Holographic Quantum Error Correcting Codes

Adrián Franco Rubio
Trabajo de Fin de Grado en Matemáticas
Universidad de Zaragoza
Perimeter Institute for Theoretical Physics

Supervisor: Daniel Gottesman
Ponente: Concepción Martínez
24 de junio de 2016

Acknowledgements / Agradecimientos

I would like to thank from these lines everyone who has helped me directly and indirectly for the realization of this project. Many thanks to my supervisor, Dr. Daniel Gottesman, for agreeing to supervise a project for a University thousands of kilometres away from Perimeter Institute, and for actively helping me to carry out the project. Thanks as well to Dr. Beni Yoshida for his guidance and for quickly answering my questions even when he was away from the Perimeter. Many thanks to my fellow students who read the draft and made comments and corrections.

Quisiera expresar mi gratitud igualmente a la Facultad de Ciencias de la Universidad de Zaragoza por ponerlo todo a mi favor para que pudiera culminar mis estudios de grado a distancia, incluida la realización de este trabajo de fin de grado. En particular muchísimas gracias a la profesora Concepción Martínez por acceder a ejercer como mi ponente y por tomarse el máximo interés en que mi trabajo saliera adelante en los plazos previstos. Muchas gracias también a mis amigos de la Facultad, sin cuya ayuda las cosas habrían sido mucho más difíciles, y a mi familia, cuyo apoyo constante me ha impulsado durante el curso, como lleva haciendo desde siempre.

Resumen

A la hora de transmitir y procesar información hemos de tener en cuenta que la probabilidad de fallo de nuestros sistemas en el mundo real es distinta de cero, y por tanto existe la posibilidad de que durante dichos procedimientos se introduzcan errores en la información tratada. Para intentar prevenir este problema se desarrollaron los códigos correctores de errores, cuya filosofía es relativamente sencilla: la información que se pretende proteger se codifica en un número de bits mayor que el mínimo necesario, de modo que la consiguiente redundancia permita recuperar la información incluso en el caso de que algunos bits se vean sometidos a errores.

Tras la aparición de la teoría cuántica a principios del siglo XX, numerosos esfuerzos se han dedicado a explotar los principios de la misma para obtener ventajas en el ámbito del procesado de información, dando lugar al nacimiento de la teoría de la información cuántica. Al contrario que en el caso clásico, en el que los bits, unidades mínimas de información, son variables reales que toman los valores 0 y 1; en información cuántica se emplean los qubits o *quantum bits*, que son vectores en un espacio de Hilbert bidimensional, que no sólo permite los valores $|0\rangle$ y $|1\rangle$, los cuales ahora forman una base ortonormal de dicho espacio, sino también combinaciones lineales $\alpha|0\rangle + \beta|1\rangle$ de los mismos. Los axiomas de la mecánica cuántica indican además que el estado de un conjunto de qubits es de nuevo un vector en el espacio producto tensorial de los espacios de Hilbert asociados a cada qubit.

Los sistemas cuánticos son particularmente sensibles a interacciones con su entorno, lo que hace aún más relevante la corrección de errores en este caso. Un código corrector de errores cuántico es una isometría entre espacios de Hilbert que codifican el estado de información de un cierto número de qubits en un número mayor de qubits. La imagen de esta isometría es el llamado *subespacio de código*, que es isomorfo al espacio de estados de los qubits codificados, y cuyas propiedades son de relevancia para comprender cómo de útil es el código. Los códigos estabilizadores son una clase particular de códigos para los que dicho subespacio puede ser caracterizado mediante su estabilizador, un grupo de operadores lineales que deja invariantes todos los elementos del subespacio. Conocer el estabilizador de un código, y en particular un conjunto de generadores del mismo facilita significativamente parte del trabajo con el código. El código básico con el que se construyen todos los que vamos a tratar es el llamado *five qubit code* o código de cinco qubits, que codifica la información de un único qubit lógico (input) en cinco qubits físicos (output). Este código es capaz de preservar la información frente a la desaparición o borrado de dos qubits cualesquiera de los cinco, y también es capaz de corregir un error cualquiera en uno de los qubits sin necesidad de la información de cuál de ellos ha sido alterado.

Además de su uso en el ámbito del procesado cuántico de información, los códigos cuánticos correctores de errores han sido objeto de mucha atención recientemente debido a una propuesta que los conectaría con el principio holográfico y la corerspondencia AdS/CFT, una dualidad existente entre teorías gravitatorias y teorías cuánticas de campos que se está estudiando con el objetivo de entender cómo llegar a una teoría de la gravitación cuántica. En 2015, un equipo de investigadores [1] introdujo los códigos cuánticos correctores de errores holográficos como modelo de juguete para explorar la relación entre corrección de errores y holografía. Estos códigos se definen por medio de un circuito cuántico, o red tensorial, que es una representación gráfica de las operaciones de codificación a las que se someten los qubits lógicos para obtener el resultado, los qubits físicos. En concreto, en los códigos con los que trabajamos la geometría subyacente a este circuito es hiperbólica, y los construiremos a partir de una teselación del disco de Poincaré de símbolo de Schläffi (4,5) que truncaremos para que sea finita. Cada vértice de la red corresponderá a una operación de codificación como la del código de cinco

qubits.

El objetivo principal de este trabajo es el cómputo de los generadores del estabilizador de una familia particular de códigos correctores de errores cuánticos holográficos que llamamos, a falta de una nomenclatura más sistemática, códigos basados en las caras o FBC (face based codes), pues el proceso de truncado de la teselación infinita para dar un circuito finito se realiza prestando atención a las caras de la misma. De esta manera obtenemos una familia de códigos indexada por los números naturales y construida añadiendo sucesivas capas (que incluyen nuevos qubits lógicos y físicos, y nuevas operaciones de codificación) a una red creciente. El método que empleamos está inspirado en [1] y [2] y consiste en construir el estabilizador el código "empujando" hacia la frontera del mismo los estabilizadores de los códigos más pequeños que lo componen. La última es por supuesto una frase muy poco técnica que será reformulada de forma rigurosa en el texto principal del trabajo.

Para determinar el estabilizador de los citados códigos comenzamos centrándonos en el primero de ellos. Con el método empleado podemos generar elementos del estabilizador de manera relativamente sencilla a partir de lo que llamamos *construcciones de estabilizador*. A partir de ellas obtenemos generadores muy localizados que son claramente independendientes pues actúan sobre qubits físicos distintos. También encontramos otros generadores con soportes más amplios y probamos que son independientes entre sí y de los demás en una aplicación sencilla de la teoría de grupos. Puesto que sabemos cuál es el tamaño del estabilizador, encontrar suficientes generadores que se pueden hallar a partir de construcciones de estabilizador nos permite probar que podemos obtener todo el grupo de esta manera.

Pasamos posteriormente a generalizar el procedimiento para cualquiera de estos códigos (cualquier número de capas). Para ello combinamos las dos estrategias que empleamos en el caso de una sola capa. En primer lugar, construimos generadores muy localizados que sólo actúan sobre unos pocos qubits físicos. En segundo lugar aprovechamos la existencia de un homomorfismo entre el estabilizador de un código y el del código con una capa menos para dar una construcción recursiva de los generadores restantes hasta cubrir el total necesario.

El estudio del estabilizador de este tipo de códigos no sólo plantea las ventajas habituales en la caracterización de los mismos como códigos correctores de errores, sino que abre la puerta a analizar fenómenos como la localización de los generadores del estabilizador (el hecho de que actúen de manera no trivial en sólo una fracción reducida de los qubits de la frontera), que podría proveernos de intuición a la hora de entender la relación entre estos modelos de juguete y la correspondencia AdS/CFT. Por otra parte, quedan abiertos problemas como la determinación de las *fault-tolerant gates* o puertas lógicas a prueba de errores, como se denominan aquellas implementaciones a nivel de código de puertas lógicas que optimizan las propiedades de corrección de errores del mismo.

Contents

Chapter 1

Quantum Error Correction

It is the aim of this chapter to introduce the formalism of quantum error correction that will be used in the second part of this work, where the original contributions will be presented. No knowledge of quantum mechanics is assumed from the reader, and hence we will intend to keep everything self-contained. Most of what is explained here can be found in references [3, 4, 5].

1.1 Basic definitions and notation

1.1.1 Quantum states

In quantum mechanics, one of the most fundamental objects we encounter is the **state** of a quantum system. Anything that behaves according to quantum mechanics (such as an electron or a photon, or any collection of them) will be a quantum system that will find itself in a particular state. This state contains all the information about every degree of freedom of the system (such as the position of a particle, or its magnetic moment) and belongs to a set of states that must have a particular structure:

Definition 1.1. The **space of states** of a quantum system is a Hilbert space $(\mathscr{H}, \langle , \rangle)$, whose elements represent the possible physical states in which we can find the system. In particular, if the Hilbert space is two-dimensional, that is $\mathscr{H} \cong \mathbb{C}^2$, the system will be called a **two level system** or a **qubit**.

Throughout this thesis we will use the standard Dirac notation, in which vectors belonging to a space of states $\mathscr{H}$ are written in the so-called *ket*-form: $|\psi\rangle \in \mathscr{H}$. Their associated linear forms in the dual of $\mathscr{H}$ will therefore assume the *bra*-form: $\langle\psi| \in \mathscr{H}^*$, so that the inner product of vectors $|\chi\rangle, |\psi\rangle \in \mathscr{H}$ is given by $\langle\chi|\psi\rangle$, that is the bra $\langle\chi|$ acting on the ket $|\psi\rangle$. Also, in this notation $|\psi\rangle\langle\chi|$ is a linear operator on $\mathscr{H}$ which will map a vector $|\xi\rangle$ to the vector $\langle\chi|\xi\rangle|\psi\rangle$. In particular, this means that:

$$\frac{|\psi\rangle\langle\psi|}{\langle\psi|\psi\rangle}$$

is a rank-one projector on the one-dimensional subspace generated by $|\psi\rangle$.

Depending on what system we are dealing with, the space of states will have different aspect. For example, if we consider a particle which can move in one dimension, its space of states is spanned by the states $\{|x\rangle\}_{x\in\mathbb{R}}$, where x denotes the position of the particle. Clearly this is an infinite dimensional Hilbert space, and requires some knowledge of functional analysis to be dealt with rigorously. But let us go back to qubits. Any physical system that has two fundamentally different possible quantum states, such as an electron's intrinsic angular momentum or *spin* (which can point "up" or "down"), or a photon's polarization axis (horizontal or vertical) realizes the abstract notion of a qubit. Qubits play in quantum information science the same role that bits do in classical information science as the fundamental units of information. Let $\mathscr{H}$ be the Hilbert space of a qubit, then we can choose a pair of orthonormal vectors that will constitute a basis of such space and that we will denote in a very suggestive way:

$$\mathscr{H} = \text{span}\{|0\rangle, |1\rangle\} \qquad \langle 0|1\rangle = 0$$

Indeed, a qubit in the state $|0\rangle$ will be the analogue of a classical bit of value 0, while a qubit in the state $|1\rangle$ will be the analogue of the classical bit having value 1. This is called the **computational basis** The rules of quantum mechanics, nevertheless, allow for more general states of the form $\alpha|0\rangle + \beta|1\rangle \in \mathscr{H}$, for $\alpha, \beta \in \mathbb{C}$. These will in general not correspond to classical bit states, specifically when $\alpha, \beta \neq 0$, in which case we will say that our qubit is in a **superposition**. The following is also standard notation for another orthonormal basis in $\mathscr{H}$:

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \qquad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

The fact that we are considering orthonormal bases goes a little bit further than just mere computational convenience: in quantum mechanics, states which are proportional to each other are considered to be indistinguishable from a physical point of view. In more precise words, the space of *different* states of a quantum system is the space of rays in the associated Hilbert space, i. e., the **projective Hilbert space** $P\mathscr{H}$. In general, to work in the quantum framework we normalize our states, that is, we use a representative of the equivalence class that has norm 1, and that is what we will assume in the rest of this work (otherwise whatever we computed with the theory, such as probabilities of different outcomes of measurements, would need to be renormalized for them to be independent of the representative used).

