



HAL
open science

Quantum Pin Codes

Christophe Vuillot, Nikolas P. Breuckmann

► **To cite this version:**

| Christophe Vuillot, Nikolas P. Breuckmann. Quantum Pin Codes. 2019. hal-02351417v1

HAL Id: hal-02351417

<https://hal.science/hal-02351417v1>

Preprint submitted on 6 Nov 2019 (v1), last revised 17 Jun 2022 (v2)

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.



Quantum Pin Codes

Christophe Vuillot, Nikolas P. Breuckmann

► **To cite this version:**

| Christophe Vuillot, Nikolas P. Breuckmann. Quantum Pin Codes. 2019. hal-02351417

HAL Id: hal-02351417

<https://hal.archives-ouvertes.fr/hal-02351417>

Submitted on 6 Nov 2019

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.

Quantum Pin Codes

Christophe Vuillot and Nikolas P. Breuckmann

Abstract—We introduce quantum pin codes: a class of quantum CSS codes. Quantum pin codes are a vast generalization of quantum color codes and Reed-Muller codes. A lot of the structure and properties of color codes carries over to pin codes. Pin codes have gauge operators, an unfolding procedure and their stabilizers form multi-orthogonal spaces. This last feature makes them interesting for devising magic-state distillation protocols. We study examples of these codes and their properties.

I. INTRODUCTION

THE REALIZATION of a fault-tolerant universal quantum computer is a tremendous challenge. At each level of the architecture, from the hardware implementation up to the quantum software, there are difficult problems that need to be overcome. Hovering in the middle of the stack, quantum error correcting codes influence both hardware design and software compilation. They play a major role not only in mitigating noise and faulty operations but also in devising protocols to distill the necessary resources granting universality to an error corrected quantum computer [1]. The study and design of quantum error correcting codes is therefore one of the major tasks to be undertaken on the way to universal quantum computation.

A well-studied class of quantum error correcting codes are Calderbank-Shor-Steane codes (CSS codes) [2], [3], which are a kind of stabilizer quantum codes [4], [5]. The advantage of CSS codes is their close connection to linear codes which have been studied in classical coding theory. A CSS code can be constructed by combining two binary linear codes. Roughly speaking, one code performs parity checks in the Pauli X -basis and the other performs parity checks in the Pauli Z -basis. Not any two binary linear codes can be used: it is necessary that any two pairs of code words from each code space have to have even overlap. Common classical linear code constructions, e.g. random constructions, do not sit well with this restriction and can therefore not be applied to construct CSS codes. Several families of CSS codes have been devised based on geometrical, homological or algebraic constructions [6]–[17], however, it is still open which parameters can be achieved.

Besides being able to protect quantum information, quantum error correcting codes must also allow for some mechanism to process the encoded information without lifting the protection. It is always possible to find some operations realizing a desired action on the encoded information but these operations may spread errors in the system. One should restrict themselves to fault-tolerant operations which do not spread errors. For instance, acting separately on each qubit of a code cannot spread single qubit errors to multi-qubit errors. This is called

a transversal gate, but not any code admits such gates. More generally, for many codes in the CSS code family it is possible to fault-tolerantly implement Clifford operations, which are all unitary operations preserving Pauli operators under conjugation. Clifford operations by themselves do not form a universal gate set. Several techniques to obtain a universal gate set, by supplementing the non-Clifford T gate to Cliffords for example, have been devised [18], [19], among which magic state distillation is currently the most promising candidate.

In this work we introduce a new class of CSS codes, which we call *quantum pin codes*. These codes form a large family while at the same time have structured stabilizer generators, namely they form multi-orthogonal spaces. This structure is necessary for codes to admit transversal phase gates and it can be leveraged to obtain codes that can be used within magic state distillation protocols. Moreover the construction of pin codes differs substantially from previous approaches making it an interesting space to explore further.

In section II, after introducing some notations and terminology, we define quantum pin codes, explain their relation to quantum color codes and give some concrete approaches to construct them. In section III, we discuss the conditions for transversal implementation of phase gates on a CSS code and magic state distillation. In section IV, we investigate the properties of pin codes. Finally in section V, we study concrete examples of pin codes obtained from Coxeter groups and chain complexes as well as applications for magic state distillation.

II. PIN CODES

A. Terminology and formalism

Consider $D+1$ finite, disjoint sets, (C_0, \dots, C_D) which we call *levels*. The elements in each of the levels are called *pins*. If a pin c is contained in a set C_j then j is called the *rank* of c . Since all the C_j are disjoint each pin has a unique rank.

Consider a $(D+1)$ -ary relation on the $D+1$ levels C_0, \dots, C_D , that is to say a subset of their Cartesian product $F \subset C_0 \times \dots \times C_D$. The tuples in the relation F will be called *flags*.

A subset of the ranks, $t \subset \{0, \dots, D\}$, is called a *type*. We will consider tuples of pins coming from a subset of the levels selected by a type t and call them *collection* of pins of type t . A collection of pins of type $t = \{j_1, \dots, j_k\}$, is therefore an element $s \in C_{j_1} \times \dots \times C_{j_k}$. Note that we can interchangeably view a collection of pins as a tuple or a set as long as no two pins come from the same level in the set.

We now define specific subsets of flags, called pinned sets, using projections.

C. Vuillot, QuTech, TU Delft, c.vuillot@tudelft.nl. This paper was presented at QEC19 in London.

N. P. Breuckmann, University College London, n.breuckmann@ucl.ac.uk.

Definition 1 (Projection of type t). Given a set of flags F and a type $t = \{j_1, \dots, j_k\}$, the projection, Π_t , is defined as the natural Cartesian product projection acting on the flags, F

$$\begin{aligned} \Pi_t : \quad F &\rightarrow C_{j_1} \times \dots \times C_{j_k} \\ (c_0, \dots, c_D) &\mapsto (c_{j_1}, \dots, c_{j_k}). \end{aligned}$$

Note that the projection of empty type, Π_\emptyset , is also well defined: for any $f \in F$ we have $\Pi_\emptyset(f) = ()$.

Definition 2 (Pinned set). Let F be a set of flags, s be a collection of pins of type t and Π_t be the corresponding projection as defined above. We define the pinned set of type t and collection of pins s , $P_t(s)$, as the preimage of s under the projection Π_t ,

$$P_t(s) = \Pi_t^{-1}(s) \subset F.$$

In words: a pinned set is the set of flags whose projection of a given type t yields a given collection of pins, s . A definition of a pinned set which is equivalent to the one given above is

$$P_t(c_{j_1}, \dots, c_{j_k}) = F \cap C_0 \dots \times \{c_{j_1}\} \times \dots \times \{c_{j_k}\} \times \dots \times C_D. \quad (1)$$

The pinned set with respect to the empty type is none other than the full set of flags, F . For convenience, we will refer to a pinned set defined by a collection with k pins as a k -pinned set.

If one wants to form a mental image one can imagine a pin-board with pins of different colors for each levels on it. Then the flags can be represented by cords each attached to one pin of each level, see Fig. 1 as an example.

