



**HAL**  
open science

## Decoding color codes by projection onto surface codes

Nicolas Delfosse

► **To cite this version:**

Nicolas Delfosse. Decoding color codes by projection onto surface codes. *Physical Review A : Atomic, molecular, and optical physics [1990-2015]*, 2014, 89, pp.012317. 10.1103/PhysRevA.89.012317. hal-00855003

**HAL Id: hal-00855003**

**<https://hal.science/hal-00855003>**

Submitted on 28 Aug 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Decoding color codes by projection onto surface codes

Nicolas Delfosse

delfosse@lix.polytechnique.fr

LIX & INRIA Saclay - École Polytechnique, 91128 Palaiseau, France

August 28, 2013

## Abstract

We propose a new strategy to decode color codes, which is based on the projection of the error onto three surface codes. This provides a method to transform every decoding algorithm of surface codes into a decoding algorithm of color codes. Applying this idea to a family of hexagonal color codes, with the perfect matching decoding algorithm for the three corresponding surface codes, we find a phase error threshold of approximately 8.7%. Finally, our approach enables us to establish a general lower bound on the error threshold of a family of color codes depending on the threshold of the three corresponding surface codes. These results are based on a chain complex interpretation of surface codes and color codes.

## 1 Introduction

Classical Low Density Parity-Check (LDPC) codes are of high practical interest for classical error correction because they are both high rate and endowed with an efficient decoding algorithm. It is therefore natural to investigate their quantum generalizations [25]. Topological constructions of quantum codes lead to some of the most promising families of quantum LDPC codes including Kitaev's toric code [20], surface codes [4, 33, 9], color codes [5] and other geometrical constructions [8, 30, 22, 15, 1]. Besides their appeal for quantum error-correction, topological codes also exhibit some interesting features for fault tolerant quantum computing, such as the property that some gates can be implemented topologically. In this paper, we are interested in decoding algorithms for topological codes, which is one essential ingredient of fault tolerance to avoid error accumulation during a computation.

The most natural way to decode quantum LDPC codes is to adapt classical strategies [27]. Unfortunately, the unavoidable presence of short cycles in the Tanner graph of CSS codes makes this iterative decoding algorithm difficult to adapt to the quantum setting. We focus here on the decoding problem for two families of quantum LDPC codes based on tilings of surfaces: surface codes and color codes. In this special case, different strategies making use of the geometrical structure of the code have been proposed.

Topological codes appeared with Kitaev's toric code defined from a square tiling of the torus [20]. More generally, we can associate a quantum code to every tiling of surface [5]. These surface codes form a family of quantum LDPC codes because they are defined by stabilizers acting on a small number of qubits but they are also physically local, reducing the error during the syndrome measurement. A first approach based on a perfect matching algorithm was proposed by Dennis, Kitaev, Landahl and Preskill to decode surface codes [10].

Their algorithm computes a most likely error using a minimum weight perfect matching algorithm. It has been applied to fault-tolerant quantum computing by Wang, Fowler, Stephen and Hohenberg [32]. More recently, inspired by ideas from statistical physics, Duclos-Cianci and Poulin introduced a renormalization group decoding algorithm, for the toric code [11].

Color codes, introduced by Bombin and Martin-Delgado, are also derived from tilings of surfaces [4]. For fault-tolerance, they offer an advantage over surface codes in that we can apply, topologically, a larger number of operations on the encoded qubits. The perfect matching decoding algorithm has been generalized to color codes by Wang, Fowler, Hill and Hollenberg, but, the decoding problem becomes more difficult with color codes because it corresponds to a hypergraph matching [31]. A message passing algorithm which shares some characteristics of the renormalization group decoding algorithm has been proposed by Sarvepalli and Raussendorf [28]. The fault tolerance of a family of color codes based on the square-octagonal lattices is studied by Landahl, Andersen and Rice in [23].

In the present work, we relate the decoding problem for color codes to the decoding problem for surface codes. This is done by the projection of an error acting on a color code onto three surface codes. This strategy allows us to transform every decoding algorithm for surface codes into a decoding algorithm for color codes. As an example, we consider the decoding algorithm for color codes deduced from the perfect matching decoding algorithm for surface codes. Our numerical results on the hexagonal color codes exhibit a phase error threshold of 8.7%, higher than the threshold of 7.8% observed by Sarvepalli and Raussendorf for the same codes [28].

Our main tool is a chain complex interpretation of CSS codes and topological codes. A similar approach has also been used recently in quantum information by Freedman and Hastings [15] and by Audoux [1]. The projection from the color code to the surface codes is a morphism of chain complexes. Therefore, it conserves the whole structure of the chain complex representing the color code.

The optimal error threshold of some particular families of surface codes and color codes was previously estimated using a mapping onto a Ising model [10, 19, 26, 2]. Our approach allows us to compare the thresholds of color codes and surface codes achievable using an efficient decoding algorithm. Using the fact that the projection onto the surface codes conserves the error model, *i.e.* the depolarizing channel for a color code is sent onto a depolarizing channel over the surface codes, we derive a general lower bound on the threshold of a family of color codes as a function of the threshold of the three corresponding surface codes.

The remainder of this article is organized as follows. The definition of CSS codes and the error model are recalled in Section 2. Section 3 is devoted to the construction of CSS codes and surface codes from chain complexes. This point of view will be essential to our study of color codes. We describe color codes with the language of chain complexes in Section 4. Then we introduce a projection from the color code chain complex to the surface code chain complex in Section 5. Our decoding algorithm for color codes, and the analysis of its performance are presented in Section 6.

## 2 Background

### 2.1 The CSS construction of quantum codes

In this section, we review the combinatorial construction of CSS codes, their syndrome function and the noise model.

A stabilizer code of length  $n$  is defined to be the set of fixed point of a family of commuting Pauli operators  $S_1, S_2, \dots, S_r$  acting over  $n$  qubits (that is over  $(\mathbb{C}^2)^{\otimes n}$ ). When this family is composed of  $r_X$  operators of  $\{I, X\}^{\otimes n}$  and  $r_Z$  operators of  $\{I, Z\}^{\otimes n}$ , it suffices to check the commutation relations between operators of different kind. Recall that  $I$  is the identity matrix of size 2,  $X$  is the Pauli matrix representing the bit flip error and  $Z$  is the Pauli matrix representing the phase error. This construction has been introduced by Calderbank, Shor [7] and Steane [29]. It is the family of *CSS codes*. The isomorphisms between  $\{I, X\}$ ,  $\{I, Z\}$  and the field  $\mathbb{F}_2$  enables us to relate classical and quantum codes. By these isomorphisms, the stabilizers  $S_i$  correspond to binary vectors and the commutation relation corresponds to the orthogonality relation in  $\mathbb{F}_2^n$ . This leads to a combinatorial description of CSS codes that is recalled in the next paragraph.

A CSS code is defined by two binary matrices  $\mathbf{H}_X \in \mathcal{M}_{r_X, n}(\mathbb{F}_2)$  and  $\mathbf{H}_Z \in \mathcal{M}_{r_Z, n}(\mathbb{F}_2)$  satisfying  $\mathbf{H}_X \mathbf{H}_Z^t = 0$ , *i.e.* with orthogonality between rows of  $\mathbf{H}_X$  and rows of  $\mathbf{H}_Z$ . The rows of these two matrices correspond to the stabilizers  $S_i$  in the previous definition. The integer  $n$  is the *length* of the quantum code, it is the number of qubits used to describe the encoded states. The number of encoded qubits is  $k = n - \text{rank } \mathbf{H}_X - \text{rank } \mathbf{H}_Z$ . Two classical codes are associated with such a quantum code. Denote by  $C_X$  the kernel of the matrix  $\mathbf{H}_X$  and denote by  $C_Z$  the kernel of  $\mathbf{H}_Z$ . Remark that  $C_X^\perp$  is the space generated by the rows of  $\mathbf{H}_X$  and  $C_Z^\perp$  is the space generated by the rows of  $\mathbf{H}_Z$ .

A *quantum error* acting on this CSS code is defined to be a pair of vectors  $E = (E_X, E_Z) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ . It simply corresponds to a Pauli operator, acting over  $n$  qubits, decomposed following  $X$  and  $Z$ . When an error  $(E_X, E_Z)$  corrupts a quantum state of the code, we can measure its *syndrome*. It is the pair of vectors  $(s_X, s_Z) \in \mathbb{F}_2^{r_Z} \times \mathbb{F}_2^{r_X}$  defined by  $s_X = \mathbf{H}_Z E_X^t$  and  $s_Z = \mathbf{H}_X E_Z^t$ . The syndrome can be measured and it is the only information that we possess about the error which occurs.

By construction of stabilizer codes and CSS codes, a group of errors fixes the quantum code. It is the set of errors satisfying  $E_X \in C_X^\perp$  and  $E_Z \in C_Z^\perp$ . This set of operators corresponds to the subgroup, called the stabilizer group, generated by the operators  $S_i$ . As a consequence, all the errors of the coset  $(E_X + C_X^\perp, E_Z + C_Z^\perp)$  behave like  $(E_X, E_Z)$ . This property of stabilizer codes, referred to as the *degeneracy*, allows us to correct all the errors of this coset using the same procedure.

Let us now introduce the error model. We consider independently the two components  $E_X$  and  $E_Z$  of a quantum error and we assume that these two vectors of  $\mathbb{F}_2^n$  are subjected to a binary symmetric channel of probability  $p$ . It is a simplified version of the depolarizing channel of probability  $3p/2$ . We assume that we are able to measure the syndrome without measurement errors.

The properties of errors for CSS codes are summarized in Table 1.

Table 1: Error representation for CSS codes.

	X-component	Z-component
error	$E_X \in \mathbb{F}_2^n$	$E_Z \in \mathbb{F}_2^n$
syndrome	$s_X = \mathbf{H}_Z E_X^t \in \mathbb{F}_2^{r_Z}$	$s_Z = \mathbf{H}_X E_Z^t \in \mathbb{F}_2^{r_X}$
stabilizer	$E_X \in C_X^\perp$	$E_Z \in C_Z^\perp$

Our goal is to compute a most likely error coset  $(\tilde{E}_X + C_X^\perp, \tilde{E}_Z + C_Z^\perp)$  reaching a given syndrome  $(s_X, s_Z)$ . To simplify, we often look for a most likely error  $(E_X, E_Z)$  reaching a given syndrome. The  $X$ -component and the  $Z$ -component of the error can be considered

separately.

## 2.2 Tilings of surfaces

The quantum codes studied in this paper are based on topological objects, that we introduce in this part.

A *graph* is defined as a pair  $(V, E)$ , where  $V$  is a set and  $E$  is a set of pairs of elements of  $V$ . The elements of  $V$  are called *vertices* and the elements of  $E$  are called *edges*. The two vertices included in an edge  $e = \{u, v\}$  are the *endpoints* of  $e$ . The edge  $e$  is said to be incident to the two vertices  $u$  and  $v$ . A *hypergraph* is defined similarly but we allow edges to contain more than two vertices.

A *tiling of surface* is a triple  $G = (V, E, F)$ , where  $(V, E)$  is a finite graph embedded in a compact 2-manifold (surface), without boundary, without overlapping edges, and  $F$  is the set of faces defined by this embedding. A face is given as the set of edges on its boundary. For example a face  $f = \{e_1, e_2, \dots, e_m\}$  contains  $m$  edges. This face  $f$  is said to be incident to the edge  $e_i$ , for all  $i$  such that  $1 \leq i \leq m$ . Without loss of generality, we can assume that this surface is smooth and connected. We also assume that the graph  $(V, E)$  contains neither loops nor multiple edges.