To be able to perform nontrivial quantum computations we should be able to deal with systems bigger than just one qubit, in the same fashion that classical computers do not carry out calculations using just one bit. Quantum mechanics provides us with a rule of thumb (technically an axiom) as to how to build the space of states of systems composed of smaller subsystems:

Axiom 1.1. *The space of states $\mathscr{H}_{AB}$ of a system composed of subsystems A and B whose respective spaces of states are $\mathscr{H}_A$ and $\mathscr{H}_B$ is the tensor product of these:*

$$\mathscr{H} = \mathscr{H}_A \otimes \mathscr{H}_B$$

Hence, the space of states of a set of n qubits (from now on we will use the notation $\mathscr{H}_n$ to refer to such Hilbert spaces) will be isomorphic to

$$\mathscr{H}_n \cong \underbrace{\mathbb{C}^2 \otimes \ldots \otimes \mathbb{C}^2}_{n} = (\mathbb{C}^2)^{\otimes n} = \mathbb{C}^{2^n}$$

And the computational basis will be given by the corresponding tensor product of the computational bases of the individual qubits, as seen here for the case $n = 2$:

$$|0\rangle \otimes |0\rangle = |0\rangle|0\rangle = |00\rangle \qquad |0\rangle \otimes |1\rangle = |0\rangle|1\rangle = |01\rangle$$
$$|1\rangle \otimes |0\rangle = |1\rangle|0\rangle = |10\rangle \qquad |1\rangle \otimes |1\rangle = |1\rangle|1\rangle = |11\rangle$$

1.1.2 Quantum operations

When it comes to the physical interpretation of quantum mechanics, the inner product structure of the Hilbert space is of the utmost importance. Indeed, the axiom known as the *Born rule*, which we will introduce below, produces probabilities for the result of a measurement in terms of the inner product. As a consequence, the allowed operations[1] that we can perform on a quantum state are those which preserve this structure, usually called **unitary operations**:

Definition 1.2. The **unitary group** $\mathscr{U}(\mathscr{H})$ on the Hilbert space $\mathscr{H}$ is the set of linear operators defined on it $U : \mathscr{H} \longmapsto \mathscr{H}$ which preserve the inner product, that is

$$\langle U\chi | U\psi \rangle = \langle \chi | \psi \rangle \qquad \forall |\psi\rangle, |\chi\rangle \in \mathscr{H} \tag{1.1}$$

[1] To be fair, the most general quantum operation that is allowed by the rules of quantum mechanics is a completely positive, trace preserving linear map or CPTP map, but for the purpose of this essay we can safely restrict ourselves to unitary operations.

In the case of a qubit, the unitary group is the set of 2×2 complex unitary matrices:

$$\mathscr{U}(\mathbb{C}^2) = \mathscr{U}(2,\mathbb{C}) = \{U \in M(2,\mathbb{C}) | U^{-1} = U^\dagger\} \tag{1.2}$$

Of this group we would like to single out a very particular discrete subgroup. Let us start by defining the following maps:

Definition 1.3. The **bit flip** map, denoted X, is the following element of $\mathscr{U}(\mathbb{C}^2)$:

$$X = |1\rangle\langle 0| + |0\rangle\langle 1|$$

The **phase flip** map, denoted Z, is the following element of $\mathscr{U}(\mathbb{C}^2)$:

$$Z = |0\rangle\langle 0| - |1\rangle\langle 1|$$

Notice that X implements a NOT operation: it flips a qubit in the state $|0\rangle$ to the state $|1\rangle$ and viceversa. Z on the other hand changes the relative phase between the $|0\rangle$ and $|1\rangle$ components of the vector it acts on, and thus it is only relevant whenever there is a superposition. The composition of these two maps is usually denoted $Y = iXZ$. Their matrix representations in the computational basis are given by:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{1.3}$$

It is easy to see that these operators anticommute pairwise:

$$XY = -YX = iZ \qquad YZ = -ZY = iX \qquad XZ = -ZX = iY \tag{1.4}$$

X, Y and Z are also Hermitian, which, together with them being unitary implies that they are idempotent

$$X^2 = Y^2 = Z^2 = \mathbb{1} \tag{1.5}$$

Equations (1.4) and (1.5) will be very useful in determining the elements of the Pauli group:

Definition 1.4. The **Pauli group** $\mathscr{G}$ on one qubit is the subgroup of $\mathscr{U}(\mathbb{C}^2)$ generated by i, X and Z, where $i \equiv i\mathbb{1}$ is the imaginary unit, and X and Z are defined as above.

This group is finite and has the following 16 elements:

$$\mathscr{G} = \{\pm\mathbb{1}, \pm X, \pm Y, \pm Z, \pm i\mathbb{1}, \pm iX, \pm iY, \pm iZ\}$$

Remark 1.1. *Any 2×2 unitary U can be written as $U = a\mathbb{1} + bX + cY + dZ$, with $a,b,c,d \in \mathbb{C}$, thus these operators are relevant since they form a (complex) basis for $\mathscr{U}(\mathbb{C}^2)$. Also, $\mathscr{U}(\mathbb{C}^2)$ is a Lie group, whose Lie algebra $\mathfrak{u}(\mathbb{C}^2)$ is be given by the 2×2 Hermitian operators[2], for which a (this time real) basis is again given by $\{\mathbb{1}, X, Y, Z\}$. Their matrix representations are usually called* ***Pauli matrices***

$$\sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

and they satisfy the following commutation relations:

$$[\sigma_j, \sigma_k] = \sum_{l=1}^{3} i\varepsilon_{jkl}\sigma_l$$

where ε_{jlk} takes the value 1 if jlk is an even permutation of 123, -1 if it is an odd permutation, and 0 otherwise. This makes these operators a set of so-called angular momentum operators*, and they therefore play a key role in quantum physics.*

[2] This is true as long as we define the exponential map as

$$\mathfrak{u} \longmapsto \mathscr{U}$$
$$\alpha \longmapsto e^{i\alpha}$$

Alternatively, there is a convention in which the imaginary unit is removed from the exponent. In this convention, we would have the elements of the Lie algebra to be anti-Hermitian operators.

1.1.3 Quantum measurement

Any physical theory that we want to validate must be able to make predictions that can be confirmed or ruled out by experiment. One of the key aspects of quantum mechanics involves precisely the predictions that it makes when it comes to measuring physical quantities. Contrary to the previous conceptions that we had of our Universe, where perfect knowledge of the state of a system guaranteed us perfect predictability of the outcome of any experiment, quantum mechanics turns out to be a **probabilistic** theory: indeed, the result of a measurement is a random variable with a certain probability distribution, rather than a single, deterministic value. But let us first define what it is that we can measure within this framework:

Definition 1.5. An **observable** A of a quantum system is a Hermitian operator[3] defined on the space of states of the system. Its eigenvectors are usually referred to as **eigenstates** of the observable.

Every physical magnitude that we can think of as measurable in a particular system, such as the energy, or the component of angular momentum along a certain direction, will have its associated observable. In our abstract theoretical analysis we will not need, in principle, to worry about the reciprocal: we will consider any observable of the space of linear operators of the Hilbert space of the system as susceptible of being measured in a laboratory, and hence we will seldom refer to the actual physical meaning of the observable. This might in fact depend on the actual physical implementation. For example, in optical implementations of quantum information processing it is usual to have the observable Z represent the polarization of a photon, measured in the horizontal-vertical basis. On the other hand, when using spin 1/2 particles, such as in NMR (nuclear magnetic resonance) quantum information processing, the observable Z is associated to the direction at which the magnetic moment of the particle points, along a given axis (spin 1/2 particles are those whose magnetic moment is quantized to have only two possible values when measured along any axis, hence they act as qubits for their space of states is two-dimensional).

Let us now focus on the actual measurement process. There are two aspects of it that we need to care about: first, the probability distribution of the outcome of the measurement; second, the state update rule, namely, how we determine the state of the system after the measurement has been made[4]. The following can be taken as definitions, or axioms of quantum mechanics. We present them in its finite dimensional version, i.e., we only consider spaces of states of finite dimension for our systems.

Axiom 1.2. *(Born's rule) Let $|\psi\rangle \in \mathscr{H}$ be a state of a particular quantum system, and let A be an observable of this system. Let $\{a_i\}_{i=1}^r$ be the set of different eigenvalues of A, and let $\{P_i\}_{i=1}^r$ denote the set of projectors over the corresponding eigenspaces. For instance, if a_i is an eigenvalue of geometric multiplicity one (we will say that it is **nondegenerate**), and we call $|a_i\rangle$ the corresponding normalized eigenvector:*

$$A|a_i\rangle = a_i|a_i\rangle \qquad P_i = |a_i\rangle\langle a_i|$$

*Then the **outcome** of a measurement of A when the system is in the state $|\psi\rangle$ is a real-valued random variable X_A with a discrete probability distribution given by*

$$\mathscr{P}(X_A = x) = \begin{cases} \langle\psi|P_i|\psi\rangle = \||P_i|\psi\rangle\|^2 & x = a_i \quad i = 1,\ldots,r \\ 0 & \textit{otherwise} \end{cases} \tag{1.6}$$

where $\|\cdot\|$ denotes the norm induced on $\mathscr{H}$ by its inner product structure. Note that if a_i is nondegenerate we can write:

$$\mathscr{P}(X_A = a_i) = |\langle a_i|\psi\rangle|^2 \tag{1.7}$$

[3] Whenever we consider infinite dimensional systems, this definition needs to be revised, but we will not need to worry about such technicalities for this work.

[4] Yes, in quantum mechanics measuring a system causes it to potentially change its state! This is one of the unusual properties of quantum mechanics that underlies the so-called *measurement problem*

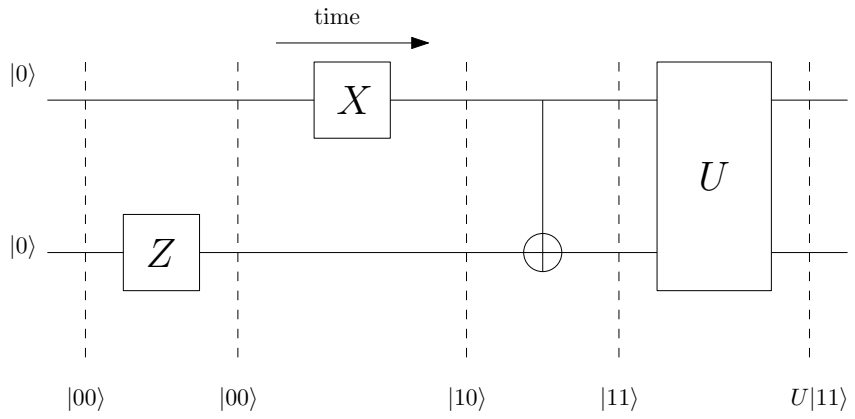


Figure 1.1: A simple example of a quantum circuit. Below, the state of the system after each step of computation.

Note that since A is an observable, we know its eigenvalues are real, and hence the outcome is well-defined. From the interpretation of these eigenvalues as the possible outcomes of, say, a measurement of the energy of a system, it is clear why we impose that these eigenvalues be real.

