controlled- $z$  gate with maximal intermediate leakage operating at the speed limit in a superconducting quantum processor,” *Phys. Rev. Lett.* **126** (2021) 220502.
- [136] Y. Ye, S. Cao, Y. Wu, X. Chen, Q. Zhu, S. Li, F. Chen, M. Gong, C. Zha, H.-L. Huang, Y. Zhao, S. Wang, S. Guo, H. Qian, F. Liang, J. Lin, Y. Xu, C. Guo, L. Sun, N. Li, H. Deng, X. Zhu, and J.-W. Pan, “Realization of high-fidelity controlled-phase gates in extensible superconducting qubits design with a tunable coupler,” *Chinese Physics Letters* **38** no. 10, (2021) 100301. 6
- [137] W. Huang, C. H. Yang, K. W. Chan, T. Tanttu, B. Hensen, R. C. C. Leon, M. A. Fogarty, J. C. C. Hwang, F. E. Hudson, K. M. Itoh, and et al., “Fidelity benchmarks for two-qubit gates in silicon,” *Nature* **569** no. 7757, (2019) 532. 6
- [138] F. Borjans, X. Mi, and J. Petta, “Spin digitizer for high-fidelity readout of a cavity-coupled silicon triple quantum dot,” *Phys. Rev. Applied* **15** (2021) 044052. 6
- [139] D. P. L. Aude Craik, N. M. Linke, M. A. Sepiol, T. P. Harty, J. F. Goodwin, C. J. Ballance, D. N. Stacey, A. M. Steane, D. M. Lucas, and D. T. C. Allcock, “High-fidelity spatial and polarization addressing of ca+43 qubits using

- 
- near-field microwave control,” *Physical Review A* **95** no. 2, (2017) 022337. 6, 60, 74, 84, 102, 169
- [140] A. C. Hughes, V. M. Schäfer, K. Thirumalai, D. P. Nadlinger, S. R. Woodrow, D. M. Lucas, and C. J. Ballance, “Benchmarking a high-fidelity mixed-species entangling gate,” *Phys. Rev. Lett.* **125** (2020) 080504. 166, 168
- [141] A. Erhard, J. J. Wallman, L. Postler, M. Meth, R. Stricker, E. A. Martinez, P. Schindler, T. Monz, J. Emerson, and R. Blatt, “Characterizing large-scale quantum computers via cycle benchmarking,” *Nature Communications* **10** no. 1, (2019) 5347.
- [142] T. P. Harty, M. A. Sepiol, D. T. C. Allcock, C. J. Ballance, J. E. Tarlton, and D. M. Lucas, “High-fidelity trapped-ion quantum logic using near-field microwaves,” *Phys. Rev. Lett.* **117** (2016) 140501.
- [143] J. P. Gaebler, T. R. Tan, Y. Lin, Y. Wan, R. Bowler, A. C. Keith, S. Glancy, K. Coakley, E. Knill, D. Leibfried, and D. J. Wineland, “High-fidelity universal gate set for  ${}^9\text{Be}^+$  ion qubits,” *Phys. Rev. Lett.* **117** (2016) 060505. 166
- [144] Y. Wang, M. Um, J. Zhang, S. An, M. Lyu, J.-N. Zhang, L.-M. Duan, D. Yum, and K. Kim, “Single-qubit quantum memory exceeding ten-minute coherence time,” *Nature Photonics* **11** no. 10, (2017) 646.
- [145] V. M. Schäfer, C. J. Ballance, K. Thirumalai, L. J. Stephenson, T. G. Ballance, A. M. Steane, and D. M. Lucas, “Fast quantum logic gates with trapped-ion qubits,” *Nature* **555** (2018) 75. 6, 60
- [146] M. J. Biercuk, H. Uys, A. P. Vandevender, N. Shiga, W. M. Itano, and J. J. Bollinger, “High-fidelity quantum control using ion crystals in a Penning trap,” *Quantum Info. Comput.* **9** no. 11, (2009) 920. 6, 62
- [147] D. Nigg, M. Müller, E. A. Martinez, P. Schindler, M. Hennrich, T. Monz, M. A. Martin-Delgado, and R. Blatt, “Quantum computations on a topologically encoded qubit,” *Science* **345** no. 6194, (2014) 302. 6, 25, 37, 60, 83, 167

## REFERENCES

---

- [148] M. Müller, A. Rivas, E. A. Martínez, D. Nigg, P. Schindler, T. Monz, R. Blatt, and M. A. Martin-Delgado, “Iterative phase optimization of elementary quantum error correcting codes,” *Phys. Rev. X* **6** (2016) 031030.
- [149] A. Bermudez, X. Xu, R. Nigmatullin, J. O’Gorman, V. Negnevitsky, P. Schindler, T. Monz, U. G. Poschinger, C. Hempel, J. Home, F. Schmidt-Kaler, M. Biercuk, R. Blatt, S. Benjamin, and M. Müller, “Assessing the progress of trapped-ion processors towards fault-tolerant quantum computation,” *Phys. Rev. X* **7** (2017) 041061. 6, 61, 62, 72, 73, 74, 94, 102, 168
- [150] P. Schindler, J. T. Barreiro, T. Monz, V. Nebendahl, D. Nigg, M. Chwalla, M. Hennrich, and R. Blatt, “Experimental repetitive quantum error correction,” *Science* **332** no. 6033, (2011) 1059. 29
- [151] D. Kielpinski, A. Ben-Kish, J. Britton, V. Meyer, M. Rowe, W. Itano, D. Wineland, C. Sackett, and C. Monroe, “Recent results in trapped-ion quantum computing at NIST,” *Quantum Information & Computation* **1** (12, 2001) 113. 62
- [152] J. Chiaverini, D. Leibfried, T. Schaetz, M. D. Barrett, R. B. Blakestad, J. Britton, W. M. Itano, J. D. Jost, E. Knill, C. Langer, R. Ozeri, and D. J. Wineland, “Realization of quantum error correction,” *Nature* **432** (2004) 602.
- [153] T. Olsacher, L. Postler, P. Schindler, T. Monz, P. Zoller, and L. M. Sieberer, “Scalable and parallel tweezer gates for quantum computing with long ion strings,” *PRX Quantum* **1** (2020) 020316.
- [154] V. Negnevitsky, M. Marinelli, K. K. Mehta, H. Y. Lo, C. Flühmann, and J. P. Home, “Repeated multi-qubit readout and feedback with a mixed-species trapped-ion register,” *Nature* **563** (2018) 527.
- [155] L. Egan, D. M. Debroy, C. Noel, A. Risinger, D. Zhu, D. Biswas, M. Newman, M. Li, K. R. Brown, M. Cetina, and C. Monroe, “Fault-tolerant operation of a quantum error-correction code,” (2021) , [arXiv:2009.11482](https://arxiv.org/abs/2009.11482). 36
- [156] D. Zhu, N. M. Linke, M. Benedetti, K. A. Landsman, N. H. Nguyen, C. H. Alderete, A. Perdomo-Ortiz, N. Korda, A. Garfoot, C. Brecque, L. Egan,