Given a tiling of surface  $G = (V, E, F)$ , we construct its *dual tiling*  $G^* = (V^*, E^*, F^*)$ . It is the graph of vertex set  $V^* = F$ , such that two vertices of  $V^*$  are joined by an edge if and only if the corresponding faces share an edge in the graph  $G$ . In other word, every edge of  $G$  corresponds to an edge of its dual  $G^*$ . The set of edges incident to a vertex  $v$  of  $G$  induces a face of the dual tiling. This leads to a one-to-one correspondence between the faces of the dual graph  $G^*$  and the vertices of the graph  $G$ . In some degenerated cases, this dual graph could have loops or multiple edges. We assume that this does not happen here.

## 3 Construction of CSS codes from chain complexes

The orthogonality relations required to define CSS codes can be deduced from the properties of chain complexes [24]. In this section, we describe these quantum codes with the language of chain complexes. This formalism will be essential to study color codes and to decompose their decoding problem.

### 3.1 Definition of 2-complexes

Let us recall the definition of a 2-complex.

**Definition 3.1.** A 2-complex is a sequence of three  $\mathbb{F}_2$ -vector spaces  $C_0, C_1, C_2$ , endowed with two  $\mathbb{F}_2$ -linear applications  $\partial_2 : C_2 \rightarrow C_1$  and  $\partial_1 : C_1 \rightarrow C_0$  such that:

$$\partial_1 \circ \partial_2 = 0. \tag{1}$$

The applications  $\partial_1$  and  $\partial_2$  are called the *boundary applications*. The vectors of the space  $\text{Ker } \partial_i$  are called *i-cycles* or *cycles* and the vectors of the space  $\text{Im } \partial_i$  are called *i-boundaries* or *boundaries*. In what follows, we consider only finite dimensional spaces  $C_i$ .

To define a chain complex from a tiling of surface, we will use the  $\mathbb{F}_2$ -linear structure of the power set  $\mathcal{P}(X)$  of a finite set  $X$ . The power set  $\mathcal{P}(X)$  is naturally a  $\mathbb{F}_2$ -linear vector

space for the symmetric difference. Alternatively, it can be represented as the  $\mathbb{F}_2$ -linear space of formal sums of elements of  $X$ :

$$\bigoplus_{x \in X} \mathbb{F}_2 x = \left\{ \sum_{x \in X} \lambda_x x \mid \lambda_x \in \mathbb{F}_2 \right\},$$

endowed with the componentwise addition:  $\sum_x \lambda_x x + \sum_x \mu_x x = \sum_x (\lambda_x + \mu_x) x$ . A subset  $Y \subset X$  corresponds to the vector  $\sum_{y \in Y} y$  in the space  $\bigoplus_{x \in X} \mathbb{F}_2 x$ . Inversely, a vector  $\sum_x \lambda_x x$  of the formal sum is the indicator vector of the subset  $Y = \{x \in X \mid \lambda_x = 1\}$ . Remark that the weight of  $Y$ , regarded as a binary vector, corresponds to the cardinality of the set  $Y$ . In what follows, it will be convenient to regard a subset as a binary vector. The space  $\mathcal{P}(X)$  is a  $\mathbb{F}_2$ -linear space of dimension  $|X|$ . A canonical basis is given by the set of all the singleton  $\{x\} = Y$  included in  $X$ . We abusively say that the elements of  $X$  form a basis of this space.

In order to study and classify manifolds, different chain complexes have been introduced. For example, the *cellular homology complex* associated with a tiling of surface  $G = (V, E, F)$  is the chain complex defined on the spaces

$$C_2 = \bigoplus_{f \in F} \mathbb{F}_2 f, \quad C_1 = \bigoplus_{e \in E} \mathbb{F}_2 e, \quad C_0 = \bigoplus_{v \in V} \mathbb{F}_2 v$$

and the boundary maps  $\partial_2$  and  $\partial_1$ , which are the  $\mathbb{F}_2$ -linear applications

$$C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0$$

such that  $\partial_2(f) = \sum_{e \in f} e$  and  $\partial_1(e) = \sum_{v \in e} v$ . The canonical basis of the space  $C_2$  (respectively  $C_1$  and  $C_0$ ) is composed of the faces  $f \in F$  (respectively the edges and the vertices) of the tiling  $G$ . That is to say,  $C_2$  (respectively  $C_1$  and  $C_0$ ) is the power set of  $F$  (respectively  $E$  and  $V$ ). Geometrically, the application  $\partial_2$  sends a face onto the set of edges on its topological boundary and the application  $\partial_1$  sends an edge onto its endpoints. This explains the term boundary applications. Remark that the boundary of a path (*i.e.* its image under  $\partial_1$ ) is composed of its two *terminal points*. We can easily check that the composition of the two boundary applications is zero. In the next section, we introduce the surface codes from this 2-complex.

### 3.2 The CSS code associated with a 2-complex

From the structure of chain complexes, we immediately obtain the orthogonality relations needed to define a CSS code:

**Proposition 3.2.** *Every 2-complex  $C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0$ , based on finite dimensional spaces, defines a CSS code of length  $n = \dim C_1$ , based on the matrices  $\mathbf{H}_X = \text{Mat}(\partial_1, B_1, B_0)$  and  $\mathbf{H}_Z = \text{Mat}(\partial_2, B_2, B_1)^t$ , where  $B_i$  is a basis of the space  $C_i$ .*

*Proof.* Equation (1) can be translated matricially into  $\mathbf{H}_X \mathbf{H}_Z^t = 0$  which proves the orthogonality relations. The length of the resulting quantum code is given by the dimension of  $C_1$  because it is the number of columns of the matrices  $\mathbf{H}_X$  and  $\mathbf{H}_Z$ .  $\square$

The properties of such a CSS code depend on the 2-complex but also on the *dual complex*. Let us recall the definition of this chain complex. It is the 2-complex defined on the dual

spaces  $C_i^*$  (the space of  $\mathbb{F}_2$ -linear forms over  $C_i$ ), and endowed with the transposed applications  $\partial_i^* : C_{i-1}^* \rightarrow C_i^*$ , which sends  $\phi \in C_{i-1}^*$  onto  $\partial_i^*(\phi) = \phi \circ \partial_i$ . This leads to the dual complex

$$C_2^* \xleftarrow{\partial_2^*} C_1^* \xleftarrow{\partial_1^*} C_0^*.$$

In the dual basis, the matrix of the application  $\partial_1^*$  is the matrix  $\mathbf{H}_X^t$  and the matrix of the application  $\partial_2^*$  is the matrix  $\mathbf{H}_Z$ . Based on this representation of the matrices defining a CSS code, Table 2 describe the errors, the syndrome function and the stabilizers with the language of chain complexes. It is the translation of Table 1.

Table 2: Error representation for 2-complex codes.

	X-component	Z-component
error	$E_X \in C_1^*$	$E_Z \in C_1$
syndrome	$s_X = \partial_2^*(E_X) \in C_2^*$	$s_Z = \partial_1(E_Z) \in C_0$
stabilizer	$E_X \in \text{Im } \partial_1^*$	$E_Z \in \text{Im } \partial_2$

Using the linear isomorphism  $C_i \simeq C_i^*$ , which conserves the weight, this dual complex can be regarded in the spaces  $C_i$ :

$$C_2 \xleftarrow{\partial_2^*} C_1 \xleftarrow{\partial_1^*} C_0.$$

More precisely, remark that the space  $C_i$ , of basis  $B_i$ , can be seen as the power set of  $B_i$ , using the structure introduced in the previous section. This allows us to define the  $\mathbb{F}_2$ -linear application  $\partial_i^*$  by

$$\partial_i^*(x) = \sum_{\substack{y \in B_i \\ x \in \partial_i(y)}} y, \quad (2)$$

for all  $x \in B_{i-1}$ .

Let us apply this construction to a cellular homology complex. From Proposition 3.2, the cellular homology complex associated with a tiling  $G = (V, E, F)$  defines a CSS code of length  $n = |E|$ . It is the *surface code* associated with  $G$ . To see that this construction of surface codes coincides with the original definition of Kitaev [20], and Bombin and Martin-Delgado [5], it suffices to remark that the stabilizers, given in Table 2, correspond to the plaquette operators and the site operators introduced by Kitaev. By definition of the cellular homology complex, the space  $\text{Im } \partial_2$  is generated by the faces of the graph. These stabilizers correspond to the plaquette operators. We can see that the space  $\text{Im } \partial_1^*$  corresponds to the site operators by using the definition of the application  $\partial_1^*$  given in Equation (2). Thus, the corresponding stabilizer group is generated by the face operators in  $Z$  and the site operators in  $X$ .

### 3.3 The decoding problem for surface codes

Before coming to a detailed study of the decoding problem for color codes, let us recall the basic ideas involved in the decoding algorithm for surface codes.

By symmetry between the cellular homology complex and its dual, it is sufficient to consider the  $Z$ -component of the error. Indeed, in the special case of cellular homology complexes, the dual complex is the cellular homology complex associated with the dual graph [16]:

$$C_2(G^*) \xrightarrow{\partial_2^{G^*}} C_1(G^*) \xrightarrow{\partial_1^{G^*}} C_0(G^*).$$

This leads to the simplified framework presented in Table 3. In this table, we consider only the  $Z$ -component of the error, denoted by  $x$ . We give a chain complex point of view and a graphical point of view. To work with the  $X$ -component of the error, replace the graph  $G$  by its dual  $G^*$ .

Table 3: Error representation for surface codes.

	2-complex point of view	graphical point of view
error	$x \in C_1$	$x \subset E$ is a subset of the edge set
syndrome	$s = \partial_1(x) \in C_0$	$s \subset V$ is the set of terminal vertices of $x$
stabilizer	$x \in \text{Im } \partial_2$	$x$ is a boundary

Hence, our goal is to recover a set of edges  $x$ , from the knowledge of its set of terminal vertices  $s$ , and it is sufficient to determine  $x$  up to a boundary. Recall that the terminal vertices of a set of edges are the vertices which are reached an odd number of times by an edge. To get a most likely error, that is to say an error of minimum weight, it suffices to choose this set  $x$  with minimum cardinality.

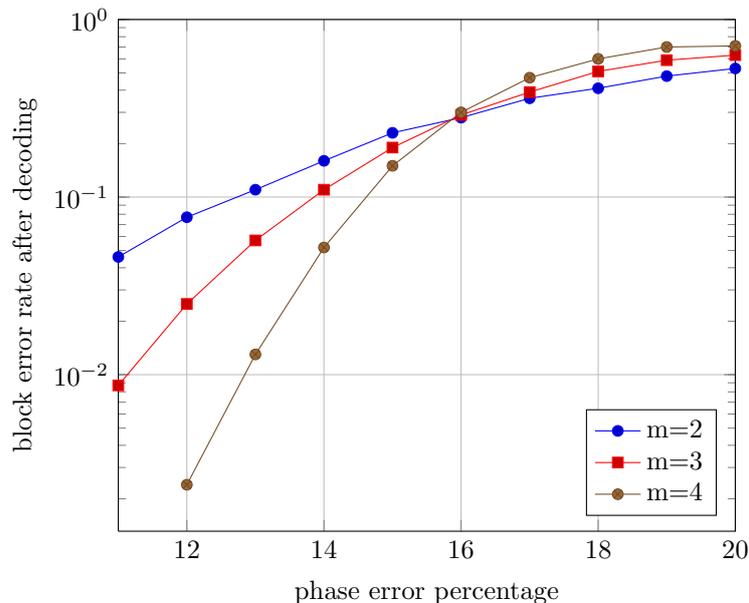


Figure 1: Bloc error rate after decoding as a function of the phase error rate, for hexagonal toric codes of length  $9.2^{2m}$ .