Axiom 1.3. *(State update rule) Let $|\psi\rangle$ and A be as above. Upon a measurement of A, the state $|\psi\rangle$ is updated by action of the projection operator P_i associated to the eigenvalue observed in the measurement. Hence, after a measurement where the outcome turns out to be a_i, the state of the system will be given by the following (normalized) vector in $\mathscr{H}$:*

$$\frac{P_i|\psi\rangle}{\|P_i|\psi\rangle\|} = \frac{P_i|\psi\rangle}{\sqrt{\mathscr{P}(X_A)}}$$

1.1.4 Quantum circuits and tensor networks

In quantum computation it is useful to employ some diagrammatical notation to represent quantum operations applied to qubits. Figure 1.1 represents a simple example of a quantum circuit. Each line represents a particular qubit, and the quantum operations applied on them are generally drawn as squares over these lines. Wheneverfone of these squares does not overlap with a particular qubit's line, it is interpreted as the corresponding quantum operation acting trivially on said qubit, that is, the unitary is a tensor product of the identity, acting on that qubit, and a less trivial unitary acting on the rest of qubits. In this context, these quantum operations are usually called **quantum gates**, due to the analogy with logical gates in computer science. Certain gates have special notations due to them being particularly important or widespread. It is the case of the CNOT (controlled-NOT) gate which is depicted in figure 1.1, and which is defined as:

$$U_{\text{CNOT}}(|x_1\rangle|x_2\rangle) = |x_1\rangle|x_1 \oplus x_2\rangle$$

where $\oplus$ denotes the boolean sum of bit, which can also be understood as the sum over the field $\mathbb{Z}_2$. Note that this operator flips the second qubit if the first one is in the state $|1\rangle$ and does not do anything otherwise. Two qubit unitaries, in general, can introduce entanglement (quantum correlations) among the two qubits, as can be seen if we apply a CNOT to the following product state:

$$U_{\text{CNOT}}|+0\rangle = U_{\text{CNOT}}\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}|0\rangle\right) = \left(\frac{U_{\text{CNOT}}|00\rangle + U_{\text{CNOT}}|10\rangle}{\sqrt{2}}\right) = \frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)$$

A slightly more general version of a quantum circuit is given by a tensor network. There are many ways in which one can define a tensor, and in particular it can be done intrinsically, but here we will deem more practical to just speak of a tensor, once bases are fixed, as a multi-indexed array of complex numbers that encode a multilinear map (basically the generalization of a matrix):

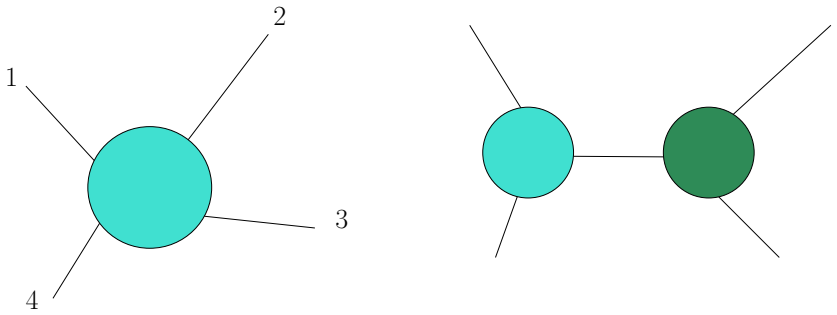


Figure 1.2: Left: pictorial depiction of a tensor. Note that given out of context, this tensor can be interpreted as a map from Hilbert spaces $\mathscr{H}_1 \otimes \mathscr{H}_2$ to $\mathscr{H}_3 \otimes \mathscr{H}_4$, $\mathscr{H}_3$ to $\mathscr{H}_1 \otimes \mathscr{H}_2 \otimes \mathscr{H}_4$,..., even as a map from $\mathbb{C}$ to $\mathscr{H}_1 \otimes \mathscr{H}_2 \otimes \mathscr{H}_3 \otimes \mathscr{H}_4$, which is equivalent to choosing a particular state (that of norm 1 in the image subspace of the said map), in the total Hilbert space. Right: a minimal example of a tensor network representing the contraction of two three-legged tensors to give a four-legged tensor.

Definition 1.6. Let $H_1, \ldots, H_r, H_{r+1}, \ldots H_n$ be Hilbert spaces of dimensions $d_1, \ldots, d_r, d_{r+1}, \ldots, d_n$ and choose respective bases $\{b_j^1\}_{j=1,\ldots,d_1}, \ldots, \{b_j^n\}_{j=1,\ldots,d_n}$ for each of them. A **tensor** $T_{a_1 a_2 \ldots a_n}$ $(1 \leq a_i \leq d_i)$ is a multi-indexed array of complex numbers that encode a multilinear map $T : H_1 \otimes \ldots \otimes H_r \longrightarrow H_{r+1} \otimes \ldots \otimes H_n$

$$T|b_{j_1}^1 b_{j_2}^2 \ldots b_{j_r}^r\rangle = \sum_{\substack{j_i=1,\ldots,d_i \\ r+1 \leq i \leq n}} T_{j_1 j_2 \ldots j_n} |b_{j_{r+1}}^{r+1} b_{j_{r+2}}^{r+2} \ldots b_{j_n}^n\rangle \tag{1.8}$$

for any choice of $j_i, 1 \leq i \leq r$.

From now on we will assume that all of the Hilbert spaces involved correspond to qubits. Tensors are very easily depicted as more or less blob-like bodys with one leg per index / Hilbert space (see figure 1.2). This allows us to represent tensor contractions (the generalization of matrix products) easily:

Definition 1.7. Let T_{ab}, S_{cd} be tensors (we assume them to have two indices only for notational simplicity, the definition is general), such that they have indices, say b and c associated to the same basis of the same Hilbert space of dimension D. Then the **contraction** of T and S along that index is another tensor U_{ad} given by

$$U_{ad} = \sum_{i=1}^{D} T_{ai} S_{id}$$

Definition 1.8. A **tensor network** is a tensor resulting of the contraction of other tensors placed on the vertices of a graph, so that the edges of said graph represent tensor contractions.

As it is explained in [1], tensors can be view as maps from the Hilbert space associated to any subset of their legs to the Hilbert space associated to the remaining legs. The tensors we will be using satisfy a particular property:

Definition 1.9. A tensor with an even number of indices of the same dimension $T_{a_1 \ldots a_{2n}}$ is **perfect** if for any partition of its set of indices/legs $\{A, A^c\}$ with $|A| \leq |A^c|$, T is proportional to an *isometric* tensor from A to A^c, i.e., a tensor that codifies an isometry from the Hilbert space associated to A to the Hilbert space associated to A^c.

1.2 Quantum Error Correcting Codes

The theory of quantum information provides us with many applications that exploit the singular features of quantum mechanics to improve our information treatment capabilities. Nevertheless, when it comes to putting these ideas into practice, a new difficulty shows up: systems that exhibit quantum behaviour

are extremely delicate. They are vulnerable to noise and the influence of the environment they are placed in may lead to very quick **decoherence** (basically defined, without being very precise here, as the loss by a system of its quantum properties, thus rendering it useless for the purpose of quantum information processing).

To solve this problem and hence be able to implement quantum algorithms in physical devices, the ideas of **quantum error correction** and **fault-tolerant quantum computation** were developed. The principle of quantum error correction codes is very simple, and can be understood without recurring to any quantum formalism, as the very widespread example that we now present shows [3]. Assume that Alice has to send a message to Bob consisting of a single bit, 0 or 1. Unfortunately, the channel they possess to communicate is noisy, and bits sent through it have a certain (hopefully small) probability of being "flipped" from 0 to 1 or 1 to 0. She fears that her message might be totally changed, so she agrees with Bob that she will send three copies of her bit to Bob, with the hope that at least two of them will reach him unchanged. That way, even if one of the bits is flipped, Bob will still be able to retrieve the message by taking the value that is repeated a majority of times. This simple example contains all the main steps of quantum error correction protocols: first, Alice **encodes** her message (called the **logical** bit) in a series of additional bits (called the **physical** bits, since these are in principle the only ones that we need to realize physically). Then she sends the physical bits through the noisy channel, which flips some of them. These bits now get to Bob, who performs a procedure of **error detection** (he notices one of the bits disagrees with the others, and assumes that a bit-flip error has occurred to this bit) and **correction** (he dismisses the conflicting bit's minority report and turns it back into the majoritary value). Finally, Bob inverts the coding operation and hence **decodes** the message from Alice.

$$\begin{array}{ccccccccc} & \textit{Encoding} & & \textit{Noise} & & \textit{Correction} & & \textit{Decoding} & \\ 0 & \longrightarrow & 000 & \longrightarrow & 001 & \longrightarrow & 000 & \longrightarrow & 0 \end{array}$$

This method, of course, is not infalible. In a very unlucky case, two or more of the bits sent by Alice could be affected by errors, and hence Bob could fail in his correction procedure, and get the wrong message our of the whole process. Nevertheless, the probability of this happening is, assuming each bit is flipped independently with probability p:

$$\mathscr{P}(\text{“Two or more flips”}) = 3p^2(1-p) + p^3$$

which is strictly smaller than p for $p \in (0, 1/2)$, thus providing an improvement on the scenario without coding.

Suppose Alice now wants to transmit a qubit to Bob. She might decide to encode it in a very similar fashion:

$$|0\rangle \longrightarrow |000\rangle \qquad\qquad |1\rangle \longrightarrow |111\rangle$$

However, now our spectrum of possibilities is greatly enriched. For example, Alice might want to send not a state of the computational basis, but a superposition like $\alpha|0\rangle + \beta|1\rangle$, which she will encode, by linearity of the encoding map (see definition below) as $\alpha|000\rangle + \beta|111\rangle$. Also the spectrum of possible errors that affect our qubits is increased. The noisy channel could, for instance, perform a Z operation on one of the qubits, resulting in Bob getting the state $\alpha|000\rangle - \beta|111\rangle$ and decoding it to $\alpha|0\rangle - \beta|1\rangle$ without being able to spot the error. Thus, the same idea of error correction can be applied in the quantum case to render quantum information processing feasible, but must be improved upon for it to give us better protection against errors, all depending on the resources that we have available (for example, quantum mechanics gives us entanglement (quantum correlations) as a resource to improve our capabilities, and in fact, contrary to our very simple example, must encoded states are usually highly entangled).

Let us then now start by defining what we mean by a quantum error correcting code:

Definition 1.10. Let $k \leq n$ be natural numbers, and $\mathscr{H}_L \cong \mathscr{H}_k, \mathscr{H}_P \cong \mathscr{H}_n$ be the state spaces associated to k and n qubits respectively. A $[[n,k]]$-**quantum error correcting code** (or **QECC** for short) is an isometric linear map $\Phi : \mathscr{H}_L \longmapsto \mathscr{H}_P$. The spaces $\mathscr{H}_L$ and $\mathscr{H}_P$ will be called **logical space** and **physical**

space respectively, whereas the image of Φ, denoted by $\mathscr{C} = \mathrm{Im}\Phi = \Phi(\mathscr{H}_L) \subset \mathscr{H}_P$ will be called the **code subspace**.

One of the main ideas behind error correcting codes is that states outside the code subspace are not valid codewords, and hence when detected they provide evidence for the presence of an error that can then be identified and corrected.

1.2.1 Error models

It is impossible to find a code that will protect our information against any kind of errors. Indeed, consider an error $\mathscr{E}$ consistent on the total erasure of the information of the physical qubits:

$$\mathscr{E}(|\psi\rangle_{\text{phys}}) = |00\ldots0\rangle \qquad \forall |\psi\rangle_{\text{phys}}$$

This map is not invertible, and hence once it happens it is impossible to retrieve any of the original information. This is an extreme example, but it tells us that our best hope is to build codes that are effective against a particular set of errors and hope that the probability of any other perturbation occurring is negligible. The errors that the code can deal with are called **correctable errors**. We can characterize the correctable errors by the following property:

Theorem 1.1. *Let $\{E_i\}_{i\in\mathscr{I}}$ be a set of error operators acting on the physical Hilbert space $\mathscr{H}_P$ of a QECC Φ with code subspace $\mathscr{C}$. Then*

$$PE_iE_j^\dagger P = \alpha_{ij}P \qquad \forall i,j \in \mathscr{I} \tag{1.9}$$

where P is the projector onto $\mathscr{C}$, is a necessary and sufficient condition for any error $\mathscr{E}$ that is a linear combination of elements in $\{E_i\}_{i\in\mathscr{I}}$ to be correctable, that is, for the existence of an algorithmic procedure that restores the system to its original state.

Proof. See [3]. □

Error models that assume independence of the errors on different physical qubits are quite typical. Suppose that we choose our set $\{E_i\}$ to be given by the Pauli operators on the j-th physical qubit $\{\mathbb{1}, X_j, Y_j, Z_j\}$. This operators span the space of unitaries that act on the j-th qubit, hence if the relation (1.9) is satisfied for these Pauli operators, the code allows to correct any single-qubit error acting on the j-th qubit.