- 
- O. Perdomo, and C. Monroe, “Training of quantum circuits on a hybrid quantum computer,” *Science Advances* **5** no. 10, (2019) eaaw9918. 6, 60
- [157] J. Zhang, R. Laflamme, and D. Suter, “Experimental implementation of encoded logical qubit operations in a perfect quantum error correcting code,” *Phys. Rev. Lett.* **109** (2012) 100503. 6, 25
- [158] E. Knill, R. Laflamme, R. Martinez, and C. Negrevergne, “Benchmarking quantum computers: The five-qubit error correction code,” *Phys. Rev. Lett.* **86** no. 18, (2001) 5811. 6
- [159] J. Kelly, R. Barends, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, I.-C. Hoi, C. Neill, P. J. J. O’Malley, C. Quintana, P. Roushan, A. Vainsencher, J. Wenner, A. N. Cleland, and J. M. Martinis, “State preservation by repetitive error detection in a superconducting quantum circuit,” *Nature* **519** (2015) 66. 6, 25, 29
- [160] M. Takita, A. D. Córcoles, E. Magesan, B. Abdo, M. Brink, A. Cross, J. M. Chow, and J. M. Gambetta, “Demonstration of weight-four parity measurements in the surface code architecture,” *Phys. Rev. Lett.* **117** (2016) 210505.
- [161] N. Ofek, A. Petrenko, R. Heeres, P. Reinhold, Z. Leghtas, B. Vlastakis, Y. Liu, L. Frunzio, S. Girvin, L. Jiang, *et al.*, “Extending the lifetime of a quantum bit with error correction in superconducting circuits,” *Nature* **536** no. 7617, (2016) 441.
- [162] C. K. Andersen, A. Remm, S. Lazar, S. Krinner, N. Lacroix, G. J. Norris, M. Gabureac, C. Eichler, and A. Wallraff, “Repeated quantum error detection in a surface code,” *Nature Physics* **16** (2020) 875. 25
- [163] S. Resch and U. R. Karpuzcu, “Effectiveness of variable distance quantum error correcting codes,” (2021) , [arXiv:2112.10044](https://arxiv.org/abs/2112.10044). 6, 167
- [164] G. Waldherr, Y. Wang, S. Zaiser, M. Jamali, T. Schulte-Herbrüggen, H. Abe, T. Ohshima, J. Isoya, J. F. Du, P. Neumann, and et al., “Quantum error

## REFERENCES

---

- correction in a solid-state hybrid spin register,” *Nature* **506** no. 7487, (2014) 204. 6, 25
- [165] T. Unden, P. Balasubramanian, D. Louzon, Y. Vinkler, M. B. Plenio, M. Markham, D. Twitchen, A. Stacey, I. Lovchinsky, A. O. Sushkov, M. D. Lukin, A. Retzker, B. Naydenov, L. P. McGuinness, and F. Jelezko, “Quantum Metrology Enhanced by Repetitive Quantum Error Correction,” *Physical Review Letters* **116** (2016) 230502. 6, 25
- [166] D. Castelvecchi, “Ibm’s quantum cloud computer goes commercial,” *Nature News* **543** no. 7644, (2017) 159. 6
- [167] K. J. Satzinger, Y.-J. Liu, A. Smith, C. Knapp, M. Newman, C. Jones, Z. Chen, C. Quintana, X. Mi, A. Dunsworth, *et al.*, “Realizing topologically ordered states on a quantum processor,” *Science* **374** no. 6572, (2021) 1237. 6, 37, 166
- [168] J. F. Marques, B. M. Varbanov, M. S. Moreira, H. Ali, N. Muthusubramanian, C. Zachariadis, F. Battistel, M. Beekman, N. Haider, W. Vlothuizen, A. Bruno, B. M. Terhal, and L. DiCarlo, “Logical-qubit operations in an error-detecting surface code,” *Nat. Phys.* (2021) . 6
- [169] L. Postler, S. Heußen, I. Pogorelov, M. Rispler, T. Feldker, M. Meth, C. D. Marciniak, R. Stricker, M. Ringbauer, R. Blatt, P. Schindler, M. Müller, and T. Monz, “Demonstration of fault-tolerant universal quantum gate operations,” (2021) , [arXiv:2111.12654](https://arxiv.org/abs/2111.12654). 6, 8, 25, 36, 37, 50, 60, 73, 92, 101, 102, 103, 104, 166, 167, 168
- [170] J. Hilder, D. Pijn, O. Onishchenko, A. Stahl, M. Orth, B. Lekitsch, A. Rodriguez-Blanco, M. Müller, F. Schmidt-Kaler, and U. Poschinger, “Fault-tolerant parity readout on a shuttling-based trapped-ion quantum computer,” (2021) , [arXiv:2107.06368](https://arxiv.org/abs/2107.06368). 6, 25, 167
- [171] C. Ryan-Anderson, J. G. Bohnet, K. Lee, D. Gresh, A. Hankin, J. P. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, *et al.*, “Realization of real-time fault-tolerant quantum error correction,” (2021) , [arXiv:2107.07505](https://arxiv.org/abs/2107.07505). 6, 8, 25, 37, 50, 60, 62, 73, 88, 92, 93, 101, 103, 166, 167, 168

- 
- [172] S. Krinner, N. Lacroix, A. Remm, A. Di Paolo, E. Genois, C. Leroux, C. Hellings, S. Lazar, F. Swiadek, J. Herrmann, *et al.*, “Realizing repeated quantum error correction in a distance-three surface code,” (2021) , [arXiv:2112.03708](#).
- [173] C. K. Andersen, A. Remm, S. Lazar, S. Krinner, N. Lacroix, G. J. Norris, M. Gabureac, C. Eichler, and A. Wallraff, “Repeated quantum error detection in a surface code,” *Nature Physics* **16** no. 8, (2020) 875. 37
- [174] G. Q. AI, “Exponential suppression of bit or phase errors with cyclic error correction,” *Nature* **595** no. 7867, (2021) 383. 6, 29
- [175] E. T. Campbell, B. M. Terhal, and C. Vuillot, “Roads towards fault-tolerant universal quantum computation,” *Nature* **549** no. 7671, (2017) 172. 6, 167, 168
- [176] C. J. Trout, M. Li, M. Gutiérrez, Y. Wu, S.-T. Wang, L. Duan, and K. R. Brown, “Simulating the performance of a distance-3 surface code in a linear ion trap,” *New Journal of Physics* **20** no. 4, (2018) 043038. 6, 73, 74
- [177] M. Gutiérrez, M. Müller, and A. Bermúdez, “Transversality and lattice surgery: Exploring realistic routes toward coupled logical qubits with trapped-ion quantum processors,” *Phys. Rev. A* **99** (2019) 022330. 72
- [178] A. Bermudez, X. Xu, M. Gutiérrez, S. C. Benjamin, and M. Müller, “Fault-tolerant protection of near-term trapped-ion topological qubits under realistic noise sources,” *Phys. Rev. A* **100** (2019) 062307. 72, 73, 74, 75, 76, 77, 78, 79, 82, 88, 90, 94, 102
- [179] R. Harper, S. T. Flammia, and J. J. Wallman, “Efficient learning of quantum noise,” *Nature Physics* **16** no. 12, (2020) 1184. 6, 74, 102, 163, 169, 172
- [180] R. Jozsa, “On the simulation of quantum circuits,” (2006) , [arXiv:quant-ph/0603163](#). 7
- [181] C. Ryan-Anderson, *Quantum Algorithms, Architecture, and Error Correction*. PhD thesis, The University of New Mexico, 2018. 50, 57, 74

## REFERENCES

---

- [182] S. Bravyi, D. Browne, P. Calpin, E. Campbell, D. Gosset, and M. Howard, “Simulation of quantum circuits by low-rank stabilizer decompositions,” *Quantum* **3** (2019) 181. 7, 51, 104
- [183] A. J. Landahl, J. T. Anderson, and P. R. Rice, “Fault-tolerant quantum computing with color codes,” (2011) , [arXiv:1108.5738](#). 7, 25, 41, 42, 48
- [184] E. Berlekamp, R. McEliece, and H. van Tilborg, “On the inherent intractability of certain coding problems (corresp.),” *IEEE Transactions on Information Theory* **24** no. 3, (1978) 384. 7, 25
- [185] V. Kolmogorov, “Blossom v: a new implementation of a minimum cost perfect matching algorithm,” *Mathematical Programming Computation* **1** (2009) 43. 7, 46, 47, 106, 134
- [186] A. G. Fowler, “Minimum weight perfect matching of fault-tolerant topological quantum error correction in average  $o(1)$  parallel time,” (2014) , [arXiv:1307.1740](#). 46, 48
- [187] G. Duclos-Cianci and D. Poulin, “Fault-tolerant renormalization group decoder for abelian topological codes,” *Quantum Info. Comput.* **14** no. 9 & 10, (2014) 721. 9, 132, 163, 170, 172
- [188] G. Duclos-Cianci and D. Poulin, “Fast Decoders for Topological Quantum Codes,” *Phys. Rev. Lett.* **104** no. 5, (2010) 050504. 7, 135, 170
- [189] P. Sarvepalli and R. Raussendorf, “Efficient decoding of topological color codes,” *Physical Review A* **85** no. 2, (2012) 022317. 7, 8, 9, 48, 107, 132, 133, 135, 138, 143, 144, 145, 157, 170
- [190] D. S. Wang, A. G. Fowler, C. D. Hill, and L. C. L. Hollenberg, “Graphical algorithms and threshold error rates for the 2d colour code,” *Quant. Inf. Comp.* **10** (2010) 780. 48
- [191] A. M. Stephens, “Efficient fault-tolerant decoding of topological color codes,” (2014) , [arXiv:1402.3037](#). 48