The perfect matching decoding algorithm [10] determines a most likely error by matching pairs of vertices of the syndrome  $s$ , using Edmonds' minimum weight perfect matching algorithm [12, 13]. To illustrate the performance of this decoding algorithm, we consider a family of surface codes based on hexagonal tilings of the torus. Our results are represented in Figure 1. We observe a threshold closed to 15.9%. This threshold will be related to the performance of a family of color codes in section 6.4. We use the implementation "blossom V" of the minimum weight perfect matching algorithm, due to Kolmogorov [21]. The worst case complexity of this decoding algorithm is in  $O(|V|^3|E|)$ , but the typical complexity is definitely better. An improved version of the perfect matching decoding algorithm, with a better complexity, has been proposed recently by Fowler, Whiteside and Hollenberg [14].

The rest of this paper is devoted to the study of the decoding problem for color codes.

## 4 A 2-complex definition of color codes

In this section, we propose a new strategy to decode color codes, which allows us to transform every surface code decoding algorithm into a color code decoding algorithm. The basic idea is to project the error onto three surface codes. These surface codes are constructed using the 3-coloration. Our strategy is illustrated in Figure 5 where the different steps of our decoding algorithm are described in a hexagonal color code. We will refer to this example throughout this section.

Color codes are usually defined from a 3-regular tiling of surface  $G$ , whose faces are 3-colorable [4]. Recall that the faces of a graph are said to be 3-colorable if and only if there exists a 3-coloration of the faces such that two faces sharing an edge do not support the same color. We will consider a chain complex definition of color codes based on the dual tiling  $G^*$ . This point of view will be more appropriate for our decomposition of the decoding problem.

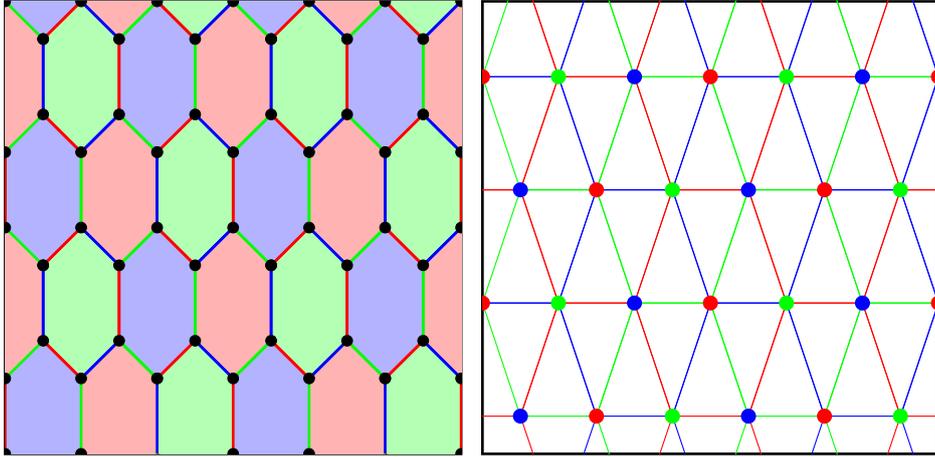


Figure 2: A tiling of the torus with 3-colorable faces and its dual. The vertices of the dual inherit of the 3-coloration.

In what follows,  $G = (V, E, F)$  is a 3-regular tiling of surface. We suppose that the faces of this graph are 3-colorable. As a consequence, the dual graph  $G^* = (V^*, E^*, F^*)$  is equipped with triangle faces and its vertices are 3-colorable (without neighbors supporting the same color). We assume that, these graphs,  $G$  and  $G^*$ , have neither loop nor multiple edge. An example of hexagonal tiling with 3-colored faces and its dual are represented in Figure 2.

### 4.1 Definition of the hypergraph

Our first objective is to recall the definition of color codes with the formalism of chain complexes. This is done by considering the dual tiling  $G^*$  as a hypergraph  $\mathcal{H}$ . By definition, a hypergraph is a pair  $(\mathcal{V}, \mathcal{E})$  where  $\mathcal{V}$  is a finite vertex set and  $\mathcal{E}$  is composed of subsets of  $\mathcal{V}$  (not necessarily of cardinality 2). Just as in the definition of a tiling of surface, the hypergraph is equipped of a set of faces  $\mathcal{F}$ , which is composed of subsets of  $\mathcal{E}$ .

**Definition 4.1.** The hypergraph associated with  $G^*$  is defined to be the hypergraph  $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$  such that:

- The vertex set  $\mathcal{V}$  of  $\mathcal{H}$  is equal to the vertex set  $V^*$  of  $G^*$ .
- The hyperedges are the triples of vertices  $\mathcal{E}(f)$  included in a face  $f$  of  $G^*$ . The set of hyperedges is  $\mathcal{E} = \{\mathcal{E}(f) \mid f \in F^*\}$ .
- The hyperfaces are the sets  $\mathcal{F}(v)$  of hyperedges incident to a vertex  $v$  of  $G^*$ . The set of hyperfaces is  $\mathcal{F} = \{\mathcal{F}(v) \mid v \in V^*\}$ .

Since the graph  $G^*$  is chosen to have triangle faces, the hyperedges always contain 3 vertices. That means that  $\mathcal{H}$  is a 3-uniform hypergraph. These 3 vertices being neighbors, their colors are different. More precisely, every face  $f$  of the graph  $G^*$  is a triple of edges  $f = \{\{u_R, u_G\}, \{u_G, u_B\}, \{u_B, u_R\}\}$ , such that  $u_{\mathbf{c}} \in V^* = \mathcal{V}$  is a vertex of color  $\mathbf{c}$ . The hyperedge  $\mathcal{E}(f) \in \mathcal{E}$  associated with this face is

$$\mathcal{E}(f) = \{u_R, u_G, u_B\}. \quad (3)$$

When the graph  $G^*$  is  $m$ -regular, all the hyperfaces contain  $m$  hyperedges. Let us describe these hyperfaces. The tiling  $G^*$  is a triangulation, therefore the vertex  $v$  has  $m$  neighbors  $v_1, v_2, \dots, v_m$ , ordered such that  $v_i$  and  $v_{i+1}$ , and  $v_m$  and  $v_1$ , are neighbors. The  $m$  hyperedges of  $\mathcal{H}$ , containing the vertex  $u$  are given by  $e_i = \{u, v_i, v_{i+1}\}$ , for  $1 \leq i \leq m-1$ , and  $e_m = \{u, v_m, v_1\}$ . The hyperface  $\mathcal{F}(u)$  is

$$\mathcal{F}(u) = \{e_1, e_2, \dots, e_m\}. \quad (4)$$

The hypergraph associated with the hexagonal tiling of Figure 2 is represented in Figure 3. Its hyperfaces are composed of 6 hyperedges.

## 4.2 Definition of color codes from hypergraph 2-complexes

Using the 2-dimensional structure of the hypergraph  $\mathcal{H}$ , we can define a 2-complex:

**Definition 4.2.** The 2-complex associated with the hypergraph  $\mathcal{H}$  is defined to be the complex composed of the three spaces

$$C_2(\mathcal{H}) = \bigoplus_{f \in \mathcal{F}} \mathbb{F}_2 f, \quad C_1(\mathcal{H}) = \bigoplus_{e \in \mathcal{E}} \mathbb{F}_2 e, \quad C_0(\mathcal{H}) = \bigoplus_{v \in \mathcal{V}} \mathbb{F}_2 v$$

and the boundary maps  $\partial_2^{\mathcal{H}}$  and  $\partial_1^{\mathcal{H}}$ , which are the  $\mathbb{F}_2$ -linear applications

$$C_2(\mathcal{H}) \xrightarrow{\partial_2^{\mathcal{H}}} C_1(\mathcal{H}) \xrightarrow{\partial_1^{\mathcal{H}}} C_0(\mathcal{H})$$

such that  $\partial_2^{\mathcal{H}}(f) = \sum_{e \in \mathcal{E}} e$ , for all  $f \in \mathcal{F}$  and  $\partial_1^{\mathcal{H}}(e) = \sum_{v \in \mathcal{V}} v$ , for all  $e \in \mathcal{E}$ .

From Equation (3), every hyperedge  $e$  of  $\mathcal{H}$  can be write  $e = \{u_R, u_G, u_B\}$ , such that  $u_{\mathbf{c}}$  is a vertex of color  $\mathbf{c}$ . The boundary of this hyperedge is

$$\partial_1^{\mathcal{H}}(e) = u_R + u_G + u_B \in C_0(\mathcal{H}). \quad (5)$$

Using the notation of Equation (4), the boundary of a hyperface  $f = \{e_1, e_2, \dots, e_m\}$  is the sum

$$\partial_2^{\mathcal{H}}(f) = \sum_{i=1}^m e_i \in C_1(\mathcal{H}). \quad (6)$$

As in the case of the cellular homology complex defined in Section 3, a vector of  $C_2(\mathcal{H})$  (respectively  $C_1(\mathcal{H})$  and  $C_0(\mathcal{H})$ ) can be regarded as a subset of  $\mathcal{F}$  (respectively  $\mathcal{E}$  and  $\mathcal{V}$ ). The image of a subset of  $\mathcal{F}$ , by the application  $\partial_2^{\mathcal{H}}$ , is the set of hyperedges included in an odd number of hyperfaces this subset. Similarly, the image of a subset of hyperedge  $x \subset \mathcal{E}$ , by the application  $\partial_1^{\mathcal{H}}$ , is the set of vertices of  $\mathcal{H}$  that are contained in an odd number of hyperedges of  $x$ . By analogy with the cellular homology complex, these vertices are called the *terminal vertices* of  $x$ .

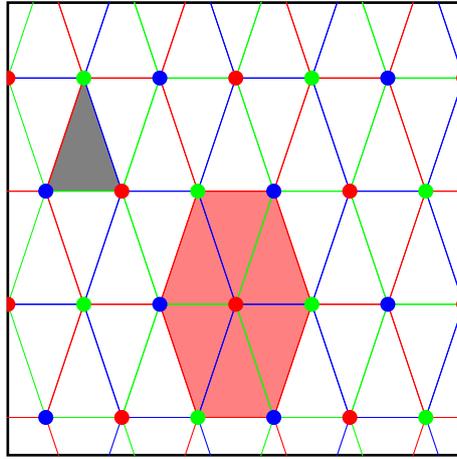


Figure 3: In grey, a hyperedge of the hypergraph  $\mathcal{H}$  associated with an hexagonal tiling of the torus. In red, a hyperface of this hypergraph.

This sequence of vector spaces is a chain complex because the composition of the boundary maps  $\partial_1^{\mathcal{H}}$  and  $\partial_2^{\mathcal{H}}$  is zero. Indeed, it suffices to check that the image of a hyperface  $\mathcal{F}(v)$  by  $\partial_1^{\mathcal{H}} \circ \partial_2^{\mathcal{H}}$  is zero. This proves that  $\partial_1^{\mathcal{H}} \circ \partial_2^{\mathcal{H}} = 0$  by linearity. From Proposition 3.2, this relation enables us to define a CSS code. This quantum code is called the *color code* associated with the graph  $G$ . This definition is equivalent to the original definition of color codes [4].