1.2.2 Encoded operations

In the same way as when we perform classical computation, we not only would like to store the information represented by our qubits, but we would also like to process it. This involves applying linear operations or gates to the logical qubits, such as bit flips X or controlled nots U_{cNOT}. It would be, however, against the philosophy of the whole quantum error correction scheme if we had to decode the state, apply the corresponding gate, and reencode the state every time that we wanted to manipulate the information. Physically, this would mean that during some time, while we are applying the gate, the information is in its original form and hence unprotected against errors. Back to our original example, suppose that Bob wants to perform a bit flip (logical negation) on the message he received from Alice before sending it to a third party, Charlie. In our non-ideal world, his bit flip operation might fail with some nonvanishing probability $p < 1/2$. If he first decoded Alice's message, then applied the gate and reencoded the message, the whole procedure would produce the wrong output with probability p. On the other hand, Bob might apply the bit flip to each of the three bits the previous error correction procedure outputted, without decoding. This operation we call **encoded bit flip**. Assume one of the three independent bit flips fails, then the output of the operation will be something like 101, which is correctable to 111. Only when two or more bit flips fail does the whole scheme fail, and this again happens with probability $3p^2(1-p) + p^3 < p$.

Definition 1.11. Let Φ be a quantum error correcting code with $\mathrm{Im}\Phi = \mathscr{C}$, and $U : \mathscr{H}_L \longmapsto \mathscr{H}_L$ a quantum operation on the logical qubits. We say that $V : \mathscr{C} \longmapsto \mathscr{C}$ is an **encoded** version of U if the following diagram

$$\begin{array}{ccc} \mathscr{H}_L & \xrightarrow{U} & \mathscr{H}_L \\ \downarrow{\scriptstyle \Phi} & & \downarrow{\scriptstyle \Phi} \\ \mathscr{C} & \xrightarrow{V} & \mathscr{C} \end{array}$$

commutes.

Note that in general the encoded version of a logical gate will not be unique. An active area of research is, given a particular code, the study of how to implement quantum gates **fault-tolerantly**. The concept of fault-tolerance is quite broad, and a precise definition depends on the particular setting, but in general fault-tolerant gates are those which optimize the protection against errors. For example, if our error models mainly assume that errors affect single qubits, it is a good idea to choose encoded gates to be **transversal**, that is, acting independently on each qubit (analogously to the encoded bit flip from the example, which was applied to every qubit independently of the others).

1.3 Stabilizer codes

In this section, we will introduce a particular family of QECC, called **stabilizer codes**. These codes present a series of advantages bLet's begin by giving some definitions.

Definition 1.12. The **Pauli group on n qubits** $\mathscr{G}_n$ is given by the tensor product[5] of the Pauli groups acting on each particular qubit, i.e. it is the tensor product of n copies of the Pauli group.

Lemma 1.1. *The Pauli group on n qubits is generated by*

$$\{i, X_1, Z_1, X_2, Z_2, \ldots, X_n, Z_n\}$$

where i is the imaginary unit:

$$i \equiv i\underbrace{\mathbb{1} \otimes \mathbb{1} \otimes \ldots \otimes \mathbb{1}}_{n}$$

and X_j, Z_j are the corresponding Pauli operators on the j-th qubit:

$$X_j \equiv \mathbb{1} \otimes \ldots \otimes \mathbb{1} \otimes \underbrace{X}_{j} \otimes \mathbb{1} \ldots \otimes \mathbb{1}$$

Proof. It follows from the definitions of Pauli group on one and n qubits. □

Definition 1.13. The **weight** of an element of the Pauli group is the number of qubits on which it acts nontrivially (i.e. with X, Y or Z rather than $\mathbb{1}$.).

The Pauli group on n qubits is a subgroup of the group of unitary operators on $\mathscr{H}_n$ qubits, and therefore it acts naturally on this space.

Definition 1.14. Let Φ be a $[[n,k]]$ quantum error correcting code of code subspace $\mathscr{C}$, and assume there exists $S \leq \mathscr{G}_n$ such that

$$\mathscr{C} = \{|x\rangle \in \mathscr{H}_n \,|\, M|x\rangle = |x\rangle, \forall M \in S\} \tag{1.10}$$

Then Φ will be called a **stabilizer code** and S will be called the **stabilizer** of the code.

Lemma 1.2. *The stabilizer of a $[[n,k]]$ QECC has $n-k$ generators.*

[5] Before any reader gets confused, let us specify that this tensor product of groups is a particular, physicist notation for the direct (Cartesian) product of groups, inherited in some way from the fact that the direct product of Pauli groups is going to act on the tensor product of the Hilbert spaces of the qubits.

Proof. We adapt the proof from [3]. By the definition of the Pauli group, we know that all of its elements square to either $\mathbb{1}^{\otimes n}$ or $-\mathbb{1}^{\otimes n}$. In the first case, their minimal polinomial is x^2-1, hence their eigenvalues are either 1 or -1. In the second case, the minimal polynomial is x^2+1 and the eigenvalues are i or $-i$. If we want our code subspace to be nontrivial, these are not acceptable elements of the stabilizer, so every element of the stabilizer is diagonalizable with eigenvalues ± 1. Given a set of independent, commuting generators $g_1,\dots,g_r \in \mathscr{G}_n$ we can build a projector on their common $+1$-eigenspace, $\mathscr{C}$:

$$P = \frac{1}{2^r}\prod_{i=1}^{r}(\mathbb{1}+g_i)$$

Indeed, adding the identity and dividing by two shifts the eigenvalues from +1,-1 to +1,0 while preserving the eigenspaces. Now, for tensor products of operators it holds:

$$\mathrm{tr}(A\otimes B) = \mathrm{tr}(A)\mathrm{tr}(B)$$

Thus every nontrivial element of the stabilizer has trace 0, for X,Y,Z have trace 0. If we take the trace at both sides of (1.3), having first expanded the product on the right hand side, we see than only the term that arises from the product of all the identities has nonvanishing trace (equal to 2^n, the dimension of the operators). Hence

$$\mathrm{tr}P = \dim\mathscr{C} = 2^k = \frac{2^n}{2^r} \implies r = n-k$$

□

Knowing a set of generators of the stabilizer of a stabilizer code can be very useful when applying quantum error correction with errors belonging to the Pauli group on the physical qubits, or linear combinations of these. Of course, not all operations on the physical qubits will be correctable. Indeed, we also want the freedom to perform encoded operations, which by definition will be operations that, while preserving the code subspace, will not fix every one of its elements, as happens with the stabilizer elements.

Lemma 1.3. *Every pair of elements in the Pauli group over n qubits either commute or anticommute.*

Proof. We know that every element in the Pauli group squares to $\pm\mathbb{1}^{\otimes n}$. Hence, for $a,b\in\mathscr{G}_n$:

$$ab = \sigma(bb)ab(aa) = \sigma b(baba)a = \sigma\sigma' ba$$

where $\sigma,\sigma' \in \{+1,-1\}$. □

Proposition 1.1. *The subgroup of $\mathscr{G}_n$ that preserves $\mathscr{C}$ is the centralizer of S in $\mathscr{G}_n$, $Z(S)$, which also coincides with the normalizer of S in $\mathscr{G}_n$, $N(S)$.*

Proof. Let $g\in Z(S)$, $s\in S$ and $|\psi\rangle\in\mathscr{C}$. The centralizer of S is the subgroup of all the elements of $\mathscr{G}_n$ that commute with every element in S. Hence we have

$$sg|\psi\rangle = gs|\psi\rangle = g|\psi\rangle \forall s\in S \implies g\mathscr{C}\subset\mathscr{C}$$

Conversely, if $g\mathscr{C}\subset\mathscr{C}$, we have

$$sg|\psi\rangle = g|\psi\rangle = gs|\psi\rangle \quad \forall s\in S \implies [g,S]=0 \implies g\in Z(S)$$

where we have used the previous lemma. The normalizer of S is defined as the subgroup of the elements that leave S invariant under conjugation:

$$N(S) = \{g\in\mathscr{G}_n \,|\, gsg^{-1}\in S \quad \forall s\in S\}$$

Clearly $Z(S)\subset N(S)$ but also $N(S)\subset Z(S)$ since every pair of elements of $\mathscr{G}_n$ either commute or anticommute, and for $g\in N(S)$ anticommutation with any $s\in S$ is forbidden, since it would lead to

$$gsg^{-1} = -sgg^{-1} = -s\in S \implies -\mathbb{1}\in S$$

which is a contradiction because $-\mathbb{1}$ can never be in a stabilizer (it does not have eigenstates of eigenvalue +1). □

Hence in the Pauli group we find three kinds of operators: those in S, which leave every codeword invariant; those in $N(S)\backslash S$, which preserve the codespace as a whole but act nontrivially on it, and those in $\mathscr{G}_n\backslash N(S)$, which do not preserve the code subspace. This is due to them anticommuting with at least one of the generators g_i of the stabilizer (remember that in the Pauli group any two elements commute or anticommute). This makes them susceptible of being identified, in the following way. In every error detection step, we measure the observables $g_1,\dots,g_{n-k}$. Since all these observables have spectrum $\{+1,-1\}$, we will obtain an $n-k$ -tuple of ±1's which is called a **syndrome**. If the state $|\psi\rangle$ of the system belongs to $\mathscr{C}$, it is an eigenstate of eigenvalue +1 of all of them, hence the outcome of all of these measurements will be a string of +1's. Now suppose that an error $E\notin N(S)$ occurs. E anticommutes with at least one generator, say g_j. Then $E|\psi\rangle$ is an eigenstate of eigenvalue -1 of g_j, since

$$g_jE|\psi\rangle=-Eg_j|\psi\rangle=-E|\psi\rangle$$

and the outcome of this particular measurement will be -1 with total certainty. Thus, a syndrome where some of the outcomes are -1 signals that an error outside the normalizer has occurred, taking the state outside of the code subsystem. If the correspondence between syndrome and error is unique, we will be able to deduce which error happened and correct it. It could also be the case that the state became a superposition of the correct state and the one affected by the error: $\frac{1}{\sqrt{2}}(|\psi\rangle+E|\psi\rangle)$. In this case, the state update rule tells us that, after measuring g_j, the new state will be the projection of the former state on the corresponding eigenspace of g_j: if we get $+1$, the new state is $|\psi\rangle$ and we have involuntarily already corrected the error; if we get -1, the new state is $E|\psi\rangle$ but we know there is an error, so we proceed as before. Note that errors belonging to $N(S)$ cannot be detected (therefore also not corrected) in this way.

Definition 1.15. The **distance** of a stabilizer QECC is the minimal weight of the elements in $N(S)\backslash S$. It is sometimes included in the double square bracket notation $[[n,k,d]]$ for the code.

Proposition 1.2. *A stabilizer QECC of distance $2t+1$ can correct up to t single qubit errors.*

Proof. See for example [5]. □

1.3.1 The five qubit code

The five qubit code is going to be the basic building block of the holographic codes we will be studying in the second part of this essay, so it is worth introducing it now to become familiar with it. The five qubit code is a $[[5,1,3]]$ QECC, that is, it encodes one logical qubit in five physical qubits, and has distance 3. This last part implies that it can correct any single qubit error. The five qubit code has the smallest number of physical qubits that allow for this property. Indeed, an intuitive argument for this is the following: the number of possible single qubit errors in a code with n physical qubits is $3n$, since we can have X, Y and Z errors in each of the qubits. If we are encoding one logical qubit, we will have $n-1$ generators of the stabilizer, and hence $2^{n-1}-1$ possible error syndromes. If we want to be able to identify which error has occurred unambiguously, so that we are able to correct it, we need this number to be bigger than the number of possible errors, which happens for the first time for $n=5$. The five qubit code can also correct any two erasure errors (i.e., those for which we know which qubits have been affected, for example because we lost them).

A set of generators for the stabilizer of the five qubit code is given by[6]:

$$S_0=\langle X\mathbb{1}XZZ, ZX\mathbb{1}XZ, ZZX\mathbb{1}X, XZZX\mathbb{1}\rangle$$

Since S_0 has four independent generators of order two, we have $|S_0|=16$. In fact, we can give a very brief description of it. Consider the following two automorphisms of $\mathscr{G}_5$:

$$\begin{array}{rccc} R_5: & \mathscr{G}_5 & \longrightarrow & \mathscr{G}_5 \\ & M_1M_2M_3M_4M_5 & \longmapsto & M_2M_3M_4M_5M_1 \end{array}$$

[6] We omit the $\otimes$ symbols between the single qubit Pauli operators to save space.