- 
- [192] N. Delfosse, “Decoding color codes by projection onto surface codes,” *Physical Review A* **89** no. 1, (2014) 012317. 47, 107, 118, 119, 128, 163
- [193] N. Delfosse and N. H. Nickerson, “Almost-linear time decoding algorithm for topological codes,” *Quantum* **5** (2021) 595. 8, 9, 48, 106, 107, 108, 109, 116, 118, 119, 126, 128, 129, 134, 162, 170, 171
- [194] N. Delfosse and G. Zémor, “Linear-time maximum likelihood decoding of surface codes over the quantum erasure channel,” *Physical Review Research* **2** no. 3, (2020) 033042. 48, 106, 112
- [195] A. Kubica and J. Preskill, “Cellular-automaton decoders with provable thresholds for topological codes,” *Physical Review Letters* **123** no. 2, (2019) 020501. 48
- [196] A. Kubica and N. Delfosse, “Efficient color code decoders in  $d \geq 2$  dimensions from toric code decoders,” (2019) , [arXiv:1905.07393](https://arxiv.org/abs/1905.07393). 7, 47, 48, 107, 118, 119, 126, 128, 129, 130, 134, 135, 162, 163, 171, 172
- [197] P. Baireuther, M. D. Caio, B. Criger, C. W. J. Beenakker, and T. E. O’Brien, “Neural network decoder for topological color codes with circuit level noise,” *New Journal of Physics* **21** no. 1, (2019) 013003. 7, 48, 50, 56, 133, 170
- [198] S. Varsamopoulos, B. Criger, and K. Bertels, “Decoding small surface codes with feedforward neural networks,” *Quantum Science and Technology* **3** no. 1, (2017) 015004.
- [199] G. Torlai and R. G. Melko, “Neural decoder for topological codes,” *Physical review letters* **119** no. 3, (2017) 030501.
- [200] Y.-H. Liu and D. Poulin, “Neural belief-propagation decoders for quantum error-correcting codes,” *Phys. Rev. Lett.* **122** (2019) 200501.
- [201] N. Maskara, A. Kubica, and T. Jochym-O’Connor, “Advantages of versatile neural-network decoding for topological codes,” *Physical Review A* **99** no. 5, (2019) 052351. 48, 133, 170

## REFERENCES

---

- [202] N. P. Breuckmann and X. Ni, “Scalable Neural Network Decoders for Higher Dimensional Quantum Codes,” *Quantum* **2** (2018) 68.
- [203] X. Ni, “Neural Network Decoders for Large-Distance 2D Toric Codes,” *Quantum* **4** (2020) 310.
- [204] C. Chinni, A. Kulkarni, D. M. Pai, K. Mitra, and P. K. Sarvepalli, “Neural decoder for topological codes using pseudo-inverse of parity check matrix,” (2019) , [arXiv:1901.07535](https://arxiv.org/abs/1901.07535). 7, 48
- [205] M. B. Plenio and S. S. Virmani, *An Introduction to Entanglement Theory*, p. 173. Springer International Publishing, Cham, 2014. 17
- [206] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, “Quantum entanglement,” *Reviews of Modern Physics* **81** no. 2, (2009) 865–942.
- [207] D. Amaro and M. Müller, “Design and experimental performance of local entanglement witness operators,” *Phys. Rev. A* **101** (2020) 012317. 17
- [208] A. Y. Kitaev, “Quantum computations: algorithms and error correction,” *Russian Mathematical Surveys* **52** no. 6, (1997) 1191. 18, 24
- [209] C. Dawson and M. Nielsen, “The solovay-kitaev algorithm,” *Quantum Inf. Comput.* **6** (2006) 81. 18
- [210] K. Mølmer and A. Sørensen, “Multiparticle entanglement of hot trapped ions,” *Phys. Rev. Lett.* **82** no. 9, (1999) 1835. 18, 67, 68
- [211] D. Gottesman, *Stabilizer codes and quantum error correction*. PhD thesis, California Institute of Technology, 1997. 19, 26, 50, 51
- [212] D. Gottesman, “The heisenberg representation of quantum computers,” (1998) , [quant-ph/9807006](https://arxiv.org/abs/quant-ph/9807006). 19, 51
- [213] A. R. Calderbank and P. W. Shor, “Good quantum error-correcting codes exist,” *Phys. Rev. A* **54** (1996) 1098. 24, 30, 135
- [214] A. M. Steane, “Error correcting codes in quantum theory,” *Phys. Rev. Lett.* **77** no. 5, (1996) 793. 24, 30, 38, 42, 73, 135

- 
- [215] D. P. DiVincenzo and P. Aliferis, “Effective Fault-Tolerant Quantum Computation with Slow Measurements,” *Phys. Rev. Lett.* **98** (2007) 020501, [arXiv:quant-ph/0607047](#). 24, 43, 73
- [216] E. Knill, R. Laflamme, and W. H. Zurek, “Resilient quantum computation: error models and thresholds,” *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* **454** no. 1969, (1998) 365. 25
- [217] D. Vodola, M. Rispler, S. Kim, and M. Müller, “Fundamental thresholds of realistic quantum error correction circuits from classical spin models,” (2021) , [arXiv:2104.04847](#). 25, 37
- [218] H. G. Katzgraber, H. Bombin, and M. A. Martin-Delgado, “Error threshold for color codes and random three-body ising models,” *Physical Review Letters* **103** no. 9, (2009) 090501. 48, 133, 134, 170
- [219] M. Ohzeki, “Accuracy thresholds of topological color codes on the hexagonal and square-octagonal lattices,” *Phys. Rev. E* **80** (2009) 011141.
- [220] R. S. Andrist, H. G. Katzgraber, H. Bombin, and M. A. Martin-Delgado, “Tricolored lattice gauge theory with randomness: fault tolerance in topological color codes,” *New Journal of Physics* **13** no. 8, (2011) 083006. 25, 37, 48, 133, 134, 170
- [221] J. Napp and J. Preskill, “Optimal bacon-shor codes,” *Quantum Info. Comput.* **13** no. 5–6, (2013) 490. 26
- [222] J. Von Neumann, “Probabilistic logics and the synthesis of reliable organisms from unreliable components,” in *Automata Studies*, p. 43. Princeton University Press, 1955. 32
- [223] B. Eastin and E. Knill, “Restrictions on transversal encoded quantum gate sets,” *Phys. Rev. Lett.* **102** no. 11, (2009) 110502. 33, 41, 133
- [224] S. Bravyi and A. Kitaev, “Universal quantum computation with ideal clifford gates and noisy ancillas,” *Physical Review A* **71** no. 2, (2005) 022316. 34, 35, 39, 41, 133, 168