### 4.3 The decoding problem for color codes

In this part, we present the decoding problem for color codes. In what follows, the sequence

$$C_2(\mathcal{H}) \xrightarrow{\partial_2^{\mathcal{H}}} C_1(\mathcal{H}) \xrightarrow{\partial_1^{\mathcal{H}}} C_0(\mathcal{H})$$

denotes a 2-complex defining a color code.

As in the case of surface codes, the decoding problem for surface codes can be simplified by exploiting the symmetries between the 2-complex and its dual.

**Lemma 4.3.** *The dual complex of  $C_2(\mathcal{H}) \xrightarrow{\partial_2^{\mathcal{H}}} C_1(\mathcal{H}) \xrightarrow{\partial_1^{\mathcal{H}}} C_0(\mathcal{H})$  is the same complex.*

*Proof.* As explained in Section 3.2, from the isomorphism between the space  $C_i$  and its dual, the dual complex can be regarded over the spaces  $C_i(\mathcal{H})$ :

$$C_2(\mathcal{H}) \xleftarrow{(\partial_2^{\mathcal{H}})^*} C_1(\mathcal{H}) \xleftarrow{(\partial_1^{\mathcal{H}})^*} C_0(\mathcal{H}). \quad (7)$$

From the one-to-one correspondence between  $\mathcal{V}$  and  $\mathcal{F}$ , the space  $C_0(\mathcal{H})$  and  $C_2(\mathcal{H})$  are in bijection, thus we can permute these two spaces in Equation (7). Then, using the definition of the transposed application given in Equation (2), we recover the original complex.  $\square$

Thus the two components of the error can be decoded using the same procedure. In what follows we only consider the  $Z$ -component of the error. Table 4 summarizes the properties of errors on color codes. The 2-complex description of the error is deduced immediately from Table 2 and Definition 4.2.

Table 4: Error representation for color codes.

	2-complex point of view	graphical point of view
error	$x \in C_1(\mathcal{H})$	$x \subset \mathcal{E}$ is a subset of the hyperedge set
syndrome	$s = \partial_1^{\mathcal{H}}(x) \in C_0(\mathcal{H})$	$s \subset \mathcal{V}$ is the set of terminal vertices of $x$
stabilizer	$x \in \text{Im } \partial_2^{\mathcal{H}}$	$x$ is a boundary

The problem of computing a most likely error for a color code can thus be reduced to the determination of a set of hyperedges  $x \subset \mathcal{E}$ , whose terminal vertices are the vertices of  $s$ , for a given set  $s \subset \mathcal{V}$ . Moreover, we are interested in the determination of  $x$  up to the boundaries.

The analogous problem for surface codes, stated in Section 3.3, was solved by using a minimum weight perfect matching algorithm. However, the hypergraph structure makes this problem difficult for color codes. For example, the 3-dimensional matching problem is NP-complete [18]. To decode color codes, our basic idea consists of projecting the error acting on a color code onto three surface codes. Our next goal is to introduce these surface codes and to study the projection onto these codes.

## 5 The projection as a morphism of 2-complexes

In this section, we start with some basic properties of the hypergraph complex. Then, we introduce the three surface codes associated with a color code. Our purpose is to transfer the decoding problem for a color code in these surface codes. Thus we have to transfer the whole 2-complex structure of the color code in the surface codes. To this end we will prove that the projection onto the surface codes is a morphism of 2-complexes.

### 5.1 Relations between the hypergraph complex and the graph complex

One of the main difficulty of this section is that we will deal with several different 2-complexes. The first one is the hypergraph chain complex:

$$C_2(\mathcal{H}) \xrightarrow{\partial_2^{\mathcal{H}}} C_1(\mathcal{H}) \xrightarrow{\partial_1^{\mathcal{H}}} C_0(\mathcal{H})$$

This 2-complex is denoted by  $\mathcal{C}(\mathcal{H})$ . A second one is the cellular homology complex associated with the graph  $G^*$ , denoted by  $\mathcal{C}(G^*)$ :

$$C_2(G^*) \xrightarrow{\partial_2^*} C_1(G^*) \xrightarrow{\partial_1^*} C_0(G^*).$$

This part introduces basic tools to connect these complexes.

The one-to-one correspondence between  $\mathcal{E}$  and  $F^*$  and between  $\mathcal{V}$  and  $V^*$  can be extended to the vectors spaces  $C_i$ :

**Lemma 5.1.** *We have the following equalities:*

- $C_0(\mathcal{H}) = C_0(G^*)$ .
- $C_1(\mathcal{H}) = C_2(G^*)$ ,

The second item of this lemma allows us to apply the transformation  $\partial_2^* : C_2(G^*) \rightarrow C_1(G^*)$ , of the cellular homology complex of  $G^*$ , to every vector of  $x \in C_1(\mathcal{H})$ . In the following lemma, we compute the image of  $x \in C_1(\mathcal{H})$  under  $\partial_2^*$ , when  $x$  corresponds to a hyperedge  $e \in \mathcal{E}$ , and when  $x$  corresponds to the boundary of a hyperface  $\partial_2^{\mathcal{H}}(\mathcal{F}(v))$ .

**Lemma 5.2.** *With the notations of Equation (3) and Equation (4), we have:*

- if  $e = \{u_R, u_G, u_B\}$  is a hyperedge of  $\mathcal{H}$ , then

$$\partial_2^*(e) = \{u_R, u_G\} + \{u_G, u_B\} + \{u_B, u_R\} \in C_1(G^*).$$

Moreover, the edge  $\{u_{\mathbf{c}}, u_{\mathbf{c}'}\}$  has color  $\mathbf{c}'' \neq \mathbf{c}, \mathbf{c}'$ .

- if  $f = \mathcal{F}(u) = \{e_1, e_2, \dots, e_m\}$  is the hyperface of  $\mathcal{H}$ , then

$$\partial_2^*(\partial_2^{\mathcal{H}}(f)) = \{v_1, v_2\} + \{v_2, v_3\} + \dots + \{v_{m-1}, v_m\} + \{v_m, v_1\} \in C_1(G^*),$$

where  $e_i = \{u, v_i, v_{i+1}\}$ , for  $i \in \{1, \dots, m-1\}$ , and  $e_m = \{u, v_m, v_1\}$ .

Moreover, all the edges  $\{v_i, v_{i+1}\}$  and  $\{v_m, v_1\}$  have the same color. It is the color of  $u$ .

*Proof.* In the first equation, the hyperedge  $e \in \mathcal{E}$  is regarded as a face of  $G^*$ . This face contains the 3 edges  $\{u_R, u_G\}$ ,  $\{u_G, u_B\}$  and  $\{u_B, u_R\}$  of  $G^*$ . Then, the definition of  $\partial_2^*$  proves the first item.

To prove the second equality, write  $\partial_2^{\mathcal{H}}(f) = \sum_i e_i$ , and use the linearity of  $\partial_2^*$ . To determine the color of the edges of  $\partial_2^*(\partial_2^{\mathcal{H}}(f))$ , observe that if  $u$  has color  $\mathbf{c}$ , then the vertices  $v_i$  share the two other colors alternatively, *i.e.* with  $color(v_i) \neq color(v_{i+1})$ .  $\square$

From this lemma, we observe that the image of a hyperfaces under  $\partial_2^* \circ \partial_2^{\mathcal{H}}$  is a monochromatic cycle of length  $m$ . In the next section, we construct a new tiling, included in  $G^*$ , using these cycles as faces. For example, we can see in Figure 3, a hyperface (the hexagon in red) composed of 6 hyperedges (triangles). Its image under  $\partial_2^*$  is the red cycle of length 6, which is at the boundary of this hexagon.

## 5.2 The three surface codes associated with a color code

Based on the 3-coloration of the graph  $G^*$ , we will construct three surface codes. The 3-coloration of the vertices of  $G^*$  induces a 3-coloration of the edges. An edge  $\{u, v\}$  of  $G^*$  inherits of the color which is absent from its endpoints  $u$  and  $v$ . We restrict our attention to the subgraph of  $G^*$ , induced by the edges colored with  $\mathbf{c}$ , for  $\mathbf{c} \in \{R, G, B\}$ .

**Definition 5.3.** *The graph  $G^*(\mathbf{c})$  is defined to be the subgraph of  $G^*$  induced by the edges colored with  $\mathbf{c}$ .*

A graph  $G^*$  and its red subgraph  $G^*(R)$  are drawn in Figure 4. The following proposition proves that these subgraphs inherit of the tiling structure of the graph  $G^*$ :

**Proposition 5.4.** *The graph  $G^*(\mathbf{c})$ , equipped with the set of faces of the form  $\partial_2^* \circ \partial_2^{\mathcal{H}}(\mathcal{F}(v))$ , where  $v$  is a vertex colored with  $\mathbf{c}$ , is a tiling of surface.*

*Proof.* We will prove that  $G^*(\mathbf{c})$  is constructed from the tiling of surface  $G^*$  by gluing faces of  $G^*$ . Therefore, it inherits of the structure of tiling of surface of  $G^*$ .

This graph is connected since  $G^*$  is supposed to be connected. From Lemma 5.2, the vectors  $\partial_2^* \circ \partial_2^{\mathcal{H}}(\mathcal{F}(v))$  are well cycles. Now, we have to prove that these cycles of the graph  $G^*(\mathbf{c})$  are obtained by gluing the faces of the tiling  $G^*$ . By definition, a hyperface  $\mathcal{F}(v)$ , is a union of  $\deg(v)$  disjoint faces of  $G^*$  (hyperedges), and a face of  $G^*$  appears in exactly one hyperface of the form  $\mathcal{F}(v)$  such that  $v$  is colored with  $\mathbf{c}$ . This concludes the proof.  $\square$

In the definition of graphs and tilings of surfaces in Section 2.2, we assumed that the graph and its dual contain neither loop nor multiple edges. To avoid these configurations in the graph  $G^*(\mathbf{c})$  and its dual, it suffices to assume that the length of the shortest cycle which is not a boundary is at least 5.

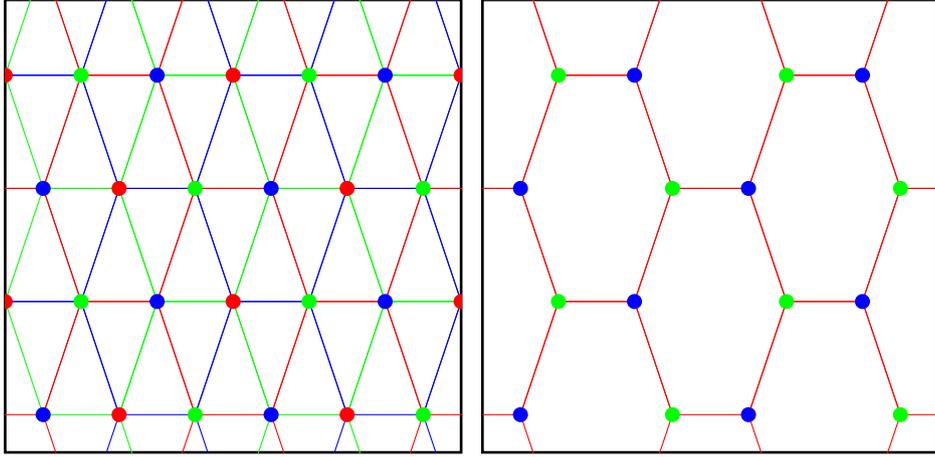


Figure 4: A triangular tiling of the torus  $G^*$  with 3-colored vertices and its subtiling  $G^*(R)$  induced by the red edges.

