POLITECNICO
MILANO 1863

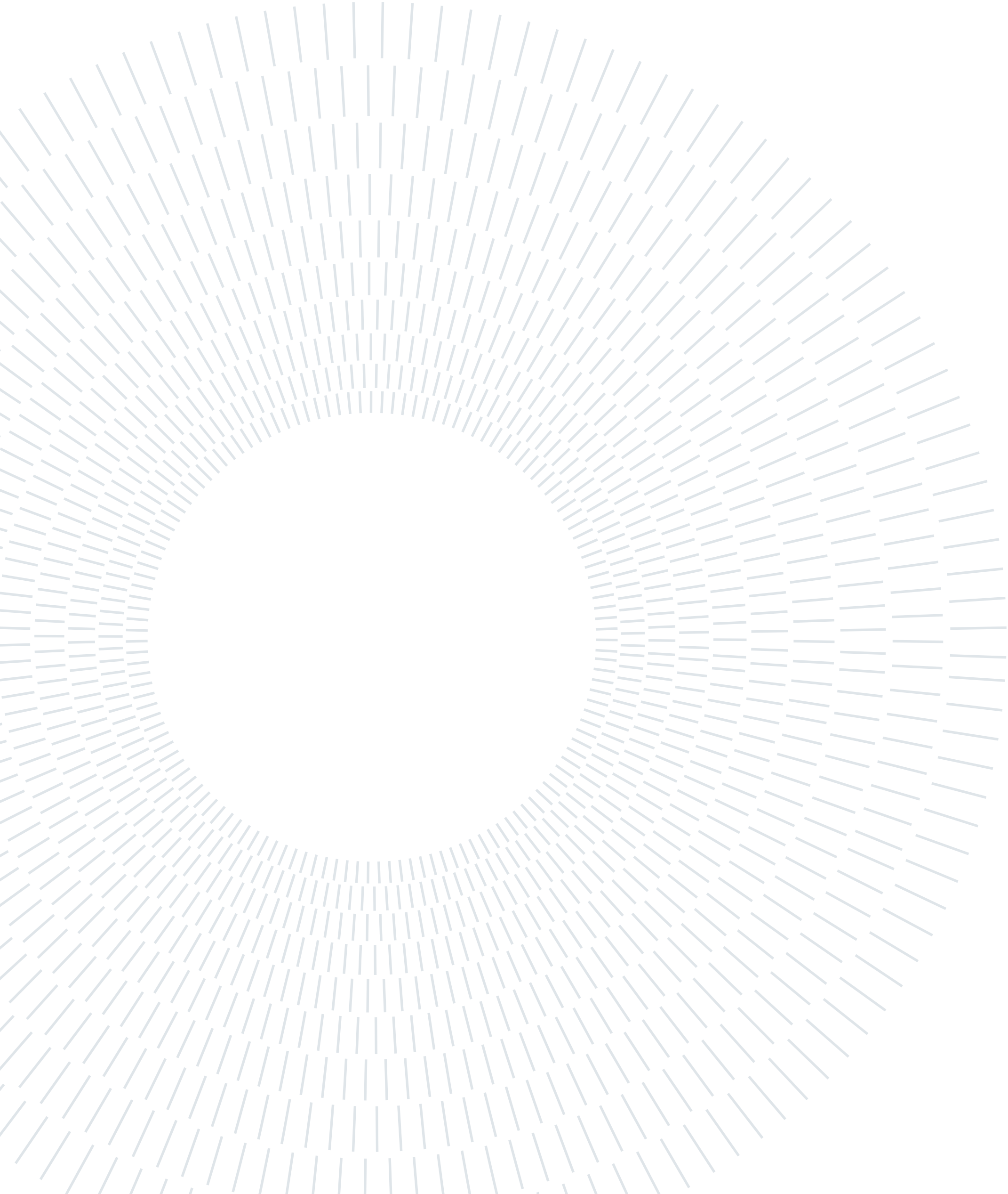
SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

QUANTUM COMPUTATION ON THE TORIC CODE

Tesi di Laurea Magistrale in
Computer Science and Engineering - Ingegneria Informatica

Author: **Alessia Conca Roncari**

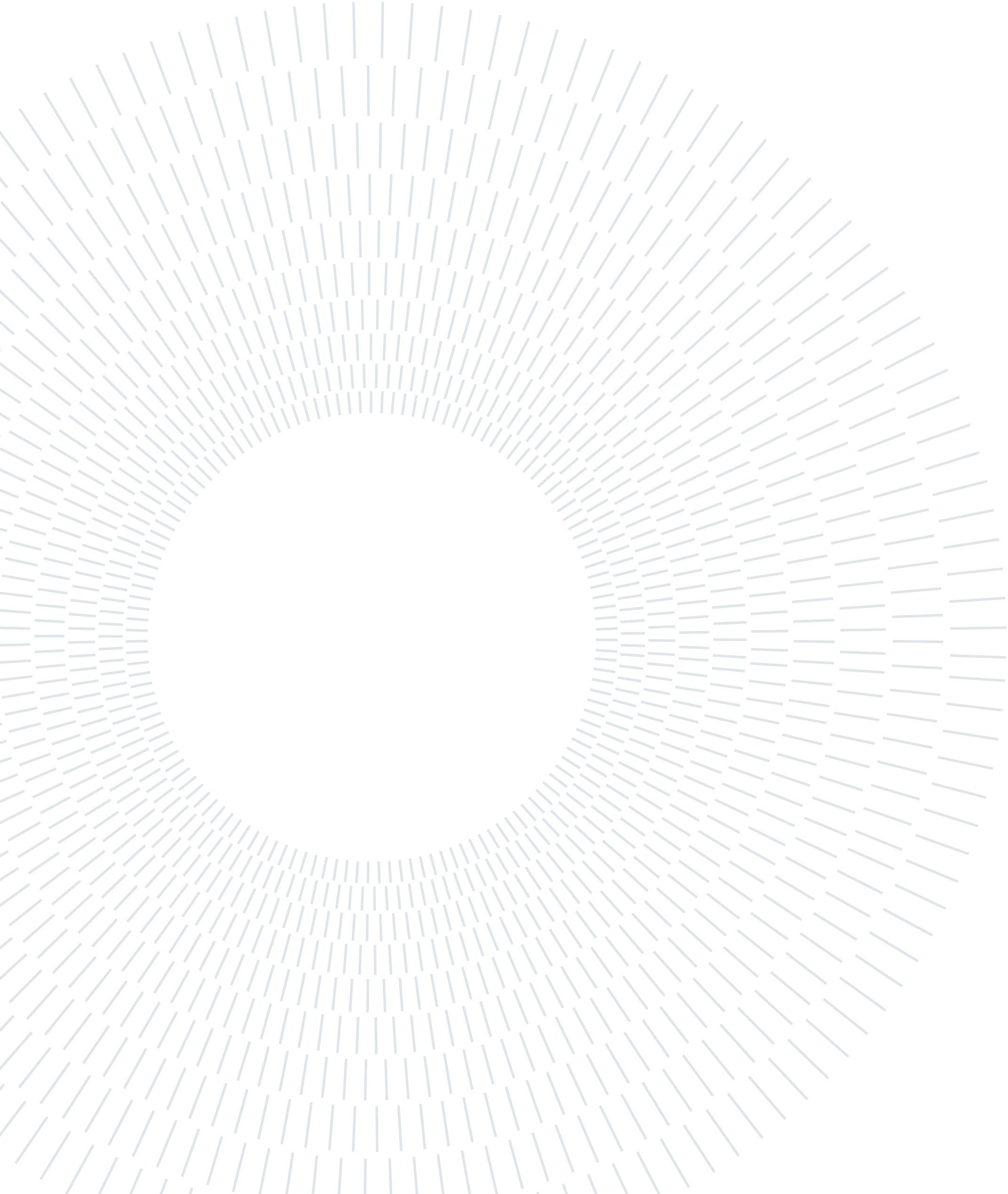
Student ID: 996809
Advisor: Prof. Michele Correggi
Co-advisors: Dr. Massimo Moscolari
Academic Year: 2023-24



Abstract

This thesis provides a detailed description of the toric code model, which plays an important role in the field of quantum computation as a paradigmatic example of quantum error correction code. After a preliminary introductory section on spin observables in Quantum Mechanics, the first chapter of the thesis contains the description of the model of the toric code Hamiltonian followed by the computation of its ground states and excited states, which are classified in terms of their topological properties. In particular, we will show that the excitations are represented by two species of quasi-particles obeying to mutual fractional statisctics. After a brief overview of classical and quantum error correction, the analysis of the anyonic excitations of the toric code done in the first chapter is then used to explicitly describe the strategy behind quantum error correction for logical qubits encoded in the ground states of the toric code Hamiltonian.

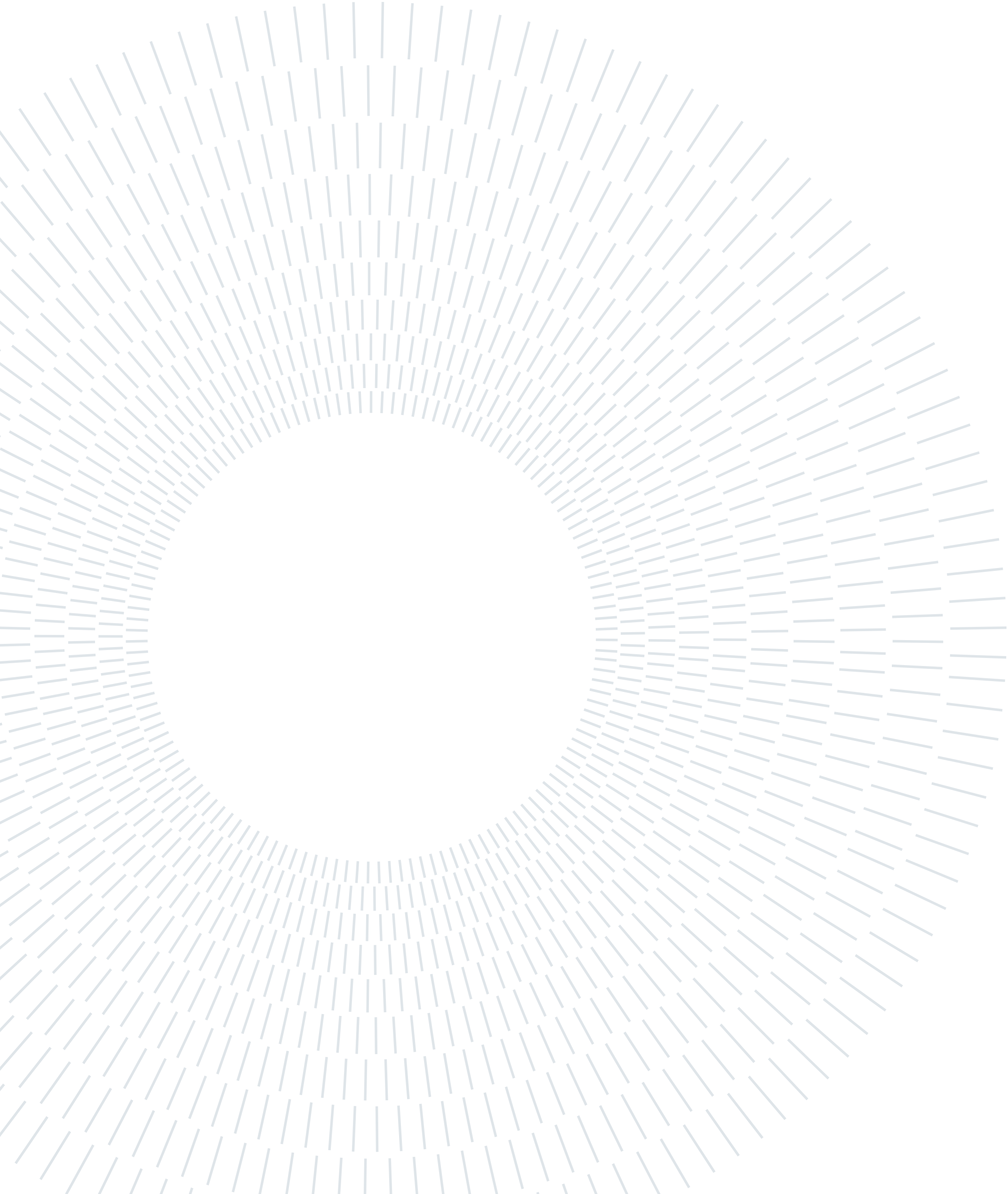
Keywords: Quantum computing, Spin lattices, Anyons, Quantum error correction.



Abstract in lingua italiana

Questa tesi fornisce una descrizione dettagliata del modello del toric code. Questo modello è importante nel campo della computazione quantistica in quanto rappresenta l'archetipo di codice topologico per la correzione di errori quantistici. Dopo una sezione introduttiva preliminare sulle osservabili di spin in Meccanica Quantistica, il primo capitolo della tesi contiene la descrizione del modello e della sua Hamiltoniana seguito dal calcolo dei suoi stati fondamentali e stati eccitati, i quali vengono classificati in base alle loro proprietà topologiche. In particolare, mostreremo che le eccitazioni sono rappresentate da due tipi di quasi-particelle che obbediscono ad una statistica frazionaria reciproca. Dopo una breve panoramica sulla correzione degli errori classica e quantistica, l'analisi delle eccitazioni anioniche del toric code svolta nel primo capitolo viene utilizzata per descrivere esplicitamente la strategia dietro la correzione degli errori quantistici per qubit logici codificati negli stati fondamentali dell'Hamiltoniana del toric code.

Parole chiave: Computazione quantistica, Reticoli di spin, Correzione quantistica degli errori.



Contents

Abstract i

Abstract in lingua italiana iii

Contents v

Introduction 1

1 Toric Code 3

1.1 Spin Observables in Quantum Mechanics . . . 3

1.1.1 Self-adjointness and unitarity . . . 3

1.1.2 The spectrum . . . 4

1.1.3 Commutation and anticommutation . . . 4

1.1.4 Spin-$\frac{1}{2}$ particles example . . . 5

1.1.5 Tensor product of Hilbert spaces . . . 5

1.2 Description of the Model . . . 7

1.3 Ground States . . . 16

1.4 Excited States . . . 24

1.4.1 Fermionic and bosonic exchange statistics in 3D and 2D . . . 27

1.5 Anyonic Excitations . . . 33

2 Coding on the Toric Code 35

2.1 Classical Error Correction . . . 35

2.2 Quantum Error Correction (QEC) . . . 40

2.2.1 Fault-tolerant computation . . . 43

2.3 QEC in the Toric Code . . . 45

2.3.1 Error detection and correction . . . 47

Bibliography **55**

List of Figures **57**

Acknowledgements **61**

Introduction

Being able to leverage quantum phenomena to perform computations has brought significant advancements to the field of quantum computing. Quantum systems possess unique properties promising exponential speedup over classical algorithms for specific tasks [Nie06]. However, realizing such potential requires overcoming significant challenges, such as the exposure to noise of a physical system. As a consequence, quantum error correction has emerged as an important area of research. Traditional error correction techniques developed for classical computers cannot be directly applied to quantum systems due to the intrinsic fragility of quantum information [Cha06]. Therefore, the main goal of quantum error correction is to mitigate errors by encoding quantum information into carefully designed quantum states. Several quantum computing models have been studied theoretically in order to assess this issue and give access to their practical realizations. Among various quantum error correcting codes, topological codes are of particular interest due to their robustness against local perturbations and their ability to encode information in a non-local manner [Kit02].

The toric code was one of the first topological quantum error correcting codes that has been proposed. It was introduced by Alexei Kitaev in 1997 as a two-dimensional lattice model, which is particularly notable for its robustness against local errors thanks to its topological properties. In fact, what makes the toric code special is the way it exploits the topological features of a quantum system to robustly encode information and protect it from the damaging effects of noise. In particular, not only does the toric code exhibit topological order, which manifests itself through the degeneracy of the ground state manifold and the occurrence of excitations given by quasi-particles obeying to exotic mutual statistics (anyons).

In this thesis, we initially give a brief mathematical description of spin observables in Quantum Mechanics by studying their mathematical properties. These properties are later used to give a precise definition of the toric code model along with the stabilizer operators encoded in its Hamiltonian. Such operators, called vertex and plaquette op-

erators, play a crucial role in the characterization of the ground states of the system, which are the states with the lowest possible energy. In particular, vertex and plaquette operators are simultaneously diagonalizable. This allows to explicitly compute the ground states, which form a manifold of dimension 4 [Her20]. Such a degeneracy is in fact due to the topological features of the toric code model, which are key in order to encode quantum information in the ground states of the system by providing redundancy. Such topological features of the toric code have a two-fold relevance: on one hand, a quantum computation on it would be protected from local perturbations, and, on the other hand, there would be a natural correction algorithm to implement [Zbi19]. After the discussion of the ground states of the Hamiltonian, we study excited states that can be obtained through the application of string operators. In fact, they contain pairwise excitations: *magnetic vorteces* and *electric charges*. Ultimately, we focus on giving a brief description of identical particles in Quantum Mechanics including anyons in two dimensional systems and explain how mutual anyonic statistics arise in the toric code model. Unlike conventional particles, anyons exhibit nontrivial phase changes when exchanged. By leveraging the braiding of anyons, we can encode quantum information in the ground states of the Hamiltonian and perform fault-tolerant operations [Rao01, Rao16].
In the second part of the thesis, the first sections are dedicated to give an overview of classical and quantum error models. In doing so, we focus on error detection and correction techniques both in the classical and quantum setting. We ultimately aim to build an analogy with the quantum error correction on the toric code. In fact, the last section focuses precisely on explaining why the anyonic excitations derived in the previous section are so important to be able to encode quantum qubits in the ground states of the Hamiltonian and why this is mathematically possible [Zbi19].

In conclusion, this thesis provides a comprehensive description of the toric code together with its implications to quantum error correction in topological quantum computation. Through the careful study of the theoretical features of the toric code model, this work investigates some practical applications of quantum information processing and error correction methodologies as well as the possibility of effectively perform fault-tolerant quantum computation.

1 | Toric Code

1.1. Spin Observables in Quantum Mechanics

In the context of quantum mechanics, an observable is a quantity that can be observed in a quantum system. Some examples could be: position, momentum or energy of a system. The latter is represented by the Hamiltonian, which not only describes the energy but also encodes the rules that govern the dynamic of the system. Formally, an observable is represented by an Hermitian operator over an Hilbert space $\mathscr{H}$.

Definition 1.1. *(Hilbert space) An Hilbert space* $\mathscr{H}$ *over* $\mathbb{C}$ *is a complex vector space on* $\mathbb{C}$ *with a scalar product* $\langle\cdot|\cdot\rangle\colon \mathscr{H}\times\mathscr{H}\to\mathbb{C}$ *and which is complete with respect to the norm induced by the scalar product.*

In particular, a 'spin' observable is a physical quantity associated with the intrinsic angular momentum of elementary particles, such as electrons. The spin is a fundamental property of particles and is characterized by a quantum number typically denoted by S, representing the spin quantum number. In quantum mechanics, spin observables represent the possible measurements that can be made on the spin of a particle. Moreover, the spin is described using spin operators, which are, as anticipated above, hermitian matrices representing the components of the angular momentum of the spin along different directions.