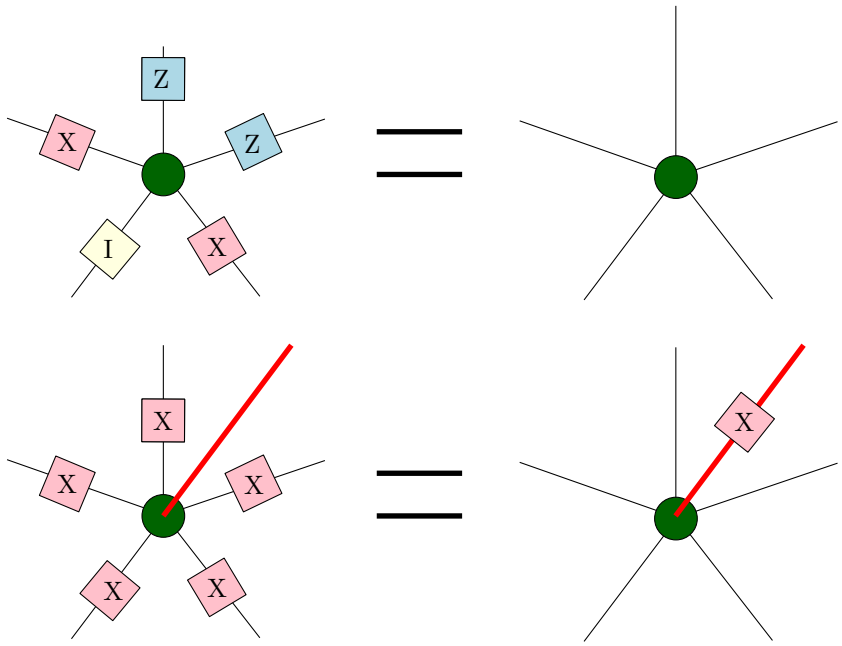


Figure 1.3: Above: the composition of the encoding map Φ with an element of the stabilizer results in the same map Φ. Below: Encoding of a logical X operation as five physical X operations.

$$\begin{array}{cccc} T_5: & \mathscr{G}_5 & \longrightarrow & \mathscr{G}_5 \\ & M_1M_2M_3M_4M_5 & \longmapsto & T(M_1)T(M_2)T(M_3)T(M_4)T(M_5) \end{array}$$

where T is the automorphism of $\mathscr{G}_1$ given by the following action on the generators:

$$X \longrightarrow Y \longrightarrow Z \longrightarrow X \tag{1.11}$$

Note that R_5 has order 5 and T_5 has order 3. Then if we call $\mathscr{S} = \langle R_5, T_5 \rangle \leq \mathrm{Aut}(\mathscr{G}_5)$, we have

$$S_0 = \{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\} \cup \mathrm{Orb}_{\mathscr{S}}(X\mathbb{1}XZZ)$$

Noting this is useful because: a) this two automorphisms will also preserve the stabilizer of our holographic codes, and b) it gives us a way to think quickly about all the elements of the stabilizer, which will be useful when we start to represent them pictorially. Indeed, we will depict this code as a tensor with one input leg (which we will many times omit in the drawings) and five output legs corresponding to the five physical qubits (see figure 1.3). The five-fold rotational symmetry R_5 is manifest in the drawing. The five qubit code allows for very easy to remember encoded logical operators. Indeed, if we call Φ the encoding isometry, it holds:

$$\Phi \circ X = XXXXX \circ \Phi \qquad \Phi \circ Z = ZZZZZ \circ \Phi$$

Hence a logical Pauli operator can be applied by just applying that same Pauli operator to all of the physical qubits. Imagine though that we only had access to the first three qubits. Could we still, say, perform a logical Z on the encoded qubit. It turns out, we could. Clearly, composing an encoded logical operator with an element of the stabilizer results in another encoded version of the same logical operator. For example,

$$(X\mathbb{1}XZZ)(ZZZZZ) = -YZY\mathbb{1}\mathbb{1}$$

and we can perform an encoded Z having access to only the first three qubits. In fact, this applies to *any* three qubits. It does not, though, if we only have access to two qubits, since we know that encoded operations belong to $N(S)\backslash S$ and the minimal weight of the operators in this set is the distance, which is three: we need to act nontrivially on three qubits to be able to manipulate the logical, encoded state. Good thing is, as we said, those can be any three qubits we want. This property will turn out to be important soon.

1.3.2 Quantum secret sharing

Let us finish this chapter with an interesting remark related with what we just saw. As an interesting side application of quantum error correction, let us review a particular scheme called *quantum secret sharing*. Suppose that five parties, Alice, Bob, Charlie, Debbie and Erica, are selected by the government to custody some particular, very important information. However, the officials involved suspect that some of them might not be trustworthy, and should not be granted access to the information. To minimize risks, they agree on a scheme such that each of the five will be given some piece of information, which is useless by itself when it comes to revealing the original secret, but which allows to recover it when three or more of the secret holders put their pieces together. If they find a way to do so they will have designed a (3,5)-threshold scheme.

Definition 1.16. Let k,n be natural numbers, with $k \leq n$. A $(\mathbf{k},\mathbf{n})$**-threshold scheme** is a scheme for information sharing among n parties that allows for any k of them to access the secret while any $k-1$ of them will be unable to retrieve any information at all.

The five qubit code turns out to provide one such scheme: it suffices to encode the logical bit of information and give one physical qubit to each of the parties [6, 7]. It can be proved that holding two of the five qubits does not allow the retrieval of any information, while any three parties coming together could use the code to correct for the absence of the two missing qubits and recover the state. As we have seen, they could as well perform encoded logical operations on the secret encoded bit without the need of the other two physical qubits to do it.

Chapter 2

Holographic Quantum Error Correcting Codes

In this chapter we turn to what will be our object of study for this essay: holographic quantum error correcting codes (HQECC). We will start with a brief motivation as to why these codes came to be an object of consideration. We will then take one of the most simple examples of such a code to develop a strategy to compute the stabilizer. Finally we will generalize the construction to a whole family of HQECC of arbitrary size.

2.1 Why HQECC?

The motivation to study these systems comes from a seemingly very unrelated area of theoretical physics called the AdS/CFT correspondence. Without the aim of being too technical but more illustrative, we will discuss briefly what this is about in this section, which is unnecessary to understand the rest of this work.

One of the research lines to which a great amount of effort is devoted in current theoretical physics is the unification of quantum theory, which describes microscopical phenomena, and general relativity, which describes gravity and black holes. The AdS/CFT correspondence is a particular duality among both theories that was discovered in the context of string theory. AdS stands for **Anti-de Sitter** spacetime[1]: it arises from solving Einstein's equations of general relativity with a negative cosmological constant, which grants it negative curvature. A d-dimensional AdS spacetime possesses a $(d-1)$-dimensional boundary at infinity, which is whay this kind of spacetime is sometimes called a "gravitational box". On the other hand, CFT stands for Conformal Field Theory, which is a particular kind of quantum field theory with a large amount of symmetries (those included in the conformal group). The correspondence happens between an AdS spacetime (the *bulk*) and a CFT defined on its boundary, which is why it is usually qualified as holographic, meaning that the CFT is a hologram, a representation in one less dimension of the bulk. One of the aspects which manifest this correspondence is the existence of a mapping between operators defined on the Hilbert space associated to the bulk and operators defined on the Hilbert space associated to the boundary. In particular, there exists the so-called *AdS-Rindler reconstruction* [8], which is a mapping between a local operator defined on some point x in the bulk, and an integral over local operators supported[2] on a region of the boundary such that x lies in its *causal wedge*. (In two dimensions (one-dimensional boundary), the causal wedge of a region R of the boundary is the region of the bulk enclosed by the geodesic that starts and ends at the extremes of R, see figure 2.1.)

[1] Generally by spacetime we understand a d-dimensional Lorentzian manifold, i.e. a smooth manifold endowed with a metric of Lorentzian signature (one negative eigenvalue and $d-1$ positive eigenvalues or viceversa).

[2] We say that an operator is supported on a particular region if its action is trivial outside this region, e.g., it commutes with the algebra of local operators outside this region. This is the continuous analogue of saying that the operator $XYZ\mathbb{1}\mathbb{1}$ is supported on the first three of the five qubits.

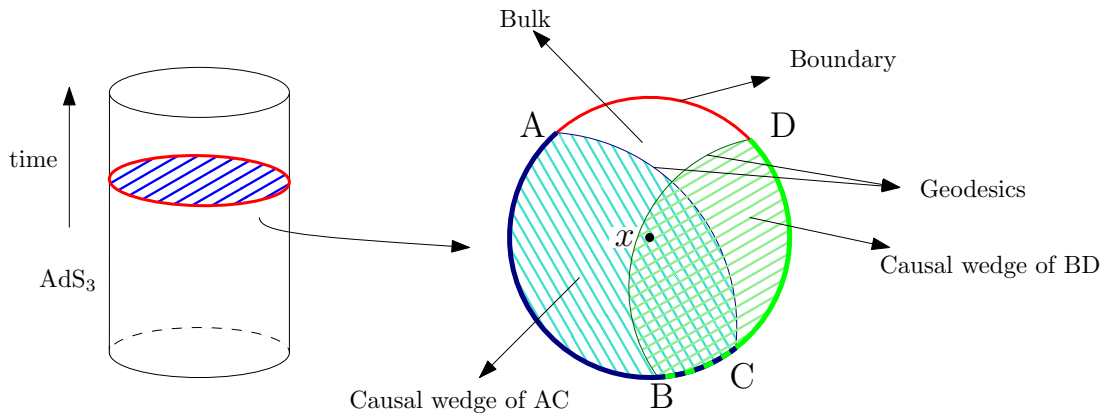


Figure 2.1: We take a time slice of AdS_3 to end up with a 2-dimensional bulk (Poincaré disc) and a 1-dimensional boundary. Point x in the bulk lies on the causal wedge of boundary regions AC and BD, so we should be able to reconstruct local operators on x over both of them.

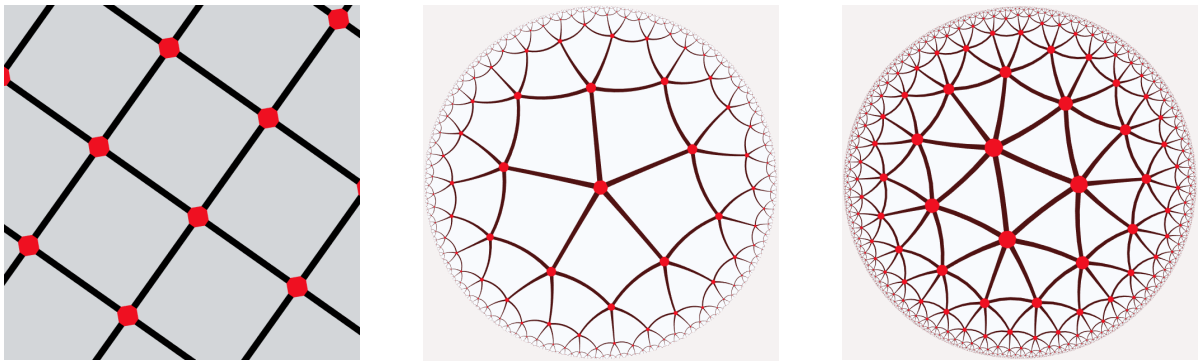
The main problem addressed in [9] is that we can find many regions of the boundary such that a particular x lies in their causal wedge. The mapping will then give rise to boundary operators supported in these different regions, and hence in principle different, even though they are associated to the same operator on the bulk, and therefore in some sense, they should be equivalent. The proposal of [9] is that all this should be interpreted in a quantum error correction context. The operator of the bulk should be seen as a logical operator acting on a logical degree of freedom, and the different operators it is mapped to in the boundary should be interpreted as different encodings of the same logical operator, which may be supported on different regions, just as it happened in the five qubit code that we presented at the end of the previous chapter. All these operators would hence act equivalently on the code subspace, which would correspond in this interpretation to the low energy subspace of the CFT. In this context, holographic quantum error correcting codes were introduced as a toy model to try to understand better this proposed connection between the AdS/CFT correspondence and quantum error correction [1]. In particular, they were proposed in the context of the AdS_3/CFT_2 duality, meaning that we consider AdS spacetime in 3 dimensions (two spatial ones plus time), and hence its boundary, where the CFT is defined, will be two-dimensional (one spatial dimension plus time). Additionally we consider a time slice, namely we fix the value of the time coordinate, so that the resulting bulk is two-dimensional and endowed with hyperbolic geometry (a Poincaré disc), and its boundary is one-dimensional. To further simplify the setting we proceed to discretize bulk and boundary by choosing a hyperbolic tiling.