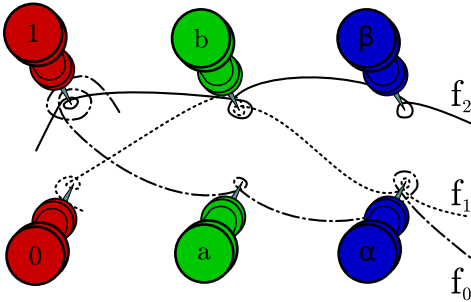


Fig. 1. Illustration of three levels (red, green and blue) each containing two pins and a relation containing three flags (f_0 , f_1 and f_2) symbolized by cords attached to the pins. The pinned set $P_{\text{green}}(b)$ is composed of the flags f_1 and f_2 . The pinned set $P_{(\text{red,blue})}(1, \alpha)$ only contains the flag f_0 .

The structure of pinned sets layed out above is such that they intersect and decompose nicely. This is captured by the following two propositions.

Proposition 1 (Intersection of pinned sets). Let s_1 and s_2 be two collections of pins of types t_1 and t_2 respectively. Then the intersection of the two pinned sets $P_{t_1}(s_1)$ and $P_{t_2}(s_2)$ is either empty or a pinned set of type $t_1 \cup t_2$ characterized by the collection of pins $s_1 \cup s_2$,

$$P_{t_1}(s_1) \cap P_{t_2}(s_2) = \begin{cases} P_{t_1 \cup t_2}(s_1 \cup s_2) & \text{if } |s_1 \cup s_2| = |t_1 \cup t_2| \\ \emptyset & \text{otherwise.} \end{cases}$$

Proof. The proof follows directly from the alternative characterization of pinned sets given in Eq. (1). \square

Proposition 2 (Pinned set decomposition). Let s be a type t collection of pins and let t' be a type containing t , i.e. $t' \supset t$. The pinned set $P_t(s)$ is partitioned into some number, say m , of pinned sets, each characterized by a type t' collections of pins containing s , i.e. $s'_j \supset s$,

$$P_t(s) = \bigsqcup_{j=1}^m P_{t'}(s'_j).$$

Proof. Let s be a type t collection of pins and let t' be a type such that $t' \supset t$. Define the following set of collections of pins

$$S = \Pi_{t'}(P_t(s)),$$

then it is the case that

$$P_t(s) = \bigsqcup_{s' \in S} P_{t'}(s').$$

Indeed, since $t' \supset t$ and $\forall s' \in S, s' \supset s$ we have that $\forall s' \in S, P_{t'}(s') \subset P_t(s)$ showing the right to left inclusion. The left to right inclusion follows from the definition of S . Finally the fact that the union is disjoint follows from Prop. 1. \square

B. Definition of a (x, z) -pin code

Equipped with the notions layed out in the previous section, we now construct quantum codes. They are defined by a choice of flags F and two natural integers x and z which fulfill the condition $x + z \leq D$. The flags are identified with qubits, so that $n = |F|$. The X -stabilizers generators are defined by all the x -pinned sets and the Z -stabilizers generators by the z -pinned sets in the following way. For each x -pinned set we define the Pauli operator acting as the Pauli- X operator on the qubits corresponding to the flags in the pinned set and as the identity on the rest; we then add it to the generating set of the X -stabilizers. We do similarly for the z -pinned sets to form the generating set of the Z -stabilizers. In order to ensure the correct commutation relations between the X - and Z -stabilizers it is sufficient to enforce the following condition on the relation F .

Definition 3 (Pin code relation). A $(D+1)$ -ary relation, F , is a pin code relation if all D -pinned sets have even cardinality.

This property is sufficient for the overlap between two pinned sets to be even every time they are pinned by a small enough number of pins. This is summarized in the following proposition.

Proposition 3 (Even overlap). Let F be a $(D+1)$ -ary pin code relation as per Definition 3. Let x and z be two natural integers such that,

$$x + z \leq D.$$

Then, the intersection between any x -pinned set and any z -pinned set has even cardinality.

Proof. By Proposition 1, the intersection of a x -pinned set and a z -pinned set is either empty or a pinned set with at most $x + z \leq D$ pins. In turn by Proposition 2, the intersection can be partitioned into D -pinned sets which all have even cardinality since the relation F is a pin code relation. \square

Using a pin code relation we can therefore define a CSS code as follows.

Definition 4 ((x, z) -pin code). *Given a pin code relation, F , on $(D + 1)$ sets and two natural integers $(x, z) \in \mathbb{N}^2$, such that $x + z \leq D$, we define the corresponding (x, z) -pin code by associating the elements of F with qubits, all the x -pinned sets with X -stabilizer generators and all the z -pinned sets with Z -stabilizer generators. The defined code is a valid CSS code.*

A first remark is that the choice of $x = 0$ (or $z = 0$) is not particularly interesting since in this case there is a single X -stabilizer (or Z -stabilizer) acting on all the qubits.

A second remark is that in the strict case, where $x + z < D$, the code will contain many small logical operators which are naturally identified as gauge operators. This is explained in details in Sec. IV-C.

Before explaining how to construct pin code relations we show that quantum color codes are a subclass of pin codes.

C. Relation to quantum color codes

The formalism and definitions above can be viewed as a generalization of quantum color codes without boundaries [11], [20], [21]. However it is also possible to integrate the notion of boundaries into the general framework of pin codes but we delay these considerations to Sec. IV-E. A D -dimensional color code is defined by a homogeneous, D -dimensional, simplicial complex, triangulating a D -manifold, whose vertices are $(D + 1)$ -colorable. A D -dimensional, simplicial complex is called homogeneous if every simplex of dimension less than D is a face of some D -simplex. The (Poincaré) dual of a simplicial complex as defined above is sometimes called a colex [20], it consists in a tessellation where the vertices are $(D + 1)$ -valent and the D -cells are $(D + 1)$ -colorable. In the original simplicial complex, the qubits are identified with the D -simplices and one chooses two natural integers, $(\tilde{x}, \tilde{z}) \in \mathbb{N}^2$ with $\tilde{x} + \tilde{z} \leq D - 2$, to define the X - and Z -checks using the \tilde{x} - and \tilde{z} -simplices in the following way. Each X -stabilizer generator is identified with a \tilde{x} -simplex which acts as Pauli- X on all D -simplices in which it is contained. Similarly, each Z -stabilizer generator operate on all the D -simplices as Pauli- Z in which the corresponding \tilde{z} -simplex is contained, respectively.

This definition can be restated in the language of pin codes: the $D + 1$ levels, C_0, \dots, C_D , are indexed by the $D + 1$ colors and each level contains the vertices of a given color. The flags, F , are defined using the D -simplices (each containing $D + 1$ vertices); this defines a relation, $F \subset C_0 \times \dots \times C_D$ thanks to the colorability condition as each D -simplex will not contain two vertices of the same color. One can further check that the relation F is a pin code relation as stated in Def. 3. Indeed, D -pinned sets correspond to $(D - 1)$ -simplices which are contained in exactly two D -simplices since the simplicial-complex triangulates a D -manifold without boundaries. This shows that it is a pin code relation. Then subsets of the $(D + 1)$ colors correspond to types and any k -simplex corresponds directly to a collection of pins of type given by the $(k + 1)$

different colors of the $(k + 1)$ vertices of the k -simplex. The corresponding $(k + 1)$ -pinned set contains all the D -simplices containing the original k -simplex. As such all the non-empty $(k + 1)$ -pinned sets are given by all the collection of pins and type corresponding to all the k -simplices. With these consideration, we see that choosing $x = \tilde{x} + 1$ and $z = \tilde{z} + 1$, the corresponding (x, z) -pin code is the same as the original color code.