1.1.1. Self-adjointness and unitarity

We denote as $\mathscr{B}(\mathscr{H})$ the space of bounded linear operators on $\mathscr{H}$. The self-adjointness of an operator can be stated as follows:

Definition 1.2. *(Hermitian operator) Let* $\mathscr{H}$ *be an Hilbert space and* $A\colon \mathscr{H}\to\mathscr{H}$ *be a bounded linear operator. An operator* A *is called Hermitian if for all* $\psi,\phi\in\mathscr{H}$ *we have that*

$$\langle A\psi|\phi\rangle = \langle\psi|A^\dagger\phi\rangle = \langle\phi|A\psi\rangle^*.$$

The symbol † denotes the Hilbert adjoint, which in the case of a finite-dimensional Hilbert space coincides with the adjoint for matrices, i.e. for $\mathscr{H}$, finite-dimensional, and any $A \in \mathscr{B}(\mathscr{H})$, $A^\dagger$ is such that $(A^\dagger)_{ij} = A^*_{ij}$. A self-adjoint square matrix A of complex numbers, representing a linear operator, is also called hermitian. One important consequence of self-adjointness is that it ensures that the spectrum of the operator is real. Notice also that, for real matrices, it is sufficient to compute only the transpose of the matrix to verify hermiticity.

In order to understand the dynamics of observables, we introduce another important property: unitarity. As previously said, observables such as position, momentum and energy are represented by hermitian operators, which ensure the existence of real eigenvalues and orthogonal eigenvectors. Instead, unitarity governs the evolution in time of quantum states and operators. In particular, it ensures that the total probability of all possible outcomes of a measurement remains conserved over time, preserving the fundamental principles of quantum mechanics.

Definition 1.3. *(Unitary operator) Let $\mathscr{H}$ be an Hilbert space and let $U \in \mathscr{B}(\mathscr{H})$. We say that U is a unitary operator if for all $\psi, \phi \in \mathscr{H}$ we have that*

$$\langle U\psi | U\phi \rangle = \langle \psi | \phi \rangle.$$

In terms of the conjugate transpose, we can define a complex matrix U to be unitary if $(U^*)^T = U^{-1}$. We can also define that a real matrix U, representing a linear operator, is unitary if $(U)^T = U^{-1}$ or, equivalently, if $(U)^T U = I$.

Overall, self-adjointness and unitarity provide a comprehensive framework for describing the behavior of observables in quantum mechanics.

1.1.2. The spectrum

The spectrum of an observable refers to the set of possible values that can be obtained when measuring the physical quantity represented by the observable.

Definition 1.4. *(Spectrum) Given an operator $A \in \mathscr{B}(\mathscr{H})$, the spectrum $\sigma(A)$ is the set of $z \in \mathbb{C}$ such that (A - z) is not invertible.*

1.1.3. Commutation and anticommutation

Given two observables represented by bounded self-adjoint operators A and B, we define the commutator as follows

$$[A, B] = AB - BA.$$

and the anticommutator as

$$\{A, B\} = AB + BA.$$

The two observables commute if

$$AB = BA$$

while, they anticommute if

$$AB = -BA.$$

1.1.4. Spin-$\frac{1}{2}$ particles example

In order to exemplify the above properties of the observables, we consider spin-$\frac{1}{2}$ particles as discussed in [Cor23], i.e. electrons, and define the components of the spin in three dimensions. In this case the Hilbert space is bidimensional and the observables can be written in terms of Pauli matrices in the basis $\{|\frac{1}{2}, \pm\frac{1}{2}\rangle\}$ (spin-up and spin-down) for $\hbar = 1$:

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} .$$

We are also going to consider $\sigma_1 = \sigma_x$, $\sigma_2 = \sigma_y$ and $\sigma_3 = \sigma_z$. Pauli matrices are self-adjoint and unitary. Moreover they satisfy some specific commuation and anticommutation properties.

$$[\sigma_i, \sigma_j] = \sigma_i\sigma_j - \sigma_j\sigma_i = 2i\epsilon_{ijk}\sigma_k$$

where ϵ_{ijk} is the Levi-Civita symbol, and

$$\{\sigma_i, \sigma_j\} = \sigma_i\sigma_j + \sigma_j\sigma_i = 2\delta_{ij}\mathbb{I}$$

where δ_{ij} is the Kronecker delta and $\mathbb{I}$ is the identity matrix.

1.1.5. Tensor product of Hilbert spaces

Given two observables A and B acting on a bidimensional Hilbert space, their tensor product $A \otimes B$ over $\mathbb{C}^2 \otimes \mathbb{C}^2$ can be expressed on the corresponding basis: combining the two basis elements of each copy of $\mathbb{C}^2$ we obtain four combinations:

$$(e_1 \otimes e'_1), (e_1 \otimes e'_2), (e_2 \otimes e'_1), (e_2 \otimes e'_2)$$

which correspond to:

$$\begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \quad .$$

Then, we can then write:

$$u = \sum_{j=1}^{2} \sum_{i=1}^{2} c_{ij}(e_i \otimes e'_j) = \sum_{j=1}^{4} d_j v_j$$

where c_{ij} is a complex coefficient in $\mathbb{C}$. Here u represents a general state in the tensor product space $\mathbb{C}^2 \otimes \mathbb{C}^2$ in terms of the linear combination of basis vectors v_j.

Considering that A acts only an the first Hilbert space $A \otimes \mathbb{I}$ and that B acts only on the second $\mathbb{I} \otimes B$, we can rename:

$$A \otimes \mathbb{I} = A_1$$
$$\mathbb{I} \otimes B = B_2$$

and write the tensor product as a simple product of A_1 and B_2:

$$A \otimes B = A_1 B_2.$$

This is because the operators acts locally, so we can ignore the identity.
Similarly if we compute the scalar product of two factorized states explicitly, it becomes a simple product of the scalar products between the components of the basis:

$$(u_1 \otimes w_1) \cdot (u_2 \otimes w_2) = \langle u_1 | w_1 \rangle \langle u_2 | w_2 \rangle.$$

1.2. Description of the Model

The toric code model is defined on a square lattice with periodic boundary conditions in both directions. These latter characteristics are typical of what is known as a torus topology or simply a torus, after which the model takes name. Our lattice $\mathcal{L}$, is a subset of $\mathbb{Z}^2$, such that each lattice point is identified with a pair of integers. The periodic boundary conditions are imposed by specifying that for any site (i, j) in the lattice, the neighboring sites are going to be: $((i+1)\text{mod}N, j)$, $((i-1)\text{mod}N, j)$, $(i, (j+1)\text{mod}N)$ and $(i, (j-1)\text{mod}N)$, where N is the side-length of the torus. We are also going to consider a dual lattice, which will be denoted as $\mathcal{L}'$ and which is given by

$$\mathcal{L}' = \{(i \text{ mod}N + \frac{1}{2}), (j \text{ mod}N + \frac{1}{2})\}.$$

Each unitary step on the lattice is identified with an edge; given the vertices or lattice points $\vec{v} = (i, j) \in V$ where V coincides with $\mathcal{L}$ and the cells $\vec{p} = (i + \frac{1}{2}, j + \frac{1}{2}) \in P$ where P coincides with $\mathcal{L}'$ of the lattice as shown in figure 1.1, we can define the set of all edges for the lattice:

Definition 1.5. *(Edges of $\mathcal{L}$) We define the set of all edges of the lattice as:* $E = \{e = (\vec{v_1}, \vec{v_2}) \text{ s.t. } \vec{v_2} = \vec{v_1} + \hat{e}_1 \text{ or } \vec{v_2} = \vec{v_1} + \hat{e}_2\}$.

Notice that we can identify each edge $e = ((i, j), (i', j')) \in E$ with the vector that points at the middle point of the edge, namely with the vector of coordinates $\vec{e} = (i + (i' - i)/2, j + (j' - j)/2)$. In the rest of the thesis we will make use of this identification without explicit mention. We can also define the set of edges of the dual lattice as:

Definition 1.6. *(Edges of $\mathcal{L}'$) We define the set of all edges of the dual lattice $\mathcal{L}'$ as:* $E' = \{e' = (\vec{p_1}, \vec{p_2}) \text{ s.t. } \vec{p_2} = \vec{p_1} + \hat{e}_1 \text{ or } \vec{p_2} = \vec{p_1} + \hat{e}_2\}$.

On each edge is located a spin-$\frac{1}{2}$ particle each belonging to a bidimensional Hilbert space $\mathbb{C}^2$. As a consequence we need to define a global Hilbert space for the lattice that takes into account every edge:

Definition 1.7. *(Space of states) The Hilbert space of the system is:* $\mathscr{H} = \bigotimes_{e \in E} \mathbb{C}^2$.

For each cell in $\mathcal{L}$ we are going to consider two spins, therefore the total number of spins is $2N^2$, where N^2 represents the total number of cells in P and vertices in V and N the side length of the lattice. Notice that, all these spins are identical particles, which will play an important role in the next sections to study the behaviour of the system.

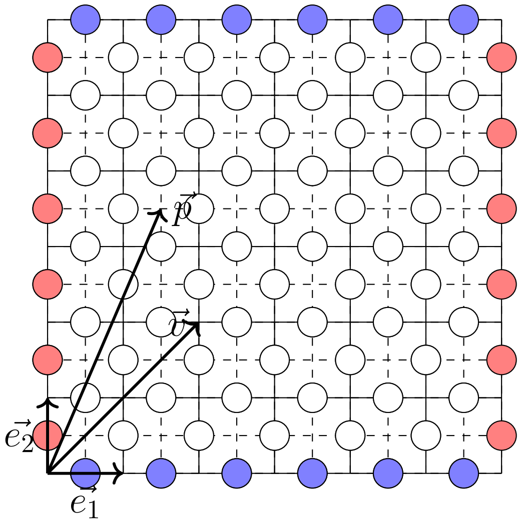


Figure 1.1: Square lattice with boundary conditions. The spins are represented with an empty circle while circles shaded in blue/red represent the boundary conditions. $\hat{e}_1$ and $\hat{e}_2$ are unitary vectors identifying the coordinate system used to identify the sites of the lattice through vectors $\vec{v}$ and $\vec{p}$ with application point in the origin of the coordinate system. In the following sections we are going to indicate the boundary conditions in gray.

The key part of the toric code are the so called *vertex* and *plaquette* operators that are going to be placed, respectively, on the vertices and cells of $\mathcal{L}$. Such operators can be defined formally as in definition 1.10 and implemented by means of Pauli matrices as exemplified in figure 1.2. In order to give definition 1.10 we define the following:

Definition 1.8. *(Edge operator) We define a generic operator* A *acting on an edge* e *of* $\mathcal{L}$ *as:* $A_e{=}A_{(i+\frac{i'-i}{2},j+\frac{j'-j}{2})}$ *where the tuples* $(i,j),(i',j')$ *uniquely identify an edge* $e \in E$*.*

Definition 1.9. *(Groups of neighbouring spins of* $\mathcal{L}$*) We define the quartets of spins sorrounding a vertex* $\vec{v}$ *and a plaquette* $\vec{p}$ *of* $\mathcal{L}$*, respectively, as the following groups of spins:* $star(\vec{v}) = \{\vec{v} \pm \frac{1}{2}\hat{e}_1, \vec{v} \pm \frac{1}{2}\hat{e}_2\}$ *and* $bdy(\vec{p}) = \{\vec{p} \pm \frac{1}{2}\hat{e}_1, \vec{p} \pm \frac{1}{2}\hat{e}_2\}$*.*

Then we can define vertex and plaquette operators as:

Definition 1.10. *(Vertex and Plaquette operators) Given a vertex* $\vec{v}$ *and a plaquette* $\vec{p}$ *we can define, respectively, vertex and plaquette operators as products over Pauli operators, each acting locally on four neighbouring spins of* $\mathcal{L}$

$$A_{\vec{v}} = \prod\nolimits_{e \in star(\vec{v})} Z_e$$

$$B_{\vec{p}} = \prod\nolimits_{e \in bdy(\vec{p})} X_e.$$

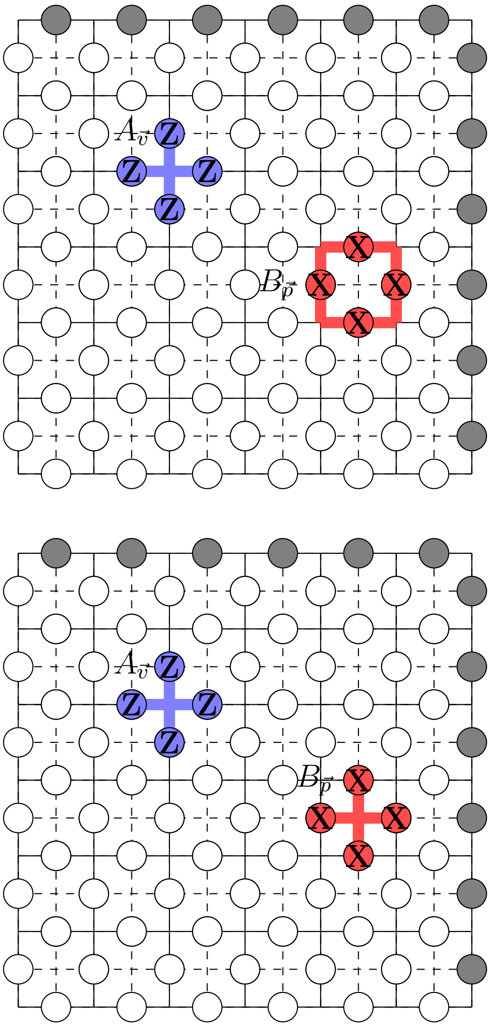

$A_{\vec{v}}$
Z
Z
Z
Z
$B_{\vec{p}}$
X
X
X
X
$A_{\vec{v}}$
Z
Z
Z
Z
$B_{\vec{p}}$
X
X
X
X

Figure 1.2: In the first picture: vertex (blue) and plaquette (red) operators applied in a vertex and cell of $\mathcal{L}$ respectively. In the second picture: vertex (blue) and plaquette (red) operators applied in a vertex of $\mathcal{L}$ and a vertex of $\mathcal{L}'$, respectively.

Where we have defined each edge operator Z_e and X_e to be equal to the following tensor proucts:

$$Z_e = \mathbb{I}_{E\setminus e} \otimes \sigma_e^z$$

$$X_e = \mathbb{I}_{E\setminus e} \otimes \sigma_e^x \ .$$

For the sake of simplicity, we will usually omit the tensor product because there is no risk of ambiguity in the interpretation. In fact, since the vertex and plaquette operators act locally, we can omit writing down the tensor product with the identity for sites that are not part of the support of the operator.

All the elements that constitute the model of the toric code are brought together by its Hamiltonian. definition 1.2 describes the Hamiltonian of the toric code according to [Kit02]. For recent reviews on the subject see [Her20].

Definition 1.11. *(Toric Code Hamiltonian) Given vertex and plaquette operators, we can define the following Hamiltonian for the system:*

$$H = -\sum_{\vec{v}\in V} A_{\vec{v}} - \sum_{\vec{p}\in P} B_{\vec{p}}$$

Each one of the $A_{\vec{v}}$ and $B_{\vec{p}}$ operators that appear in the Hamiltonian share some important properties. Firstly, notice that $A_{\vec{v}}$ and $B_{\vec{p}}$ operators are Hermitian and square to the identity. Given the considerations made in definition 1.2 and 1.3 for bounded operators, we can easily prove the first following proposition:

Proposition 1.1. *(Properties of $A_{\vec{v}}$ and $B_{\vec{p}}$ operators) The operators $A_{\vec{v}}$ and $B_{\vec{p}}$ are involutions, i.e. $A_{\vec{v}}^2 = B_{\vec{p}}^2 = \mathbb{I}$.*

Proof. We know that the operator $A_{\vec{v}} = \prod_{star(\vec{v})} Z_e$ and that $B_{\vec{p}} = \prod_{bdy(\vec{p})} X_e$.
Firstly, recall the form of the σ_x and σ_z matrices representing the Pauli gates X and Z:

$$\mathrm{Z} = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

$$\mathrm{X} = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

It can be proved that they are Hermitian:

$$(\sigma_z)^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}^T = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \sigma_z,$$

$$(\sigma_x)^\dagger = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^\dagger = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^T = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \sigma_x.$$

since Pauli matrices are real matrices and thus $A^\dagger = A^T$. Hence, by the properties of the tensor product of Hilbert spaces,

$$(A_{\vec{v}})^\dagger = (\sigma_{x_1}\sigma_{x_2}\sigma_{x_3}\sigma_{x_4})^\dagger = \sigma_{x_1}^\dagger\sigma_{x_2}^\dagger\sigma_{x_3}^\dagger\sigma_{x_4}^\dagger = \sigma_{x_1}\sigma_{x_2}\sigma_{x_3}\sigma_{x_4} = A_{\vec{v}}$$

$$(B_{\vec{p}})^\dagger = (\sigma_{z_1}\sigma_{z_2}\sigma_{z_3}\sigma_{z_4})^\dagger = \sigma_{z_1}^\dagger\sigma_{z_2}^\dagger\sigma_{z_3}^\dagger\sigma_{z_4}^\dagger = \sigma_{z_1}\sigma_{z_2}\sigma_{z_3}\sigma_{z_4} = B_{\vec{p}}.$$

Moreover, since the square of all Pauli matrices is the identity:

$$(\sigma_z)^2 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \mathbb{I}$$

$$(\sigma_x)^2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \mathbb{I}$$

Then, again $A_{\vec{v}}$ and $B_{\vec{p}}$ are involutions

$$(A_{\vec{v}})^2 = (\sigma_{x_1}\sigma_{x_2}\sigma_{x_3}\sigma_{x_4})^2 = \sigma_{x_1}^2\sigma_{x_2}^2\sigma_{x_3}^2\sigma_{x_4}^2 = \mathbb{I}$$

$$(B_{\vec{p}})^2 = (\sigma_{z_1}\sigma_{z_2}\sigma_{z_3}\sigma_{z_4})^2 = \sigma_{z_1}^2\sigma_{z_2}^2\sigma_{z_3}^2\sigma_{z_4}^2 = \mathbb{I}.$$

□

Given proposition 1.1, we can now further characterize the spectrum of the operators. First we prove the property of involution.

Proposition 1.2. *(Eigenvalues of $A_{\vec{v}}$ and $B_{\vec{p}}$ operators) Given Hermitian operators $A_{\vec{v}}$ and $B_{\vec{p}}$, which satisfy the involution, their possible eigenvalues are ± 1.*

Proof.
Let us start by writing the expression for the eigenvalues $A_{\vec{v}}|\xi\rangle = \lambda|\xi\rangle$ and take as hypothesis that $|\xi\rangle \neq 0$. Then, by means of the scalar product:

$$\langle\xi|A_{\vec{v}}|\xi\rangle = \lambda\langle\xi|\xi\rangle,$$

$$\lambda = \frac{\langle\xi|A_{\vec{v}}\xi\rangle}{\langle\xi|\xi\rangle} = \frac{(\langle A_{\vec{v}}\xi|\xi\rangle)^*}{||\xi||^2} = \frac{\langle\xi|A_{\vec{v}}^{\dagger}\xi\rangle}{||\xi||^2} = \lambda^*$$

Notice that we have applied antilinearity of the adjoint in the penultimate equality:

$$(\langle\xi|A_{\vec{v}}^{\dagger}|\xi\rangle)^{\dagger} = |\xi\rangle^{\dagger}(A_{\vec{v}}^{\dagger})^{\dagger}\langle\xi|^{\dagger} = \langle\xi|A_{\vec{v}}|\xi\rangle^*.$$

Therefore, we have derived that the eigenvalues of $A_{\vec{v}}$ are $\lambda = \lambda^*$, which implies that they must be real.