## REFERENCES

---

- [225] C. Chamberland and K. Noh, “Very low overhead fault-tolerant magic state preparation using redundant ancilla encoding and flag qubits,” *npj Quantum Information* **6** no. 1, (2020) 91. 36, 73
- [226] A. Krishna and J.-P. Tillich, “Towards low overhead magic state distillation,” *Phys. Rev. Lett.* **123** (2019) 070507. 39
- [227] D. Litinski, “Magic state distillation: Not as costly as you think,” *Quantum* **3** (2019) 205. 34, 41, 133, 167, 168
- [228] C. Chamberland and K. Noh, “Very low overhead fault-tolerant magic state preparation using redundant ancilla encoding and flag qubits,” *npj Quantum Information* **6** no. 1, (2020) 91. 35, 43
- [229] B. W. Reichardt, “Fault-tolerant quantum error correction for steane’s seven-qubit color code with few or no extra qubits,” *Quantum Science and Technology* **6** no. 1, (2020) 015007. 36, 43, 45, 46, 73, 74, 75, 88, 89, 90, 100, 101, 102, 103
- [230] R. Versluis, S. Poletto, N. Khammassi, B. Tarasinski, N. Haider, D. J. Michalak, A. Bruno, K. Bertels, and L. DiCarlo, “Scalable quantum circuit and control for a superconducting surface code,” *Phys. Rev. Applied* **8** (2017) 034021. 37
- [231] M. Takita, A. D. Córcoles, E. Magesan, B. Abdo, M. Brink, A. Cross, J. M. Chow, and J. M. Gambetta, “Demonstration of weight-four parity measurements in the surface code architecture,” *Phys. Rev. Lett.* **117** (2016) 210505. 37
- [232] C. D. de Albuquerque, G. G. L. Guardia, R. P. J. au2, C. R. de Oliveira Quilles Queiroz, and V. L. Vieira, “Euclidean and hyperbolic asymmetric topological quantum codes,” (2021) , [arXiv:2105.01144](https://arxiv.org/abs/2105.01144). 38
- [233] A. J. Landahl and C. Ryan-Anderson, “Quantum computing by color-code lattice surgery,” (2014) , [arXiv:1407.5103](https://arxiv.org/abs/1407.5103). 39
- [234] H. Bombin, C. Dawson, R. V. Mishmash, N. Nickerson, F. Pastawski, and S. Roberts, “Logical blocks for fault-tolerant topological quantum computation,” (2021) , [arXiv:2112.12160](https://arxiv.org/abs/2112.12160). 39

- 
- [235] D. Amaro, *Characterization and implementation of robust quantum information processing*. PhD thesis, Swansea University, 2020. 40
- [236] T. J. Yoder and I. H. Kim, “The surface code with a twist,” *Quantum* **1** (2017) 2. 41
- [237] M. E. Beverland, A. Kubica, and K. M. Svore, “Cost of universality: A comparative study of the overhead of state distillation and code switching with color codes,” *PRX Quantum* **2** no. 2, (2021) 020341. 41, 130, 133, 167, 168
- [238] H. Bombín, “Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes,” *New Journal of Physics* **17** no. 8, (2015) 083002. 41, 42, 168, 171, 172
- [239] T. Jochym-O’Connor and S. D. Bartlett, “Stacked codes: Universal fault-tolerant quantum computation in a two-dimensional layout,” *Phys. Rev. A* **93** (2016) 022323. 41
- [240] S. Bravyi and A. Cross, “Doubled color codes,” (2015) , [arXiv:1509.03239](https://arxiv.org/abs/1509.03239). 41, 168, 172
- [241] C. Chamberland, A. Kubica, T. J. Yoder, and G. Zhu, “Triangular color codes on trivalent graphs with flag qubits,” *New Journal of Physics* **22** no. 2, (2020) 023019. 41, 42, 43, 47, 73
- [242] D. S. Wang, A. G. Fowler, and L. C. L. Hollenberg, “Surface code quantum computing with error rates over 1%,” *Phys. Rev. A* **83** (2011) 020302. 42
- [243] M. Li, M. Gutiérrez, S. E. David, A. Hernandez, and K. R. Brown, “Fault tolerance with bare ancillary qubits for a  $[[7,1,3]]$  code,” *Phys. Rev. A* **96** (2017) 032341. 43, 44, 73
- [244] C. Chamberland and M. E. Beverland, “Flag fault-tolerant error correction with arbitrary distance codes,” *Quantum* **2** (2018) 53. 43, 73
- [245] R. Chao and B. W. Reichardt, “Quantum Error Correction with only Two Extra Qubits,” *Physical Review Letters* **121** (2018) 050502. 43, 73, 74

## REFERENCES

---

- [246] R. Chao and B. Reichardt, “Fault-tolerant quantum computation with few qubits,” *npj Quantum Information* **4** (2017) 42. 43, 73, 74
- [247] J. Edmonds, “Paths, trees, and flowers,” *Can. J. Math.* **17** (1965) 449. 46, 47, 106, 134
- [248] H. N. Gabow, “The weighted matching approach to maximum cardinality matching,” *Fund. Inform.* **154** no. 1-4, (2017) 109. 47, 134
- [249] R. Duan, S. Pettie, and H.-H. Su, “Scaling algorithms for weighted matching in general graphs,” *ACM Trans. Algorithms* **14** no. 1, (2018) 1. 47, 106, 134
- [250] C. Chamberland and P. Ronagh, “Deep neural decoders for near term fault-tolerant experiments,” *Quantum Science and Technology* **3** no. 4, (2018) 044002. 48, 133, 170
- [251] A. Davaasuren, Y. Suzuki, K. Fujii, and M. Koashi, “General framework for constructing fast and near-optimal machine-learning-based decoder of the topological stabilizer codes,” *Physical Review Research* **2** no. 3, (2020) 033399. 48, 133, 170
- [252] E. Magesan, D. Puzzuoli, C. E. Granade, and D. G. Cory, “Modeling quantum noise for efficient testing of fault-tolerant circuits,” *Physical Review A* **87** no. 1, (2013) 012324. 50, 51
- [253] T. O’Brien, B. Tarasinski, and L. DiCarlo, “Density-matrix simulation of small surface codes under current and projected experimental noise,” *npj Quantum Information* **3** no. 1, (2017) 1. 50, 56
- [254] P. Aliferis, D. Gottesman, and J. Preskill, “Quantum accuracy threshold for concatenated distance-3 codes,” (2005) , [arXiv:quant-ph/0504218](https://arxiv.org/abs/quant-ph/0504218). 50
- [255] D. Poulin, A. Qarry, R. Somma, and F. Verstraete, “Quantum simulation of time-dependent hamiltonians and the convenient illusion of hilbert space,” *Phys. Rev. Lett.* **106** (2011) 170501. 50
- [256] Y. Zhou, E. M. Stoudenmire, and X. Waintal, “What limits the simulation of quantum computers?,” *Physical Review X* **10** no. 4, (2020) 041038. 50

- 
- [257] D. Pérez-García, F. Verstraete, M. M. Wolf, and J. I. Cirac, “Matrix product state representations,” *Quantum Inf. Comput.* **7** no. 5, (2007) 401. 50
- [258] M. Suzuki, *Quantum Monte Carlo methods in condensed matter physics*. World scientific, 1993. 50
- [259] Z.-X. Li and H. Yao, “Sign-problem-free fermionic quantum monte carlo: Developments and applications,” *Annual Review of Condensed Matter Physics* **10** no. 1, (2019) 337.
- [260] E. Andriyash and M. H. Amin, “Can quantum monte carlo simulate quantum annealing?,” (2017) , [arXiv:1703.09277](https://arxiv.org/abs/1703.09277).
- [261] L. T. Brady and W. van Dam, “Quantum monte carlo simulations of tunneling in quantum adiabatic optimization,” *Physical Review A* **93** no. 3, (2016) 032304.
- [262] M. B. Plenio and P. L. Knight, “The quantum-jump approach to dissipative dynamics in quantum optics,” *Rev. Mod. Phys.* **70** (1998) 101. 50, 52
- [263] R. Martoňák, G. E. Santoro, and E. Tosatti, “Quantum annealing by the path-integral monte carlo method: The two-dimensional random ising model,” *Phys. Rev. B* **66** (2002) 094203. 50
- [264] A. D. King, J. Raymond, T. Lanting, S. V. Isakov, M. Mohseni, G. Poulin-Lamarre, S. Ejtemaee, W. Bernoudy, I. Ozfidan, A. Y. Smirnov, *et al.*, “Scaling advantage over path-integral monte carlo in quantum simulation of geometrically frustrated magnets,” *Nature communications* **12** no. 1, (2021) 1. 50
- [265] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science* **220** no. 4598, (1983) 671. 50
- [266] G. B. Mbeng, L. Privitera, L. Arceci, and G. E. Santoro, “Dynamics of simulated quantum annealing in random ising chains,” *Physical Review B* **99** no. 6, (2019) 064201. 50