An example for $D = 2$, based on the hexagonal color code, is shown in figure 2a. To summarize: to go from color codes to pin codes one just forgets the geometry, keeping only the $(D + 1)$ -ary relation given by the $(D + 1)$ -colored D -simplices.

Importantly, pin codes are more general, as there are pin code relations which are not derived from these specific simplicial complexes. In the next section after recalling a concrete color code construction we give two more general constructions of pin code relations.

D. Constructing pin codes

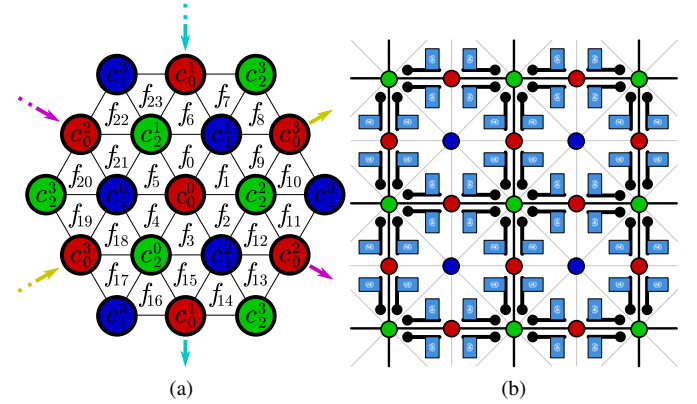


Fig. 2. (a) An example of flag set based on the triangular lattice: $F = \{f_0, \dots, f_{23}\}$ from $D + 1 = 3$ levels: $C_0 = \{c_0^0, \dots, c_0^3\}$ in red, $C_1 = \{c_1^0, \dots, c_1^3\}$ in blue, $C_2 = \{c_2^0, \dots, c_2^3\}$ in green. The figure wraps around on itself according to the arrows so that there are no boundaries, and the surface obtained is a torus. (b) Schematic representation of the flags of the square lattice and the corresponding pins. Each triple of incident vertex (green pin), edge (red pin) and face (blue pin) is a flag. They are symbolized in the picture as actual flags put closest to the elements in the triple (vertex, edge, face) they stand for. The corresponding color code is the well known 4.8.8 color code: there are eight flags around each vertex, four around each edge and eight around each face.

1) *Color codes from tilings:* In [20], the authors explain how to obtain a colex from any tiled D -manifold. The idea is to successively inflate the $(D - 1)$ -cells, $(D - 2)$ -cells, \dots , 0-cells into D -cells. The dual of the tiling obtained is then a $(D + 1)$ -colorable triangulation of the D -manifold, see Appendix A of [20]. This can also be understood directly, without inflating the cells, as follows: Separate all the cells into $(D + 1)$ sets, C_0, \dots, C_D , according to their dimensions, i.e. C_j contains all the j -cells. We can now define a $(D + 1)$ -ary relation on the cells via the incidence relation. Two cells of different dimension are incident if and only if one is a sub-cell of the other. An element of this relation, i.e. a $(D + 1)$ -tuple containing a 0-cell (a vertex), a 1-cell (an edge), etc. \dots up to a D -cell, is called a flag. See for example Fig. 2b for a representation of the flags of the square lattice. The flag

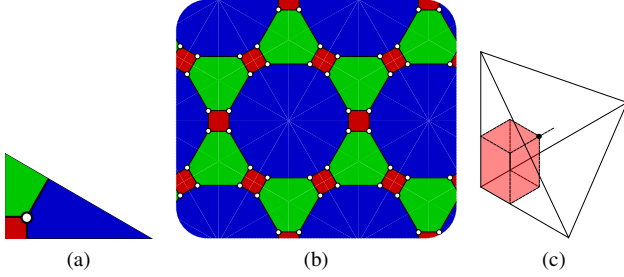


Fig. 3. (a) The fundamental triangle of the Wythoff construction. A vertex (white circle) is placed in the middle and three edges are drawn to the boundary. The three regions are colored red, green and blue. (b) Reflecting the fundamental triangle along its sides creates a three colored lattice. In this case the 4.6.12 lattice. Note that we can obtain the red/blue/green shrunk lattice by moving the vertex into the red/blue/green corner, see Sec. IV-B. (c). The Wythoff construction can be generalized to higher dimensions as well. This example shows the vertex put into the middle of a 3D simplex, a tetrahedron, and one of the four corner cells colored in red (see main text for more information).

relation obtained this way is the same as the one after going through the inflating procedure and it is a pin code relation.

A similar way to construct $(D+1)$ -colorable tessellations in D dimensions directly is to use the Wythoff construction. The construction is quite general, but for simplicity, let us start on the 2D euclidean plane. Consider a right-angled triangle and draw a point into its interior. From this point we draw three lines, each intersecting a boundary edge in a right angle (see Fig. 3a). This creates three regions in the triangle which we assign three different colors. We can now reflect the triangle along its boundary edges. The internal points of the original and the reflected triangles become the vertices of a uniform tiling. The faces of the tiling are colored by the three colors and by construction no two faces of the same color are adjacent (see Fig. 3b). If the angles of the triangle are $2\pi/r$, $2\pi/s$ and $2\pi/l$ then the result will be a $r.s.l$ -tiling, meaning that the three faces around a vertex will have r , s and l number of sides.

This idea readily generalizes to higher dimensions by placing a vertex into a D -dimensional simplex and drawing lines to the mid-point of the $D-1$ -dimensional faces of the simplex (see Fig. 3c for the case $D = 3$). The faces of the simplex are simplices themselves, so this process can be iterated until $D = 2$. Reflecting along the faces of the D -simplex gives rise to a uniform tiling of the D -dimensional space with D -cells being colored by $D+1$ colors and no two cells of the same color sharing a $D-1$ -dimensional face. The number of vertices of the D -cells is then determined by the orbit of the reflections along all but one of the sides.

The color codes from regular tilings can be obtained this way, for example, in 2D Euclidean space, the hexagonal, 4.8.8. or 4.6.12. color codes or more generally both Euclidean and hyperbolic tilings in any dimension.

The classification of what initial simplex can be used, also called fundamental domain, in order to tile the spherical, Euclidean or hyperbolic spaces amounts to studying the groups of symmetries of the tilings. These groups of symmetries are also called Coxeter groups.