Then we exploit unitarity while taking as hypothesis that $|\xi\rangle \neq 0$. We write the expression $A_{\vec{v}}|\xi\rangle = \lambda|\xi\rangle$ and its self-adjoint $\langle\xi|A_{\vec{v}}^{\dagger} = \lambda^*\langle\xi|$. Recall that $A_{\vec{v}}^{\dagger} = A_{\vec{v}}^{-1}$, thus by means of the scalar product we obtain

$$\langle\xi|A_{\vec{v}}A_{\vec{v}}^{\dagger}|\xi\rangle = \lambda\lambda^*\langle\xi|\xi\rangle$$

$$\langle\xi|I|\xi\rangle = \lambda\lambda^*\langle\xi|\xi\rangle$$

$$\langle\xi|\xi\rangle = \lambda\lambda^*\langle\xi|\xi\rangle$$

$$\lambda\lambda^* = 1$$

$$|\lambda|^2 = 1$$

Which means that the eigenvalues of $A_{\vec{v}}$ have modulus equal to one, as expected. Though, this can happen for two real choices of λ which are $+1$ and -1. Therefore, putting together hermiticity and unitarity,

$$\begin{cases} \lambda = \lambda^* \\ |\lambda|^2 = 1 \end{cases}$$

we obtain that the only two remaining possibilities for the eigenvalues of $A_{\vec{v}}$ are ± 1.

The same reasoning can be carried out for the plaquette operators.

□

Given the result of proposition 1.2, is easy to find the spectrum of the operators:

Proposition 1.3. *(Spectrum of $A_{\vec{v}}$ and $B_{\vec{p}}$ operators) The spectrum of $A_{\vec{v}}$ and $B_{\vec{p}}$ operators is equal to* $\{-1, +1\}$.

Proof.
If we choose the eigenvalues of the σ^z and σ^x making up, respectively, the vertex and plaquette operator, to be positive and equal to $+1$ we obtain the following eigenvalues assosciated with the vertex and plaquette operator:

$$A_{\vec{v}}|\psi\rangle = |\psi\rangle$$
$$B_{\vec{p}}|\psi\rangle = |\psi\rangle.$$

On the other hand, if we choose an odd numer of eigenvalues, associated to the σ^z and σ^x, to be equal to -1 we will obtain the following:

$$A_{\vec{v}}|\psi\rangle = -|\psi\rangle$$
$$B_{\vec{p}}|\psi\rangle = -|\psi\rangle$$

Overall, if we compute the corresponding eigenvalue of $A_{\vec{v}}$ (or $B_{\vec{p}}$), this can either assume value -1 or $+1$ according to the fact that the Pauli matrices that make up the operator anticommute, respectively, with an odd or even number of edges.

☐

Given proposition 1.3, we can in fact prove that the following relationships between vertex and plaquette operators holds:

Proposition 1.4. *(Commutation of $A_{\vec{v}}$ and $B_{\vec{p}}$ operators) The commutator between two vertex or plaquette operators is always zero:*

$$[A_{\vec{v}}, A_{\vec{v}}] = 0$$

$$[B_{\vec{p}}, B_{\vec{p}}] = 0$$

$$[A_{\vec{v}}, B_{\vec{p}}] = 0$$

Proof.
Fix the origin of the coordinate system in the bottom left corner of the lattice as shown in figure 1.3.

Then, we can define the two vectors representing the site of application of the vertex and plaquette operator, respectively over the lattice $\mathcal{L}$ and dual lattice $\mathcal{L}'$

$$\vec{v} = n\hat{e_1} + m\hat{e_2}, \text{ where } n, m \in \mathbb{Z}$$

$$\vec{p} = (n + \tfrac{1}{2})\hat{e_1} + (m + \tfrac{1}{2})\hat{e_2}, \text{ where } n, m \in \mathbb{Z}.$$

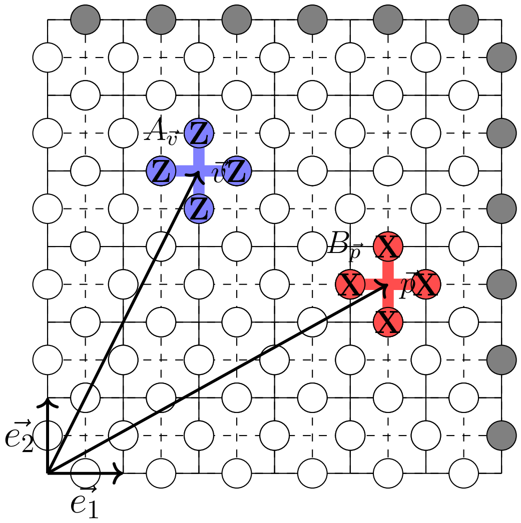


Figure 1.3: Vertex and plaquette operators according to the coordinate system with origin in (0,0). Here the plaquette operator is implemented on $\mathcal{L}'$ in order to coherently define its edges from the point of view of $\mathcal{L}$ using vector $\vec{v}$.

Rewrite the operators as follows:

$$A_{\vec{v}} = \sigma^z_{\vec{v}+\frac{1}{2}\hat{e_1}} \sigma^z_{\vec{v}+\frac{1}{2}\hat{e_2}} \sigma^z_{\vec{v}-\frac{1}{2}\hat{e_1}} \sigma^z_{\vec{v}-\frac{1}{2}\hat{e_2}}$$

$$B_{\vec{p}} = \sigma^x_{\vec{p}+\frac{1}{2}\hat{e_1}} \sigma^x_{\vec{p}+\frac{1}{2}\hat{e_2}} \sigma^x_{\vec{p}-\frac{1}{2}\hat{e_1}} \sigma^x_{\vec{p}-\frac{1}{2}\hat{e_2}}.$$

In order to simplify the calculations we rewrite $B_{\vec{p}}$ on the lattice $\mathcal{L}$ by rewriting the indeces in terms of vector $\vec{v}$

$$(n\hat{e_1} + m\hat{e_2}) + \tfrac{1}{2}\hat{e_2} = n\hat{e_1} + (m + \tfrac{1}{2}\hat{e_2})$$

$$(n\hat{e_1} + m\hat{e_2}) + \tfrac{1}{2}\hat{e_1} = (n + \tfrac{1}{2})\hat{e_1} + m\hat{e_2}$$

$$(n\hat{e_1} + m\hat{e_2}) + \tfrac{1}{2}\hat{e_1} + \hat{e_2} = (n + \tfrac{1}{2})\hat{e_1} + (m+1)\hat{e_2}$$

$$(n\hat{e_1} + m\hat{e_2}) + \tfrac{1}{2}\hat{e_2} + \hat{e_1} = (n+1)\hat{e_1} + (m + \tfrac{1}{2})\hat{e_2}.$$

Then the $B_{\vec{p}}$ operator becomes:

$$B_{\vec{p}} = \sigma^x_{n\hat{e_1}+(m+\frac{1}{2}\hat{e_2})} \sigma^x_{(n+\frac{1}{2})\hat{e_1}+m\hat{e_2}} \sigma^x_{(n+\frac{1}{2})\hat{e_1}+(m+1)\hat{e_2}} \sigma^x_{(n+1)\hat{e_1}+(m+\frac{1}{2})\hat{e_2}}.$$

Now calcuate the commutator $[A_{\vec{v}}, B_{\vec{p}}] = A_{\vec{v}}B_{\vec{p}} - B_pA_v$ by focusing on the first term:

$$A_{\vec{v}}B_{\vec{p}} = \sigma^z_{(n+\frac{1}{2})\hat{e_1}+m\hat{e_2}}\sigma^z_{n\hat{e_1}+(m+\frac{1}{2})\hat{e_2}}\sigma^z_{(n-\frac{1}{2})\hat{e_1}+m\hat{e_2}}\sigma^z_{n\hat{e_1}+(m-\frac{1}{2})\hat{e_2}} \cdot$$
$$\sigma^x_{n\hat{e_1}+(m+\frac{1}{2}\hat{e_2})}\sigma^x_{(n+\frac{1}{2})\hat{e_1}+m\hat{e_2}}\sigma^x_{(n+\frac{1}{2})\hat{e_1}+(m+1)\hat{e_2}}\sigma^x_{(n+1)\hat{e_1}+(m+\frac{1}{2})\hat{e_2}}.$$

Matrices do not always commute but for Pauli matrices we have the following commutation relationship :

$$\sigma^x_{\vec{v}}\sigma^z_{\vec{v}'} = \sigma^z_{\vec{v}'}\sigma^x_{\vec{v}} + 2\sigma^x_{\vec{v}}\sigma^z_{\vec{v}'}\delta_{\vec{v}\vec{v}'}$$

where $\delta_{\vec{v}\vec{v}'} = \begin{cases} 1, & if \quad \vec{v} = \vec{v}' \\ 0, & if \quad \vec{v} \neq \vec{v}' \end{cases}$

which states that $\sigma^x_{\vec{v}}\sigma^z_{\vec{v}'}$ commutes for $\vec{v} \neq \vec{v}'$ but anticommutes for $\vec{v} = \vec{v}'$. This is follows from the anticommutation relationship of Pauli matrices $\sigma^x\sigma^z = -\sigma^x\sigma^z$. Thus, for an even numer of overlapping edges the commutator becomes:

$$[A_{\vec{v}}, B_{\vec{p}}] = 2\sigma^x_{n\hat{e_1}+(m+\frac{1}{2}\hat{e_2})}\sigma^x_{(n+\frac{1}{2})\hat{e_1}+m\hat{e_2}}\sigma^x_{(n+\frac{1}{2})\hat{e_1}+(m+1)\hat{e_2}}\sigma^x_{(n+1)\hat{e_1}+(m+\frac{1}{2})\hat{e_2}} \cdot$$
$$\sigma^z_{(n+\frac{1}{2})\hat{e_1}+m\hat{e_2}}\sigma^z_{n\hat{e_1}+(m+\frac{1}{2})\hat{e_2}}\sigma^z_{(n-\frac{1}{2})\hat{e_1}+m\hat{e_2}}\sigma^z_{n\hat{e_1}+(m-\frac{1}{2})\hat{e_2}}.$$

Instead, for an odd number of edges the commutator vanishes: $[A_{\vec{v}}, B_{\vec{p}}] = 0$.

In the same way, we notice that two vertex/plaquette operators applied on different sites always commute; this is because they either do not share any edge or they share only one.

□

Now that we have described the toric code model and its operators, we proceed to compute its ground state, which is at the heart of its fault-tolerant behaviour. The ground state arises from the interaction of vertex and plaquette operators acting on the system's lattice. Through the characterization of these operators and the understanding of their role in enforcing some additional constraints, we gain insight into the structure of the ground state manifold.

In the following section, we derive the unique features of the ground state, including its topological order.

1.3. Ground States

The ground states of the toric code represent the system's lowest energy configurations. What makes these ground state particularly intriguing is its highly degenerate nature, meaning that there are numerous distinct configurations that all share the same minimum energy. The ground states exhibit topological order, characterized by non-local correlations between spins [Kit02]. This means that the model is robust even when the system is subject to local perturbations. In fact, quantum entanglement between spins is not confined to neighboring pairs, instead, it extends over long distances and involves collective behaviours of spins that cannot be understood by considering them individually. Not only, as we will see, these features are intimately related to the underlying topology of the torus. This topological protection ensures also the robustness of quantum information against local perturbations, giving the toric code the potential to be feasibly applied in quantum information processing (see Chapter 3).

We start by explicitly defining the ground state by making use of the properties of the vertex and plaquette operators that we have derived in the previous section.

Proposition 1.5. *(Ground states) The ground states of the toric code Hamiltonian are the simultaneous* $+1$ *eigenstates of all the* $A_{\vec{v}}$ *and* $B_{\vec{p}}$ *operators.*

Proof.
Recall the form of the Hamiltonian as given in section 1.3:

$$H = -\sum_{\vec{v}\in V} A_{\vec{v}} - \sum_{\vec{p}\in P} B_{\vec{p}}$$

we already know that commuting hermitian matrices are simultaneously diagonalizable, moreover $A_{\vec{v}}$ and $B_{\vec{p}}$ operators commute for an even number of edges, thus there exists a common base of eigenvectors with their respective $\lambda_{\vec{v}}$ and $\lambda_{\vec{p}}$ eigenvalues. We also know from proposition 1.2 that such eigenvalues can assume two values $\{-1, +1\}$ due to the specific properties of our operators.

To determine the ground state of the Hamiltonian we have to determine the minimum energy of the system. Therefore we compute all the possible combinations of the eigenvalues associated to the vertex and palquette operators to obtain the following range of values:

$$\sigma(H) = \begin{cases} 2N^2, & if\ all\ \lambda_{\vec{v}} = \lambda_{\vec{p}} = 1,\ \forall\ \vec{v} \in V, \vec{p} \in P \\ \dots \\ intermediate\ energies, \\ \dots \\ -2N^2+1 \\ -2N^2, & if\ all\ \lambda_{\vec{v}} = \lambda_{\vec{p}} = 1,\ \forall\ \vec{v} \in V, \vec{p} \in P \end{cases}$$

The minimum is reached at $-2N^2$ on the eigenstates associated to the eigenvalues $\lambda_{\vec{v}} = \lambda_{\vec{p}} = 1,\ \forall\ \vec{v} \in V, \vec{p} \in P$. Such states form the ground state manifold of the system.

□

In order to determine which are the admissible configurations of eigenstates that we can use to form the ground state, we treat the $A_{\vec{v}}$ and $B_{\vec{p}}$ operators as constraint equations over the configuration space. As a consequence, all of our configurations will have to satisfy the two following conditions:

$$A_{\vec{v}}|\psi\rangle = +1|\psi\rangle \tag{1.1}$$

$$B_{\vec{p}}|\psi\rangle = +1|\psi\rangle \tag{1.2}$$

This means that configuration $|\psi\rangle$ needs to be be an eigenvector for all the vertices and plaquette operators.

In order to satisfy equation (1.1), we look for loop configurations such that if we apply $A_{\vec{v}}$ to that state, the result would still yield an overall positive value.

Graphically, we identify the loop configurations through strings of 'occupied' edges (shaded in black in figure 1.4 and 1.5).

Definition 1.12. *(Occupied edges) Given a state $|\psi\rangle$, we say that in the state $|\psi\rangle$ the edge e of the lattice is occupied if $\sigma^z_e|\psi\rangle = -|\psi\rangle$.*

Then, if we apply $A_{\vec{v}}$ at one of the open ends of such strings, we have two possibilities: leaving the string of occupied edges open (figure 1.4) or closing the string over the $A_{\vec{v}}$ operator (figure 1.5). In the first case, what we get is $\sigma^z|\psi\rangle$ at one endpoint of the string, which implies obtaining $-1|\psi\rangle$ eigenvalue for that endpoint; as a result we violate constraint (1.1). In the second case, instead, we end up with positive eigenvalues for each

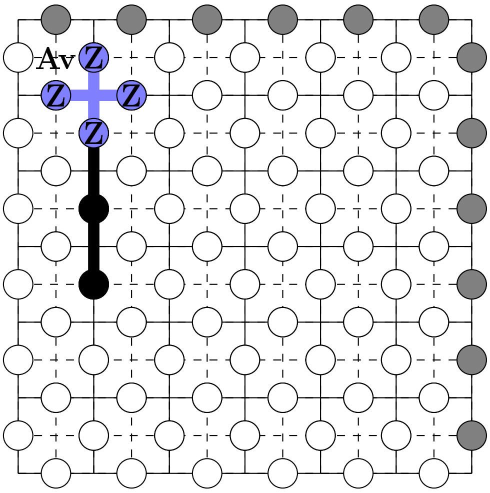


Figure 1.4: Operator A_v applied on an open string of occupied edges.

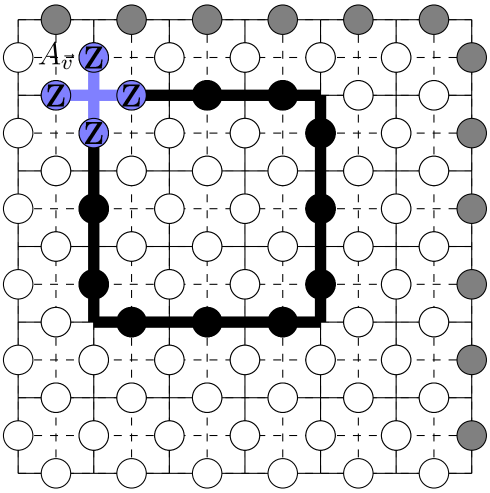


Figure 1.5: String of occupied edges closed on an operator $A_{\vec{v}}$.

edge, this is because the negative signs obtained at the endpoints cancel themselves out. Overall, in order to respect contraint (1.1) we are interested only in closed loops on $A_{\vec{v}}$. Notice that such loops will always have an even length [Her20].

If we consider only closed loops, we can identify different configurations having such characteristic. The illustrations in figure 1.6 show some examples of them. Each of these loops, having such properties, is an eigenstate of all $A_{\vec{v}}$ since, no matter how we locally apply $A_{\vec{v}}$, it will always preserve the sign of the state.

Now we focus on constraint (1.2). We want to apply the plaquette operator $B_{\vec{p}}$ only to eigenstates of $A_{\vec{v}}$, which we have determined above. The illustration in figure 1.7 shows an example of how to do it in practice with one of the eigenstates of $A_{\vec{v}}$. Notice that, after the transformation, we do always end up with a valid eigenstate of $A_{\vec{v}}$ but any such state cannot be an eigenstate of $B_{\vec{p}}$ by itself.

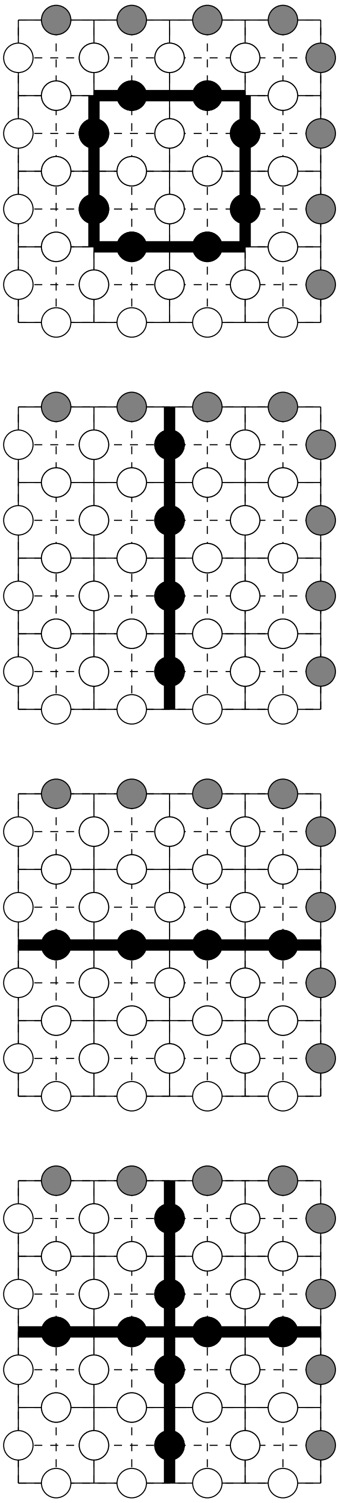


Figure 1.6: Examples of closed loops yielding +1 eigenvalue for $A_{\vec{v}}$

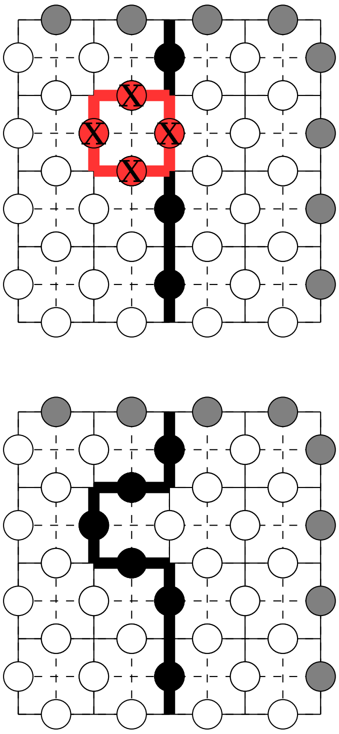


Figure 1.7: Examples of $B_{\vec{p}}$ to a closed loop.