2.2 Hyperbolic tilings

Our quantum error correcting code will be given by a tensor network that we will place in the Poincaré disc. This we will do by choosing a particular **regular hyperbolic tiling** of the plane. A regular hyperbolic tiling is given by a covering of the hyperbolic plane by regular polygons which do not overlap or leave gaps. A possible way to describe these tilings or tessellations is by means of their **Schläffi symbol**. A two dimensional tessellation of Schläffi symbol $\{p,q\}$ consists of a covering with regular p-gons such that q of them meet at each vertex. Examples are given in figure **??**.

Lemma 2.1. *A two-dimensional regular tiling of Schläffi symbol $\{p,q\}$ is hyperbolic if and only if $q > \frac{2p}{p-2}$, and Euclidean if and only if $q = \frac{2p}{p-2}$.*

Proof. If q equal polygons are to meet at each vertex, their inner angles must be of $\frac{2\pi}{q}$. A regular p-gon in the Euclidean plane has inner angles of $\frac{(p-2)\pi}{p}$. If this amount exceeds $\frac{2\pi}{q}$, we will need p-gons whose angles are smaller than in the Euclidean case, hence the underlying geometry will be hyperbolic. If both



$\{4,4\}$ $\{4,5\}$ $\{3,7\}$

Figure 2.2: Some regular two dimensional tilings with their corresponding Schläffi symbols. Note that the first one is Euclidean, while the other two are hyperbolic.

quantities are equal, then we are in the Euclidean case, and we need no intrinsic curvature to construct the tessellation. □

2.3 A first HQECC

The tiling that we will choose for this work has Schläffi symbol $\{4,5\}$ and it is depicted in the central part of figure 2.2. We will build a tensor network by placing a five-qubit code tensor in each of the vertices[3] of this tiling. One of the legs of the tensor will be thought of as pointing "upwards" (input leg), while the other five will be contracted with other tensors in the network. Of course we cannot work with the whole, infinite tessellation, so we will have to truncate it at some point. This will leave uncontracted legs in the boundary that will represent the output of the encoding map Φ, the physical qubits. We will also call them **boundary legs/qubits**, while the logical, input legs will be called **bulk legs/qubits**. This toy model was proposed in [1].

Definition 2.1. We will call any tensor such that n of its legs are boundary qubits an n**-cluster**.

In the codes we will work with there will only be 2- and 3-clusters.

Our first example of holographic code is shown in figure 2.3, and corresponds to the truncating the network after a first layer of faces (squares) around the central tensor, which we denote v_0. The bulk legs are shown in red and the boundary ones in blue. We insist: the tensor network we are drawing is nothing but a representation of the encoding isometry $\Phi : \mathscr{H}_L \longrightarrow \mathscr{H}_P$, decomposed in terms of "smaller" multilinear maps. We count 11 logical qubits and 25 physical qubits, that is $\mathscr{H}_L \cong \mathscr{H}_{11}, \mathscr{H}_P \cong \mathscr{H}_{25}$ and our code is a $[[25,11]]$ QECC, and it being a stabilizer code[4] it will have 25-11=14 generators of its stabilizer S, which we want to determine.

The tactic that we will follow to determine the stabilizer is inspired in [9] and [2]. Remember from figure 1.3 that we can include elements of the stabilizer around pentagon code tensors without changing the tensor. If we choose cleverly which elements of the stabilizer we apply around each tensor, we can get them to cancel each other in the bulk (due to X,Y,Z squaring to the identity). This is depicted in figure 2.4. Once we have removed all of the unitaries in the bulk, we are left with an element of the Pauli group acting on the boundary qubits. But everything we did was composing the five qubit code isometries with elements of their stabilizer, what leaves them unchanged. Thus the action of just the Pauli operators remaining on the boundary qubits does not affect the encoding isometry of the HQECC:

[3] From now on we might be a little bit sloppy and identify terms like "vertex" and "tensor" or "network" and "code" whenever the subtleties of the difference between both terms are not important.

[4] The fact that we can expect the holographic code to be a stabilizer code is nontrivial, and is proved in [1].

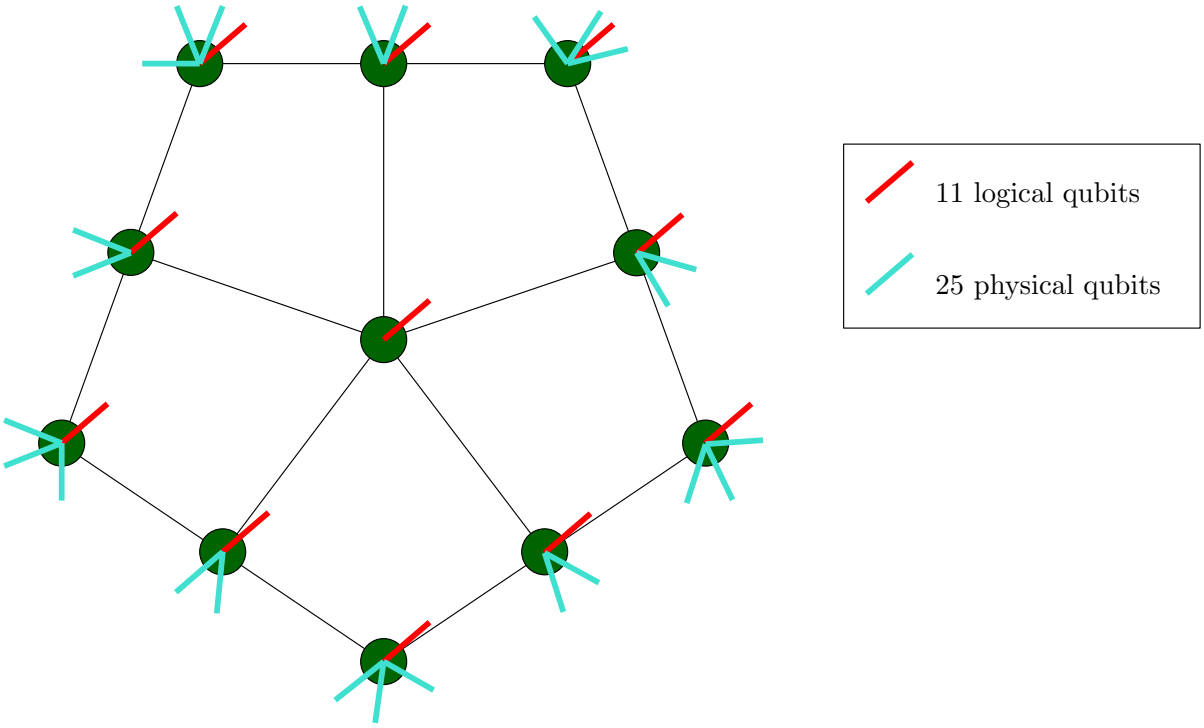


Figure 2.3: The first HQECC we will work with

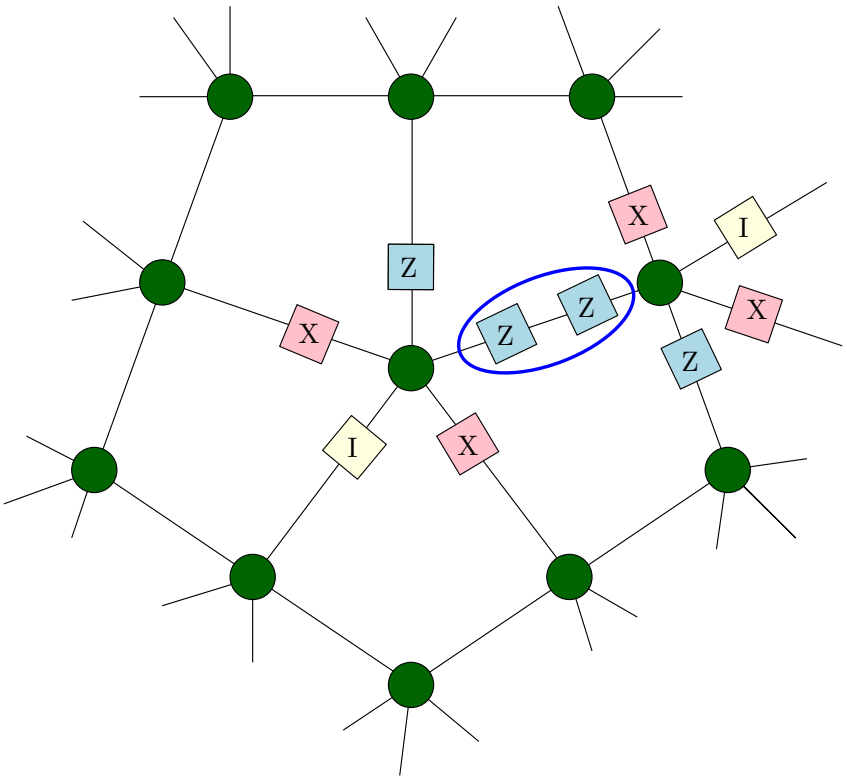


Figure 2.4: We choose to act with the appropiate five qubit code stabilizer elements so that all Pauli operators except those acting on boundary qubits cancel each other.

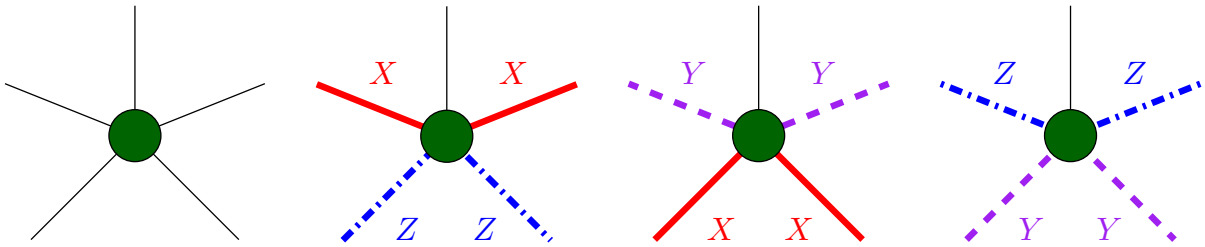


Figure 2.5: The only four possibilities, up to rotation, that we have for each vertex when choosing a stabilizer construction.

it is hence a member of the stabilizer. Because of the structure of the five qubit code stabilizer, there are, up to rotation, four possibilities for each vertex in the network, as may be seen in figure 2.5 were we denote the corresponding operators X,Y,Z by colouring the edges. Hence, we define

Definition 2.2. A **stabilizer construction** is a map $\phi : V \longmapsto S_0$, where V is the set of vertices (tensors) of the holographic code tensor network, such that the Pauli operators associated to the same edge of the network are the same (see figure 2.4). We call the set of all stabilizer constructions Σ.

Proposition 2.1. *The element s_ϕ of $\mathscr{G}_{25}$ built from taking the Pauli operators associated to boundary qubits from a stabilizer construction $\phi \in \Sigma$ belongs to the stabilizer of the holographic code S. We call the set of these elements S_c.*

Proof. It follows from the reasoning above. □

Proposition 2.2. *Σ and S_c can both be endowed with a group structure that turns*

$$\begin{aligned} \zeta : \Sigma &\longrightarrow S_c \\ \phi &\longmapsto s_\phi \end{aligned}$$

into an isomorphism.