2) *Coxeter group approach*: In this section we present how to obtain pin code relations directly from Coxeter groups [22], [23], or more generally from finite groups which are generated by elements with even order. A Coxeter group is a finitely presented group with reflections as generators, denoted as

$$G = \langle a_0, \dots, a_D | a_0^2 = \dots = a_D^2 = (a_i a_j)^{k_{ij}} = r_k = \dots = 1 \rangle,$$

where r_k are additional relations between generators and 1 is the trivial element. Define the subgroups, H_j for $j \in \{0, \dots, D\}$, as

$$H_j = \langle \{a_0, \dots, a_D\} \setminus \{a_j\} \rangle.$$

Define the levels, C_j , as the sets of left cosets for each H_j , i.e.

$$C_j = \{gH_j | g \in G\}.$$

The cosets of a subgroup always form a partition of the full group. So for every $j \in \{0, \dots, D\}$, a group element $g \in G$ uniquely defines a coset $c_j \in C_j$ such that $g \in c_j$. Hence, each group element defines a $(D+1)$ -tuple of cosets, $(c_0, \dots, c_D) \in C_0 \times \dots \times C_D$. Taking the set of all such tuples defines a $(D+1)$ -ary relation on the cosets, $F \subset C_0 \times \dots \times C_D$, which is a pin code relation. The fact that this F is a pin code relation can be verified by the following argument. A k -pinned set here correspond to the intersection of k different cosets with respect to k different subgroups H_j . It always holds that the intersection of several cosets is either empty or is a coset with respect to the intersection of the subgroups of the original cosets. Hence non-empty k -pinned sets are cosets with respect to a subgroup, H_{j_1, \dots, j_k} ,

$$H_{j_1, \dots, j_k} = \bigcap_{i=1}^k H_{j_i}.$$

Each subgroup H_{j_i} is generated by all generators of G except one, this means that H_{j_1, \dots, j_k} contains at least a subgroup generated by $D - k + 1$ of the generators,

$$H_{j_1, \dots, j_t} \supseteq \langle \{a_0, \dots, a_D\} \setminus \{a_{j_1}, \dots, a_{j_t}\} \rangle. \quad (2)$$

In particular the D -pinned sets are cosets with respect to a subgroup which contains $\langle a_j \rangle$ for some j , which has even order since a_j is a reflection. Therefore D -pinned sets have even order. In well behaved cases the containment in Eq. (2) will actually be an equality but this is not guaranteed depending on the relations between generators.

In the case where the Coxeter group describes the symmetries of a tiling this is equivalent to the Wythoff construction described above. But one also obtains more general pin codes when considering Coxeter groups not defining tilings or more general finite groups with generators of even order.

In Sec. V-A we explore in more details the construction of pin codes from 3D hyperbolic Coxeter groups and give some explicit examples.

3) *Chain complex approach*: An other way of obtaining a pin code relation is from \mathbb{F}_2 chain complexes of length $D+1$. These algebraic objects are composed of $(D+1)$ vector spaces over \mathbb{F}_2 , say $\mathcal{C}_0, \dots, \mathcal{C}_D$, together with D linear maps called boundary maps, $\partial_j : \mathcal{C}_j \rightarrow \mathcal{C}_{j-1}$, which are such that

$$\forall j \in \{0, \dots, D-1\}, \partial_j \circ \partial_{j+1} = 0. \quad (3)$$

For example the tiling of a D -manifold can be seen as a chain complex, taking the j -cells as a basis for the \mathcal{C}_j vector space and the natural boundary map. We have shown how to get a pin code relation from such a tiling by taking its flags, but it can as well be obtained from any \mathbb{F}_2 chain complex.

The construction works as follows: choose a basis set \mathcal{C}_j for each vector space \mathcal{C}_j . The \mathcal{C}_j basis sets are the levels and the basis elements the pins. Then use the boundary map, ∂ , to define binary relations, $R_{j,j+1} \subset \mathcal{C}_j \times \mathcal{C}_{j+1}$, where $(c_j, c_{j+1}) \in R_{j,j+1}$ if c_j appears in the decomposition of $\partial(c_{j+1})$ over the basis set \mathcal{C}_j . Then the relation $F \subset \mathcal{C}_0 \times \dots \times \mathcal{C}_D$ is defined as follows

$$F = \{(c_0, \dots, c_D) \mid \forall j, (c_j, c_{j+1}) \in R_{j,j+1}\}.$$

The relation F obtained like this is almost a pin code relation. All the pinned sets of type $t = \{0, \dots, D\} \setminus \{j\}$ with $0 < j < D$ have even cardinality since their size is given by the number of paths between the pin c_{j+1} and the pin c_{j-1} which has to be even by the property of the boundary map ∂ given in Eq. (3). For pinned sets of type $t = \{1, \dots, D\}$ or $t = \{0, \dots, D-1\}$ it is not generally the case that they have even cardinality. Although this can be easily fixed by adding at most two pins: the idea is then to add one rank-0 pin, b_0 , in the level \mathcal{C}_0 and add all pairs (b_0, c^*) such that

$$|\{c_0 \mid (c_0, c^*) \in R_{0,1}\}| = 1 \pmod{2},$$

to the new relation $R_{0,1}$. Then do the same for the level D , adding b_D in \mathcal{C}_D . After this modification the resulting flag relation F is a pin code relation.

Note that this way of obtaining a quantum code from any \mathbb{F}_2 chain complex is fundamentally different from the usual homological code construction. In the homological code construction one chooses one of the levels, say \mathcal{C}_j , and identify its elements with qubits. Then the Z -stabilizer generators are given by the boundary of the elements in \mathcal{C}_{j+1} and the X -stabilizer generators by the coboundary of the elements in \mathcal{C}_{j-1} . These are different from the flags and pinned sets used to define a pin code.

In Sec. V-B we give some explicit pin codes constructed from chain complexes.

E. Remarks

While some flag relations F obtained from Coxeter groups can be equivalently viewed as coming from some \mathbb{F}_2 chain complex, the converse does not necessarily hold. Indeed not every multi-ary relation can be decomposed into a sequence of binary relations, the hexagonal lattice depicted in Fig. 2a is an example of such a relation which cannot be decomposed this way. The other way around, not all flag relations obtained from a \mathbb{F}_2 chain complex can be seen as coming from a Coxeter

groups as in general they would lack the regular structure required.

Depending on the pin code relation, F , it can happen that some pinned sets can in fact be safely split when defining the stabilizers. That is to say, one can separate them into several disjoint sets of flags defining each an independent stabilizer still commuting with the rest of the stabilizers. For example this is the case for Coxeter groups for which (2) is strict, i.e.

$$\langle \{a_{i_1}, \dots, a_{i_s}\} \rangle \cap \langle \{a_{j_1}, \dots, a_{j_t}\} \rangle \supsetneq \langle \{a_{i_1}, \dots, a_{i_s}\} \cap \{a_{j_1}, \dots, a_{j_t}\} \rangle. \quad (4)$$

In this case the cosets with respect to the first group can be further split into cosets with respect to the second one without harming the commutation relations. Groups generated by reflections for which (2) is always an equality are called C-groups [24]. If the stabilizers are still defined as whole pinned sets, in cases where they could be split, then these smaller sets of qubits would be logical operators which would be detrimental to the overall distance of the code.

Another remark is that Reed-Muller codes can be simply expressed using specific pin code relations. This fact is detailed in Appendix A.

III. TRANSVERSAL GATES AND MAGIC STATE DISTILLATION

In this section we present independently of pin codes what structure is desirable for CSS codes to admit transversal phase gates of different levels of the Clifford hierarchy. The presentation here is close in spirit to that of [25], [26]. It is included here to set terminology and for self containment purposes.