Furthermore, any new configuration that we obtain through the application of the plaquette operator to eigenstates of $A_{\vec{v}}$ simply yields a permutation of the edges of the initial state, provided that the topological characteristics of the loop are preserved.

This means that if we firstly partition the eigenstates of $A_{\vec{v}}$ in the following fours classes:

1. class 0 : contains all closed loops as well as the state without loops; in other words, all of the closed loops can be continuously deformed into the state without loops.
2. class 1 : contains loops that wind the torus all the way in the horizontal direction an odd number of times;
3. class 2 : contains loops that wind the torus all the way in the vertical direction an odd number of times;
4. class 3 : contains loops that wind the torus all the way in both directions an odd number of times. Notice also that in class 3 the vertical loop must be taken on the dual lattice to yield a valid configuration.

Then, applying a plaquette operator to any of the eigenstates belonging to one of the classes above must yield an eigenstate that lies in the same class. This is formally expressed by stating that the class is invariant under the action of the operator $B_{\vec{p}}$.

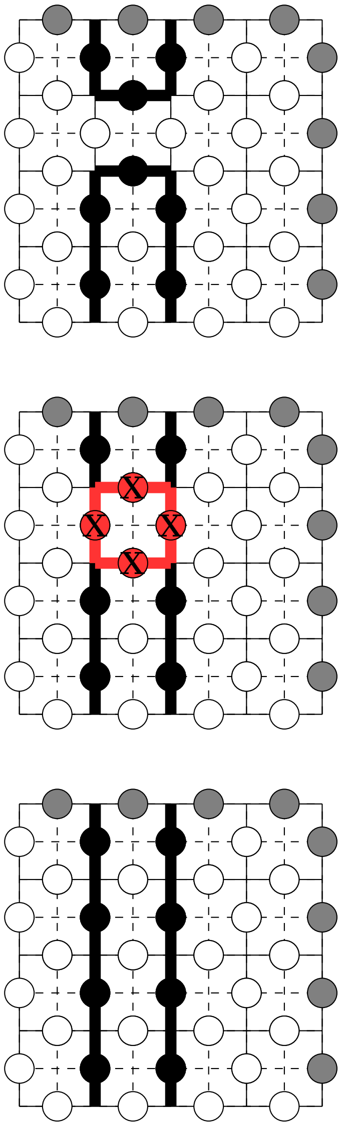


Figure 1.8: Apply $B_{\vec{p}}$ to a closed loop.

Proposition 1.6. *(Invariance under $B_{\vec{p}}$) The four classes of eigenstates of $A_{\vec{v}}$ are invariant under the action of the $B_{\vec{p}}$ operators, for each $\vec{p} \in P$.*

We do not provide a complete proof of the above result but illustrate various examples. If we take the loop illustrated in figure 1.8 and close it by means of a plaquette operator as illustrated, we would be brought to believe that there indeed exist a way to apply the operator $B_{\vec{p}}$ such that we exit class 0 and land in class 1. Though, this is not possible.
In fact, we can show the impossibility of the above action by defining two topological indeces to label the four classes of eigenstates. These indeces can assume values in $\mathbb{Z}_2 = \{\overline{0}, \overline{1}\}$, whose elements respectively represent the 'sets' of even and odd numbers. In our context such numbers identify the number of vertical and horizontal loops of a configuration intersecting the vertical and horizontal winding of the torus. Notice that the above mentioned

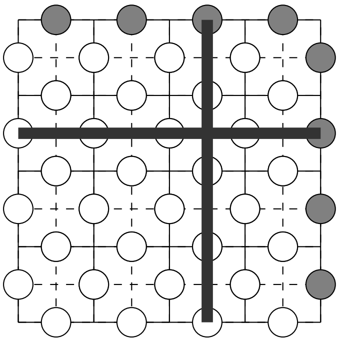


Figure 1.9: One horizontal circle and one vertical circle passing passing through the spins of the dual lattice.

intersections are meant to be computed by fixing two circles (figure 1.9): one horizontal circle and one vertical circle passing passing through the spins of the main lattice.

The above indeces can be expressed more clearly as follows:

$$n_x = (number\ of\ vertical\ intersections) \bmod 2 \ \in \ 0,1 \ with\ the\ right\ boundary$$

$$n_y = (number\ of\ horizontal\ intersections) \bmod 2 \ \in \ 0,1 \ with\ the\ upper\ boundary$$

in order to create a correspondence with the actual definition of the torus over $(\mathbb{Z} \times \mathbb{Z})$.

We can then label the four classes as : $(\overline{0},\overline{0})$, $(\overline{0},\overline{1})$, $(\overline{1},\overline{0})$, $(\overline{1},\overline{1})$.

If we go back to the configuration in figure 1.8, after the application of the operator $B_{\vec{p}}$ we do not land in class 1, but we remain in class 0; this is because we have an even number of horizontal intersections and an even number of vertical intersections, i.e. the indeces equal $(\overline{0},\overline{0})$.

Hence all the classes are $B_{\vec{p}}$*-invariant* due to the topological properties of the torus. More formally, given $|\psi_1\rangle, ..., |\psi_i\rangle, ..., |\psi_n\rangle$ the eigenstates of $A_{\vec{v}}$, $\forall \vec{v} \in V$ all belonging to the same class, we can construct a state $|\psi\rangle$:

$$|\psi\rangle = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} |\psi_j\rangle$$

Then since we know that for every $\{i \in 1...n\}$ $B_{\vec{p}}|\psi_i\rangle = +|\psi_j\rangle$ for some $j \in \{1...n\}$ we get:

$$|\Xi\rangle = B_{\vec{p}} \frac{1}{\sqrt{n}} \sum_{j=1}^{n} |\psi_i\rangle = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} B_{\vec{p}}|\psi_i\rangle = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} |\psi_J\rangle = |\Xi\rangle, \forall \vec{p} \in P$$

because, $B_{\vec{p}}$ simply permutes all $|\psi_i\rangle$, thus we get back our initial state. This result leads to the following proposition:

Proposition 1.7. *(Eigenstates of $B_{\vec{p}}$) A completely symmetric superposition of all the eigenstates of $A_{\vec{v}}$ belonging to the same class, is an eigenstate of $B_{\vec{p}}$.*

From Proposition 1.6 and Proposition 1.7, also naturally follows the degeneracy (and dimension) of the ground state manifold:

Proposition 1.8. *(Degeneracy of the ground state) The degeneracy of the ground state manifold is* 4.

Proof.
Because the four classes of eigenstates are $B_{\vec{p}}$-*invariant*, we can construct a valid ground state through the configurations belonging to one the four classes, as these configurations respect both constraint (1.1) and (1.2).

□

Now that we have characterized the ground state configurations of the toric code, we focus on studying its excited states. These states arise from perturbations or interactions within the system and offer valuable insights into the dynamic behavior of the toric code beyond its ground state equilibrium. While the ground state embodies the system's lowest energy configuration and serves as the foundation of its stability, the excited states provide an insight into the system's response to external influences. Through the systematic study of these excitations we discover some intrinsic features of the toric code that will serve us later to explain its fault-tolerant nature.

1.4. Excited States

Taking into account the results of the previous chapters, we now proceed to describe, firstly, what are the excitations of the ground states and what are their exchange statistics. In general, the low-energy excitations of a quantum system, often referred to as quasiparticles, represent deviations from the system's ground state. These excitations are associated to elementary quantum particles that can be created or annihilated with relatively low energy. The properties of these quasiparticles, including their statistics, are crucial for gaining insights into the system's ground state degeneracy, topological properties and response to external perturbations.
In the particular case of the toric code, such low-energy excitations can be created through the application of a string operator on the ground state (figure 1.10). Let us start by introducing some definitions that will be useful in the following.

Definition 1.13. *(Adjacent edges) We say that the edges* $e_1 = (v_1, v_1')$, $e_2 = (v_2, v_2') \in E$ *are adjacent if* $e_1 \neq e_2$ *and* $\{\vec{v}_1, \vec{v'}_1\} \cap \{\vec{v}_2, \vec{v'}_2\} \neq \emptyset$.

Definition 1.14. *(Length of a string) We define the length of a string as* $\ell_\Sigma = \max\{|(\vec{v'}_n - \vec{v}_1)_y|, |(\vec{v}_n - \vec{v'}_1)_y|, |(\vec{v'}_n - \vec{v}_1)_x|, |(\vec{v}_n - \vec{v'}_1)_x|\}$

Definition 1.15. *(String* Σ*) A string* Σ *of the lattice consists of a set of numbered edges* $\{e_1, \ldots, e_i, \ldots e_n\}$ *where* e_i *is adjacent to* e_{i+1} *and such that the overall length of the string is equal to* $\ell_\Sigma < N$.

Definition 1.16. *(Endpoints of a string) Given a string* Σ *we identify its endpoints with the initial vertex* $\vec{v_1}$ *and its final vertex* $\vec{v'_n}$.

Definition 1.17. *(Macroscopic loop) A macroscopic loop consists of a set of* $\{e_1, \ldots, e_n\}$ *such that* $e_1 = e_n$ *and* $\ell_\Sigma \geq N$.

Analogously, we can define the notion of adjacent edges, strings and length of a string and macroscopic loop also on the dual lattice. We give here explicitly only the definition of a string on the dual lattice:

Definition 1.18. *(Dual string* Σ'*) A dual string* Σ' *of the dual lattice consists of a set of numbered dual edges* $\{e'_1, \ldots, e'_i, \ldots e'_n\}$ *where* e'_i *is adjacent to* e'_{i+1} *and such that the overall length of the string is equal to* $\ell_{\Sigma'} < N$.

Definition 1.19. *(*S^x_Σ *string operator) Given a string* Σ*, the string operator is given by*

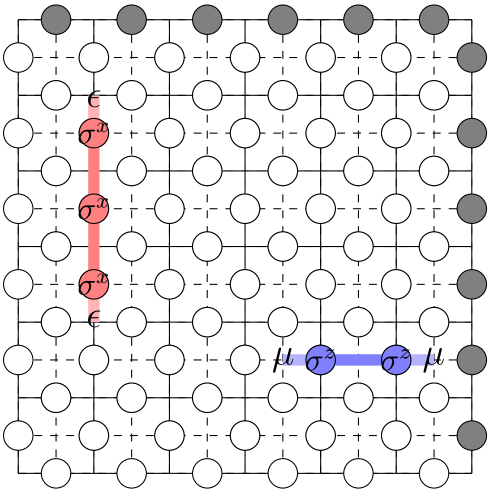


Figure 1.10: Graphical representation of ϵ and μ excitations on the ground state.

$S^x_\Sigma = \prod_{j=1}^n \sigma^x_{e_j}$.

Definition 1.20. *($S^z_{\Sigma'}$ string operator) Given a string Σ', the string operator is given by* $S^z_{\Sigma'} = \prod_{j=1}^n \sigma^z_{e'_j}$.

The application of string operators creates quasi-particles at their endpoints. In particular we can get two distinct types of excitations (figure 1.10):

Definition 1.21. *(Electric charges) The pairs of particles generated by the application of a string operator S^x_Σ on a ground state are called electric charges ϵ.*

Definition 1.22. *(Magnetic vortices) The pairs of particles generated by the application of a string operator $S^z_{\Sigma'}$ on a ground state are called magnetic vorteces μ.*

These excitations can be transported on the lattice via the action of vertex and plaquette operators and interact with each other so characterizing the low energy spectrum.
In order to understand this, we study the relationships between string operators and vertex/plaquette operators as we have defined them (see section 1.2).

We can indeed prove that a string operator S^x_Σ operator anticommutes with $A_{\vec{v}}$ for each one of its endpoints.

Proposition 1.9. *(Anticommutation of S^x_Σ) Given a string Σ, the string operator S^x_Σ anticommutes with $A_{\vec{v}}$, for any $\vec{v} \in \{\vec{v_1}, \vec{v'_n}\}$ where $\vec{v_1}, \vec{v'_n}$ are the endpoints of the string Σ.*

Proof.
Let $\vec{v}$ be one of the endpoints of Σ. We want to show that $A_{\vec{v}} S^x_\Sigma + S^x_\Sigma A_{\vec{v}} = 0$. Knowing

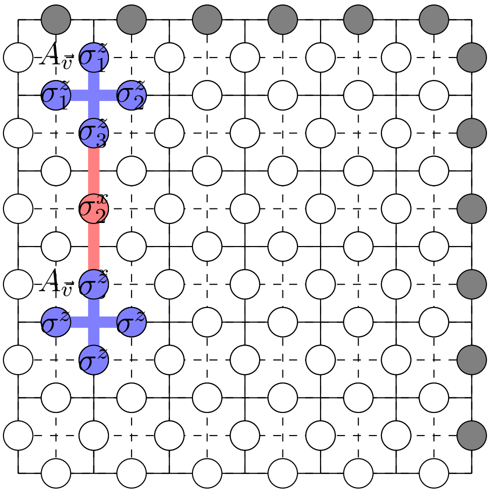


Figure 1.11: String operator anticommutes with two $A_{\vec{v}}$ each.

that for $\vec{v}$

$$A_{\vec{v}} = \prod_{i=1}^{4} \sigma_i^z$$
$$S_\Sigma^x = \prod_{j=1}^{n} \sigma_{e_j}^x$$

by substitution we obtain the following expression:

$$\prod_{i=1}^{4} \sigma_i^z \prod_{j=1}^{n} \sigma_j^x + \prod_{j=1}^{n} \sigma_j^x \prod_{i=1}^{4} \sigma_i^z = 0 \tag{1.3}$$

remeber that only one σ_i^z will overlaps with the endpoints of the string S_Σ^x; thus, only one σ_i^z anticommutes with the endpoints of the string S_Σ^x. Knowing that Pauli matrices acting on the same edge anticommute

$$\sigma_{v'}^x \sigma_{v'}^z = -\sigma_{v'}^z \sigma_v^x \text{ for } v = v'$$
$$\sigma_v^x \sigma_{v'}^z = \sigma_{v'}^z \sigma_v^x \text{ for } v \neq v'$$

if for simplicity we fix $N = 2$ we can easily see that :

$$(\sigma_1^z \sigma_2^z \sigma_3^z \sigma_4^z)(\sigma_1^x \sigma_2^x) = -(\sigma_1^x \sigma_2^x)(\sigma_1^z \sigma_2^z \sigma_3^z \sigma_4^z)$$

having supposed that, for example, the extremity σ_1^x overlaps with the σ_3^z edge part of the $A_{\vec{v}}$ operator (figure 1.11). This can then be easily generalized to N Pauli operators. Finally, if we substitute such expression in (1.3) we obtain that the overall equation is in fact equal to zero.

This procedure should be iterated also for the other extremity of the string S_Σ^x to show that the operator commutes with two $A_{\vec{v}}$ operators, one for each endpoint.

□

In the same way, we could prove that, if we take a string $S^z_{\Sigma'}$ it anticommutes with one $B_{\vec{p}}$ for each one of its endpoints.

Proposition 1.10. *(Anticommutation of $S^z_{\Sigma'}$) The $S^z_{\Sigma'}$ operators anticommute with $B_{\vec{p}}$ for any $\vec{p} \in \{\vec{p}_1, \vec{p'}_n\}$ where $\vec{p}_1$ and $\vec{p'}_n$ are the endpoints of the dual string.*

The effect of putting such strings on the ground states correspond to raising the associated energy of the ground states by 2. In fact, coherently with what said in [Her20], it is impossible to create unitary excitations on the ground states.

Proposition 1.11. *(Magnitude of the excitations) Acting with a string operator on the ground state corresponds to raising the associated energy of the ground states by 2.*

Proof.
Recall that for a string operator, here we will choose S^x_Σ, both equations hold:

$$\begin{cases} A_{\vec{v}_1} S^x_\Sigma + S^x_\Sigma A_{\vec{v}_1} = 0 \\ A_{\vec{v}_2} S^x_\Sigma + S^x_\Sigma A_{\vec{v}_2} = 0 \end{cases}$$

If we apply the string operator to the ground state, taking into account anticommutativity stated above, we obtain that:

$$\begin{cases} A_{\vec{v}_1} S^x_\Sigma |GS\rangle = -S^x_\Sigma A_{\vec{v}_1} |GS\rangle = -S^x_\Sigma |GS\rangle \\ A_{\vec{v}_2} S^x_\Sigma |GS\rangle = -S^x_\Sigma A_{\vec{v}_2} |GS\rangle = -S^x_\Sigma |GS\rangle \end{cases}$$

since $A_{\vec{v}_1}|GS\rangle = +1|GS\rangle$ and $A_{\vec{v}_2}|GS\rangle = +1|GS\rangle$. Summing up term by term :

$$\Big\{ A_{\vec{v}_1} S^x_\Sigma |GS\rangle + A_{\vec{v}_2} S^x_\Sigma |GS\rangle = -2 S^x_\Sigma A_{\vec{v}_1} |GS\rangle$$

Thus, if the energy of the ground state is $-2N^2$ by acting with a string operator on it we obtain a final energy equal to $-2N^2 + 2$.

□

1.4.1. Fermionic and bosonic exchange statistics in 3D and 2D

We give here a brief description of the statistics of fermions and bosons in three and two dimensions.
Firstly, we introduce indistinguishable particles by means of their physical properties

Definition 1.23. *(Indistinguishable particles) Indistinguishable particles in Quantum Mechanics are particles that share the same mass, electric charge and spin.*

However, we can characterize identical particles by means of their wavefunctions, which are mathematical functions that represent the probability amplitude of finding a particle at a certain position. In fact, in Quantum Mechanics, due to the uncertainty principle, we know that it is not possible to measure any two physical properties, such as position and momentum of a particle, together and with infinite precision. Indeed, this would correspond to the progressive spreading and overlap of the wavefunctions of the particles in the system, leading to indistinguishability. Therefore, what we do is calculating a probabilistic location of the particles. As a consequence, in order to study the behaviour of the particles we look at the overall acquired phase of the wavefunction of the system of which they are part of. Observing such phase allows us to classify particles by assigning them bosonic or fermionic statistics, which correspond to symmetric or antisymmetic wavefunctions meaning that the system will acquire a phase corresponding to +1 or -1, respectively. As we will see, there also exist cases where we obtain mutual statistics that fall out of the bosonic or fermionic ones. By controlling the acquired phase of the wavefunction, we can deduce interference patterns between particles, which is important to perform quantum computations.