From such a tiling  $G^*(\mathbf{c})$ , we introduce the cellular homology complex  $\mathcal{C}(G^*(\mathbf{c}))$ :

$$C_2(G^*(\mathbf{c})) \xrightarrow{\partial_2^{G^*(\mathbf{c})}} C_1(G^*(\mathbf{c})) \xrightarrow{\partial_1^{G^*(\mathbf{c})}} C_0(G^*(\mathbf{c})).$$

This defines the three surface codes associated with a color code. The next part introduces the projection onto these surface codes.

### 5.3 The projection onto the three surface codes

Our purpose, in this section, is to introduce a projection from color codes to surface codes and to show that it defines a morphism of 2-complex. This proves that the projection transfer the whole structure of the color code to the surface code.

A natural way to define an application from the hypergraph complex to the graph  $G^*(\mathbf{c})$  is to send the hypergraph onto the graph  $G^*$  and then to restrict ourself to the subgraph  $G^*(\mathbf{c})$ . For example, to define the first projection which sends  $C_1(\mathcal{H})$  onto the space  $C_1(G^*(\mathbf{c}))$ , we send a hyperedge  $e \in \mathcal{E}$  onto its boundary  $\partial_2(e) \in C_1(G^*)$ , which is a set of edges of the graph  $G^*$ . Then, we restrict  $\partial_2(e)$  to the edges of  $G^*(\mathbf{c})$ , that is to say, to the edges colored with  $\mathbf{c}$ . This leads to the following definition.

**Definition 5.5.** *The projection onto the graph  $G^*(\mathbf{c})$  is the triple of operators  $\pi_{\mathbf{c}} = (\pi_{\mathbf{c}}^0, \pi_{\mathbf{c}}^1, \pi_{\mathbf{c}}^2)$  such that*

$$\pi_{\mathbf{c}}^i : C_i(\mathcal{H}) \longrightarrow C_i(G^*(\mathbf{c}))$$

and

$$\begin{aligned} \forall v \in \mathcal{V}, \quad \pi_{\mathbf{c}}^0(v) &= \begin{cases} v & \text{if } \text{color}(v) \neq \mathbf{c} \\ 0 & \text{if } \text{color}(v) = \mathbf{c} \end{cases} \\ \forall e = \{v_R, v_G, v_B\} \in \mathcal{E}, \quad \pi_{\mathbf{c}}^1(e) &= \{v_{\mathbf{c}'}, v_{\mathbf{c}''}\}, \text{ with } \mathbf{c}', \mathbf{c}'' \neq \mathbf{c} \\ \forall f = \mathcal{F}(v) \in \mathcal{F}, \quad \pi_{\mathbf{c}}^2(f) &= \begin{cases} \partial_2^* \circ \partial_2^{\mathcal{H}}(f) & \text{if } \text{color}(v) = \mathbf{c} \\ 0 & \text{if } \text{color}(v) \neq \mathbf{c} \end{cases} \end{aligned}$$

The motivation behind this definition of the projection is the following theorem which proves that the projection conserves the structure of the 2-complexes:

**Theorem 5.6.** *The projection  $\pi_{\mathbf{c}}$  is a morphism of 2-complexes from  $\mathcal{C}(\mathcal{H})$  to  $\mathcal{C}(G^*(\mathbf{c}))$ , i.e. the following diagram is commutative:*

$$\begin{array}{ccccc} C_2(\mathcal{H}) & \xrightarrow{\partial_2^{\mathcal{H}}} & C_1(\mathcal{H}) & \xrightarrow{\partial_1^{\mathcal{H}}} & C_0(\mathcal{H}) \\ \downarrow \pi_{\mathbf{c}}^2 & & \downarrow \pi_{\mathbf{c}}^1 & & \downarrow \pi_{\mathbf{c}}^0 \\ C_2(G^*(\mathbf{c})) & \xrightarrow{\partial_2^{G^*(\mathbf{c})}} & C_1(G^*(\mathbf{c})) & \xrightarrow{\partial_1^{G^*(\mathbf{c})}} & C_0(G^*(\mathbf{c})) \end{array}$$

More precisely, we have  $\partial_2^{G^*(\mathbf{c})} \circ \pi_{\mathbf{c}}^2 = \pi_{\mathbf{c}}^1 \circ \partial_2^{\mathcal{H}}$  and  $\partial_1^{G^*(\mathbf{c})} \circ \pi_{\mathbf{c}}^1 = \pi_{\mathbf{c}}^0 \circ \partial_1^{\mathcal{H}}$ .

*Proof.* Let us prove that  $\partial_1^{G^*(\mathbf{c})} \circ \pi_{\mathbf{c}}^1(x) = \pi_{\mathbf{c}}^0 \circ \partial_1^{\mathcal{H}}(x)$ , for all  $x$  of  $C_1(\mathcal{H})$ . Without loss of generality we can assume that  $\mathbf{c} = R$ . By linearity, it is enough to prove the proposition when  $x$  corresponds to a hyperedge  $e$  of  $\mathcal{H}$ . This hyperedge is a triple  $e = \{v_R, v_G, v_B\}$  of vertices of the hypergraph  $\mathcal{H}$ , the colors of these vertices being indicated by their indices. By Equation (5), the image of  $e$  under  $\partial_1^{\mathcal{H}}$  is the sum  $v_R + v_G + v_B \in C_0(\mathcal{H})$ . Then, the application of  $\pi_R^0$  gives  $\pi_R^0 \circ \partial_1^{\mathcal{H}}(e) = v_G + v_B$ .

Let us compute  $\partial_1^{G^*(R)} \circ \pi_R^1(e)$ , for  $e = \{v_R, v_G, v_B\}$ . By Definition 5.5, the vector  $\pi_R^1(e)$  is  $\{v_G, v_B\}$ . Applying  $\partial_1^{G^*(R)}$  to this vector, we obtain  $\partial_1^{G^*(R)} \circ \pi_R^1(e) = \partial_1^{G^*(R)}(\{v_G, v_B\}) = v_G + v_B$ . This proves that  $\partial_1^{G^*(\mathbf{c})} \circ \pi_{\mathbf{c}}^1 = \pi_{\mathbf{c}}^0 \circ \partial_1^{\mathcal{H}}$ . The second equality can be proved similarly.  $\square$

This Theorem is the key ingredient for our decoding algorithm of color codes. It allows us to transport the 2-complex structure, and therefore the quantum code structure, from the color code to the surface code. This algorithm is described in the next section.

## 6 Decoding color codes

In this section, we consider a color code, associated with a hypergraph complex  $\mathcal{C}(\mathcal{H})$ . From Table 4, an error is a vector  $x \in C_1(\mathcal{H})$ . Our goal is to recover  $x$ , from the knowledge of its syndrome  $s = \partial_1^{\mathcal{H}}(x)$  and it is enough to identify  $x$ , up to the space  $\text{Im } \partial_2^{\mathcal{H}}$  of stabilizers. Our basic idea is to send this error  $x$  on surface codes by using the projections introduced in Section 5.3. Then, we decode this projected error in the surface codes and we lift the result in the color code.

### 6.1 Decoding of the projection of the error

The projection  $\pi_{\mathbf{c}}$  transforms the hypergraph complex  $\mathcal{C}(\mathcal{H})$  in the chain complex  $\mathcal{C}(G^*(\mathbf{c}))$  introduced in Section 5.2. This cellular homology complex defines a surface code. Recall that, from Table 3, an error on this surface code is a vector  $b_{\mathbf{c}} \in C_1(G^*(\mathbf{c}))$ , its syndrome is the vector  $\partial_1^{G^*(\mathbf{c})}(b_{\mathbf{c}})$  and the stabilizers are the vectors of  $\text{Im } \partial_2^{G^*(\mathbf{c})}$ .

The projection  $\pi_{\mathbf{c}}^1(x)$  can be regarded as an error vector on the surface code associated with  $G^*(\mathbf{c})$ . The following proposition enables us to compute its syndrome in the surface code.

**Proposition 6.1.** *Let  $x \in C_1(\mathcal{H})$  be an error for a color code and let  $s \in C_0(\mathcal{H})$  be its syndrome. The projection  $\pi_{\mathbf{c}}^1(x)$  of the error  $x$ , is an error for the surface code associated with  $G^*(\mathbf{c})$  and its syndrome is the projection  $\pi_{\mathbf{c}}^0(s)$ , of the syndrome of  $x$ .*

*Proof.* From the description of the errors, given in Table 3, the projection  $\pi_{\mathbf{c}}^1(x)$  is well an error for this surface codes. Moreover, its syndrome is the vector  $\partial_1^{G^*(\mathbf{c})}(\pi_{\mathbf{c}}^1(x))$ . By Theorem 5.6, we have  $\partial_1^{G^*(\mathbf{c})} \circ \pi_{\mathbf{c}}^1(x) = \pi_{\mathbf{c}}^0 \circ \partial_1^{\mathcal{H}}(x)$ . Since the vector  $\partial_1^{\mathcal{H}}(x)$  is the syndrome of the error  $x$ , we obtain  $\partial_1^{G^*(\mathbf{c})} \circ \pi_{\mathbf{c}}^1(x) = \pi_{\mathbf{c}}^0(s)$ .  $\square$

Thanks to this proposition, we are able to compute the syndrome of a projection  $\pi_{\mathbf{c}}^1(x)$ , using only the measured syndrome  $s$ . Then, we estimate the projection of the error by applying a surface code decoding algorithm in the tiling  $G^*(\mathbf{c})$ . This step is illustrated in Figure 5b and Figure 5c.

### 6.2 Lifting in the color codes

The previous section provides a method to estimate the three projections  $\pi_{\mathbf{c}}(x)$  of the error  $x$  acting on the color code. Our purpose is now to recover the error in the color codes from this information.

First, let us recall a basic property of the chain space  $C_1(G^*)$ .

**Lemma 6.2.**

$$C_1(G^*) = \bigoplus_{\mathbf{c}} C_1(G^*(\mathbf{c})).$$

Denote by  $\tilde{b}_{\mathbf{c}}$  the estimation of the projection  $\pi_{\mathbf{c}}(x)$ , returned by the surface codes decoding algorithm. The previous lemma allows us to construct the vector  $\tilde{b} = \sum_{\mathbf{c}} \tilde{b}_{\mathbf{c}}$  of  $C_1(G^*)$ , from the three estimations of the projection. Our goal is to determine  $x$  from the knowledge of the vector  $\tilde{b}$ .

Our strategy is motivated by the following property of the error.