Proof. Let $\phi_1, \phi_2 \in \Sigma$, and define $(\phi_1 \cdot \phi_2)(v) = \phi_1(v) \cdot \phi_2(v)$.This is clearly another stabilizer construction. (Note that the map

$$\begin{aligned} \Sigma &\longrightarrow (S_0)^{\times 11} \\ s &\longmapsto (s(v_0), s(v_1), \ldots, s(v_{10}) \end{aligned}$$

where the v_i are the different vertices of the code, is a monomorphism, so that Σ is isomorphic to its image, a subgroup of $(S_0)^{\times 11}$). We can get a group structure on S_c by demanding the mapping ζ to be an isomorphism. Note it is surjective by construction. It is also injective since the Pauli operators applied on the boundary qubits of the 2-clusters already determine the whole stabilizer construction. Indeed, given two single qubit Pauli operators (X,Y,Z) acting on two output qubits of the same tensor there is one and only one element of S_0 whose restriction to those two qubits is precisely those two Pauli operators (this is not difficult to check from, for example, figure 2.5). Hence s_ϕ determines $\phi(v)$ for all v boundary tensor (since they are n-clusters with $n \geq 2$), and this completely constraints $\phi(v_0)$. □

Remark 2.1. *Note that the group operation we just defined on S_c is the same as the one inherited from S, namely the composition of linear operators or matrix product.*

Figure 2.6 represents one such construction. Now, we would like to obtain a set of 14 independent generators, that will represent the stabilizer slightly more compactly than the enumeration of its 2^{14} elements. A possible strategy to find independent generators is starting by the elements of S which have minimal weight.

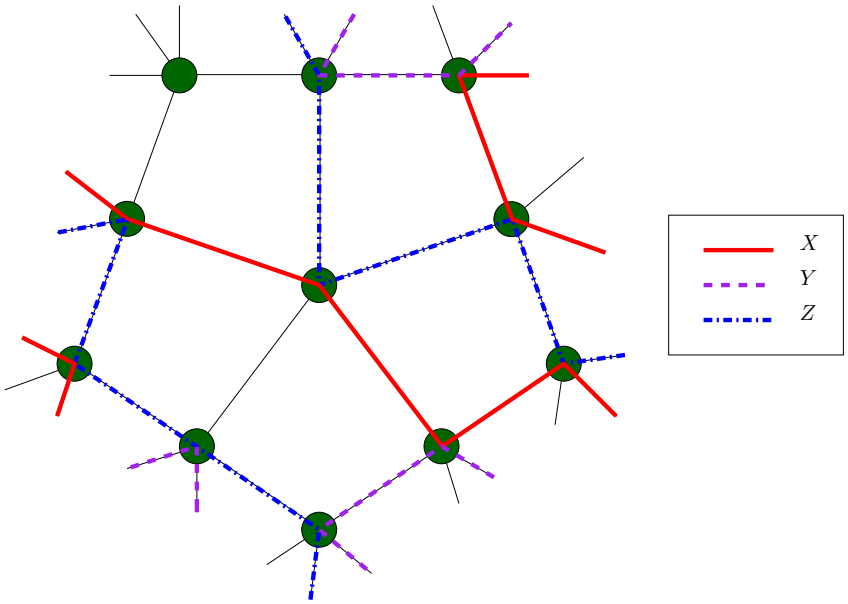


Figure 2.6: Starting from the tensor in the upper left corner, this stabilizer construction builds the stabilizer element $\mathbb{1}\mathbb{1}\mathbb{1}ZY\mathbb{1}YX\mathbb{1}XZX\mathbb{1}Y\mathbb{1}YZ\mathbb{1}YYX\mathbb{1}XZX$.

Proposition 2.3. *The minimal weight of a nontrivial boundary Pauli operator $s_\phi \in S_c$ built from a stabilizer construction $\phi \in \Sigma$ is 8.*

Proof. To abbreviate along this proof, we will call tensors v such that $\phi(v) \neq \mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}$ *loaded* tensors. Note the following consequences of the geometry of the network and the nontrivial S_0 elements having weight 4:

- Any loaded n-cluster provides at least $n-1$ boundary qubits acted upon nontrivially by s_ϕ.
- Any loaded boundary tensor is next to at least another loaded boundary tensor.

Assume first that the central tensor is loaded. Then at least four out of the five 2-clusters of the boundary are loaded. Hence, at least four boundary qubits, one per 2-cluster, are acted upon nontrivially by s_ϕ. Also, at least two 3-clusters next to the 2-clusters are loaded, and they provide another four boundary qubits acted upon nontrivially by s_ϕ, whose weight is hence at least 8. Now, if the central tensor is not loaded, every 3-cluster next to a loaded 2-cluster is loaded, hence two loaded 2-clusters already imply that the weight of s_ϕ is bigger than 8. Alternatively we can have only one loaded 2-cluster and its two adjacent 3-clusters, and no other loaded boundary tensors, but this requires s_ϕ to have weight exactly eight, since the two 3-clusters will have all of their boundary qubits acted on by nontrivial Pauli operators, the identity must act on the leg that connects them to their adjacent unloaded 2-clusters (we will soon give examples of this). If no 2-clusters are loaded, s_ϕ is trivial (remember the injectivity of ζ) and we are done. □

The previous proof is constructive in the sense that it provides us with a guideline on how to get minimal weight elements of S. These are interesting because it is easy to see that those will be independent generators, each being localized on their side of the boundary pentagon. We can also apply qubitwise the automorphism T that we defined at the end of the first chapter. Note that this is equivalent to choosing $T \circ \phi$ as our stabilizer construction. We get this way a set of ten independent generators, the ones in figure 2.7 and their rotated partners. Note that we cannot apply T once again to get more generators since they would turn out not to be independent:

$$XZZZZZZX \xrightarrow{T} YXXXXXXY \xrightarrow{T} ZYYYYYYZ = XZZZZZZX \cdot YXXXXXXY \tag{2.1}$$

If we manage to find four more independent generators, we will be done since we know there cannot

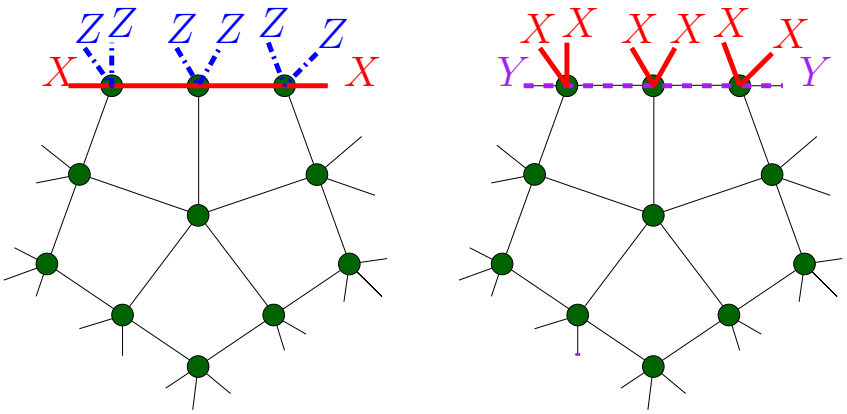


Figure 2.7: Stabilizer generators of minimal weight.

be any more. To do so, we notice that all of the previous generators come from stabilizer constructions where $\phi(v_0) = \mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}$. But we have many more possibilities for $\phi(v_0)$, some of which might lead to independent elements of the stabilizer. Indeed, we notice the existence of the following homomorphism:

Lemma 2.2. *Let $s \in S_c$. Then the map*

$$\begin{aligned} \pi : S_c &\longrightarrow S_0 \\ s &\longmapsto \zeta^{-1}(s)(v_0) \end{aligned}$$

is a homomorphism.

Proof. It is the composition of ζ^{-1}, which is an isomorphism, and

$$\begin{aligned} \zeta_0 : S_c &\longmapsto S_0 \\ \phi &\longmapsto \phi(v_0) \end{aligned}$$

which is an homomorphism by the definition of the group structure on the set of stabilizer constructions. □

If we now find elements $s_1, s_2, s_3, s_4 \in S_c$ such that they are mapped to the generators of S_0:

$$\pi(s_1) = X\mathbb{1}XZZ \qquad \pi(s_2) = ZX\mathbb{1}XZ \qquad \pi(s_3) = ZZX\mathbb{1}X \qquad \pi(s_4) = XZZX\mathbb{1}$$

it is clear that they will be independent of each other. Indeed, if there existed a way to express one in terms of the others, the homomorphism π would map it into the corresponding relationship between the generators of S_0, which we know to be independent. The same reasoning applies to prove that these four elements are independent of the other ten we had found, since these all belong to $\pi^{-1}(\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1})$, thus we can prove the following

Theorem 2.1. $S_c = S$

Proof. Figure 2.8 shows the construction ϕ associated to a possible $s_1 = \zeta(\phi)$, which then can be rotated three times to provide the rest of generators. Together with the other ten generators found before, this gives us a set of 14 different generators of $S_c \leq S$, which is the same number that S has, hence they are equal. □

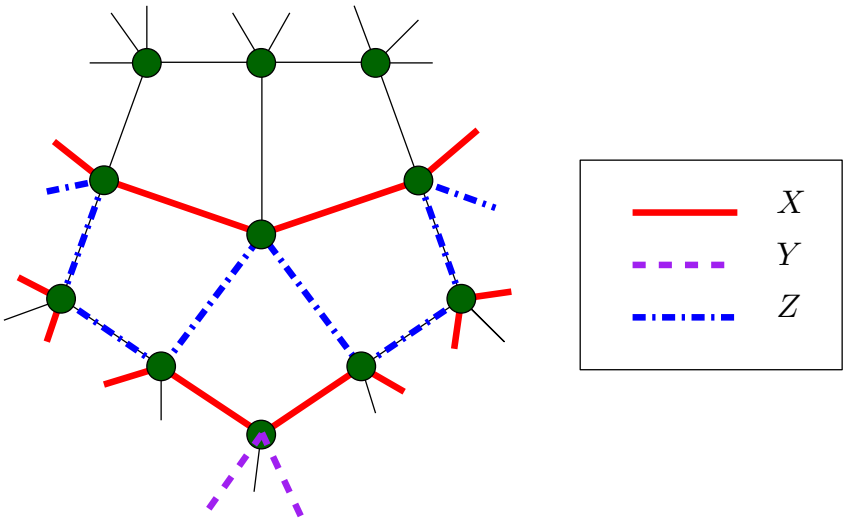


Figure 2.8: A generator which does not map to $\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}$ under π. By rotating the figure one, two and three fifths of a circumference we obtain constructions for other independent generators.

2.4 The network grows

We wonder now whether we are able to find the stabilizer for codes built from larger portions of the hyperbolic tiling. In principle we can truncate the network arbitrarily, but we expect that keeping some symmetry and regularity will help us in the long run. Therefore we will focus here on computing the stabilizer for what we will call **face based codes**, which are depicted in figure 2.9. To define them properly we need some extra notation:

Definition 2.3. We call the central tensor of the {4,5}-hyperbolic tiling its zeroth **boundary**, $B_0 = \{v_0\}$. Recursively, we define the n-th boundary of the code B_n as the set of tensors of the network which belong to the same face that at least one tensor of B_{n-1} and are not included in B_i for $i < n$.

Remark 2.2. *Note that the progressively increasing boundaries provide a partition of the set of vertices of the network.*

Remark 2.3. *This way of defining boundaries does not proceed by graph distance from the center. Instead, the n-th boundary contains tensors placed at graph distances ranging from n to 2n. If we draw the diagonals of every face and count them as edges of the graph, then the n-th boundary could actually be defined as the set of tensors at graph distance n from the center.*

Definition 2.4. The n-th face based code or FBC is the holographic QEC obtained from a truncation of the {4,5}-hyperbolic tiling along its n-th boundary. We call its set of vertices V_n and its stabilizer S_n. Note that $V_n = \cup_{i=0}^{n} B_i$.