If we take a system of indistinguishable particles in three dimensions as in [Cor23], if we exchange any two of them such that $i < j$ we can only obtain a phase $\phi = \pm\pi$ in the resulting wave function:

$$P_{(i,j)}\psi(x_1, ..., x_i, x_j, ..., x_N) = e^{i\phi}\psi(x_1, ..., x_j, x_i, ..., x_N) = \pm\psi(x_1, ..., x_j, x_i, ..., x_N)$$

where $P_{(i,j)}$ indicates the permutation operator that exchanges two particles according to the permutation group $\mathbb{P}_{\mathbb{N}}$ that characterizes fermionic and bosonic statistics.

Definition 1.24. *(Permutation operator) Given a set of N objects an operator $P \in P_N$ acts as $(P\psi)(x_1, ..., x_N) = \psi(x_{\pi(1)}, ..., x_{\pi(N)})$ with π permutation of N objects.*

The permutation group $\mathbb{P}_N$ is defined as the set of all possible permutations involving N objects. Within this group, the operation of group multiplication (exchange) corresponds to the sequential application of permutations, and the group inverse involves the inverse of the permutation. Here, executing a permutation twice on two objects results in the initial configuration of the system. Consequently, the Hilbert space of a quantum system of indistinguishable particles must be a realization of a one-dimensional representation of the permutation group; there exist only two such representation (inequivalent and irreducible), corresponding to bosons or fermions. To visualize the behaviour described above, consider the case where the relative coordinate of the particles is constrained to

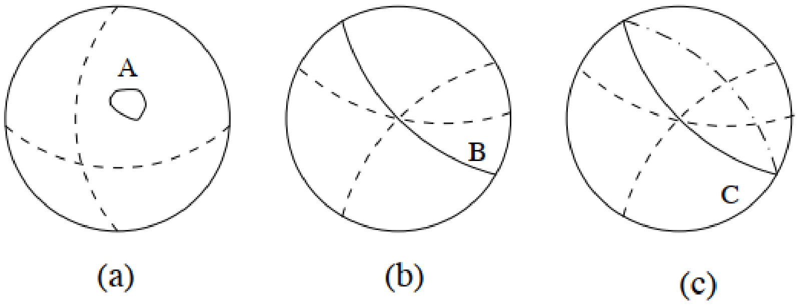


Figure 1.12: Possible exchanges of two particles in 3D. From: [Rao16].

move on a sphere. Closed paths emerge when particles return to their original positions (no exchange) or reach the antipodal point (exchange). In fact, we will have three distinct types of paths (figure 1.12):

(a) : closed loops without exchange; they can disappear if reduced to a point, implying that the wave-function cannot acquire any phase;

(b) : involve a single exchange; they connect two fixed points on the sphere, making it impossible to erase them. Consequently, this types of exchanges introduce a phase in the wave-function;

(c) : closed loops with two exchanges, as in case 1; as a result, also in this case the wavefunction does not acquire any additional phase.

Overall, we only have two classes of paths: those that do not involve any exchange as in class (a) and (c), and those that are characterized by an exchange as in class (b). Since exchanging the particles twice leads again to no exchange, we must have $e^{2i\phi} = 1$. This means that ϕ can either be 0 or π giving rise to bosonic and fermionic statistics, respectively.

In contrast to what stated above, take a system of N indistinguishable particles in two dimensions, if we exchange any two of them such that $i < j$ we can obtain any phase ϕ in the resulting wave function, based on the number of exchanges:

$$R_{(i,j)}\psi(x_1, ..., x_i, x_j, ..., x_N) = R_{(i,j)}e^{i\phi}\psi(x_1, ..., x_j, x_i, ..., x_N) =$$
$$R_{(i,j)}e^{i2\phi}\psi(x_1, ..., x_i, x_j, ..., x_N) = R_{(i,j)}e^{i3\phi}\psi(x_1, ..., x_j, x_i, ..., x_N) = ...$$
$$... = e^{in\phi}\psi(x_1, ..., x_i, x_j, ..., x_N)$$

where R represents the braiding operator, which corresponds to the exchange operator for particles in two dimensions. The resulting statistics is characterized according to the braiding group $\mathbb{B}_{\mathbb{N}}$. We can visualise the process of moving (or exchanging) particles as paths in spacetime with time being the vertical axis and space being the horizontal axis (figure 1.14). The particles can circle around each other and form closed paths by coming back to their original positions [Wil91]. For further details see [Rao16, Wil91]. To visualize the above behaviour, consider moving particles on a circle. This time several possible paths are possible, in particular closed paths. In fact, if we classify again the possible paths as we did before, this time we obtain these three classes (figure 1.13):

(a) : those that can be shrunk into a point;

(b) : those that cannot be contracted into a point as the endpoints remain fixed;

(c) : also this kind of paths cannot be shrunk into a point since they wind all the way around the circle. Notice that in three dimensions the path corresponding to two exchanges is equivalent to no exchange at all, while this is not possible in two dimensions. This will be crucial for the existence of intermidiate statistics that exist only in 2D (section 1.5).

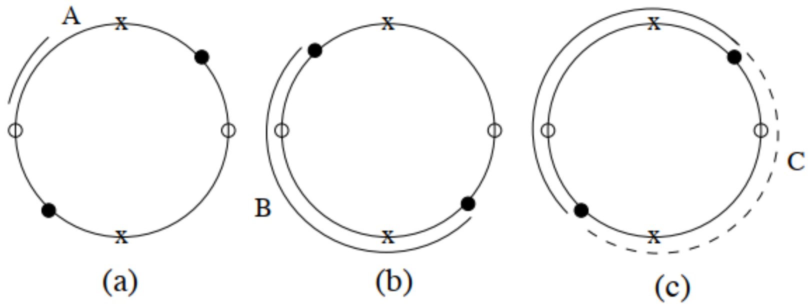


Figure 1.13: Possible exchanges of two particles in 2D [Rao16]. We denote diametrically opposite points on the circle (x, empty end black point) to indicate the starting and ending points of a half- exchange of particles along with the three possible paths denote as A,B and C. Case A represents a contractible string, case B represents a non-contractible string obtained through a half-exchange happening between two diametrically opposite points and case C represents a non-contractible string obtained by means of a double half-exchange.

Overall, for $\phi = 0, \pi$ we would obtain the usual bosons and fermions statistics, but since in general, $e^{in\phi} \neq \pm 1$ for any n (n exchanges never yield the identity), ϕ can be anything so any statistic is possible in two dimensions. Such diverse statistics in two dimensions

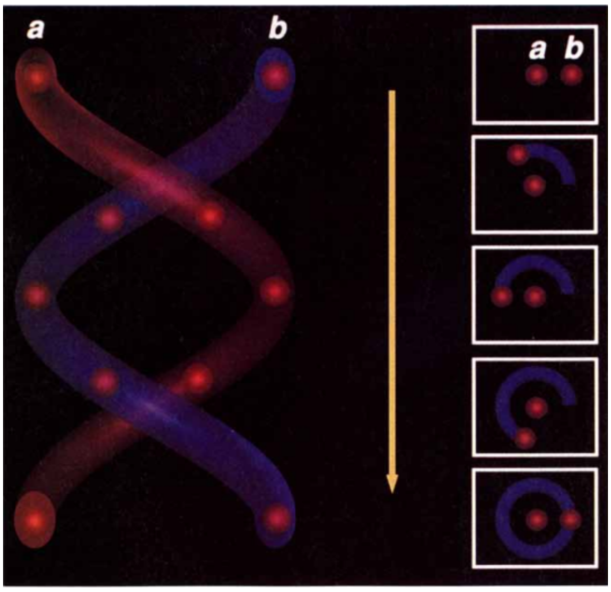


Figure 1.14: Identical particles exchange position twice as they travel in three dimensions (no exchange). Particle a and b move along a closed path in two dimesions (non contractible) [Wil91].

are called fractional or *anyonic* (see section 1.5).

We can now analyze the statistics of ϵ-particles in the context of the ground state of the toric code. If we consider the action of a string operator S^x_Σ on any ground state $|GS\rangle$, it creates two ϵ-particles of the vertices adjacent to the edge e, as shown in figure 1.15. As we are going to see, a string operator implements the exchange of these two excitations. In fact, as illustrated in figure 1.15, the exchange of these particles becomes possible by applying the string operator in such a way that it moves the excitations around the lattice until they reach the final configuration (c) shown in figure 1.15. However, due to the commutation of all σ^x with each other, after the application of a closed loop of σ^x (last step in figure 1.15), there is no overall acquired phase. Consequently, the ϵ-excitations obey bosonic statistics [Rao16, Her20]. A similar argument can be carried out for the μ-excitations, proving that they behave as bosons as well.

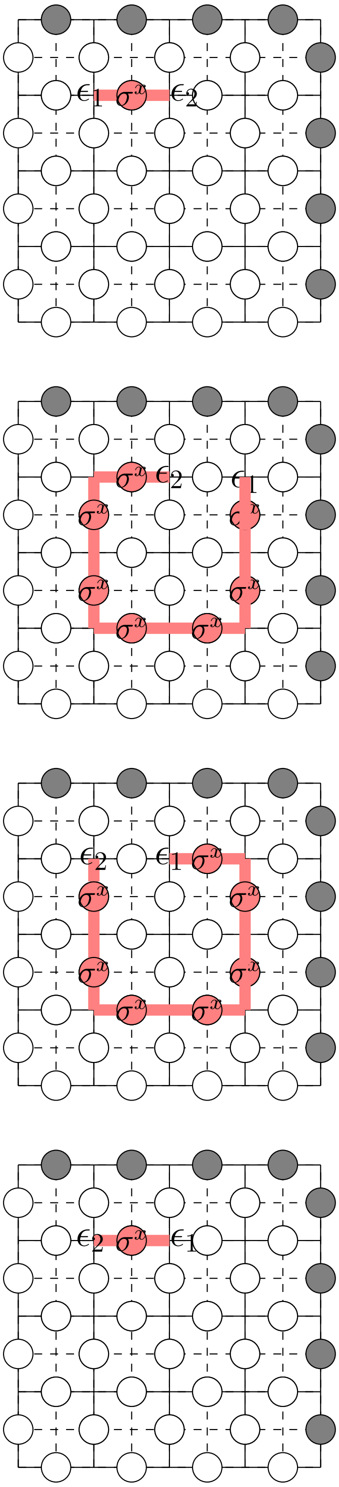

ϵ1
σx
ϵ2
σx
ϵ2
ϵ1
σx
σx
σx
σx
σx
σx
ϵ2
ϵ1
σx
σx
σx
σx
σx
σx
σx
ϵ2
σx
ϵ1

Figure 1.15: In the first picture we show the creation of a pair of ϵ-excitations on the ground state by means of σ^x operator. In the following pictures we move around the excitations using a string operator until we reach the configuration in the third picture. In the last passage we apply a closed loop of σ^x in order to go back to the first kind of configuration [Rao16].

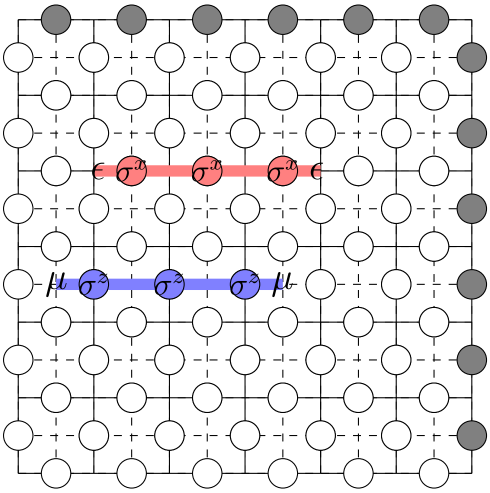


Figure 1.16: Creation of the ϵ-excitations and μ-excitations.

1.5. Anyonic Excitations

Let us examine now the mutual statistics between the ϵ and μ particles by means of an example. [Rao16]. Initially, we generate pairs of excitations at distinct chosen sites by applying two S^x_Σ and $S^z_{\Sigma'}$ operators as shown in figure 1.16. There, we separate the excitations using string operators and we move an ϵ-particle around an μ-particle, as depicted in figure 1.17. Notice that there is a site where σ^z must cross a σ^x operator, resulting in an anticommutation and thus introducing a minus sign [Kit02]. The closed loop can be eliminated, leading to the following transformation:

$$\sigma^z_{e'_i}\sigma^x_{e_i}|\psi\rangle = -\sigma^x_{e_i}\sigma^z_{e'_i}|\psi\rangle$$

if we denote the exchange operator as *the braiding operator* R (see [Rao16]) then, according to the discussion in Section 1.4.1, we have that

$$R^2\psi(\epsilon,\mu) = -\psi(\epsilon,\mu)$$

since we are going all the way around the particle, wehave performed two exchanges in 2D. Then, we find out that:

$$R^2 = -1 \rightarrow (e^{-\frac{i\pi}{2}})^2 = -1 \rightarrow R = e^{-\frac{i\pi}{2}} = \pm i$$

which cannot be described by the usual bosonic/fermionic statistics. Thus we can say that the low-energy excitations of the ground state exhibit mutual anyonic statistics [Rao01, Kha98].

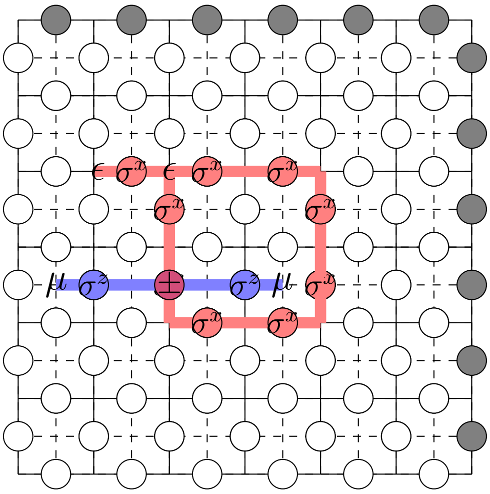

ϵ
σ^x
ϵ
σ^x
σ^x
σ^x
σ^x
μ
σ^z
σ^z
μ
σ^x
σ^x
σ^x

Figure 1.17: Creation of an anyonic excitation by braiding of ϵ and μ pairs of particles. Here we loop the ϵ-excitation around an μ-excitation. The red loop can be removed with the result that the ground state acquires a nontrivial phase.

2 | Coding on the Toric Code

2.1. Classical Error Correction

In classical computing the most elementary unit of information is the *bit*. A bit stores binary information, which can either assume value 1 or 0. When we transmit information in the classical world, we immagine to make strings of bits travel through a channel similar to the one depicted below in figure 2.1. During the transmission a bit can change its value either going from 1 to 0 or viceversa, this event is called *bit-flip* and it tells us that an error has occured in the transmission. A bit-flip can occur independently on each bit with a certain probability that we will denote in general as p: mathematically, this translates to the assumption that errors, i.e. bit-flips, are assumed to be independently identically distributed on all the bits with the same probability p. In such a case we say that errors are i.i.d. with probability p.

The noisy channel described in picture 2.1 is one of the easiest error models also known as the *Binary Symmetric Channel*. An error model, in the context of information theory and communication, is a mathematical or conceptual representation of how errors can occur during the transmission of information. It describes the types of errors that can happen, the probabilities associated to each type of error and the conditions under which these errors occur.

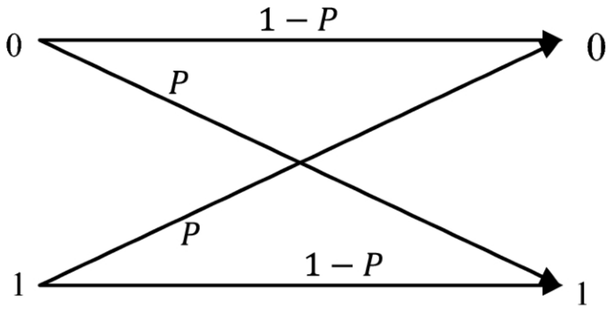


Figure 2.1: Binary symmetric channel [Cha06]

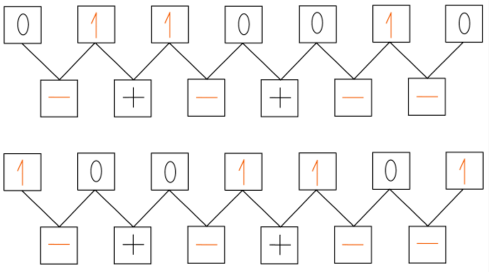


Figure 2.2: Example of error syndrome associated to a transmitted codeword of length 7. We have two equal error syndromes despite having two different codewords. Each '+' or '-' represents the parity of the bits acting as parents [Kas19]

In order to protect information what we do is introduce redundancy, which allows us to detect errors and correct them. Redundacy means duplicating bits of information before sending them through a channel. We say that we encode logical bits into *strings of bits* (physical bits). For example: the bit 0 can be encoded into the pair of bits 00 and 1 into 11. This is useful to detect errors because in this way, whenever a bit changes we know that a single error has occured since the pairs 10 and 01 should never occur. Formally speaking, we have introduced a *parity*: the strings 00 and 11 have an *even* parity while 10 and 01 have an *odd* parity. Whenever we receive a string, we perform *parity-check* and signal an error whenever we get odd parity. We will call the values obtained through the parity-check procedure *error syndromes*.

In order to also be able to correct errors, the redundancy above is not enough since we would not be able to decide a criterion to correct the corrupted information. Thus, we introduce a third redundant bit and generate the *three repetition code*; in this way, when we decode information at the destination, we can apply what is called *majority voting*. Majority voting decodes into 1 those triplets that have most bits set to 1 and decodes as 0 the ones that have most bits set to 0. Though, as soon as two bit-flips happen, majiority voting stops working. In fact, the state 000 with two errors (110) would be associated to 111 and vice-versa. Therefore we will need other tools to decode information correctly. In fact, in the case of the three repetition code, in order to perform a parity-check we look at two 'error bits' for each parity in the error syndrome, which could still yield the same error syndrome for two different bit-flip errors scenarios (figure 2.2). Though, if we can assign a fixed iid error probability of $p < \frac{1}{2}$ for each bit, we will always end up choosing to decode the codeword with less errors. This method does not erase the possibility of getting decoding mistakes but significantly decreases them [Kas19].

The above redundant code is actually an example of what is called a *linear code.* We can in fact reframe all of the above in terms of linear algebra. To do so we define what is called the generator matrix **G** as given in [Cha06, Kas19], which is the operator that we use to introduce redundancy.

Definition 2.1. *(Generator matrix) A linear code encoding k-bit messages into an m-bit codespace* $\boldsymbol{C}$ *is specified by an* $m \times k$ *matrix* $\mathbf{G}$*.*

The matrix **G** encodes the original information into codewords. For example, if we have a single bit to be encoded in a three bit codeword we can do the following:

$$\mathbf{G} = [1\ 1\ 1]^T,$$

$$\text{a} = [i], \text{ where i} = 0{,}1\ ,$$

$$[1\ 1\ 1]^T\ [0] = [0\ 0\ 0]^T.$$

A linear code encoding k-bit messages into an m-bit codespace **C** is specified by an m × k generator matrix **G** is an [m, k] code.

The generator matrix is used to encode information before sending it through a channel but we have also said that a realistic channel is noisy so it can modify the codewords we send. Thus, we also need to define an operator to specify how an error occurs and how to detect it.