**Proposition 6.3.** *The vector  $b = \sum_{\mathbf{c}} \pi_{\mathbf{c}}(x) \in C_1(G^*)$  is equal to the vector  $\partial_2^*(x)$ .*

*Proof.* To prove this proposition, let us introduce the projection  $p_{\mathbf{c}} : C_1(G^*) \rightarrow C_1(G^*(\mathbf{c}))$  corresponding to the direct sum decomposition of Lemma 6.2. By definition 5.5, we have  $\partial_{\mathbf{c}}^1 = p_{\mathbf{c}} \circ \partial_2^*$ . Therefore, the vector  $b$  can be written  $b = \sum_{\mathbf{c}} p_{\mathbf{c}} \circ \partial_2^*(x)$ . Then remark that the sum  $\sum_{\mathbf{c}} p_{\mathbf{c}}$  is the identity operator. This proves the proposition.  $\square$

In other words, Proposition 6.3 proves that  $b$  is a boundary in the tiling  $G^*$ . Our strategy is to fill the estimation  $\tilde{b}$  of the vector  $b$ , when it is possible. This corresponds to Figure 5e. When the error has sufficiently low weight, the estimation  $\tilde{b}$  is a boundary and its filling allows us to recover the original error  $x$  for the color code, up to a stabilizer. This fact is proved in the next section.

---

**Algorithm 1** Decoding algorithm for color codes

---

**Require:**  $s \in \text{Im } \partial_1^{\mathcal{H}}$  the syndrome of the error  $x$ .

**Ensure:**  $\tilde{x}$  such that  $\partial_1^{\mathcal{H}}(\tilde{x}) = s$  and  $\partial_2^*(\tilde{x})$  of minimum weight or *NOLIFTING*.

- 1: Compute the projection  $\pi_{\mathbf{c}}^0(s)$  onto  $G^*(\mathbf{c})$  for  $\mathbf{c} \in \{R, G, B\}$ .
  - 2: Decode the surface code associated with  $G^*(\mathbf{c})$ , using the syndrome  $\pi_{\mathbf{c}}^0(s)$ , for  $\mathbf{c} \in \{R, G, B\}$ . This returns the three vectors  $\tilde{b}_{\mathbf{c}}$ . Compute  $\tilde{b} = \sum_{\mathbf{c} \in \{R, G, B\}} \tilde{b}_{\mathbf{c}}$ .
  - 3: Determine  $\tilde{x} \in C_2(G^*)$  such that  $\partial_2(\tilde{x}) = \tilde{b}$ , by filling the boundary  $\tilde{b}$  in the tiling  $G^*$ . If  $\tilde{b}$  cannot be filled return *NOLIFTING*.
- 

The complexity of Algorithm 1, which allows us to decode color codes is polynomial. The most expensive step of this algorithm is the surface decoding algorithm, when we use the perfect matching algorithm. Algorithm 1 could be parallelized by combining the improved perfect matching decoding of Fowler *et al.* [14], with an approximation of the lifting of  $\tilde{b}$ .

### 6.3 Performance of our decoding algorithm for color codes

The following theorem proves that our color codes decoding algorithm corrects an error for the color code when the three surface codes decoding algorithms correct the projected errors.

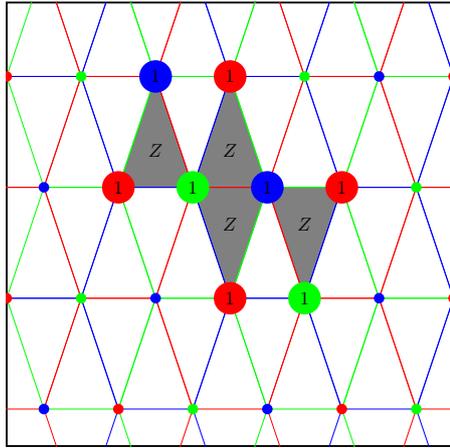
Let  $x \in C_1(\mathcal{H})$  be an error for a color code and let  $\pi_{\mathbf{c}}(x)$  be its surface code projection. Recall that a surface code decoding algorithm allows us to correct an error  $\pi_{\mathbf{c}}^1(x)$ , iff it returns a vector  $\tilde{b}_{\mathbf{c}}$ , which is a vector of the coset of  $\pi_{\mathbf{c}}^1(x)$ , modulo the subspace  $\text{Im } \partial_2^{G^*(\mathbf{c})}$ . Similarly, we say that a color codes decoding algorithm corrects an error  $x \in C_1(\mathcal{H})$  iff it finds a vector  $\tilde{x} \in C_1(\mathcal{H})$ , which is equivalent to  $x$ , modulo the subspace  $\text{Im } \partial_1^{\mathcal{H}}$ .

**Theorem 6.4.** *If the surface code decoding algorithm corrects the projection  $\pi_{\mathbf{c}}^1(x)$  of the error  $x$  for the three colors  $\mathbf{c} = R, G$  and  $B$ , then Algorithm 1 corrects the error  $x$ .*

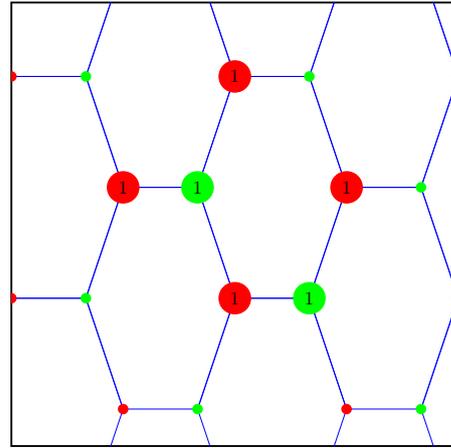
*Proof.* Assume that  $\tilde{b}_{\mathbf{c}}$  is a vector of the coset of  $\pi_{\mathbf{c}}(x)$  modulo the subspace  $\text{Im } \partial_2^{G^*(\mathbf{c})}$  for the three colors  $\mathbf{c}$ . We must prove that the vector  $\tilde{b} = \sum_{\mathbf{c}} \tilde{b}_{\mathbf{c}}$  of  $C_1(G^*)$  belongs to the space  $\text{Im } \partial_2^*$  and that all its preimages  $\tilde{x}$  are in the coset of  $x$  modulo  $\text{Im } \partial_1^{\mathcal{H}}$ . By linearity, it is enough to show that, if  $\tilde{b}_{\mathbf{c}} \in \text{Im } \partial_2^{G^*(\mathbf{c})}$  for all  $\mathbf{c}$ , then  $\tilde{b} \in \text{Im } \partial_2^*$  and its preimages under  $\partial_2^*$  belong to the space  $\text{Im } \partial_1^{\mathcal{H}}$ .

Denote by  $\pi^i$  the linear application

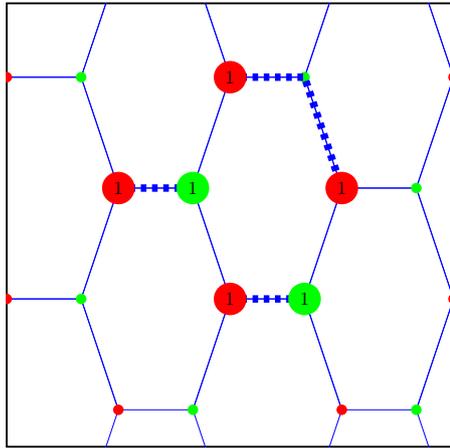
$$\pi^i : C_i(\mathcal{H}) \longrightarrow \prod_{\mathbf{c}} C_i(G^*(\mathbf{c})),$$



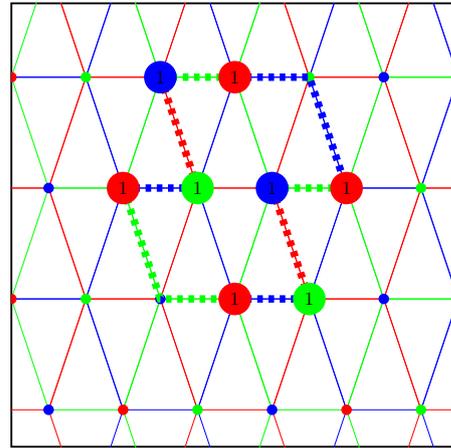
(a) An error  $x$  whose support is the set of grey hyperedges. Its syndrome  $s$  is composed of the vertices supporting a symbol 1.



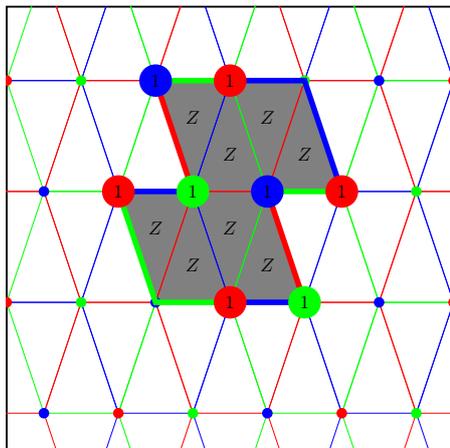
(b) We restrict the syndrome to the blue graph  $G^*(B)$



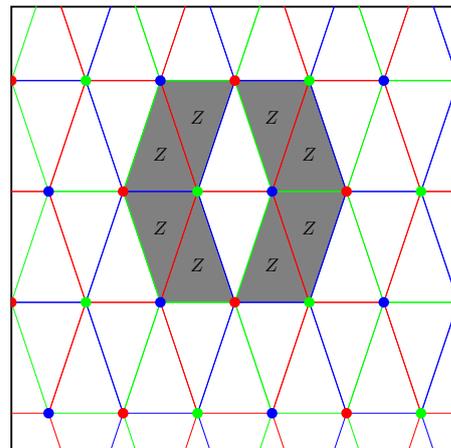
(c) We decode the projection on the surface code  $G^*(B)$ . This defines a set  $\tilde{b}_B$  of dotted blue edges.



(d) We consider the union  $\tilde{b}$  of the three sets  $\tilde{b}_c$  of edges computed in the three surface codes associated with graph  $G^*(c)$ .



(e) The estimation  $\tilde{x}$  of the error is the set of grey edges obtained by filling the cycle  $\tilde{b}$ .



(f) The sum  $x + \tilde{x}$  corresponds to the grey hyperedges. It is a sum of two hyperfaces therefore the vector  $\tilde{x}$  is a good estimation of  $x$ .

which maps  $x$  to the triple  $(\pi_R^i(x), \pi_G^i(x), \pi_B^i(x))$ . From Theorem 5.6, the following diagram is commutative:

$$\begin{array}{ccc} C_2(\mathcal{H}) & \xrightarrow{\partial_2^{\mathcal{H}}} & C_1(\mathcal{H}) \\ \downarrow \pi^2 & & \downarrow \pi^1 \\ \prod_{\mathbf{c}} C_2(G^*(\mathbf{c})) & \xrightarrow{\prod_{\mathbf{c}} \partial_2^{G^*(\mathbf{c})}} & \prod_{\mathbf{c}} C_1(G^*(\mathbf{c})) \end{array}$$

The set  $\prod_{\mathbf{c}} C_1(G^*(\mathbf{c}))$  is in one-to-one correspondence with the direct sum  $\bigoplus_{\mathbf{c}} C_1(G^*(\mathbf{c}))$ . Then, By Lemma 6.2, it is also in one-to-one correspondence with the space  $C_1(G^*)$ . Denote by  $\phi : C_1(G^*) \rightarrow \prod_{\mathbf{c}} C_1(G^*(\mathbf{c}))$  this bijection. Then, the application  $\phi \circ \partial_2^*$  coincides with the application  $\partial_1$ . Thus, it is sufficient to show that, if we have  $\tilde{b}_{\mathbf{c}} \in \text{Im } \partial_2^{G^*(\mathbf{c})}$  for all  $\mathbf{c}$ , then the triple  $(\tilde{b}_R, \tilde{b}_G, \tilde{b}_B)$  admits a preimage under  $\pi^1$ , and that the preimages of this triple are included in the space  $\text{Im } \partial_2^{\mathcal{H}}$ .