Given ℓ binary row vectors, $\mathbf{v}^1, \dots, \mathbf{v}^\ell \in \mathbb{F}_2^n$, we denote their element-wise product as, $\mathbf{v}^1 \wedge \dots \wedge \mathbf{v}^\ell$, its j^{th} entry is given by

$$\left[\bigwedge_{m=1}^{\ell} \mathbf{v}^m \right]_j = [\mathbf{v}^1 \wedge \mathbf{v}^2 \wedge \dots \wedge \mathbf{v}^\ell]_j = v_j^1 v_j^2 \dots v_j^\ell.$$

The hamming weight of a binary vector \mathbf{v} is denoted as $|\mathbf{v}|$, it is given by the sum of its entries. We also define the notions of multi-even and multi-orthogonal spaces:

Definition 5 (Multi-even space). *Given an integer, $\ell \in \mathbb{N}$, a subspace $\mathcal{C} \subset \mathbb{F}_2^n$, is called ℓ -even if all vectors in \mathcal{C} have hamming weight divisible by 2^ℓ :*

$$\forall \mathbf{v} \in \mathcal{C}, |\mathbf{v}| = 0 \pmod{2^\ell}.$$

An equivalent characterization is that for any integer $s \in \{1, \dots, \ell\}$ and any s -tuple of vectors, $(\mathbf{v}^1, \dots, \mathbf{v}^s) \in \mathcal{C}^s$, it holds that

$$|\mathbf{v}^1 \wedge \dots \wedge \mathbf{v}^s| = 0 \pmod{2^{\ell-s+1}}.$$

Definition 6 (Multi-orthogonal space). *Given an integer, $\ell \in \mathbb{N}$, a subspace $\mathcal{C} \subset \mathbb{F}_2^n$, is called ℓ -orthogonal if for any ℓ -tuple of vectors, $(\mathbf{v}^1, \dots, \mathbf{v}^\ell) \in \mathcal{C}^\ell$,*

$$|\mathbf{v}^1 \wedge \dots \wedge \mathbf{v}^\ell| = 0 \pmod{2}.$$

Binary addition is denoted by \oplus , and the following identity can be used to convert binary addition to regular integer addition

$$\bigoplus_{m=1}^r \mathbf{w}^m = \sum_{s=1}^r (-2)^{s-1} \sum_{1 \leq m_1 < \dots < m_s \leq r} \bigwedge_{i=1}^s \mathbf{w}^{m_i}. \quad (5)$$

Using this identity one can show the equivalence of the two characterizations of an ℓ -even space given in Def. 5, and that it is enough to verify the second one on a generating set. Similarly it is enough to verify that a space is ℓ -orthogonal on a generating set.

The single-qubit phase gates are denoted as

$$R_\ell = \begin{pmatrix} 1 & 0 \\ 0 & \omega_\ell \end{pmatrix}, \quad \omega_\ell = e^{i \frac{2\pi}{2^\ell}},$$

where ω_ℓ is the 2^ℓ th root of unity. For instance $R_1 = Z$, $R_2 = S$ and $R_3 = T$ in the usual notations.

A. Weighted polynomials and transversal gates

Given k qubits and some integer ℓ , we consider quantum gates, U_{F_ℓ} , acting diagonally on the computational basis, such that for $\mathbf{x} \in \mathbb{F}_2^k$,

$$U_{F_\ell} |\mathbf{x}\rangle = \omega_\ell^{F_\ell(\mathbf{x})} |\mathbf{x}\rangle,$$

where F_ℓ is a so-called weighted polynomial of the form

$$F_\ell(\mathbf{x}) = \sum_{s=1}^{\ell} 2^{s-1} \sum_{m_1 < \dots < m_s} \alpha_{m_1 \dots m_s} \cdot x_{m_1} \cdots x_{m_s}, \quad (6)$$

with coefficients $\alpha_{m_1 \dots m_s}$ in \mathbb{F}_{2^ℓ} . Any such gate U_{F_ℓ} belongs to the ℓ th level of the Clifford hierarchy [27]. Examples, and generating set for $\ell = 3$, are given in Table I.

TABLE I

WEIGHTED POLYNOMIALS CORRESPONDING TO SOME WELL KNOWN GATE, FOR $\ell = 3$, ARRANGED ACCORDING NUMBER OF QUBITS INVOLVED AND LEVEL OF THE CLIFFORD HIERARCHY.

$U_{F_3} \leftrightarrow F_3(\mathbf{x})$	1st level	2nd level	3rd level
1 qubit	$Z \leftrightarrow 4x_1$	$S \leftrightarrow 2x_1$	$T \leftrightarrow x_1$
2 qubits	-	$CZ \leftrightarrow 4x_1x_2$	$CS \leftrightarrow 2x_1x_2$
3 qubits	-	-	$CCZ \leftrightarrow 4x_1x_2x_3$

The goal is to implement such a gate on the logical level of a quantum error correcting code by the transversal application of some phase gates. Given an $[[n, k, d]]$ quantum CSS code, define G as the $r \times n$ matrix whose rows describe a generating set of the X -stabilizers of the code, and define L as the $k \times n$ matrix whose rows describe a basis for the X -logical operators. The code state in this basis corresponding to $\mathbf{x} \in \mathbb{F}_2^k$ can then be expressed as

$$|\bar{\mathbf{x}}\rangle = \frac{1}{\sqrt{2^r}} \sum_{\mathbf{y} \in \mathbb{F}_2^r} |\mathbf{x}L \oplus \mathbf{y}G\rangle.$$

Applying transversally the gate R_ℓ on this code state, $|\bar{\mathbf{x}}\rangle$, yields

$$R_\ell^{\otimes n} |\bar{\mathbf{x}}\rangle = \frac{1}{\sqrt{2^r}} \sum_{\mathbf{y} \in \mathbb{F}_2^r} \omega_\ell^{|\mathbf{x}L \oplus \mathbf{y}G|} |\mathbf{x}L \oplus \mathbf{y}G\rangle. \quad (7)$$

Using Eq. (5), the power of ω_ℓ above can be rewritten in three parts as

$$|\mathbf{x}L \oplus \mathbf{y}G| = F_\ell(\mathbf{x}) + F'_\ell(\mathbf{y}) + F''_\ell(\mathbf{x}, \mathbf{y}), \quad (8)$$

where we defined

$$F_\ell(\mathbf{x}) = |\mathbf{x}L| \quad (9)$$

$$F'_\ell(\mathbf{y}) = |\mathbf{y}G| \quad (10)$$

$$F''_\ell(\mathbf{x}, \mathbf{y}) = -2 |\mathbf{x}L \wedge \mathbf{y}G| \quad (11)$$

Using again Eq. (5), one can express these three parts as weighted polynomials whose coefficients are given by the different overlap between X -logical operator generators, between X -stabilizer generators or between both, see Appendix B.

Provided that it is possible to cancel the action of $F'_\ell(\mathbf{y})$ and $F''_\ell(\mathbf{x}, \mathbf{y})$ then the resulting operation would correspond to the gate U_{F_ℓ} on the logical qubits of the code. The following two properties of CSS codes are designed to get rid of these two unwanted parts.