Definition 2.2. *(Noise operator) We denote a noise operator as* $\boldsymbol{N}_j \in \mathcal{N}$ *where* j *denotes the element that will be modified by* $\boldsymbol{N}_j$ *in the original codeword and* $\mathcal{N}$ *represents the set of all possible error operators that could happen on the codeword.*

Overall, the codeword is changed by the noise operator into the correspondig codeword $c' = c + n_j$ where n_j is an m × 1 vector having a 1 at the j-th row and zeros elsewhere; notice that, since we work with bits, addition is always performed modulo 2.
Having defined all of the above we can come to parity-check performed after transmission.

Definition 2.3. *(Parity-check matrix) Given an m-bit codespace and a k-bit encoding, we define a parity check matrix as a matrix denoted by* $\boldsymbol{H}$ *having size* $(m - k) \times m$ *and such that the code* $\mathbf{C}$ *is the kernel of the parity-check matrix.*

To check parity we can simply act with **H** on the received codeword:

$$\mathbf{H}\text{c} = 0$$

for any valid codeword. This exactly means that the code **C** is the kernel of the parity-check matrix.
Instead, for any corrupted codeword the computation yields:

$$\mathbf{H}c' = \mathbf{H}(c + n_j) = 0 + \mathbf{H}n_j = \mathbf{H}n_j$$

which is the error syndrome. As anticipated, the error syndrome serves as an indicator of errors within a coding and decoding system. In the absence of errors, the error syndrome is represented as 0. This indicates a clean transmission. Conversely, in the case of a single error, the error syndrome is denoted as $\mathbf{H}n_j$, providing information about the specific bit j within the codeword where the error has taken place. Consequently, the corrupted codeword can be successfully decoded by correcting the jth bit. However, as already mentioned, when two errors n_i and n_j are present, the error syndrome becomes $\mathbf{H}(n_i + n_j)$. In this case error correction becomes more difficult because there could be a type of error $n_q = n_i + n_j$ that makes error detection ambiguous, as we could correct the q-th bit instead of the j-th bit. Overall, it is possible to have multiple error combinations leading to the same syndrome. To prevent ambiguity, it is essential to verify that any additional error n_q does not satisfy the condition $n_q = n_i + n_j$. If there are no such instances where n_q equals $n_i + n_j$, the decoding process can proceed with confidence. However, if such instances are detected, then ambiguity arises. One possible approach is to reject the codeword, prompting a request for retransmission from the sender. Alternatively, refining the encoding/decoding scheme may be considered.

Because of the ambiguity that can arise in error correction it is essential to discuss formally the detectability and correctability of errors according to linear codes formalism given in [Cha06] and discussed above. We have mentioned how a codeword can be rejected if corrupted and how a receiver can ask to resend information. The efficacy of the encoding/decoding scheme is evaluated based on its ability to either maintain the integrity of information or detect errors caused by each noise operator. Therefore, we can state that a code successfully detects a noise operator **N** if, for every codeword c within the code, the outcome is either $\mathbf{N}\mathrm{c} = \mathrm{c}$ or **N**c does not belong to the code **C**.

Theorem 2.1. *(Error detection)* ***N*** *is detectable by a code if and only if for each* $c_m \neq c_n$ *in the code,* $\boldsymbol{N}c_n \neq c_m$.

On the other hand, error correction involves an active process where the decoder does not only identifies errors but also corrects them. In the context of a code **C** and a set of error operators **N**, the primary goal is to establish the presence of a decoding procedure capable of correcting errors introduced by $\mathbf{N}$. Assuming the existence of distinct codewords c_m and c_n within the code, and for specific set of noise operators $\mathbf{N}$, there could be situations such that if we apply $\mathbf{N}_i$ to the codeword c_m and the error operator $\mathbf{N}_j$ to the codeword c_n we end up in the same final state c_q. In such cases the challenge lies in determining whether the original codeword was c_m or c_n. Thus, the error is unidentified meaning that it occurs in the set of error operators $\mathbf{N}$ but we do not know which specific error operator caused it. In other words, given the state c_q, we cannot say which codeword was initially sent due to the ambiguity in identifying the specific error that occurred.

Theorem 2.2. *(Error correction) The set* $\boldsymbol{N}$ *of errors is correctable if and only if for all* $c_m \neq c_n$ *in the code and for all* $i, j \in \boldsymbol{N}$*, it is true that* $\boldsymbol{N}_i c_m \neq \boldsymbol{N}_j c_n$*.*

Thanks to linear codes, we gain insights into the two essential steps of error recovery: error detection and error correction, which allows us to assess the efficacy of the code in its entirety. For further details see [Kas19]. Moreover, in order to evaluate the robustness of a given code we can define the notion of *distance of a code* :

Definition 2.4. *(Hamming distance) Given two codewords,* c_m *and* c_n*, the Hamming distance* $d(c_m; c_n)$ *is the minimum number of bits that must be flipped in order to transform* c_m *into* c_n*.*

Definition 2.5. *(Distance of a code) The distance of a code* $\mathbf{C}$ *is the minimal Hamming distance between any two distinct codewords* c_m *and* c_n *having* $m \neq n$*.*

To conclude, linear codes' representation plays a crucial role in studying encoding/decoding procedures along with their performance, offering an effective framework also to be used for the quantum world as well.

2.2. Quantum Error Correction (QEC)

Due to the no-cloning theorem, which states that it is impossible to perfectly clone an arbitrary quantum state using quantum operations, one might be brought to think that error correction cannot be performed in the quantum setting.

Theorem 2.3. *(No cloning theorem) There is no unitary operator U acting on the composite Hilbert space $\mathscr{H} \otimes \mathscr{H}$ such that for any $|\psi\rangle \in \mathscr{H}$:*

$$U(|\psi\rangle \otimes |0\rangle) = |\psi\rangle \otimes |\psi\rangle$$

where $|0\rangle$ is a reference state.

We will see that this problem can be solved, despite the two main issues that arise in quantum systems [Cha06].:

1. No cloning limitation: the attempt to implement a repetition code in the quantum setting by duplicating a quantum state multiple times violates the no-cloning theorem, which prevents the exact replication of a quantum state;
2. Information loss through measurement: in contrast to classical error-correction, the act of measuring a quantum state typically results in the loss of quantum information contained in that state, as a consequence, the recovery of the original quantum information becomes impossible after a measurement.

In the context of quantum computation, classical bits are replaced by quantum bits or qubits, which are represented by a two-level quantum system. More precisely, a qubit lives in an Hilbert space of dimension 2. A common physical realization of qubits involves spin-$\frac{1}{2}$ particles, such as electrons. The quantum state space of a qubit is spanned by two orthogonal basis vectors, conventionally denoted as $|0\rangle$ and $|1\rangle$, corresponding to the classical binary states 0 and 1. Mathematically, any qubit state $|\psi\rangle$ can be expressed as a superposition of these states as:

$$\alpha|0\rangle + \beta\,|1\rangle = [\,\alpha \quad \beta\,]^T$$

where α and β are complex numbers and satisfy the normalization condition $|\alpha|^2 + |\beta|^2 = 1$.

When it comes to qubits, computations are performed through operators, which are the quantum logical gates. Each quantum gate corresponds to a unitary operator, which is a linear transformation that preserves the norm of vectors (in this case, quantum states) and is represented by a unitary square matrix. These gates can perform various operations such as changing the phase of a qubit (phase-shift), performing logical operations like NOT(bit-flip), AND, or OR and more. In particular, a phase-shift error on a qubit alters the phase factor in the qubit's superposition state.

We shall now discuss the quantum analog of the classical error correction theory [Cha06]. Similarly to what we did with repetition codes, we can represent the bit-flip error (generally called *noise*) for qubits thrugh the gate X and associate to it a proability p.
Then, we can introduce redundancy, firstly by mapping the physical basis of qubits to their respective logical counterparts in the following way [Nie06]:

$$\alpha|0\rangle + \beta\,|1\rangle \to \alpha|000\rangle + \beta\,|111\rangle$$

where we have encoded the basis vector as:

$$\begin{aligned} |0\rangle &\to |0\rangle_L \equiv |0\rangle \otimes |0\rangle \otimes |0\rangle \equiv |000\rangle \\ |1\rangle &\to |1\rangle_L \equiv |1\rangle \otimes |1\rangle \otimes |1\rangle \equiv |111\rangle \end{aligned}$$

where L stands for *logical qubit*

If we imagine to send all of the three bits one at a time through a channel and suppose that one or fewer qubits are flipped; we can define a simple quantum error correction scheme by means of detection and analysis of error syndromes, given by projector operators [Nie06]:

1. $P_0 = |000\rangle\langle000| + |111\rangle\langle111|$, which corresponds to the projector for which no error has occurred;
2. $P_1 = |100\rangle\langle100| + |011\rangle\langle011|$, which corresponds to the projector for which an error on the 1st qubit has occurred;
3. $P_2 = |010\rangle\langle010| + |101\rangle\langle101|$, which corresponds to the projector for which an error on the 2nd qubit has occurred;
4. $P_3 = |001\rangle\langle001| + |110\rangle\langle110|$, which corresponds to the projector for which an error on the 3rd qubit has occurred.

In order to detect the syndrome we compute $\langle\psi|P_i|\psi\rangle$ in the decoded phase, which will be equal to 1 for the i-bit that has been flipped. Measuring the syndrome does not affect the state in any way, neither before nor after the measurement. Furthermore, the syndrome gives no information on the state phase. We can recover the initial state with perfect accuracy by flipping again the qubit of interest. Though, we have already mentioned that this scheme works well only if errors occur on one or less of the three qubits.
Similarly to the classical case, the minimum number of bit-flips needed to convert one valid quantum state into another represents the distance of the code. It quantifies the

code's ability to correct errors, with higher distance indicating greater resilience. We can now denote a given quantum code by means of the following three characterizing quantities: [n, k, d], where n represents the number of physical qubits, k denotes the number of logical qubits, and d stands for the code's distance.

In this paragraph, we more formally define the Quantum Error Correctio (QEC) as we did in the classical case according to [Cha06]. Let the quantum information to be a state $|\psi\rangle \in \mathscr{H}_A$. We can introduce ancillary qubits for redundancy. Ancillary qubits are replicas of the information that will be manipulated along with the data that we want to transmit. Ancillary bits are part of an Hilbert space $\mathscr{H}_B$ and are initially in a state $|b\rangle$. When we encode information what we do is mapping the tensor product of $|\psi\rangle$ and $|b\rangle$ in a new Hilbert space $\mathscr{H}_C$ (the code space) through the application of an encoder operator **E**, which should be the equivalent of the generator matrix in classic models. In the same way we can define an Hilbert space $\mathscr{H}_N$ to model the noise; this space more generally represents the noisy environment that introduces errors and is initially in a state $|n\rangle$. As given in [Cha06], the interaction between the information state and the environment is computed as the tensor product:

$$|c'\rangle = |c\rangle \otimes |n\rangle = \textstyle\sum_i (N_i c) \otimes |n_i\rangle$$

with N_i being a noise operator acting on i. When we decode information through the decoding operator **D** we want to map $|c'\rangle$ back to $|\psi\rangle \otimes |s_i\rangle$, which represent the original state $|\psi\rangle$ and its error syndrome $|s_i\rangle$. We can then use the error syndrome to correct $|\psi\rangle$ and get the initial state. However, in order to reproduce the classical case, the error syndrome should be in a bijective relationship with N_i in order to recover the respective errors; furthermore, $|s_i\rangle$ should also be independent from $|\psi\rangle$. Note that all N_i can also be made linearly independent to avoid confounding errors.

The above description of the quantum error behavior actually contains the conditions on error correctability to protect quantum information. This conditions are summed up into the equation given by the following theorem [Nie06]:

Theorem 2.4. *Let $\mathscr{H}_C$ be a quantum code and P the projector onto C. Suppose N is a quantum operation with operation elements $\{N_i\}$. A necessary and sufficient condition for the existence of an error-correction operation $\mathcal{R}$ correcting N on C is that*

$$PN_i^\dagger N_j P = \lambda_{ij} P$$

for some self-adjoint matrix (λ_{ij}), $i, j \in \{1 \ldots n\}$ with $\lambda_{ij} \in \mathbb{C}$ and for all N_i.

The projector is a mathematical operation that projects any quantum state onto the code space. It ensures that any state acted upon by the error correction process is effectively confined to the subspace defined by the error-correcting code C since, when quantum states undergo errors, these errors might cause the state to deviate from the code space. The condition essentially states that the error correction is successful if when applied to the states within C, the corrected state remains within the code space.

We could generally say that it ensures that a noisy quantum state is not *too* noisy. More precisely, the projector P forces the correction operation to act within the space where it provides reliable error-correction. In this sense, P protects against excessive noise. The Ns actually represent noise operators and if such an operation exists we say that $\{N_i\}$ forms a correctable set of errors. This would make the code correctable.

2.2.1. Fault-tolerant computation

Quantum error-correction applications concerns not only the protection of transmitted quantum information from noise but also the protection of quantum information during computations. We start by briefly highlighting two main issues in quantum computation. The first issue is due to the fact that each component of a quantum circuit is subject to noise. What is done in general to make the computation fault-tolerant is to substitute qubits with encoded blocks of qubits by using a code protected from noise and substitute faulty gates with encoded gates also referred to as fault-tolerant procedures. The second issue concerns the fact that when we perform error-correction on encoded qubits we may introduce errors through the error-correction itself.

In the context of fault-tolerant quantum computation, it is possible to carry out a comprehensive range of logical operations on encoded quantum states. This is also done in a manner that results in the effective failure probability within the encoded states, scaling as a function of p, which is the error probability associated with the underlying gates. These errors could affect a single qubit but quantum error-correction codes are designed to correct them. However, in the process of error-correction itself, new errors can be introduced, and these are typically correlated with the original errors. The probability of introducing correlated errors during the error correction process is proportional to the square of the original error probability. This means that the effective failure probability scales quadratically with the probability of errors in the underlying gates.

The above considerations lead us to the main result concerning fault tolerant compuatation: the threshold theorem [Nie06]. Essentially, the theorem states that as long as the noise present in individual quantum gates remains below a specific constant threshold (the probability that each gates fails is $p < p_t$) it becomes feasible to reliably execute a quantum computation of arbitrary length. In fact, if we apply fault-tolerant computations on encoded states, achieving a quadratic reduction of the errors $O(p^2)$, the desired error threshold achieved can be made as low as desired with an increase in the circuit size by means of code concatenation. For further details see [Nie06].

2.3. QEC in the Toric Code

Broadly speaking, errors refer to the unintentional changes or flips in the physical signals that encode the information. These errors can be caused by various factors, such as external environmental noise. In the toric code, or in quantum systems in general, errors can be caused by what are often referred to as local perturbations. Local perturbations involve small disturbances that primarily affect individual qubits or a small region of the quantum system, causing the corruption of the stored quantum information. Single qubit errors can result in different types of errors such as bit-flips errors, phase-flips errors or a combination of them. Mathematically, local perturbations in the toric code framework can be described as operators that act on qubits of the system and can be represented using combinations of Pauli operators. For example, consider the following perturbation:

$$\mathrm{V} = \lambda \sum_j \sigma_j^x$$

such that $\lambda \ll 1$[Kit02]. The ground states of the toric code Hamiltonian are protected from local perturbations of the form that we have described above but only if they are of moderate entity. In order to determine such entity we apply perturbation theory. Given an initial state $|k\rangle$ and a different orthogonal state $|\psi\rangle$ both ground states of the toric code Hamiltonian, we have

$$\langle\psi|V^n|k\rangle = 0. \tag{2.1}$$

where we assumed that the order of the perturbation is less than $\frac{N}{2}$, where N is the length of the side of the lattice. If $n < \frac{N}{2}$, then the effect of the perturbation is of the order of $O(\lambda^{2n})$; for such order the toric code is not subject to local perturbations if their size is much smaller than the size of the torus.

To summarize, we can state that in order not to diverge from the state space of which state $|k\rangle$ is part of, the perturbation should be of the order of $n < \frac{N}{2}$. The interpretation in terms of quantum error correction is that the codewords are maximally distinguishable and therefore not corrupted by noise only for a perturbation of the order of $n < \frac{N}{2}$ (see figure 2.3). Indeed, when it comes to error correction we in principle convert an erroneous state into the one which is closer to it, i.e. into the one with minimum distance from it, because it takes less effort to do so. For instance, if a perturbation is of the order of $\frac{3N}{2}$ it would be more convienient to close the error string into a non-contractible loop rather than erasing it.

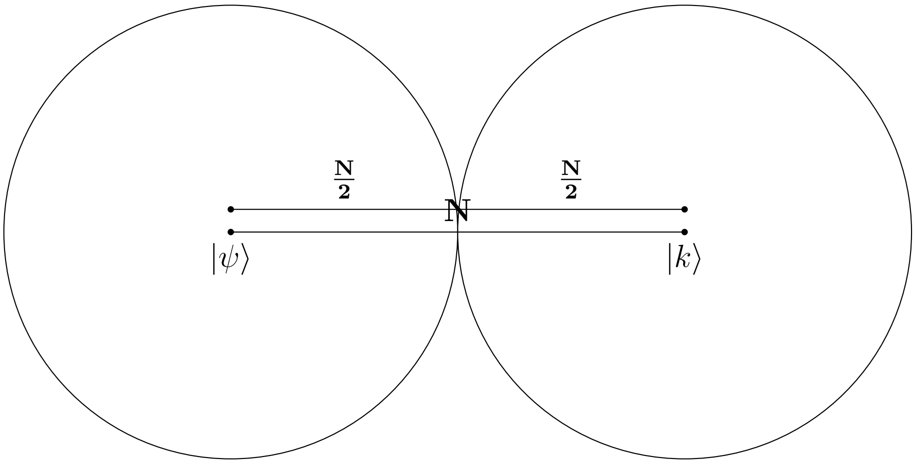

$\frac{\mathbf{N}}{\mathbf{2}}$
$\frac{\mathbf{N}}{\mathbf{2}}$
N
$|\psi\rangle$
$|k\rangle$

Figure 2.3: Consider the two big circles a visual representation of the state spaces of which $|\psi\rangle$ and $|k\rangle$ are part of. The distance between the eigenstates is N representing the length of a non-contactible loop that we need in order to be able to transition between the two states. In order to safeguard the integrity of the two states (no overlap of the circles) we have a margin equal to $\frac{N}{2}$ for each state, which corresponds to the entity of the perturbation that the system can bear.

Thus, for $n < \frac{N}{2}$ we can use the toric code to encode two qubits of information in a large number of physical qubits arranged on the squared lattice in such a way that the logical qubit can be protected from noise. The physical qubits are stored on the spins of the lattice but the logical qubits are encoded in the ground states of the Hamiltonian of the system. The method for encoding information in the toric code lies in the topological nature of the ground states. In fact, instead of directly manipulate individual qubits to store quantum information, we manipulate the entire lattice to encode and protect the logical qubit. The protection is due to the fact that errors generate the anyonic excitations presented in section 1.5. As already discussed, anyons follow specific rules, like braiding around each other, that allow us to protect the quantum information we want to store.

We can reframe all of the above physical information into the usual logical framework by relabelling the orthogonal ground states as logical sates: $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$, in such a way that we are able to identify logical gates into those represented in figure 2.4 labelled as: $\overline{X}_1$,$\overline{X}_2$,$\overline{Z}_1$,$\overline{Z}_2$.