## REFERENCES

---

- [267] S. Aaronson and D. Gottesman, “Improved simulation of stabilizer circuits,” *Physical Review A* **70** no. 5, (2004) 052328. 50, 51
- [268] M. Gutierrez, L. Svec, A. Vargo, and K. R. Brown, “Approximation of real error channels by Clifford channels and Pauli measurements,” *Phys. Rev. A* **87** (2013) 030302. 51, 77, 78, 79
- [269] C. Gidney, “Stim: a fast stabilizer circuit simulator,” *Quantum* **5** (2021) 497. 50, 51
- [270] S. R. White, “Density matrix formulation for quantum renormalization groups,” *Phys. Rev. Lett.* **69** (1992) 2863. 50
- [271] S. R. White, “Density-matrix algorithms for quantum renormalization groups,” *Phys. Rev. B* **48** (1993) 10345.
- [272] J. Prior, A. W. Chin, S. F. Huelga, and M. B. Plenio, “Efficient simulation of strong system-environment interactions,” *Phys. Rev. Lett.* **105** (2010) 050404.
- [273] F. Verstraete, D. Porras, and J. I. Cirac, “Density matrix renormalization group and periodic boundary conditions: A quantum information perspective,” *Phys. Rev. Lett.* **93** (2004) 227205.
- [274] R. Hübener, C. Kruszynska, L. Hartmann, W. Dür, F. Verstraete, J. Eisert, and M. B. Plenio, “Renormalization algorithm with graph enhancement,” *Phys. Rev. A* **79** (2009) 022317.
- [275] A. Baiardi, A. K. Kelemen, and M. Reiher, “Excited-state dmrg made simple with feast,” *Journal of Chemical Theory and Computation* (2021) .
- [276] U. Schollwöck, “The density-matrix renormalization group,” *Rev. Mod. Phys.* **77** (2005) 259. 50
- [277] P. Rall, D. Liang, J. Cook, and W. Kretschmer, “Simulation of qubit quantum circuits via pauli propagation,” *Physical Review A* **99** no. 6, (2019) 062337. 50, 57, 104



- 
- [278] A. Li, B. Fang, C. Granade, G. Prawiroatmodjo, B. Heim, M. Roetteler, and S. Krishnamoorthy, “Sv-sim: Scalable pgas-based state vector simulation of quantum circuits,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '21. Association for Computing Machinery, New York, NY, USA, 2021. 50
- [279] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F. J. Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C.-F. Chen, *et al.*, “Qiskit: An open-source framework for quantum computing.” Zenodo. 50
- [280] C. Ryan-Anderson, “Pecos: Performance estimator of codes on surfaces,” 2018. <https://github.com/PECOS-packages/PECOS>. 74
- [281] D. S. Steiger, T. Häner, and M. Troyer, “ProjectQ: an open source software framework for quantum computing,” *Quantum* **2** (2018) 49. 53, 74
- [282] “List of QC simulators.” <https://www.quantiki.org/wiki/list-qc-simulators>. Accessed: 2021-12-12. 50
- [283] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, “Trapped-ion quantum computing: Progress and challenges,” *Applied Physics Reviews* **6** (2019) 021314. 60, 63, 76, 166, 168
- [284] P. Schindler, D. Nigg, T. Monz, J. T. Barreiro, E. Martinez, S. X. Wang, S. Quint, M. F. Brandl, V. Nebendahl, C. F. Roos, M. Chwalla, M. Hennrich, and R. Blatt, “A quantum information processor with trapped ions,” *New Journal of Physics* **15** no. 12, (2013) 123012.
- [285] R. Ozeri, “The trapped-ion qubit tool box,” *Contemporary Physics* **52** no. 6, (2011) 531. 62
- [286] H. Haffner, C. Roos, and R. Blatt, “Quantum computing with trapped ions,” *Physics Reports* **469** no. 4, (2008) 155. 65

## REFERENCES

---

- [287] D. J. Wineland, M. Barrett, J. Britton, J. Chiaverini, B. DeMarco, W. M. Itano, B. Jelenković, C. Langer, D. Leibfried, V. Meyer, and et al., “Quantum information processing with trapped ions,” *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* **361** no. 1808, (2003) 1349.
- [288] D. J. Wineland, “Quantum information processing and quantum control with trapped atomic ions,” *Physica Scripta* **T137** (2009) 014007. 61
- [289] R. Blatt and D. Wineland, “Entangled states of trapped atomic ions,” *Nature* **453** (2008) 1008.
- [290] Z. D. Romaszko, S. Hong, M. Siegele, R. K. Puddy, F. R. Lebrun-Gallagher, S. Weidt, and W. K. Hensinger, “Engineering of microfabricated ion traps and integration of advanced on-chip features,” *Nature Reviews Physics* **2** no. 6, (2020) 285. 60
- [291] T. Graß, D. Raventós, B. Juliá-Díaz, C. Gogolin, and M. Lewenstein, “Quantum annealing for the number-partitioning problem using a tunable spin glass of ions,” *Nature Communications* **7** no. 1, (2016) 11524. 60
- [292] P. Hauke, H. G. Katzgraber, W. Lechner, H. Nishimori, and W. D. Oliver, “Perspectives of quantum annealing: methods and implementations,” *Reports on Progress in Physics* **83** no. 5, (2020) 054401. 166
- [293] D. Raventós, T. Graß, B. Juliá-Díaz, and M. Lewenstein, “Semiclassical approach to finite-temperature quantum annealing with trapped ions,” *Phys. Rev. A* **97** (2018) 052310. 60
- [294] J. I. Cirac and P. Zoller, “Quantum computations with cold trapped ions,” *Phys. Rev. Lett.* **74** (1995) 4091. 60
- [295] C. E. Langer, *High Fidelity Quantum Information Processing with Trapped Ions*. PhD thesis, University of Colorado at Boulder, 2006. 60
- [296] N. M. Linke, M. Gutierrez, K. A. Landsman, C. Figgatt, S. Debnath, K. R. Brown, and C. Monroe, “Fault-tolerant quantum error detection,” *Science Advances* **3** no. 10, (2017) e1701074. 60

- 
- [297] S. Jain, J. Alonso, M. Grau, and J. P. Home, “Scalable arrays of micro-penning traps for quantum computing and simulation,” *Phys. Rev. X* **10** (2020) 031027. 61, 62, 168
- [298] C. Monroe and J. Kim, “Scaling the ion trap quantum processor,” *Science* **339** no. 6124, (2013) 1164. 62
- [299] D. Kielpinski, C. Monroe, and D. J. Wineland, “Architecture for a large-scale ion-trap quantum computer,” *Nature* **417** no. 6890, (2002) 709. 61, 62, 168
- [300] K. A. Landsman, Y. Wu, P. H. Leung, D. Zhu, N. M. Linke, K. R. Brown, L. Duan, and C. Monroe, “Two-qubit entangling gates within arbitrarily long chains of trapped ions,” *Physical Review A* **100** no. 2, (2019) 022332. 62, 74, 102
- [301] S. Kim, *Acousto-Optic Devices for Optical Signal Processing and Quantum Computing*. PhD thesis, University of Colorado at Boulder, USA, 2008. 74, 102
- [302] N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, “Experimental comparison of two quantum computing architectures,” *Proceedings of the National Academy of Sciences* **114** no. 13, (2017) 3305. 62, 68, 167
- [303] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L.-M. Duan, and J. Kim, “Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects,” *Phys. Rev. A* **89** (2014) 022317. 62, 63
- [304] M. Cetina, A. Bylinskii, L. Karpa, D. Gangloff, K. M. Beck, Y. Ge, M. Scholz, A. T. Grier, I. Chuang, and V. Vuletić, “One-dimensional array of ion chains coupled to an optical cavity,” *New Journal of Physics* **15** no. 5, (2013) 053001. 62, 63
- [305] V. Kaushal, B. Lekitsch, A. Stahl, J. Hilder, D. Pijn, C. Schmiegelow, A. Bermudez, M. Müller, F. Schmidt-Kaler, and U. Poschinger, “Shuttling-based trapped-ion quantum information processing,” (2019) , [arXiv:1912.04712](https://arxiv.org/abs/1912.04712). 62
- [306] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, and et al.,