Proposition 4 (Exact transversality). *Let \mathcal{C} be a CSS code, given an integer ℓ , the code \mathcal{C} allows for the transversal application of R_ℓ if the following conditions hold:*

- (i) *The X -stabilizers form an ℓ -even space.*
- (ii) *Element-wise products of a X -logical operator and a X -stabilizer always have hamming weight divisible by $2^{\ell-1}$.*

The gate performed at the logical level is then given by the weighted polynomial in Eq. (9).

Indeed the two conditions above exactly give

$$F'_\ell(\mathbf{y}) = F''_\ell(\mathbf{x}, \mathbf{y}) = 0 \pmod{2^\ell},$$

which precisely enforce that the actions of $F'_\ell(\mathbf{y})$ and $F''_\ell(\mathbf{x}, \mathbf{y})$ are trivial. We can also settle for a weaker condition under which the unwanted part is not trivial but belong to the $(\ell - 1)$ th level of the Clifford hierarchy.

Proposition 5 (Quasi-transversality). *Let \mathcal{C} be a $[[n, k, d]]$ CSS code, with $r \times n$ generating matrix, G , for its X -stabilizers and $k \times n$ generating matrix, L , for its X -logical operators. Given an integer ℓ , the code \mathcal{C} allows for the transversal application of R_ℓ up to a $(\ell - 1)$ th-level Clifford correction if the following conditions hold:*

- (i) *The X -stabilizers form an ℓ -orthogonal space.*
- (ii) *For any choice of $s \geq 1$ X -logical operators and $t \geq 1$ X -stabilizers with $s + t \leq \ell$*

$$\left| \bigwedge_{i=1}^s \mathbf{L}^{m_i} \bigwedge_{j=1}^t \mathbf{G}^{n_j} \right| = 0 \pmod{2}.$$

The gate performed at the logical level after correction is then given by the weighted polynomial in Eq. (9).

Indeed, under this condition it follows that, see Appendix B

$$\omega_\ell^{F'_\ell(\mathbf{y}) + F''_\ell(\mathbf{x}, \mathbf{y})} = \omega_\ell^{2\tilde{F}_{\ell-1}(\mathbf{x}, \mathbf{y})} = \omega_{\ell-1}^{\tilde{F}_{\ell-1}(\mathbf{x}, \mathbf{y})}, \quad (12)$$

where $\tilde{F}_{\ell-1}$ is a properly weighted polynomial which defines a $(\ell - 1)$ th-level Clifford correction to be applied. The exact correction is given by the conjugation of $U_{\tilde{F}_{\ell-1}}$ by a decoding

circuit for \mathcal{C} , see [25]. Note that we can also define intermediate conditions so that the correction belongs to the $(\ell - q)$ th level of the Clifford hierarchy for some $1 \leq q \leq \ell$. Here we have just defined the two extreme ones, for which the correction belongs either to the 0th level or the $(\ell - 1)$ th level of the Clifford hierarchy.

B. Magic-state distillation

Given a code which exhibits exact transversality or quasi-transversality it is possible to devise magic-state distillation protocols. We describe briefly a variant here, see also [18], [25], [26], [28]–[31].

A magic state is roughly speaking a state which enables the implementation of some gate on an other unknown state. The most common example is the state $|A\rangle = T|+\rangle$ which can be used to implement a T gate on an unknown qubit state using only a CNOT gate, a measurement and possibly a S correction (see for example Figure 2 of [1]). If one uses a CSS code to encode information then the CNOT gate on the encoded level can be done transversally between two encoded blocks. The main difficulty lies in obtaining an encoded magic state of good quality.

A common protocol consists in concatenating the base code with a distillation code, say of parameters $[[n, k, d]]$, which admits transversal T gates which also correspond to logical T gates on the encoded level. Then using n possibly low fidelity magic states encoded in the base code, one applies a transversal T gate on $|+\rangle$ states at the level of the distillation code. Then measuring the checks of the distillation code conditioning on seeing a trivial syndrome and decoding to the base code one obtains k magic states encoded in the base code of better quality.

Provided that the quality of the initial magic states is not too low, repeating sufficiently many times the protocol will reach any desired accuracy. Then the amount of resources spent will directly depend on the parameters of the distillation code $[[n, k, d]]$. The efficiency of the protocol is often summarized in just one quantity:

$$\gamma = \frac{\log(n/k)}{\log(d)}, \quad (13)$$

since the average number of output distilled magic states at a desired accuracy, ϵ_{out} , per initial noisy magic state is given by $1/O(\log(\epsilon_{\text{out}}^{-1})^\gamma)$. So the smaller γ is, the more efficient the protocol is. Previously conjectured to be at least 1, it has recently been shown that $\gamma < 1$ is achievable [31].

IV. PROPERTIES OF QUANTUM PIN CODES

In this section we examine the properties of pin codes. Since their definition is fairly general, their properties depend on the precise choice of pin code relations F . We stay as general as possible and state precisely when the pin code relations need to be restricted.

A. Code parameters and basic properties

First we investigate the LDPC (Low Density Parity Check) property. A code family is LDPC if it has stabilizer checks

of constant weight and each of its qubits are acted upon by a constant number of checks. For pin codes, both properties depend on the relation F , but it is fairly easy to construct LDPC families. For instance, pin codes based on Coxeter groups with fixed relations between generators and one growing compactifying relation are LDPC, see Sec. V-A. As another example, pin codes from chain complexes with fixed length $D + 1$, sparse boundary map and growing dimension of the levels are LDPC as well.

Let us examine a simple example: choose some $D \in \mathbb{N}$, a set, \mathcal{C} , of size $2m$ for some $m \in \mathbb{N}$ and the complete relation on $D + 1$ copies of \mathcal{C} : $F = \mathcal{C}^{D+1}$. One can easily verify that the relation F is a pin code relation as \mathcal{C} has even cardinality. The number of flags is $n_q = |F| = (2m)^{D+1}$ and the number of x - and z -pinned sets are $n_x = \binom{D+1}{x} \times (2m)^x$ and $n_z = \binom{D+1}{z} \times (2m)^z$. If one considers growing m then, the code would not be LDPC, but more strikingly the ratio of number of stabilizer checks to number of qubits would go to zero. This illustrates that for a fixed D the complete relation is a poor choice, leading to very high rate and very low distance. To get interesting codes, one either needs to vary D , or find some other relations with a number of flags growing significantly slower than the complete relation.

Concerning logical operators, we first note that they have even weight.

Proposition 6 (Logical operators have even weight). *Let F be a pin code relation on $D + 1$ sets and let \mathcal{C} be the associated (x, z) -pin code for $(x, z) \in \{1, \dots, D\}^2$ with $x + z \leq D$. Then the X - and Z -logical operators of \mathcal{C} have even weight.*

Proof. Let the set $L \subset F$ represent a X -logical operator, and let t be a type of size z . Consider the set, S , of collections of pins given by the projection of type t of the set L ,

$$S = \Pi_t(L).$$

For every $s \in S$, the pinned set $P_t(s)$ correspond to a Z -stabilizer and therefore has an even intersection with L . Pinned sets of the same type but defined by two different collections of pins are disjoint. Hence, every element in L appear in exactly one of the pinned sets $P_t(s)$ for some $s \in S$ and so the cardinal of L is even. The proof for Z -logical operators is the same. \square

One can also prove the following general lower bound on the distance of pin codes.