Such gates can be seen as string operators that create pairs of excitations that self-annihilate at the endpoints by winding all the way around the torus. These non-contractible loops play a crucial role in encoding quantum information in the toric code. In fact, the logical states of the two encoded qubits are determined by whether these loops are crossed

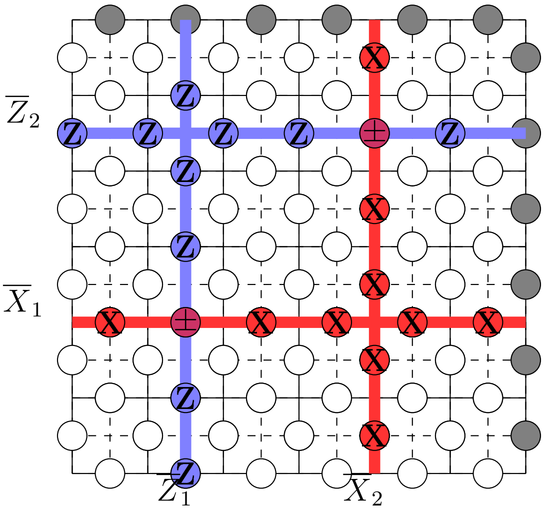


Figure 2.4: Logical gates on the toric code.

an even or odd number of times.
Overall we can determine the action of $\overline{X}_1$ as the logical gate maps $|00\rangle$ to $|10\rangle$, while $\overline{Z}_1$ as the logical gate that maps $|00\rangle$ to $|00\rangle$ and $|10\rangle$ to -$|10\rangle$.
Similarly, the action of $\overline{X}_2$ logical gate maps $|00\rangle$ to $|01\rangle$, while $\overline{Z}_2$ maps $|11\rangle$ to $|11\rangle$ and $|01\rangle$ to -$|01\rangle$. For further details see [Her20].

Errors that occur in the system can disrupt these non-contractible loops. However, the topological nature of the toric code ensures that errors affecting the logical qubits can be detected and corrected by measuring the syndrome associated with these non-contractible loops. By performing appropriate error correction procedures based on the outcomes of these measurements, the encoded quantum information can be protected against errors.

2.3.1. Error detection and correction

We know that the Hilbert space of the ground states of the toric code is composed by the states $|\psi\rangle$ satisfying the following constraints:

$$A_{\vec{v}}|\psi\rangle = |\psi\rangle$$

$$B_{\vec{p}}|\psi\rangle = |\psi\rangle.$$

When errors occur on a given ground state $|\psi\rangle$, the resulting state does not belong anymore to the ground state manifold. This is because, due to errors, there would be some vertex or plauqette operators that yield eigenvalue equal to -1.
By combining simultaneous measurements of vertex and plaquette operators with the

anyonic nature of the excitations, we can find methods to identify and correct errors whenever it is possible. In the following paragraphs, we briefly discuss how errors can be described as anyon pairs and consequently we will identify several error scenarios and study the relative error syndromes and corrective actions that can be applied.

Due to the commutation properties of string operators that we have discussed in the previous sections, we can say that vertex operators are designed to detect an odd number of bit-flips errors that occur in the neighbourhood of the vertex on which they are applied. When an error affects a qubit adjacent to such vertex, it can disrupt the commuting relationship between the Pauli operators making up the vertex operator. As a result, the measurement outcome of the vertex operator may deviate from its expected $+1$ eigenvalue, i.e.

$$A_{\vec{v}}|\phi\rangle = \text{-}|\phi\rangle.$$

This deviation indicates that an error has occurred adjacently to the vertex $\vec{v}$. However, the measurement of the a single vertex operator is not sufficient to detect the exact position of the error. Moreover, an ϵ-excitation is created in the vertex $\vec{v}$. Since excitations are always created in pairs, we can assume another excitation of the same kind is present at the other endpoint of the error string.

In view of the duality between vertex and plaquette operators, plaquette operators are designed to detect an odd number of phase-shift errors that occur within plaquette regions. When an error affects a qubit within such region, it alters the state of those qubits. As a result, the measurement outcome of the plaquette operator gives a -1 eigenvalue. This deviation indicates that an error has occurred within the plaquette region. This error is associated with the creation of an μ-excitation in the application site $\vec{p}$ of the plaquette operator.

More in general, errors on the toric code can be more complex than the simple creation of anyon pairs; though, as mentioned in [Kas19, Bro14], in order to be able to correct them we only need to know how to correct errors generated through σ^x and σ^z operators. Anyon pairs are connected through paths that we have named *error strings*. We are now going to discuss the three main error scenarios that can arise: in the first case, isolated error strings are created, resulting in anyon pairs on the lattice; in the second case, anyons pairs may self-annihilate by forming contractible closed loops, resulting in no error; in the third case, we can get a non-contactible loop and thus creates a logical operation on the quantum state.

In order to better clarify the above concepts, we take as example a bit-flip error that could occur on the lattice and always assume an odd number of errors involved in each vertex operator. There are three possible ways an error could occur, as depicted in figure 2.6:

1. Isolated error link: isolated bit-flip that can be detected through the measurement of the adjacent vertices;
2. Adjacent error links: they are composed of multiple adjacent single erroneous edges. Similary to the case above, we are only able to identify the two endpoints through simultaneous measurements (see figure 2.7);
3. Undetectable errors: these types of errors are not visible because their two endpoints overlap: they indeed have no endpoints. We can divide this latter category of errors in two : the contractible loops, which do not alter the state, and the non-contractible loops, which cause irreversible errors.

Such errors can be detected through simultaneous measurement of vertex (and plaquette) operators of the Hamiltonian (figure 2.7). The fact that the vertex and plaquette operators commute with each other (Proposition 1.4) along with the fact that the states in the ground state manifold are eigenfunctions of the vertex and plaquette operators (Proposition 1.6) and the fact that string operators commute and anticommute with $A_{\vec{v}}$ and $B_{\vec{p}}$ operators (Proposition 1.10, which states that the anyonic excitations are eigenfunctions of vertex and plaquette operators with eigwnvalues +1 or -1) allows us to measure the operators without destroying the initial given state. In practice, it is is posssible to measure, on the same physical state, all the $A_{\vec{v}}$ and $B_{\vec{p}}$ operators withouth modifying the underlying state. The resulting measurements are called *error syndrome*, which gives information about the error strings [Den02]. As a consequence what we do is using algorithms that estimate the way errors formed by connecting pairs of endpoints by selecting the shortest string that connects two -1 eigenvalues yielded by the vertex operators, i.e. we make use of the minium distance between two -1 eigenvalues.

Given a bit-flip error as above, starting from a state $|00\rangle$ where we have no loop, we also determine three different error scenarions for error-correction, as depicted in figure 2.8:

1. First scenario: the algorithm correctly identifies an error string and erases it by applying the inverse of the Pauli operator that caused it;
2. Second scenario: the algorithm connects two endpoints of an error string. Since there are several paths of the same length that can connect two endpoints, it might happen that we fail to correctly select all the edges involved and end up flipping

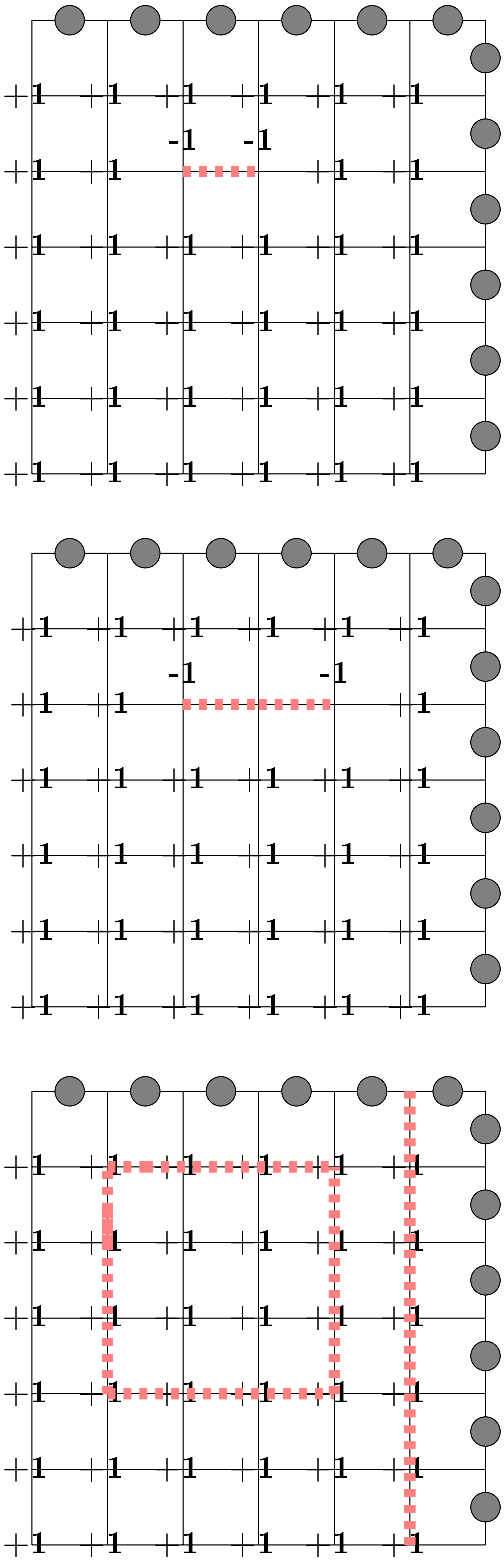

+1 +1 +1 +1 +1 +1
-1 -1
+1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
-1 -1
+1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1

Figure 2.5: Bit-flip error scenarios on the toric code. Error strings made up by σ^x Pauli operators (local bit-flips) are represented in red. To each vertex is assigned a +1 eigenvalue of the associated vertex operator. Whenever an odd numer of erroneous qubits is involved in a vertex operator we assign a -1 to that vertex (see figure 2.7).

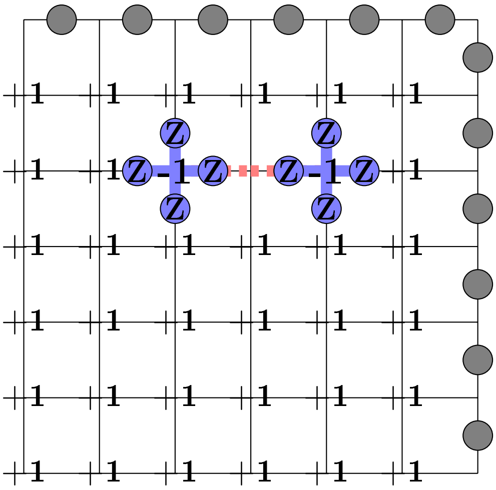


Figure 2.6: Example of error syndrome yielded by the simultaneous measurement of vertex operators after an error string as occurred (red). For the sake of clarity, the dual lattice and the spins over the edges are omitted, as they are not involved in the current matter. The boundary conditions are highlighted in gray. We represent the vertex operators involved in the error string along with their respective -1 eigenvalue, resulting from the anticommutation of $\sigma^z \sigma^x$ for an odd number of erroneous edges.

previously correct ones. This would mean propagating the error on the lattice. Moreover, we can also end up into two situations: we could obtain contractible loops or we could end up having a non-contractible loop, which irremediably damages quantum information since we would not be able to detect the two endpoints anymore due to their overlap.

3. Third scenario: we connect separate open error strings into a single one. Then, the feasibility of erasing the error can be divided in the two cases discussed in the scenario above: we form a contractible closed loop or we end up with a non-contractible loop.

Overall, the error-correction can be said to be effective whenever we end up in scenario 1 or in scenario 2 and 3 but only by forming contractible loops that leave the state unaltered.

As a consequence, the goal is to decrease the probability of uncorrectable errors. There exist two known methods to achieve this goal. The first method involves increasing the size of the torus; by doing so, we increase the number of erroneous edges needed to form a non-contractible loop. Notice that this would mean to increase the Hamming distance of the code.

The second method involves reducing the probability p with which each edge is flipped.

In this way the errors would tend not to propagate far away from where they originated and could thus be detected and corrected in an easier way. When p increases, ambiguity arises due to the fact that it would become more and more challenging to correclty identify paths connecting anyon pairs with acceptable accuracy; thus, error-correction may not succeed [Zbi19].

To summarize, the process of identifying and correcting errors relies on the relationship between the observed syndrome and the underlying errors that occurred during quantum computation. The syndrome, obtained through simultaneous measurements of vertex and plaquette operators, provides crucial information about the presence and location of errors within the quantum code. However, due to the complexity of quantum systems and the nature of errors, the mapping between the observed syndrome and the actual error is not always one-to-one. This lack of a bijective correspondence between syndromes and errors arises because multiple distinct errors can lead to identical syndromes. In other words, these errors are indistinguishable based solely on their syndromes. This ambiguity in error identification represents a significant challenge to error-correction techniques, as it complicates the task of precisely correcting the errors that occurred during the computation. Effectively addressing this challenge requires the development of sophisticated error correction procedures that can effectively handle the degeneracy related to the mapping between syndromes and error. These procedures may involve advanced decoding algorithms and error mitigation techniques.

In conclusion, the understanding of the non-trivial relationship between syndromes and errors highlights the importance of robust error-correction strategies in quantum computing. Addressing this challenge is essential for maintaining the integrity of encoded quantum information and to realize the potential of fault-tolerant quantum computation in the future.

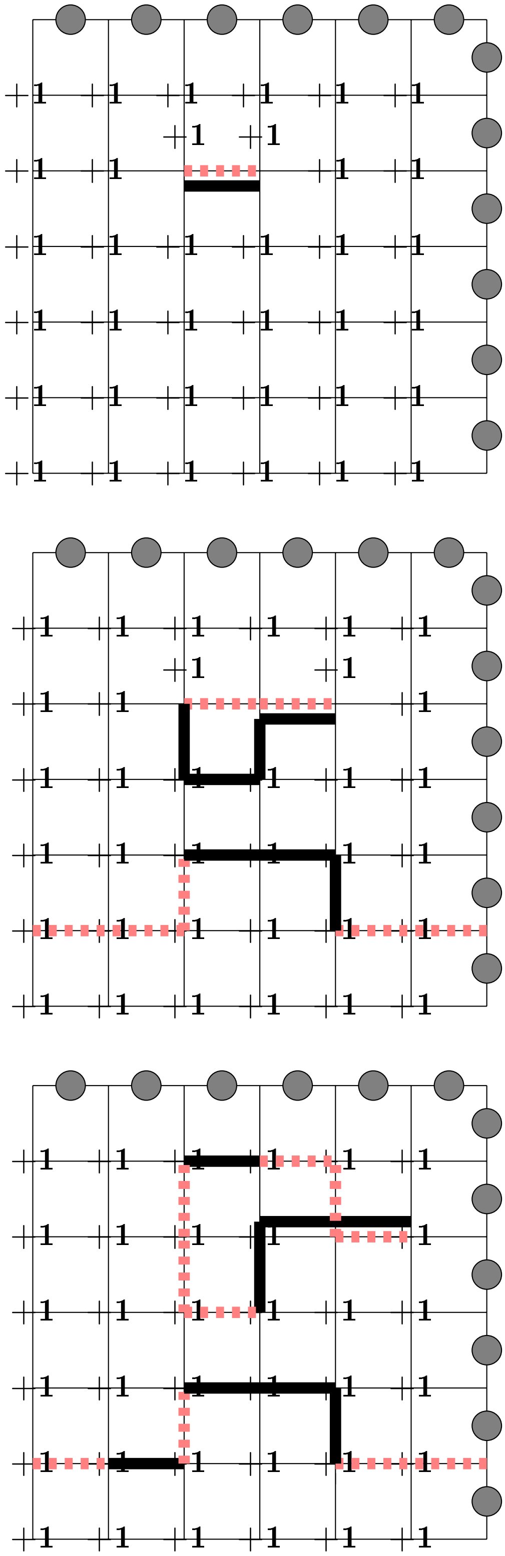
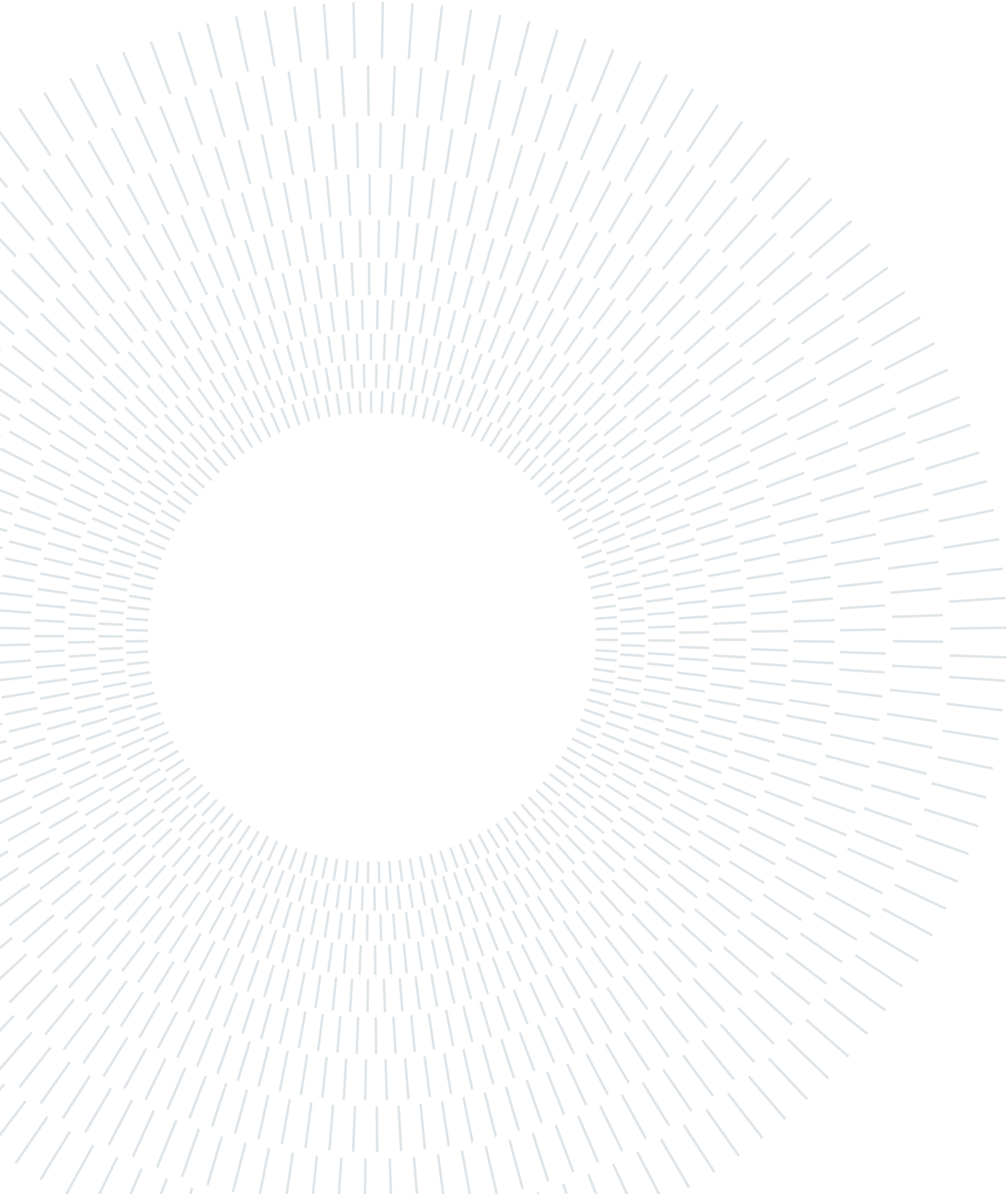

+1 +1 +1 +1 +1 +1
+1 +1
+1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1
+1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1
+1 +1 +1 +1 +1 +1

Figure 2.7: Bit-flip error correction scenarios on the toric code. The correction is shown through black strings connecting anyon pairs in several ways. The correction still consists of Pauli operators of the same type of the error string, i.e. σ^x, which represent the inverse of the bit-flip errors.