It is easy to compute how many 3-clusters and 2-clusters the n-th FBC has:

Proposition 2.4. *Let a_n, b_n be the number of 2-clusters and 3-clusters of the n-th FBC. Then we have the following recurrence relations:*

$$a_1 = 5 \qquad a_n = 2a_{n-1} + 3b_{n-1} \tag{2.2}$$

$$b_1 = 5 \qquad b_n = a_{n-1} + 2b_{n-1} \tag{2.3}$$

Proof. The values for $n = 1$ can be checked by inspection. Two adjacent physical qubits in the same cluster will become edges of a face , and the tensor opposite to the one giving the cluster will give rise to a new 3-cluster. Hence a 2-cluster gives rise to a 3-cluster, and a 3-cluster to two 3-clusters. This provides the second recurrence relation. As for the first one, notice that every physical qubit of

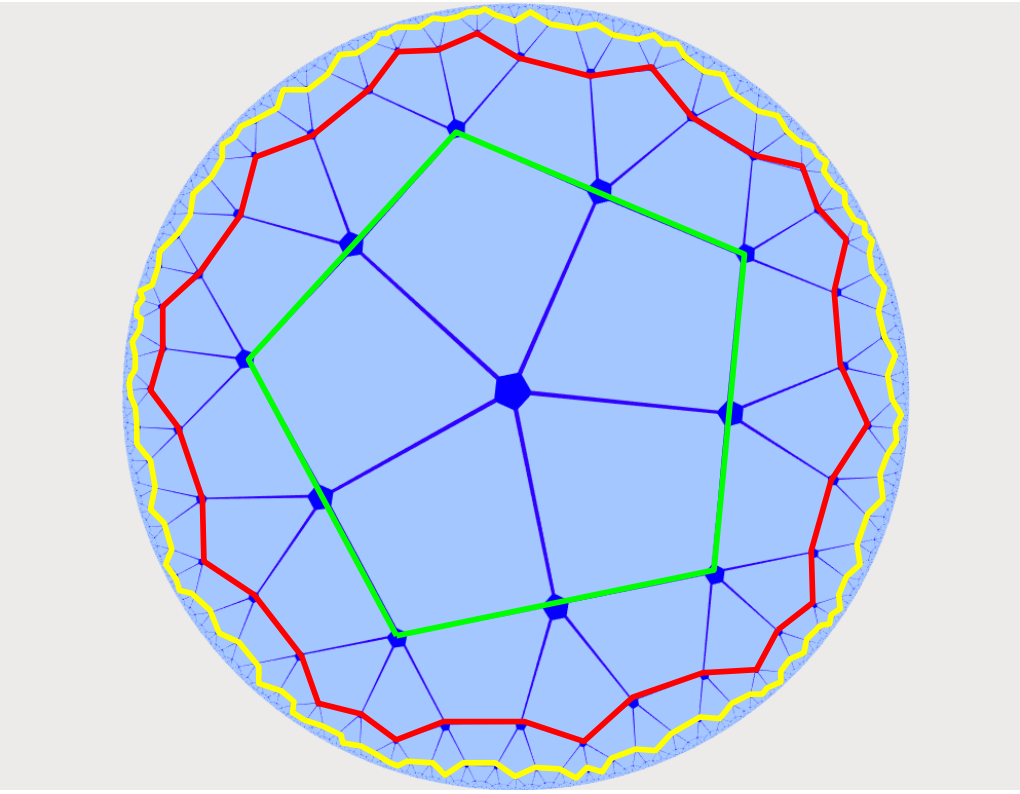


Figure 2.9: The first three FBCs, with the boundaries B_1, B_2, B_3 colored. Note that the initial example we analyzed is precisely the first of them.

the $(n-1)$-th layer will be contracted with a tensor shared by two added faces, hence this tensor will provide a 2-cluster for the next layer. □

Corollary 2.1.

$$a_n = \frac{5}{2}\left(\left(\sqrt{3}+1\right)\left(\sqrt{3}+2\right)^{n-1} - \left(2-\sqrt{3}\right)^{n-1}\left(\sqrt{3}-1\right)\right) \tag{2.4}$$

$$b_n = \frac{5}{6}\left(\left(\sqrt{3}+2\right)^{n-1}\left(\sqrt{3}+3\right) - \left(2-\sqrt{3}\right)^{n-1}\left(\sqrt{3}-3\right)\right) \tag{2.5}$$

Proof. It amounts to solving the linear recurrences of the proposition. □

From this expressions we can get the number of physical and logical qubits of each FBC easily. However, for what follows we will only need the following

Corollary 2.2. *Let g_n be the number of generators of S_n. Then* $g_{n+1} = g_n + 2b_{n+1} \qquad \forall n \in \mathbb{N} \cup \{0\}$.

Proof. Call p_n and l_n the number of physical and logical qubits respectively. From the proof of the proposition it follows that $p_n = a_{n+1}$ and it is also clear that $l_{n+1} - l_n = a_{n+1} + b_{n+1}$. Putting it all together,

$$g_{n+1} - g_n = (p_{n+1} - l_{n+1}) - (p_n - l_n) = (p_{n+1} - p_n) - (l_{n+1} - l_n) = a_{n+2} - a_{n+1} - a_{n+1} - b_{n+1} = 2b_{n+1}$$

□

2.5 The stabilizer of a general FBC

It is now time to apply what we learned in the case $n = 1$ to a general *FBC*. The generalization of the definition of a stabilizer construction is straightforward. It is our hope that again these constructions will provide us with all the generators (and thus all the elements) of the stabilizer S_n of the n-th FBC.

Remember that last time we started using stabilizer constructions whose only loaded tensors lay in a very localized region of the boundary. It turns out, these kind of stabilizer generators exist for all FBCs! Indeed, we can repeat the construction of figure 2.7 from any segment of B_n located between two 3-clusters, and it is easy to see that we can make the only loaded tensors be the ones in that segment. Thus, this provides us with $2b_n$ independent generators for S_n.

Now comes a key realization: according to Corollary 2.2 the number of generators we are missing is $g_n - 2b_n = g_{n-1}$, that is the number of generators of S_{n-1}, the stabilizer of the previous FBC. This is totally analogous to the situation for $n = 1$, when we were missing four generators, as many as S_0 has. Thus we just we just need to go through the slightly tedious generalization of the machinery we developed above to reach a quick conclusion:

Definition 2.5. A stabilizer construction for the n-th FBC is a map $\phi : V_n \longmapsto S_0$ such that the Pauli operators associated to the same edge of the graph by this map are the same. We call this set of stabilizer constructions Σ_n. It has a group structure analogous to that of Σ_1. Every $\phi \in \Sigma_n$ defines an element s_ϕ of a subset $S_{c,n}$ of S_n by selecting only the Pauli operators acting on the boundary qubits.

Proposition 2.5. *Let $n, m \in \mathbb{N} \cup \{0\}$, $n > m$.*

1. *Let $\phi \in \Sigma_n$. Then $\phi\big|_{V_m} \in \Sigma_m$ and $\phi \longmapsto \phi\big|_{V_m}$ is an epimorphism.*
2. *The map*

$$\begin{aligned} \zeta_n : \Sigma_n &\longrightarrow S_{c,n} \\ \phi &\longmapsto s_\phi \end{aligned}$$

   is a bijection that, by demanding it to be an isomorphism, induces a group structure on $S_{c,n}$ whose product coincides with the one inherited from S_n.

3. *The map*

$$\begin{aligned} \pi_{n,m} : S_{c,n} &\longrightarrow S_{c,m} \\ s &\longmapsto \zeta_m\left(\zeta_n^{-1}(s)\big|_{V_m}\right) \end{aligned}$$

   is an epimorphism.

Proof. Only the surjectivity in (1) and the injectivity in (2) might be nontrivial. For the first one, it is enough to prove it in the case $m = n - 1$. Notice that every tensor $v \in B_n$ is connected to at most one tensor in B_{n-1}, hence only one of the Pauli operators of $\phi(v)$ is determined by $\phi\big|_{V_m}$. Now choose v to be a 2-cluster adjacent to a 3-cluster and choose an arbitrary Pauli operator for the leg of this 2-cluster that is not a boundary qubit nor connects it to B_m or the 3-cluster. That way we fix $\phi(v)$. Now we proceed to the next boundary tensor v' in opposite direction to the 3-cluster: at most two of the Pauli operators of $\phi(v')$ are determined. Hence, if needed we make an arbitrary choice and move on. Eventually we will reach the 3-cluster next to where we started, and by doing so at most two of its Pauli operators will have been determined. This way we can build a valid ϕ such that its restriction to V_m is a given element of Σ_m. For the injectivity of (2), notice that every tensor in B_n is an r-cluster with $r \geq 2$, thus s_ϕ determines $\phi\big|_{B_n}$ by the properties of S_0 we used previously. But now, every tensor of B_{n-1} is connected to at least two tensors of B_n by construction of the boundaries, thus the same argument applies and $\phi\big|_{B_n}$ is determined by s_ϕ. Now we iterate the process until we reach B_1, when proposition 2.2 applies and we are done. □

Theorem 2.2. *For every $n \geq 1$, $S_{c,n} = S_n$.*

Proof. We argue by induction. For $n = 1$ it is theorem 2.1. Assume it is true for $n \leq n-1$. It is easily checked that the localized elements analogous to the ones from figure 2.7 provide us with $2b_n$ independent generators of $S_{c,n}$: the corresponding stabilizer construction assigns $\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}$ to every tensor in the network but the boundary tensors between two consecutive 3-clusters. Now the map $\pi_{n,n-1}$ is an epimorphism, hence there exist elements in $S_{c,n}$ which are mapped to the g_{n-1} generators of S_{n-1}. These elements are therefore independent of each other and of the localized ones, which belong to the kernel of $\pi_{n,n-1}$. Thus we have enough generators, we know that $S_{c,n} = S_n$ and we know how to build them. □

2.6 Conclusions and outlook

Finally we know how to build the stabilizer for FBCs. It turns out that a large proportion of the generators of S_n are actually very localized in the boundary. Then we have some which have a slightly bigger support: the (not unique) preimages by $\pi_{n,n-1}$ of the generators of S_{n-1} that are localized inside B_{n-1}. Then we get the preimages by $\pi_{n,n-2}$ of the localized elements of S_{n-2}, and so on. To obtain these preimages we just need to "push" the stabilizer elements towards the boundary, as in the first part of the proof of proposition 2.5. Thus we end up with generators from

$$\ker \pi_{n,n-1}, \ker \pi_{n,n-2} \backslash \ker \pi_{n,n-1}, \ldots$$

until we get to the preimages by $\pi_{n,0}$ of the four generators of the stabilizer of S_0, which will probably be very delocalized in the boundary. It would be interesting to study how much so, since the size of the support of the generators of the stabilizer may actually have physical consequences, not only for the practical reason that these operators need to be measured if we are to use the HQECC as a QECC, but also as a hint at the entanglement (quantum correlations) properties of the CFT states that the HQECC is modelling.

There are many other questions that can be explored when it comes to HQECC, both from a mathematical and a physical perspective. Following with the ideas of localization in the boundary, it would be interesting to characterize the fault-tolerant gates of these codes, maybe establishing as a fault-tolerance criterion the requirement that the support of the encoded logical operators does not grow too much during a computation. Another research line involves generalizing our results to other truncation schemes of the holographic codes, and studying the possible relationship, if there is any, between the algebraic properties of the stabilizer and the geometrical properties of the network. Also, we could try to obtain insights from our results with the HQECC to help us in the generalization of these ideas to a continuous setting as the one provided by AdS/CFT.

Bibliography

[1] F. Pastawski, B. Yoshida, D. Harlow, and J. Preskill, "Holographic quantum error-correcting codes: toy models for the bulk/boundary correspondence," *Journal of High Energy Physics*, vol. 2015, no. 6, pp. 1–55, 2015.

[2] B. Yoshida, "Quantum gravity from quantum error correcting-codes? [https://quantumfrontiers.com/2015/03/27/quantum-gravity-from-quantum-error-correcting-codes/]."

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

[4] D. Gottesman, "Theory of fault-tolerant quantum computation," *Phys. Rev. A*, vol. 57, pp. 127–137, Jan 1998.

[5] D. Gottesman, "Stabilizer codes and quantum error correction. Caltech Ph.D. thesis, 1997."

[6] R. Cleve, D. Gottesman, and H.-K. Lo, "How to share a quantum secret," *Phys. Rev. Lett.*, vol. 83, pp. 648–651, Jul 1999.

[7] J. Preskill, "Lecture notes on quantum computation [http://www.theory.caltech.edu/ preskill/ph219/index.htmllectur

[8] A. Hamilton, D. Kabat, G. Lifschytz, and D. A. Lowe, "Holographic representation of local bulk operators," *Phys. Rev. D*, vol. 74, p. 066009, Sep 2006.

[9] A. Almheiri, X. Dong, and D. Harlow, "Bulk locality and quantum error correction in AdS/CFT," *Journal of High Energy Physics*, vol. 2015, no. 4, pp. 1–34, 2015.

[10] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A*, vol. 52, pp. R2493–R2496, Oct 1995.