## REFERENCES

---

- “Demonstration of the trapped-ion quantum ccd computer architecture,”  
*Nature* **592** no. 7853, (2021) 209. 74, 102, 166, 168, 169
- [307] H. Kaufmann, T. Ruster, C. T. Schmiegelow, M. A. Luda, V. Kaushal, J. Schulz, D. Von Lindenfels, F. Schmidt-Kaler, and U. G. Poschinger, “Scalable Creation of Long-Lived Multipartite Entanglement,” *Physical Review Letters* **119** (2017) 150503.
- [308] Y. Wan, D. Kienzler, S. D. Erickson, K. H. Mayer, T. R. Tan, J. J. Wu, H. M. Vasconcelos, S. Glancy, E. Knill, D. J. Wineland, A. C. Wilson, and D. Leibfried, “Quantum gate teleportation between separated qubits in a trapped-ion processor,” *Science* **364** no. 6443, (2019) 875. 168
- [309] B. Lekitsch, S. Weidt, A. G. Fowler, K. Mølmer, S. J. Devitt, C. Wunderlich, and W. K. Hensinger, “Blueprint for a microwave trapped ion quantum computer,” *Science Advances* **3** no. 2, (2017) e1601540.
- [310] M. Lee, J. Jeong, Y. Park, C. Jung, T. Kim, and D. il Cho, “Ion shuttling method for long-range shuttling of trapped ions in MEMS-fabricated ion traps,” *Japanese Journal of Applied Physics* **60** no. 2, (2021) 027004. 62
- [311] D. Leibbrandt, J. Labaziewicz, R. Clark, I. Chuang, R. Epstein, C. Ospelkaus, J. Wesenberg, J. Bollinger, D. Leibfried, D. Wineland, D. Stick, J. Sterk, C. Monroe, C.-S. Pai, Y. Low, R. Frahm, and R. Slusher, “Demonstration of a scalable, multiplexed ion trap for quantum information processing,” *Quantum Information & Computation* **9** (2011) 901. 62, 63
- [312] L. S. Brown and G. Gabrielse, “Geonium theory: Physics of a single electron or ion in a penning trap,” *Rev. Mod. Phys.* **58** (1986) 233. 62
- [313] J. D. Baltrusch, A. Negretti, J. M. Taylor, and T. Calarco, “Fast and robust quantum computation with ionic wigner crystals,” *Phys. Rev. A* **83** (2011) 042319.
- [314] F. Galve, P. Fernández, and G. Werth, “Operation of a planar penning trap,” *The European Physical Journal D-Atomic, Molecular, Optical and Plasma Physics* **40** no. 2, (2006) 201.

- 
- [315] C.-C. J. Wang, A. C. Keith, and J. K. Freericks, “Phonon-mediated quantum spin simulator employing a planar ionic crystal in a penning trap,” *Phys. Rev. A* **87** (2013) 013422.
- [316] J. G. Bohnet, B. C. Sawyer, J. W. Britton, M. L. Wall, A. M. Rey, M. Foss-Feig, and J. J. Bollinger, “Quantum spin dynamics and entanglement generation with hundreds of trapped ions,” *Science* **352** no. 6291, (2016) 1297. 62
- [317] G. Ciaramicoli, F. Galve, I. Marzoli, and P. Tombesi, “Array of planar penning traps as a nuclear magnetic resonance molecule for quantum computation,” *Phys. Rev. A* **72** (2005) 042323. 62
- [318] D. Crick, S. Donnellan, S. Ananthamurthy, R. Thompson, and D. Segal, “Fast shuttling of ions in a scalable penning trap array,” *Review of Scientific Instruments* **81** no. 1, (2010) 013111. 62, 63
- [319] S. Stortini and I. Marzoli, “Composite pulses for quantum computation with trapped electrons,” *The European Physical Journal D-Atomic, Molecular, Optical and Plasma Physics* **32** no. 2, (2005) 209. 62
- [320] G. Ciaramicoli, I. Marzoli, and P. Tombesi, “Trapped electrons in vacuum for a scalable quantum processor,” *Phys. Rev. A* **70** (2004) 032301.
- [321] P. Bushev, S. Stahl, R. Natali, G. Marx, E. Stachowska, G. Werth, M. Hellwig, and F. Schmidt-Kaler, “Electrons in a cryogenic planar penning trap and experimental challenges for quantum processing,” *The European Physical Journal D* **50** no. 1, (2008) 97. 168
- [322] L. Lamata, D. Porras, J. I. Cirac, J. Goldman, and G. Gabrielse, “Towards electron-electron entanglement in penning traps,” *Phys. Rev. A* **81** (2010) 022301. 62, 168
- [323] J. W. Britton, B. C. Sawyer, A. C. Keith, C.-C. J. Wang, J. K. Freericks, H. Uys, M. J. Biercuk, and J. J. Bollinger, “Engineered two-dimensional ising interactions in a trapped-ion quantum simulator with hundreds of spins,” *Nature* **484** no. 7395, (2012) 489. 62

## REFERENCES

---

- [324] A. Safavi-Naini, R. J. Lewis-Swan, J. G. Bohnet, M. Gärttner, K. A. Gilmore, J. E. Jordan, J. Cohn, J. K. Freericks, A. M. Rey, and J. J. Bollinger, “Verification of a many-ion simulator of the dicke model through slow quenches across a phase transition,” *Phys. Rev. Lett.* **121** (2018) 040503. 62
- [325] S.-T. Wang, C. Shen, and L.-M. Duan, “Quantum computation under micromotion in a planar ion crystal,” *Scientific reports* **5** no. 1, (2015) 1. 63
- [326] Q. Yu, A. M. Alonso, J. Caminiti, K. M. Beck, R. T. Sutherland, D. Leibfried, K. J. Rodriguez, M. Dhital, B. Hemmerling, and H. Häffner, “Feasibility study of quantum computing using trapped electrons,” (2021) , [arXiv:2112.04034](https://arxiv.org/abs/2112.04034). 63, 168
- [327] C. Simon and W. T. M. Irvine, “Robust long-distance entanglement and a loophole-free bell test with ions and photons,” *Phys. Rev. Lett.* **91** (2003) 110405. 63
- [328] D. L. Moehring, P. Maunz, S. Olmschenk, K. C. Younge, D. N. Matsukevich, L.-M. Duan, and C. Monroe, “Entanglement of single-atom quantum bits at a distance,” *Nature* **449** no. 7158, (2007) 68.
- [329] N. H. Nickerson, J. F. Fitzsimons, and S. C. Benjamin, “Freely scalable quantum technologies using cells of 5-to-50 qubits with very lossy and noisy photonic links,” *Phys. Rev. X* **4** (2014) 041041.
- [330] M. Bock, P. Eich, S. Kucera, M. Kreis, A. Lenhard, C. Becher, and J. Eschner, “High-fidelity entanglement between a trapped ion and a telecom photon via quantum frequency conversion,” *Nature Communications* **9** no. 1, (2018) 1998.
- [331] L. J. Stephenson, D. P. Nadlinger, B. C. Nichol, S. An, P. Drmota, T. G. Ballance, K. Thirumalai, J. F. Goodwin, D. M. Lucas, and C. J. Ballance, “High-rate, high-fidelity entanglement of qubits across an elementary quantum network,” *Phys. Rev. Lett.* **124** (2020) 110501.
- [332] S. Todaro, V. Verma, K. McCormick, D. Allcock, R. Mirin, D. Wineland, S. Nam, A. Wilson, D. Leibfried, and D. Slichter, “State readout of a trapped ion qubit using a trap-integrated superconducting photon detector,” *Physical Review Letters* **126** no. 1, (2021) 010501. 63