Therefore, we assume that  $\tilde{b}_{\mathbf{c}} \in \text{Im } \partial_2^{G^*(\mathbf{c})}$ , for all  $\mathbf{c}$ . For each color  $\mathbf{c}$ , there exists a vector  $\tilde{\beta}_{\mathbf{c}} \in C_2(G^*(\mathbf{c}))$  such that  $\partial_2^{G^*(\mathbf{c})}(\tilde{\beta}_{\mathbf{c}}) = \tilde{b}_{\mathbf{c}}$ . The triple  $(\tilde{\beta}_R, \tilde{\beta}_G, \tilde{\beta}_B)$  is then a preimage of  $(\tilde{b}_R, \tilde{b}_G, \tilde{b}_B)$  under the application  $\prod_{\mathbf{c}} \partial_2^{G^*(\mathbf{c})}$ . To lift the vector  $(\tilde{\beta}_R, \tilde{\beta}_G, \tilde{\beta}_B)$  in the space  $C_2(\mathcal{H})$ , it suffices to remark that the application  $\pi^2$  is an isomorphism. Therefore, there exists a vector  $\alpha \in C_2(\mathcal{H})$  such that

$$\left( \prod_{\mathbf{c}} \partial_2^{G^*(\mathbf{c})} \right) \circ \pi^2(\alpha) = (\tilde{b}_R, \tilde{b}_G, \tilde{b}_B).$$

By commutativity of the diagram drawn below, we have  $\pi^1 \circ \partial_2^{\mathcal{H}}(\alpha) = (\tilde{b}_R, \tilde{b}_G, \tilde{b}_B)$ . This proves that  $\partial_2^{\mathcal{H}}(\alpha)$  is a preimage of  $(\tilde{b}_R, \tilde{b}_G, \tilde{b}_B)$ , under the application  $\pi^1$ . Therefore, the triple  $(\tilde{b}_R, \tilde{b}_G, \tilde{b}_B)$  is in the set  $\text{Im } \partial_2^*$  and its preimage is clearly in the image of  $\partial_2^{\mathcal{H}}$ .

To conclude, let us prove that the other preimages of the vector  $\tilde{b}$  are also included in the space  $\text{Im } \partial_2^{\mathcal{H}}$ . First, consider the kernel of the linear application  $\pi^1$ . If a set  $x$  of hyperedges satisfies  $\pi^1(x) = 0$ , then it corresponds to a set of faces of the graph  $G^*$ , which has an empty boundary. By connexity of the tiling  $G$ , it is either zero or the set of all the faces of  $G^*$ . Thus, we have  $\ker \pi^1 = \{0, \sum_{e \in \mathcal{E}} e\}$ . Consequently, the vector  $\tilde{b}$  has two preimages and that their difference is the vector  $\sum_{e \in \mathcal{E}} e$ . Since this sum is in the image of the application  $\partial_2^{\mathcal{H}}$ , the second preimages of  $\tilde{b}$  is also included in the space  $\text{Im } \partial_2^{\mathcal{H}}$ .  $\square$

For numerical simulations, we consider the hexagonal color codes studied by Sarvepalli and Raussendorf in 2012 [28]. Let us recall the definition of these color codes. Denote by  $H_r$  the Cayley graph of the group  $\mathbb{Z}/(3r)\mathbb{Z} \times \mathbb{Z}/(3r)\mathbb{Z}$  and the generating set:

$$\{\pm(1, 0), \pm(0, 1), \pm(1, -1)\}.$$

By definition, it is the graph whose vertices are the elements of the group and two vertices are joined by an edge iff they differ in an element of the generating set. This graph can be naturally embedded in the torus. It is endowed with triangular faces and its vertices can be 3-colored. The dual tiling of  $H_r$  is a trivalent hexagonal tiling, equipped with 3-colored faces. It defines a color code of parameters  $[[18.r^2, 4, 4r]]$ .

The performance of Algorithm 1 for the color codes based on  $H_r$ , with  $r = 2^m$ , is represented in Figure 6. We observe an error threshold close to 8.7%. The decoding algorithm proposed by Sarvepalli and Raussendorf gave a threshold of 7.8%.

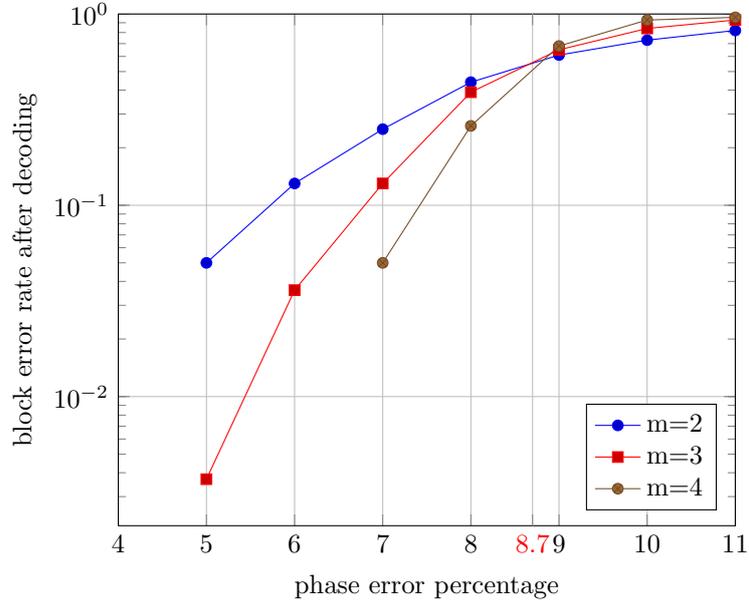


Figure 6: Bloc error rate after decoding as a function of the phase (or bit flip) error rate, for hexagonal color codes of parameters  $[[18.4^m, 4, 4.2^m]]$ .

#### 6.4 Comparison of the thresholds of color codes and surface codes

The projection morphism, studied in Section 5, enables us to transfer results from surface codes to color codes. In this section, we compare the error threshold of surface codes and color codes.

In this paper, we consider independently the bit flip error  $X$  and the phase error  $Z$ . Thus, we study the *phase error threshold* and the *bit flip error threshold*. The phase error threshold of a family of stabilizer codes  $(Q_t)_t$  is defined to be the highest phase error probability, that can be tolerate with vanishing phase error probability after decoding when  $t \rightarrow \infty$ .

By symmetry, the phase error threshold of a family of color codes coincides with its bit flip error threshold. For a family of surface codes, these thresholds can be different. It is the case for example with the surface codes based on the triangular lattice and the hexagonal lattice. The phase error threshold is observed in the original graph and the bit flip error threshold is observed in the dual graph.

Let us consider a family of color codes, denoted by  $(\mathcal{C}_t)_t$ , and the three corresponding surface codes, denoted by  $(\mathcal{C}_t(\mathbf{c}))_t$ , for  $\mathbf{c} \in \{R, G, B\}$ . Assume that we use a surface decoding algorithm and the color decoding algorithm deduced from it using Algorithm 1. Given these decoding algorithms, denote by  $p_c$  the phase error threshold of the family of color codes  $(\mathcal{C}_t)_t$  and denote by  $p_c(\mathbf{c})$  the phase error threshold of the family of surface codes  $(\mathcal{C}_t(\mathbf{c}))_t$ .

**Theorem 6.5.** *The phase error threshold  $p_c$  of a family of color codes is bounded below by a function of the phase error threshold  $p_c(\mathbf{c})$  of the three corresponding families of surface*

codes:

$$p_c \geq \min_{\mathbf{c} \in \{R, V, B\}} \left\{ \frac{1}{2} \left( 1 - \sqrt{1 - 2p_c(\mathbf{c})} \right) \right\}.$$

*Proof.* Consider an error  $x \in C_1(\mathcal{H})$  for the color code and its projection  $\pi_{\mathbf{c}}(x)$ , which is an error in the surface code corresponding to the color  $\mathbf{c}$ . When  $x$  is a random error for a binary symmetric channel of probability  $p$ , then  $\pi_{\mathbf{c}}(x)$  is a random error for a binary symmetric channel of probability  $2p(1-p)$ . To prove this, remark that every qubit of the surface code corresponds to two qubits of the color code. Thus, the probability to have an error on a qubit of the surface code is the probability to have an error on exactly one of the corresponding two qubits in the color code.

If the probability  $2p(1-p)$  is below the threshold  $p_c(\mathbf{c})$  of the three surface codes, then we are able to correct the error  $\pi_{\mathbf{c}}(x)$ , in the three surface codes, with vanishing error probability, using the surface code decoding algorithm. From Theorem 6.4, we can also correct the error  $x$  acting on the color code with vanishing error probability, in this case. This proves that  $p$  is below the phase error threshold of the family of color codes  $(C_t)_t$ . To conclude the proof, it remains only to solve the equation  $2p(1-p) = p_c(\mathbf{c})$ .  $\square$

In the special case of the hexagonal color codes, the three corresponding surface codes are based on an hexagonal tiling of the torus. The performance of the perfect matching algorithm over these surface codes was simulated in Figure 1, where we found a phase error threshold close to 15.9%. The phase error threshold of hexagonal color codes obtained in Figure 6 is approximately 8.7%. These numerical results are in excellent agreement with Theorem 6.5, indeed when  $p_c(\mathbf{c}) \approx 0.159$ , we have  $\frac{1}{2} \left( 1 - \sqrt{1 - 2p_c(\mathbf{c})} \right) \approx 0.0871$ .

## Conclusion

- We proposed a new decoding algorithm for color codes. This algorithm is based on the projection of the error onto three surface codes. This strategy allows us to transform every decoding algorithm for surface codes, such as perfect matching decoding [10], or renormalization group decoding [11] into a decoding algorithm for color codes. Using the perfect matching decoding algorithm for surface codes, we found a threshold of 8.7% for a family of hexagonal color codes. This value is higher than the threshold observed by Sarvepalli and Raussendorf for these codes [28].
- Our idea shares some common features with the method recently proposed by Bombin, Duclos-Cianci and Poulin, which is based on the decomposition of a color codes into multiple copies of toric codes [3]. The point of view developed here is more general because it can be applied to every kind of color code, not only those based on square tilings. Recall that the use of square tilings is a strong restriction, because it makes impossible the construction of positive rate topological codes with growing minimum distance [6, 9]. For algorithmic considerations, we deal with only three surface codes whereas the algorithm of Bombin *et al.* can involve a large number of copies of the toric code.
- From a theoretical point of view, this study of the decoding problem of color codes could be used to transfer results from surface codes to color codes. For example, using the

fact that our algorithm conserves the error model, we connected the threshold of color codes and surface codes in Theorem 6.5.

- One future direction is optimizing our decoding algorithm, which is not always able to estimate the most likely error. Indeed, when the boundary estimation  $\tilde{b}$  is not a vector of  $\text{Im } \partial_2^*$ , Algorithm 1 returns *NOLIFTING*. One could examine these problematic configurations. Further improvement may also be possible by using the correlations between the three projections of the error on the surface codes. Finally, we do not consider the correlation between phase errors and bit flip errors for the depolarizing channel in this paper. The study of these correlations is necessary to approach the optimal threshold of the depolarizing channel [2].