Bibliography

[Bro14] D. Browne, *Topological Codes and Computation*, Lecture notes, unpublished, 2014.

[Cor23] M. Correggi, *Aspetti Matematici della Meccanica Quantistica*, Lecture notes, unpublished, 2023.

[Cha06] H.H. Chang, "An Introduction to Error-Correcting Codes: From Classical to Quantum", *arXiv: quant-ph/0602157*, 2006.

[Den02] E. Dennis, A. Kitaev, A. Landhal, J. Preskill, "Topological Quantum Memory", *Journal of Mathematical Physics* , 2002.

[Her20] P. Herringer, "The Toric Code", *Rutgers Physics*, 2020.

[Kas19] M. Kastroyano, *Quantum Error Correction*, Lecture notes, unpublished, 2018-19.

[Kha98] A. Khare, *Fractional statistics and quantum theory*, World Scientific, 1998.

[Kit02] A. Yu. Kitaev, "Fault-toleaarnt quantum computation by anyons", *Annals of Physics*, 2-30, 2002.

[Nie06] M. A. Nielsen, I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2010.

[Odd20] L. Oddis, *PhD Thesis: Two-Anyon Schrödinger Operators*, 2019-2020.

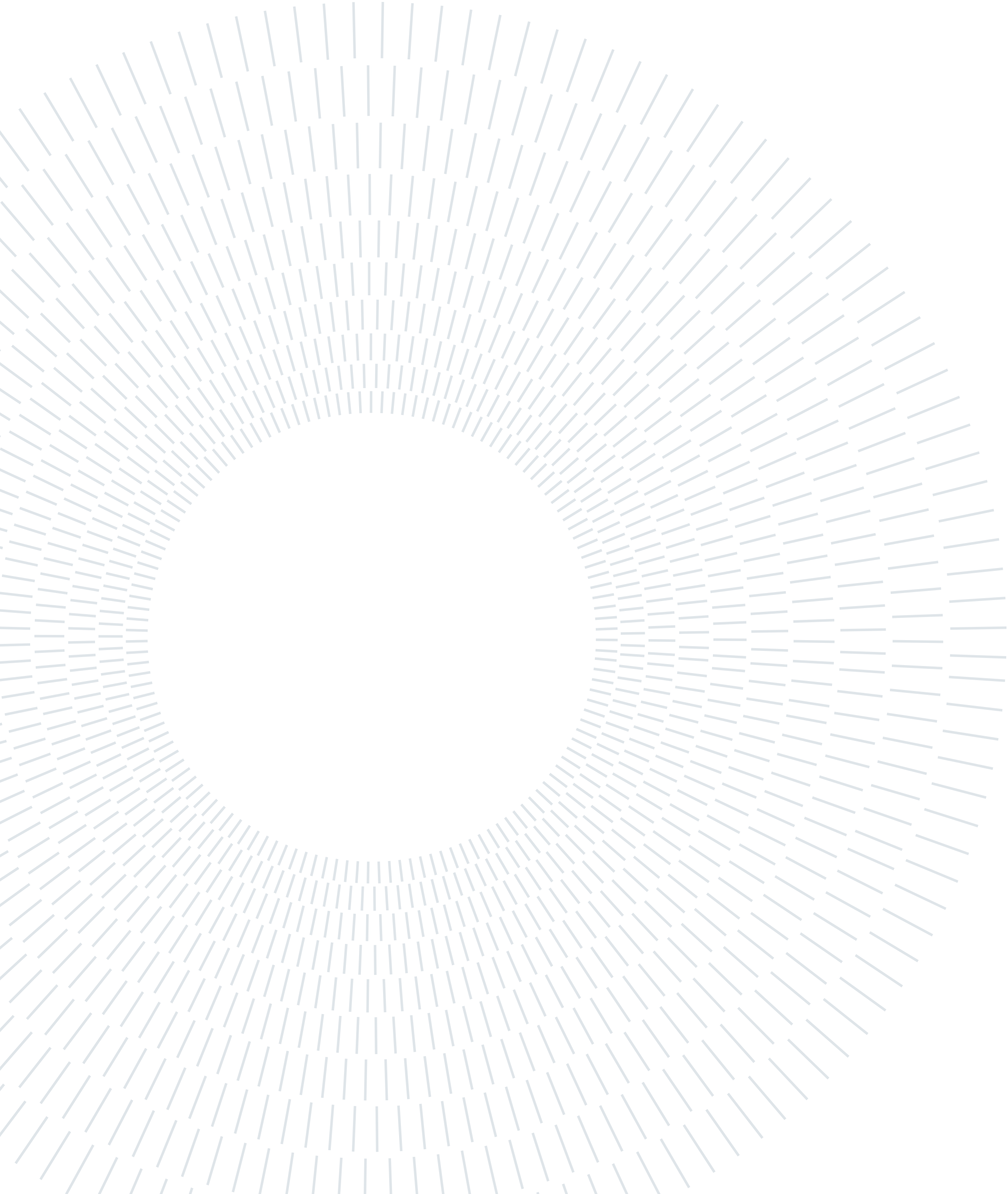
[Pre04] J. Preskill, *Lecture Notes for Physics 219: Quantum Computation*, Lecture notes, unpublished, 2019.

[Rao01] S. Rao, "An Anyon Primer", *Models and Techniques of Statistical Physics*, 2001.

[Rao16] S. Rao, "Introduction to abelian and non-abelian anyons", *Harish-Chandra Research Institute*, 1-20, 2016.

[Wil91] F. Wilczek, "Anyons", *Scientific American*, 58-65, 1991.

[Zbi19] M. Zbinden, *PhD Thesis: Quantum Error Correction for the Toric Code*, 2019.



List of Figures

1.1 Square lattice with boundary conditions. The spins are represented with an empty circle while circles shaded in blue/red represent the boundary conditions. $\hat{e}_1$ and $\hat{e}_2$ are unitary vectors identifying the coordinate system used to identify the sites of the lattice through vectors $\vec{v}$ and $\vec{p}$ with application point in the origin of the coordinate system. In the following sections we are going to indicate the boundary conditions in gray. 8

1.2 In the first picture: vertex (blue) and plaquette (red) operators applied in a vertex and cell of $\mathcal{L}$ respectively. In the second picture: vertex (blue) and plaquette (red) operators applied in a vertex of $\mathcal{L}$ and a vertex of $\mathcal{L}'$, respectively. 9

1.3 Vertex and plaquette operators according to the coordinate system with origin in (0,0). Here the plaquette operator is implemented on $\mathcal{L}'$ in order to coherently define its edges from the point of view of $\mathcal{L}$ using vector $\vec{v}$. . 14

1.4 Operator A_v applied on an open string of occupied edges. 18

1.5 String of occupied edges closed on an operator $A_{\vec{v}}$. 18

1.6 Examples of closed loops yielding +1 eigenvalue for $A_{\vec{v}}$ 19

1.7 Examples of $B_{\vec{p}}$ to a closed loop. 20

1.8 Apply $B_{\vec{p}}$ to a closed loop. 21

1.9 One horizontal circle and one vertical circle passing passing through the spins of the dual lattice. 22

1.10 Graphical representation of ϵ and μ excitations on the ground state. 25

1.11 String operator anticommutes with two $A_{\vec{v}}$ each. 26

1.12 Possible exchanges of two particles in 3D. From: [Rao16]. 29

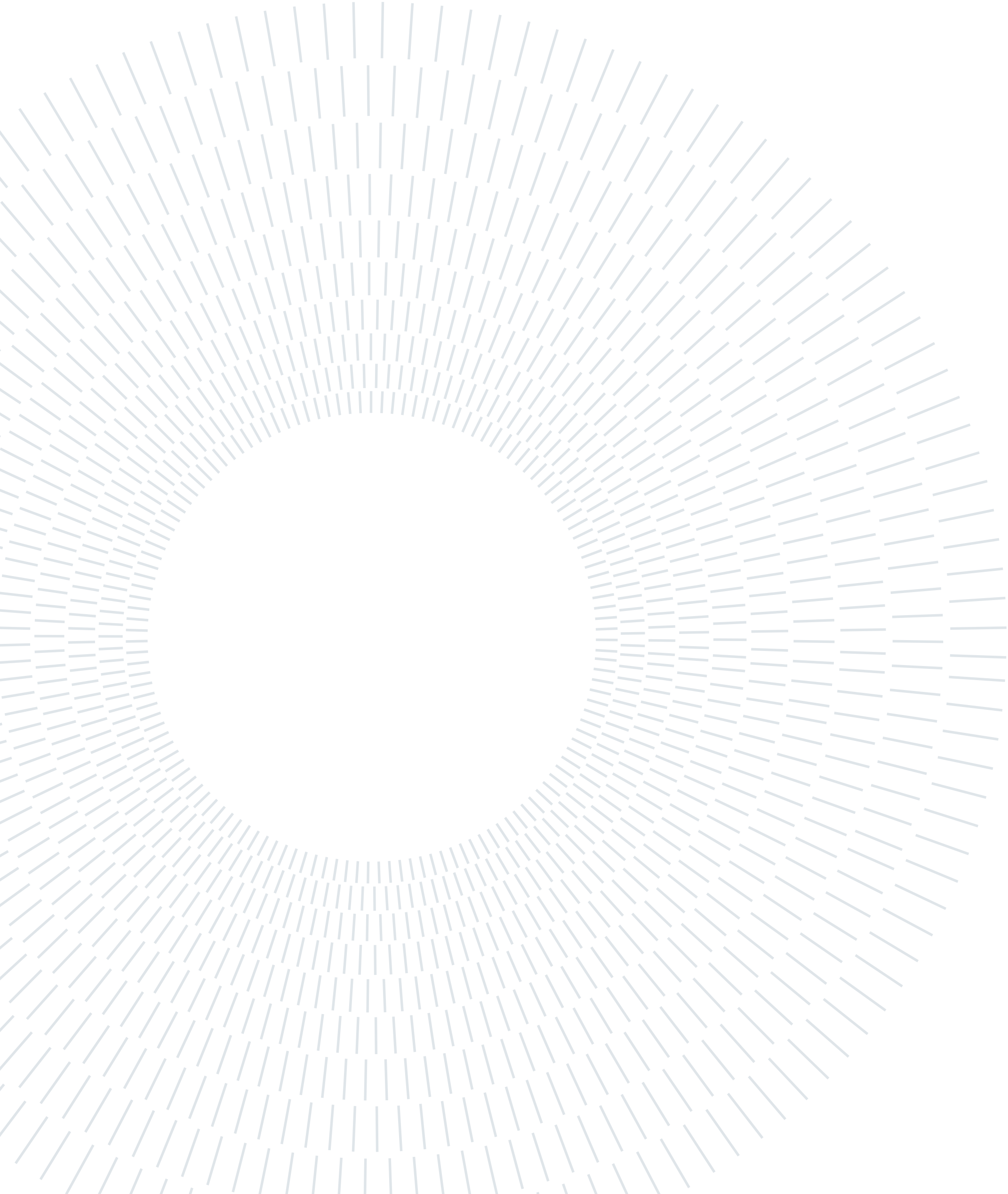
1.13 Possible exchanges of two particles in 2D [Rao16]. We denote diametrically opposite points on the circle (x, empty end black point) to indicate the starting and ending points of a half- exchange of particles along with the three possible paths denote as A,B and C. Case A represents a contractible string, case B represents a non-contractible string obtained through a half-exchange happening between two diametrically opposite points and case C represents a non-contractible string obtained by means of a double half-exchange. 30
1.14 Identical particles exchange position twice as they travel in three dimensions (no exchange). Particle a and b move along a closed path in two dimesions (non contractible) [Wil91]. 31
1.15 In the first picture we show the creation of a pair of ϵ-excitations on the ground state by means of σ^x operator. In the following pictures we move around the excitations using a string operator until we reach the configuration in the third picture. In the last passage we apply a closed loop of σ^x in order to go back to the first kind of configuration [Rao16]. 32
1.16 Creation of the ϵ-excitations and μ-excitations. 33
1.17 Creation of an anyonic excitation by braiding of ϵ and μ pairs of particles. Here we loop the ϵ-excitation around an μ-excitation. The red loop can be removed with the result that the ground state acquires a nontrivial phase. . 34

2.1 Binary symmetric channel [Cha06] 35
2.2 Example of error syndrome associated to a transmitted codeword of length 7. We have two equal error syndromes despite having two different codewords. Each '+' or '-' represents the parity of the bits acting as parents [Kas19] 36
2.3 Consider the two big circles a visual representation of the state spaces of which $|\psi\rangle$ and $|k\rangle$ are part of. The distance between the eigenstates is N representing the length of a non-contactible loop that we need in order to be able to transition between the two states. In order to safeguard the integrity of the two states (no overlap of the circles) we have a margin equal to $\frac{N}{2}$ for each state, which corresponds to the entity of the perturbation that the system can bear. 46
2.4 Logical gates on the toric code. 47

2.5 Bit-flip error scenarios on the toric code. Error strings made up by σ^x Pauli operators (local bit-flips) are represented in red. To each vertex is assigned a +1 eigenvalue of the associated vertex operator. Whenever an odd numer of erroneous qubits is involved in a vertex operator we assign a -1 to that vertex (see figure 2.7). 50

2.6 Example of error syndrome yielded by the simultaneous measurement of vertex operators after an error string as occurred (red). For the sake of clarity, the dual lattice and the spins over the edges are omitted, as they are not involved in the current matter. The boundary conditions are highlighted in gray. We represent the vertex operators involved in the error string along with their respective -1 eigenvalue, resulting from the anticommutation of $\sigma^z\sigma^x$ for an odd number of erroneous edges. 51

2.7 Bit-flip error correction scenarios on the toric code. The correction is shown through black strings connecting anyon pairs in several ways. The correction still consists of Pauli operators of the same type of the error string, i.e. σ^x, which represent the inverse of the bit-flip errors. 53



Acknowledgements

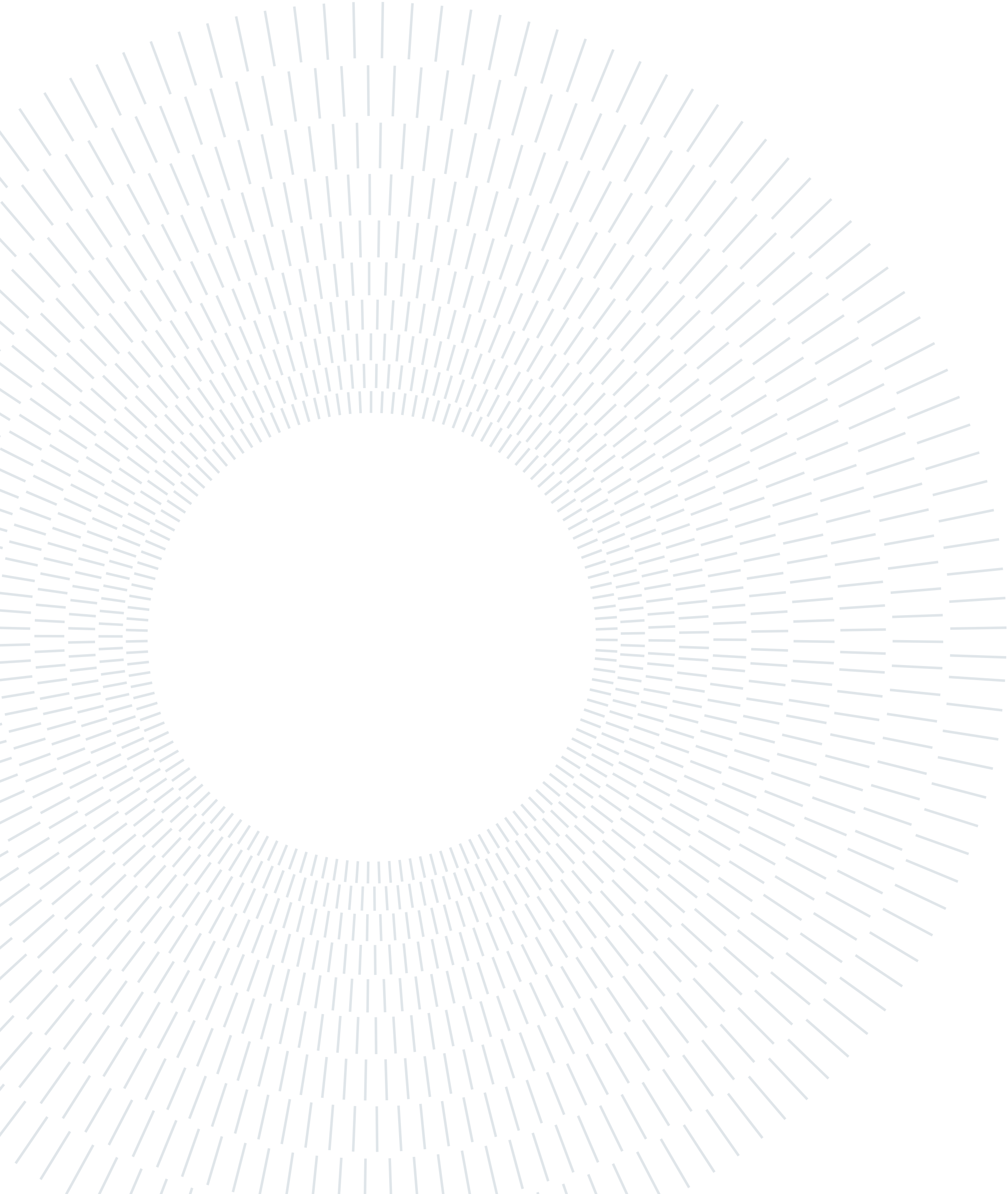
I would like to express my deepest appreciation to my advisor, Prof. Michele Correggi, for giving me not only the opportunity to write my disseration on a topic that in part diverged from my field of study but also for having introduced me to the Principles of Quantum Mechanics, despite the deeply challenging mathematical background that was required to do so. I would also like to sincerely thank my co-advisor, Dr. Massimo Moscolari, who has promptly answered to all my doubts while I was writing my thesis and who revised every piece of notation I mistaken.

I would also like to extend my sincere thanks to my life-long friend Gaia, who was always present to support me both accademically, helping me get through my first year exams, and emotionally when I felt the most undecided and lost throughout my academic journey.

Additionally I would like to mention my classmate Tommaso, without whom these five years would have been way lonlier and boring, especially during the pandemic, and for being a true friend, always present with his sound advice.

I would also like to thank my friends Anna, Ravi and Noemi for being a pillar of support by sharing both their lows and highs, always ready to celebrate our next achievements together.

I do also sincerely thank my mother and my grandmother for always being empathetic with me when, despite being with them, I either kept thinking to my exams or was absent all at once. I am particularly thankful to my mother for all the opportunities that she gave me and that she keeps giving me. I do also want to express my sincere thanks to my older sister Federica who has provided me her support and guidance.

Last but not least, I thank myself for my resilience and resourcefullness, which have always helped me look into the future with a positive outlook despite the so many times I doubted myself.