- 
- [333] M. Barrett, J. Chiaverini, T. Schätz, J. Britton, W. Itano, J. Jost, E. Knill, C. Langer, D. Leibfried, R. Ozeri, and D. Wineland, “Deterministic quantum teleportation of atomic qubits,” *Nature* **429** (2004) 737. 63
- [334] M. Müller, L. Liang, I. Lesanovsky, and P. Zoller, “Trapped rydberg ions: from spin chains to fast quantum gates,” *New Journal of Physics* **10** no. 9, (2008) 093009. 63
- [335] C. Zhang, F. Pokorny, W. Li, G. Higgins, A. Pöschl, I. Lesanovsky, and M. Hennrich, “Submicrosecond entangling gate between trapped ions via rydberg interaction,” *Nature* **580** no. 7803, (2020) 345. 63
- [336] D. J. Wineland, C. Monroe, W. M. Itano, D. Leibfried, B. E. King, and D. M. Meekhof, “Experimental issues in coherent quantum-state manipulation of trapped atomic ions,” *Journal of research of the National Institute of Standards and Technology* **103** no. 3, (1998) 259. 65
- [337] D. Leibfried, R. Blatt, C. Monroe, and D. Wineland, “Quantum dynamics of single trapped ions,” *Rev. Mod. Phys.* **75** (2003) 281. 65
- [338] A. Sørensen and K. Mølmer, “Quantum computation with ions in thermal motion,” *Physical Review Letters* **82** (1999) 1971. 67, 68
- [339] C. F. Roos, “Ion trap quantum gates with amplitude-modulated laser beams,” *New Journal of Physics* **10** (2008) 013002. 67, 68
- [340] P. O. Boykin, T. Mor, M. Pulver, V. Roychowdhury, and F. Vatan, “On Universal and Fault-Tolerant Quantum Computing: A Novel Basis and a New Constructive Proof of Universality for Shor’s Basis,” in *Proceedings of the 40th Annual Symposium on Foundations of Computer Science, FOCS ’99*, p. 486. IEEE Computer Society, Washington, DC, USA, 1999. 68
- [341] M. Sarovar, T. Proctor, K. Rudinger, K. Young, E. Nielsen, and R. Blume-Kohout, “Detecting crosstalk errors in quantum information processors,” (2019) , [arXiv:1908.09855](https://arxiv.org/abs/1908.09855). 74, 102, 169

## REFERENCES

---

- [342] A. V. C. Piltz, T. Sriarunothai and C. Wunderlich, “A trapped-ion-based quantum byte with 10<sup>-5</sup> next-neighbour cross-talk,” *Nature Communications* **5** no. 1, (2014) 4679. 84
- [343] M. Reagor, C. B. Osborn, N. Tezak, A. Staley, G. Prawiroatmodjo, M. Scheer, N. Alidoust, E. A. Sete, N. Didier, M. P. da Silva, and et al., “Demonstration of universal parametric entangling gates on a multi-qubit lattice,” *Science Advances* **4** no. 2, (2018) eaao3603. 74, 84
- [344] D. M. Debroy, M. Li, S. Huang, and K. R. Brown, “Logical performance of 9 qubit compass codes in ion traps with crosstalk errors,” *Quantum Science and Technology* **5** no. 3, (2020) 034002. 74, 94, 96, 102, 169
- [345] R. Harper, S. T. Flammia, and J. J. Wallman, “Efficient learning of quantum noise,” *Nature Physics* (2020) 1184. 74, 84
- [346] L. Gerster, F. Martínez-García, P. Hrmo, M. van Mourik, B. Wilhelm, D. Vodola, M. Müller, R. Blatt, P. Schindler, and T. Monz, “Experimental bayesian calibration of trapped ion entangling operations,” (2021) , [arXiv:2112.01411](https://arxiv.org/abs/2112.01411). 74
- [347] F. Martínez-García, L. Gerster, D. Vodola, P. Hrmo, T. Monz, P. Schindler, and M. Müller, “Analytical and experimental study of center line miscalibrations in mølmer-sørensen gates,” (2021) , [arXiv:2112.05447](https://arxiv.org/abs/2112.05447). 74, 82
- [348] T. Haner, D. S. Steiger, K. Svore, and M. Troyer, “A software methodology for compiling quantum programs,” *Quantum Science and Technology* **3** no. 2, (Feb, 2018) 020501. 74
- [349] A. G. Fowler, “Coping with qubit leakage in topological codes,” *Phys. Rev. A* **88** (2013) 042308. 79
- [350] R. Stricker, D. Vodola, A. Erhard, L. Postler, M. Meth, M. Ringbauer, P. Schindler, R. Blatt, M. Müller, and T. Monz, “Characterizing quantum instruments: from non-demolition measurements to quantum error correction,” (2021) , [arXiv:2110.06954](https://arxiv.org/abs/2110.06954). 79



- 
- [351] R. Stricker, D. Vodola, A. Erhard, L. Postler, M. Meth, M. Ringbauer, P. Schindler, T. Monz, M. Müller, and R. Blatt, “Experimental deterministic correction of qubit loss,” *Nature* **585** no. 7824, (2020) 207. 79
- [352] J. T. Merrill, S. C. Doret, G. Vittorini, J. P. Addison, and K. R. Brown, “Transformed composite sequences for improved qubit addressing,” *Phys. Rev. A* **90** (2014) 040301. 83
- [353] M. R. Geller and Z. Zhou, “Efficient error models for fault-tolerant architectures and the pauli twirling approximation,” *Physical Review A* **88** no. 1, (2013) . 85, 86
- [354] A. Katabarwa and M. R. Geller, “Logical error rate in the pauli twirling approximation,” *Scientific reports* **5** no. 1, (2015) 1. 85, 86
- [355] J. Emerson, M. Silva, O. Moussa, C. Ryan, M. Laforest, J. Baugh, D. G. Cory, and R. Laflamme, “Symmetrized characterization of noisy quantum processes,” *Science* **317** no. 5846, (2007) 1893. 86
- [356] Y. Tomita and K. M. Svore, “Low-distance surface codes under realistic quantum noise,” *Phys. Rev. A* **90** (2014) 062320. 86
- [357] P. Schindler. Private communication, 2020. 87
- [358] H. Häffner, S. Gulde, M. Riebe, G. Lancaster, C. Becher, J. Eschner, F. Schmidt-Kaler, and R. Blatt, “Precision measurement and compensation of optical stark shifts for an ion-trap quantum processor,” *Physical Review Letters* **90** no. 14, (2003) 143602. 94
- [359] D. W. Leung, I. L. Chuang, F. Yamaguchi, and Y. Yamamoto, “Efficient implementation of coupled logic gates for quantum computation,” *Physical Review A* **61** no. 4, (2000) 042310. 96
- [360] M. Müller, K. Hammerer, Y. L. Zhou, C. F. Roos, and P. Zoller, “Simulating open quantum systems: from many-body interactions to stabilizer pumping,” *New Journal of Physics* **13** no. 8, (2011) 085007. 97