## Acknowledgements

The author wishes to acknowledge Alain Couvreur, Gilles Zémor and Benjamin Smith for careful reading of this paper. This work was supported by the French ANR Defis program under contract ANR-08-EMER-003 (COCQ project). Part of this work was done while the author was supported by the Délégation Générale pour l'Armement (DGA) and the Centre National de la Recherche Scientifique (CNRS) at Bordeaux University. He is now supported by the LIX-Qualcomm Postdoctoral fellowship.

## Appendix A: Lifting of a boundary in a tiling of surface

The goal of this section is to describe an algorithm to determine a vector  $\tilde{x} \in C_2(G^*)$ , such that  $\partial_2^*(\tilde{x}) = \tilde{b}$ , given a vector  $\tilde{b} \in C_1(G^*)$ . In the original graph  $G$ , the vector  $\tilde{x}$  correspond to a set  $X$  of vertices and the vector  $\tilde{b}$  corresponds to a set  $B$  of edges. For  $X \subset V$ , we denote by  $\partial(X)$  the set of edges leaving  $X$ , *i.e.* the set of edges having exactly one end-point in  $X$ . This set  $\partial(X)$  is called the boundary of  $X$ . This leads to the following formulation of the problem: given a set of edges  $B$  of the graph  $G$ , determine a set  $X$  of vertices of the graph  $G$ , whose boundary  $\partial(X)$  is equal to  $B$ .

First, we must be able to distinguish the connected components of the graph  $G$ , limited by the edges of  $B$ . The subgraph of  $G$ , induced by the edges of  $E \setminus B$ , is denoted by  $G_{B^c}$ . We are interested in the connected components of the graph  $G_{B^c}$ .

**Definition A.1.** *Let  $C_1, C_2, \dots, C_\kappa$  be the connected components of the graph  $G_{B^c}$ . The components graph of  $G_{B^c}$  is the graph of vertex set  $V = \{1, 2, \dots, \kappa\}$  whose edges are the pairs of vertices  $\{i, j\}$ , such that the subgraphs  $C_i$  and  $C_j$  of  $G$  are linked by an edge of  $B$ .*

To find the desired set of vertices  $X$ , we choose some of the connected components of the graph  $G_{B^c}$ . We must be careful to not take two neighbors components. Otherwise, we do not get the edges of  $B$  which separate these components in the boundary  $\partial(X)$ . The next proposition characterizes the sets  $B$ , that can be lifted into a satisfying set  $X$ , and gives a method to construct the lifting  $X$ .

**Proposition A.2.** *Let  $C_{B^c}$  be the components graph of  $G_{B^c}$ . Then, there exists a set  $X \subset V$  such that  $\partial(X) = B$  if and only if the graph  $G_{B^c}$  is a bipartite graph. Moreover, if  $G_{B^c}$  is a*

bipartite graph, denote by  $V_1 \cup V_2$  the corresponding partition of the vertices. Then the sets  $X \subset V$  such that  $\partial(X) = B$  are the two sets

$$X_i = \{v \in V \mid v \in C_j \text{ and } j \in V_i\}, \quad (8)$$

for  $i = 1$  and  $i = 2$ .

*Proof.* Suppose that the set  $B$  admits a preimage  $X$  such that  $\partial(X) = B$ . Let us prove that we can partition the vertices of the components graph  $G_{B^c}$ . Let  $C_i$  and  $C_j$  be two connected components of the graph  $G_{B^c}$ , joined by an edge  $e = \{u, v\}$  of  $B$ , i.e.  $u \in C_i$  and  $v \in C_j$ . The set  $X$  contains exactly one of the end-points of  $e$ . Otherwise, the edge  $e \in B$  does not belong to the set  $\partial(X)$ . Assume that  $u \in X$  and  $v \notin X$ . Since the set  $X$  contains a vertex  $u$  of the component  $C_i$ , it also contains all the neighbors of  $x$ , which are included in  $C_i$ . Step by step, this proves that all the vertices of this connected component are in the set  $X$ . Moreover, there is no vertex of the connected component  $C_j$  included in the set  $X$ . Indeed, if  $X$  contains a vertex of  $C_j$ , then it contains the whole component  $C_j$ , and we have  $v \in X$ . This contradicts the fact that  $X$  contains exactly one of the end-points of  $e$ .

Finally, we proved that the set  $X$  is a union of connected components  $C_i$ , such that if we consider two neighbors connected components, it contains exactly one of these two components. This proves that the vector  $X$  defines a partition of the vertices of the components graph corresponding to a bipartite structure. The converse statement can be proved similarly  $\square$

---

#### Algorithm 2 Lifting of a boundary

---

**Require:** a graph  $G = (V, E)$ , a subset  $B \subset E$ .

**Ensure:** a subset  $X \subset V$  such that  $\partial(X) = B$ , or *NOLIFTING* when such a set  $X$  does not exist.

- 1: Compute the connected components of the graph  $G_{B^c}$ .
  - 2: Define the components graph  $C_{BC}$  of the graph  $G_{B^c}$ .
  - 3: Compute a bipartite decomposition  $V_1 \cup V_2$  of  $C_{BC}$ . If this graph is not bipartite return *NOLIFTING*.
  - 4: Return  $X_1 = \{v \in V \mid v \in C_j \text{ and } j \in V_1\}$ .
- 

From this proposition, we are able to determine if a set  $B \subset E$  has a lifting  $X \subset V$  such that  $\partial(X) = B$ , and we can compute efficiently this lifting  $X$ , when such a lifting exists. This result leads to Algorithm 2. The complexity of this algorithm is quadratic due to the determination of the connected components of a graph [17].

## References

- [1] AUDOUX, B. An application of Khovanov homology to quantum codes. arXiv:1307.4677 (2013).
- [2] BOMBIN, H., ANDRIST, R. S., OHZEKI, M., KATZGRABER, H. G., AND MARTIN-DELGADO, M. Strong resilience of topological codes to depolarization. *Physical Review X* 2, 2 (2012), 021004.

- [3] BOMBIN, H., DUCLOS-CIANCI, G., AND POULIN, D. Universal topological phase of two-dimensional stabilizer codes. *New Journal of Physics* 14, 7 (2012), 073048.
- [4] BOMBIN, H., AND MARTIN-DELGADO, M. Topological quantum distillation. *Physical Review Letters* 97 (2006), 180501.
- [5] BOMBIN, H., AND MARTIN-DELGADO, M. Homological error correction: Classical and quantum codes. *Journal of Mathematical Physics* 48, 5 (2007), 052105.
- [6] BRAVYI, S., POULIN, D., AND TERHAL, B. Tradeoffs for reliable quantum information storage in 2D systems. *Physical Review Letters* 104, 5 (2010), 50503.
- [7] CALDERBANK, A., AND SHOR, P. Good quantum error-correcting codes exist. *Physical Review A* 54, 2 (1996), 1098.
- [8] COUVREUR, A., DELFOSSE, N., AND ZÉMOR, G. A construction of quantum LDPC codes from Cayley graphs. *Information Theory, IEEE Transactions on* 59, 9 (2013), 6087–6098.
- [9] DELFOSSE, N. Tradeoffs for reliable quantum information storage in surface codes and color codes. To appear in *Proc. of IEEE International Symposium on Information Theory, ISIT 2013*. (2013) arXiv:1301.6588
- [10] DENNIS, E., KITAEV, A., LANDAHL, A., AND PRESKILL, J. Topological quantum memory. *Journal of Mathematical Physics* 43 (2002), 4452.
- [11] DUCLOS-CIANCI, G., AND POULIN, D. A renormalization group decoding algorithm for topological quantum codes. In *Proc. of IEEE Information Theory Workshop, ITW 2010* (2010), pp. 1–5.
- [12] EDMONDS, J. Maximum matching and a polyhedron with 0-1 vertices. *Journal of Research at the National Bureau of Standards* 69B (1965), 125–130.
- [13] EDMONDS, J. Path, trees, and flowers. *Canadian Journal of Mathematics* 17 (1965), 449–467.
- [14] FOWLER, A., WHITESIDE, A., AND HOLLENBERG, L. Towards practical classical processing for the surface code. *Physical Review Letters* 108 (2012), 180501.
- [15] FREEDMAN, M., AND HASTINGS, M. Quantum systems on non- $k$ -hyperfinite complexes: A generalization of classical statistical mechanics on expander graphs. arXiv:1301.1363 (2013).
- [16] HATCHER, A. *Algebraic topology*. Cambridge University Press, 2002.
- [17] HOPCROFT, J., AND TARJAN, R. Algorithm 447: Efficient algorithms for graph manipulation. *Communications of the ACM* 16, 6 (1973), 372–378.
- [18] JOHNSON, D., AND GAREY, M. Computers and intractability: A guide to the theory of NP-completeness. *Freeman&Co, San Francisco* (1979).

- [19] KATZGRABER, H. G., BOMBIN, H., AND MARTIN-DELGADO, M. Error threshold for color codes and random three-body ising models. *Physical review letters* 103, 9 (2009), 090501.
- [20] KITAEV, A. Fault-tolerant quantum computation by anyons. *Annals of Physics* 303, 1 (2003), 27.
- [21] KOLMOGOROV, V. Blossom V: a new implementation of a minimum cost perfect matching algorithm. *Mathematical Programming Computation* 1 (2009), 43–67.
- [22] KOVALEV, A. A., AND PRYADKO, L. P. Quantum Kronecker sum-product low-density parity-check codes with finite rate. *Physical Review A* 88, 1 (2013), 012311.
- [23] LANDAHL, A. J., ANDERSON, J. T., AND RICE, P. R. Fault-tolerant quantum computing with color codes. arXiv:1108.5738 (2011).
- [24] LANG, S. *Algebra*, 3 ed. Graduate Texts in Mathematics. Springer, 2002.
- [25] MACKAY, D. J. C., MITCHISON, G., AND MCFADDEN, P. L. Sparse-graph codes for quantum error correction. *IEEE Transaction on Information Theory* 50, 10 (2004), 2315–2330.
- [26] OHZEKI, M. Accuracy thresholds of topological color codes on the hexagonal and square-octagonal lattices. *Physical Review E* 80, 1 (2009), 011141.
- [27] POULIN, D., AND CHUNG, Y. On the iterative decoding of sparse quantum codes. *Quantum Information & Computation* 8, 10 (2008), 987–1000.
- [28] SARVEPALLI, P., AND RAUSSENDORF, R. Efficient decoding of topological color codes. *Physical Review A* 85, 2 (2012), 022317.
- [29] STEANE, A. Multiple-particle interference and quantum error correction. *Proc. of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 452, 1954 (1996), 2551–2577.
- [30] TILlich, J.-P., AND ZÉMOR, G. Quantum LDPC codes with positive rate and minimum distance proportional to  $n^{1/2}$ ; In *Proc. of IEEE International Symposium on Information Theory, ISIT 2009* (2009), pp. 799–803.
- [31] WANG, D., FOWLER, A., HILL, C., AND HOLLENBERG, L. Graphical algorithms and threshold error rates for the 2d color code. *Quantum Information & Computation* 10, 9 (2010), 780–802.
- [32] WANG, D., FOWLER, A., STEPHENS, A., AND HOLLENBERG, L. Threshold error rates for the toric and planar codes. *Quantum Information & Computation* 10, 5 (2010), 456–469.
- [33] ZÉMOR, G. On Cayley graphs, surface codes, and the limits of homological coding for quantum error correction. In *Proc. of the 2nd International Workshop on Coding and Cryptology, IWCC 2009* (2009), Springer-Verlag, pp. 259–273.