Proposition 7 (Distance at least 4). *Let F be a pin code relation on $D + 1$ sets and let \mathcal{C} be the associated (x, z) -pin code for $(x, z) \in \{1, \dots, D\}^2$ with $x + z \leq D$. Then the distance of \mathcal{C} is at least 4.*

Proof. Using the fact that the distance has to be even from Prop. 6, we just need to verify that it cannot be 2. Let f_1 and f_2 be any two flags in F , they must differ on at least one level $j \in \{0, \dots, D\}$. Pick a type t_x of size x containing j and define the collection of pins $s_x = \Pi_{t_x}(f_1)$ of type t_x . Then the pinned set $P_{t_x}(s_x)$, defining a X -stabilizer, contains f_1 but not f_2 , hence $\{f_1, f_2\}$ cannot define a Z -logical operator. Similarly by defining the type t_z of size z containing j and

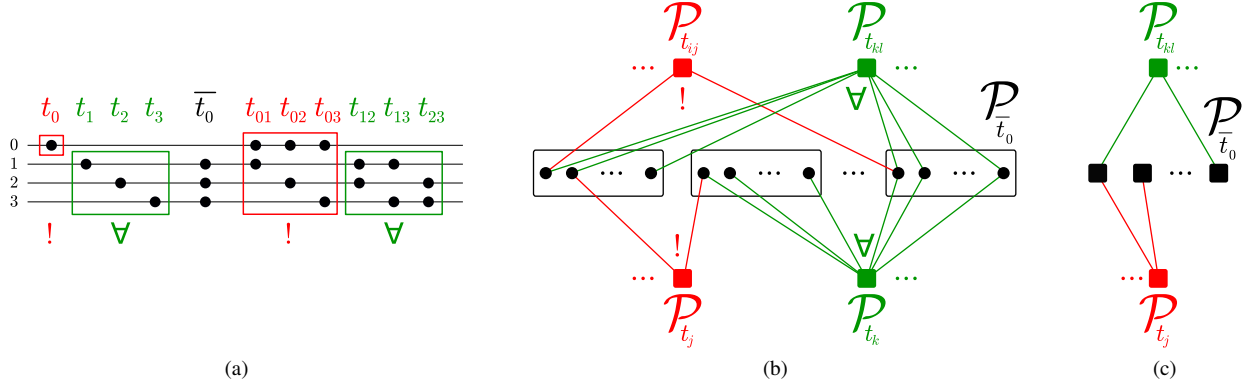


Fig. 4. (a) Schematic representation of the different types for the X - and Z -stabilizers in the case of $D = 3$. The different types are classified according to their possible intersection with the complementary type to t_0 . For some, the intersection, if not empty, is a unique flag, they are labeled by the symbol “!”. For the others, the intersection, if not empty, is the full pinned set of type \bar{t}_0 , they are labeled by the symbol “V”. (b) The chain complex corresponding to the pin code, highlighting the intersections between the different sorts of X - and Z -stabilizers and pinned sets of type \bar{t}_0 . (c) The t_0 -shrunk chain complex derived from the pin code (see main text).

the collection of pins $s_z = \Pi_{t_z}(f_1)$, the pinned set $P_{t_z}(s_z)$ is a witness that $\{f_1, f_2\}$ cannot be a X -logical operator. \square

We conjecture that better lower bounds can be obtained when taking into account the exact size of the type for the stabilizers as larger types can differentiate more flags. The proof above is optimal only for $x = z = 1$. In order to get odd weight logical operators, one has to introduce free pins, see Sec. IV-E. Note that in the presence of free pins, the proof above does not hold anymore.

Making precise statements about the dimension and distance of pin codes is difficult in general. To get closer to be able to do this we need to study the structure of the logical operators.

B. Colored logicals and unfolding

The structure of the logical operators of color codes is understood as colored string-nets or membrane nets [20] and this structure is directly linked to an unfolding procedure existing for color codes [32], [33]. This structure mostly remains for all pin codes, we recast it here.

The general idea is to group qubits into sets with even overlap with all except one sort of stabilizer which will correspond to all stabilizers defined by pinned sets of a given type. Logical operators build out of these sets then only depend on the structure of the one type of stabilizer selected. Repeating this for different choices of type of stabilizer fully covers all logical operators in the case of color codes.

Consider a pin code relation, $F \subset C_0 \times \dots \times C_D$, and the associated (x, z) -pin code. Define the complement of a type, t , denoted as \bar{t} :

$$\bar{t} = \{0, \dots, D\} \setminus t.$$

The intersection between a pinned set of type t and a pinned set of type \bar{t} is either empty or it contains exactly one flag. Furthermore for any another type with the same number of pins as t , the corresponding pinned sets have necessarily even overlap with pinned sets of type \bar{t} , see Figs. 4a and 5a for visual representations of this. This means that grouping flags according to pinned set of the complementary type \bar{t} can single

out logical operators only having to ensure commutation with pinned sets of type t . For our code, X -stabilizers are generated by x -pin sets, which come in $\binom{D+1}{x}$ different types. Take one such type, t_x , and group the qubits according to pinned sets of type \bar{t}_x . Now the Z -stabilizers are generated by z -pinned sets, which come in $\binom{D+1}{z}$ different types. Some of these types, we denote them as $t_z^{\text{inc.}}$, are fully included in \bar{t} , which means that pinned sets of such type fully contain any group of qubits they intersect. The other types only partially intersect with the groups of qubits. The situation is schematized in Fig. 4b for $D = 3$, $x = 1$ and $z = 2$. From these considerations, one can construct a chain complex for which the homology gives candidate Z -logical operators. Take the pinned sets of type t_x , for the level 0, the pinned sets of type \bar{t}_x for the level 1, and the pinned sets of types $t_z^{\text{inc.}}$ for level 2 and the boundary map is given by the overlaps of these sets. This is represented in Fig. 4c, we call it the t_x -shrunk chain complex. Then one can check that an element of the homology of this chain complex can be lifted to a potential Z -logical operator for the pin code. Indeed it would commute with all the X -stabilizer, by homology for the stabilizers of type t_x and by construction for the other X -stabilizers. It would also not be simply generated by Z -stabilizers of type $t_z^{\text{inc.}}$ by homology, and one would have to check for the other Z types. So it is a valid (potentially trivial) Z -logical operator.

The same procedure can be done for each of the X types. Symmetrically, the same can be done for the X -logical with the Z types, and this is represented in Fig. 5 in the case $D = 3$, $x = 1$ and $z = 2$.