## REFERENCES

---

- [361] V. Nebendahl, “Optimized quantum error-correction codes for experiments,” *Phys. Rev. A* **91** (2015) 022332. 97
- [362] R. E. Tarjan, “Efficiency of a good but not linear set union algorithm,” *J. ACM* **22** no. 2, (1975) 215. 106, 111, 112, 124
- [363] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd ed., 2009. 111, 117, 124
- [364] D. Amaro, J. Bennett, D. Vodola, and M. Müller, “Analytical percolation theory for topological color codes under qubit loss,” *Physical Review A* **101** no. 3, (2020) 032317. 129, 171
- [365] H. Bombin, “An introduction to topological quantum codes,” (2013) , [arXiv:1311.0277](https://arxiv.org/abs/1311.0277). 135
- [366] J. Pearl, “Reverend Bayes on inference engines: a distributed hierarchical approach,” in *AAAI’82: Proceedings of the Second AAAI Conference on Artificial Intelligence*, p. 133. AAAI Press, 1982. 144
- [367] D. J. C. MacKay and R. M. Neal, “Near Shannon limit performance of low density parity check codes,” *Electron. Lett.* **32** no. 18, (1996) 1645. 145
- [368] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. 144
- [369] D. Poulin and Y. Chung, “On the iterative decoding of sparse quantum codes,” *Quantum Inf. Comput.* **8** no. 10, (2008) 987. 145
- [370] K. Svore, A. Cross, I. Chuang, and A. Aho, “A flow-map model for analyzing pseudothresholds in fault-tolerant quantum computing,” *Quantum Information & Computation* **6** (2006) 193. 161
- [371] C. Wang, J. Harrington, and J. Preskill, “Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory,” *Ann. Phys.* **303** no. 1, (2003) 31. 161
- [372] Y. Rinott, T. Shoham, and G. Kalai, “Statistical aspects of the quantum supremacy demonstration,” (2021) , [arXiv:2008.05177](https://arxiv.org/abs/2008.05177). 166

- 
- [373] S. Jordan, “The quantum algorithm zoo.”  
<https://quantumalgorithmzoo.org/>. Accessed: 2021-12-12. 166
- [374] A. Montanaro, “Quantum algorithms: an overview,” *npj Quantum Information* **2** no. 1, (2016) 15023. 166
- [375] M. Mosca, “Quantum algorithms,” (2008) , [arXiv:0808.0369](https://arxiv.org/abs/0808.0369).
- [376] D. Bacon and W. van Dam, “Recent progress in quantum algorithms,”  
*Commun. ACM* **53** (2010) 84. 166
- [377] M. Saffman, “Quantum computing with atomic qubits and rydberg interactions: progress and challenges,” *Journal of Physics B: Atomic, Molecular and Optical Physics* **49** no. 20, (2016) 202001. 166
- [378] G. Wendin, “Quantum information processing with superconducting circuits: a review,” *Reports on Progress in Physics* **80** no. 10, (2017) 106001. 167
- [379] H.-L. Huang, D. Wu, D. Fan, and X. Zhu, “Superconducting quantum computing: A review,” (2020) , [arXiv:2006.10433](https://arxiv.org/abs/2006.10433). 167
- [380] S. Slussarenko and G. J. Pryde, “Photonic quantum information processing: A concise review,” *Applied Physics Reviews* **6** no. 4, (2019) 041303. 166, 168
- [381] K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. C. Pisi, M. Chmielewski, C. Collins, and et al., “Benchmarking an 11-qubit quantum computer,” *Nature Communications* **10** no. 1, (2019) 5464. 166
- [382] B. Barch, R. Mohseninia, and D. Lidar, “Low overhead universality and quantum supremacy using only  $\mathbf{Z}$  control,” *Physical Review Research* **3** no. 3, (2021) 033207. 166, 168
- [383] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations,” *Phys. Rev. Lett.* **103** (2009) 150502.
- [384] A. Montanaro and S. Pallister, “Quantum algorithms and the finite element method,” *Physical Review A* **93** no. 3, (2016) 032324.

## REFERENCES

---

- [385] O. Kyriienko, A. E. Paine, and V. E. Elfving, “Solving nonlinear differential equations with differentiable quantum circuits,” *Physical Review A* **103** no. 5, (2021) 052416. 166
- [386] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, “Noisy intermediate-scale quantum (nisq) algorithms,” (2021) , [arXiv:2101.08448](https://arxiv.org/abs/2101.08448). 166
- [387] E. Martín-López, A. Laing, T. Lawson, R. Alvarez, X.-Q. Zhou, and J. L. O’Brien, “Experimental realization of shor’s quantum factoring algorithm using qubit recycling,” *Nature Photonics* **6** no. 11, (2012) 773. 166
- [388] L. M. K. Vandersypen, M. Steffen, M. H. Sherwood, C. S. Yannoni, G. Breyta, and I. L. Chuang, “Implementation of a three-quantum-bit search algorithm,” *Applied Physics Letters* **76** no. 5, (2000) 646–648. 166
- [389] J. King, S. Yarkoni, M. M. Nevisi, J. P. Hilton, and C. C. McGeoch, “Benchmarking a quantum annealing processor with the time-to-target metric,” (2015) , [arXiv:1508.05087](https://arxiv.org/abs/1508.05087). 166
- [390] S. Barz, I. Kassal, M. Ringbauer, Y. O. Lipp, B. Dakić, A. Aspuru-Guzik, and P. Walther, “A two-qubit photonic quantum processor and its application to solving systems of linear equations,” *Scientific Reports* **4** no. 1, (2014) 6115. 166
- [391] J. Pan, Y. Cao, X. Yao, Z. Li, C. Ju, H. Chen, X. Peng, S. Kais, and J. Du, “Experimental realization of quantum algorithm for solving linear systems of equations,” *Physical Review A* **89** no. 2, (2014) 022313. 166
- [392] R. Blume-Kohout, J. K. Gamble, E. Nielsen, K. Rudinger, J. Mizrahi, K. Fortier, and P. Maunz, “Demonstration of qubit operations below a rigorous fault tolerance threshold with gate set tomography,” *Nature Communications* **8** no. 1, (2017) 14485. 166
- [393] S. S. Hong, A. T. Papageorge, P. Sivarajah, G. Crossman, N. Didier, A. M. Polloreno, E. A. Sete, S. W. Turkowski, M. P. da Silva, and B. R. Johnson, “Demonstration of a parametrically activated entangling gate protected from flux noise,” *Phys. Rev. A* **101** (2020) 012302. 167

- 
- [394] A. M. Childs and W. van Dam, “Quantum algorithms for algebraic problems,” *Reviews of Modern Physics* **82** no. 1, (2010) 1–52. 167
- [395] M. H. Devoret and R. J. Schoelkopf, “Superconducting circuits for quantum information: an outlook,” *Science* **339** no. 6124, (2013) 1169. 167
- [396] L. Gyongyosi and S. Imre, “A survey on quantum computing technology,” *Computer Science Review* **31** (2019) 51.
- [397] H. O. Everitt, *Experimental aspects of quantum computing*, vol. 61. Springer, 2005.
- [398] National Academies of Sciences, Engineering, and Medicine, *Quantum Computing: Progress and Prospects*. The National Academies Press, Washington, DC, 2019. 167
- [399] J. P. Gaebler, T. R. Tan, Y. Lin, Y. Wan, R. Bowler, A. C. Keith, S. Glancy, K. Coakley, E. Knill, D. Leibfried, and D. J. Wineland, “High-fidelity universal gate set for  ${}^9\text{Be}^+$  ion qubits,” *Phys. Rev. Lett.* **117** (2016) 060505. 168
- [400] M. J. Harlander, *Architecture for a scalable ion-trap quantum computer*. PhD thesis, University of Innsbruck, 2021. 168
- [401] K. R. Brown, J. Chiaverini, J. M. Sage, and H. Häffner, “Materials challenges for trapped-ion quantum computers,” *Nature Reviews Materials* **6** (2021) 892.
- [402] J. Goldman and G. Gabrielse, “Optimized planar penning traps for quantum-information studies,” *Phys. Rev. A* **81** (2010) 052335.
- [403] G. Yang, A. Fragner, G. Koolstra, L. Ocola, D. A. Czapski, R. J. Schoelkopf, and D. I. Schuster, “Coupling an ensemble of electrons on superfluid helium to a superconducting circuit,” *Phys. Rev. X* **6** (2016) 011031. 168
- [404] Error: Reference not found.
- [405] N. Daniilidis, D. J. Gorman, L. Tian, and H. Häffner, “Quantum information processing with trapped electrons and superconducting electronics,” *New Journal of Physics* **15** no. 7, (2013) 073017. 168

## REFERENCES

---

- [406] D. Vodola, D. Amaro, M. A. Martin-Delgado, and M. Müller, “Twins percolation for qubit losses in topological color codes,” *Phys. Rev. Lett.* **121** (2018) 060501. 171