Given a type t , the chain complexes constructed like this are called t -shrunk lattices in the case of color codes [20]. For color codes obtained from the Wythoff construction described in Sec. II-D1, the construction of the t -shrunk lattice is fairly direct. First move the vertex from the middle of the fundamental simplex to the corner corresponding to the first rank in the type t , then focus on the opposite face: a simplex of dimension one less which now looks exactly like the beginning of the procedure but in a lower dimension. Recursively exhaust all the ranks of t in this way by each time adding a vertex

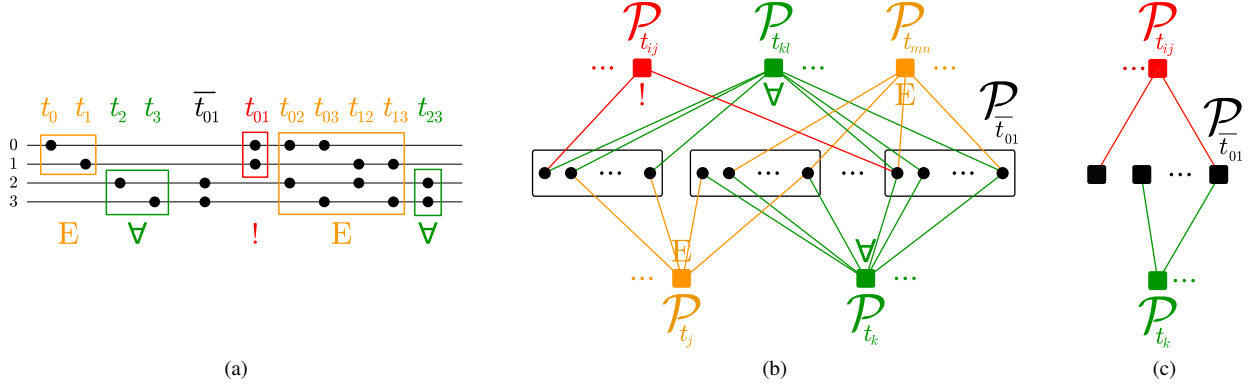


Fig. 5. (a) Schematic representation of the different types classified according to their possible intersection with the complementary type to t_{01} . The ones with unique intersection are labeled with “!”, the ones with even intersection with “E” and the one with full containment with “∇”. (b) The chain complex corresponding to the pin code, highlighting the intersections between the different sorts of X - and Z -stabilizers and pinned sets of type \bar{t}_{01} . (c) The t_{01} -shrunk (co)chain complex derived from the pin code (see main text).

in the middle of the current simplex and moving it to the corresponding corner.

These shrunk lattices are the basis for the unfolding procedure proved for color codes in all dimensions in [32]. This procedure establishes a local unitary equivalence between a color codes and the reunion of the homological codes on the shrunk lattices corresponding to all the different types for X -stabilizers except one. The local unitary acts separately on groups of qubits defined by the X -stabilizer generators of the type that is not used to produce one of the shrunk lattices. The proof of the existence of the local unitary relies on the analysis of the so called overlap groups of stabilizers restricted to the support of the X -stabilizer generators aforementioned and the corresponding groups of qubits in the shrunk lattices. The global structure is still present for general pin codes, but for the proof to hold we need to require that the linear dependency between the generators within the overlap groups in the pin code is such that the number of independent generators agrees with the number of independent generators in the corresponding shrunk lattices as an additional assumption.

These shrunk lattices are also the basis for some color code decoders [34]–[37] but these decoders rely on a lifting procedure from the shrunk lattices to the color code lattice which seems intrinsically geometric as it consists in finding a surface filling inside a boundary. So it is at this point unclear how to leverage this structure in order to decode general pin codes.

C. Gauge pin codes

In this section we define gauge pin codes from a pin code relation. Gauge pin codes can also be viewed as a generalization of gauge color codes [19].

A gauge code, or subsystem code, is a code defined by a so called gauge group instead of a stabilizer group [10], [38]. For stabilizer codes, the code states are eponymously stabilized by the stabilizer group which is an abelian subgroup of the group of Pauli operators. For gauge codes, the gauge group is not abelian and hence all gauge operators cannot share a common $+1$ -eigenspace. In this case the code states are stabilized by

the center of the gauge group. Besides gauge operators not in the center of the gauge group commute with its center and as such would qualify as logical operators in the case of a stabilizer code but are not used to encode information in the case of a gauge code.

Take a pin code relation F and two positive integers x and z such that $x + z < D$. The associated pin code has its X -stabilizer generators defined by all the x -pinned sets and its Z -stabilizers generators defined by all the z -pinned sets. Since the relation F is a pin code relation, by Prop. 3 any $(D - x)$ -pinned set has an even intersection with any x -pinned set. So all the $(D - x)$ -pinned sets correspond to some Z -logical operators. On top of that, they generate all the Z -stabilizers. Indeed using Prop. 2 and the fact that $D - x > z$ one shows that the z -pinned sets decompose into disjoint $(D - x)$ -pinned sets. As such $(D - x)$ -pinned set define naturally Z -gauge operators which can be measured individually and whose outcomes can be recombined to reconstruct the value of the Z -stabilizers defined by z -pinned sets. Symmetrically, the same happens for $(D - z)$ -pinned sets which have even overlap with z -pinned sets and generate x -pinned sets and therefore can be viewed as X -gauge operators.

In conclusion, given a pin code relation F and two natural integers x and z such that $x + z < D$; one defines the corresponding gauge pin code with X -gauge operators defined by the $(D - z)$ -pinned sets and Z -gauge operators by the $(D - x)$ -pinned sets. One can check that these operators do not all commute since $(D - x) + (D - z) > D$. The center of this gauge group, i.e. the stabilizer group, is defined by the x -pinned sets as X -stabilizer generators and z -pinned sets as Z -stabilizer generators. Note that it is not guaranteed that the number of logical qubits in a (x, z) -gauge pin code is the same as the number of logical qubits in the $(x, D - x)$ -pin code obtained from the same relation F .

The error correction procedure for a gauge code with only fully X -type or fully Z -type gauge operators is conveniently performed in two parts. In one part, one measures the X -gauge operators, reconstructs the syndrome for the X -stabilizers and uses it to correct Z -errors. In the other part, one measures the Z -gauge operators, reconstructs the syndrome for the Z -

stabilizers and uses it to correct X -errors.

The advantages of this procedure in the case of gauge pin codes are two-fold. First, the weight of the gauge generators, i.e. the number of qubits involved in each generator, is reduced compared to the weight of the stabilizer generators making their measurement easier and less error prone. Second, the record of gauge operator measurements contains the information of the stabilizer measurements with redundancy. To understand this redundancy consider a x -pinned set and define $k = (D - z) - x$. This is the number of additional levels to pin in order to decompose the x -pinned set into $(D - z)$ -pinned sets. There are $\binom{D+1-x}{k}$ different ways to choose these additional levels to pin and therefore that many different ways to reconstruct the x -pinned set. This redundancy permits a more robust syndrome extraction procedure which can even become in some cases single-shot, meaning that the syndrome measurements do not have to be repeated to reliably decode [39]. Meaning that even when the measurements are noisy one can measure the gauge operators only once and process the obtained information to reduce the noise enough and proceed with the computation.

D. Transversality

We examine here pin codes in regards of Prop. 4 and Prop. 5. Nicely, x -pinned sets always have some multi-orthogonality property.

Proposition 8 (Multi-orthogonality of pinned sets). *Let F be a $(D + 1)$ -ary pin code relation. For any $x \in \{1, \dots, D\}$, the x -pinned sets seen as binary vectors in \mathbb{F}_2^F generate a $\lfloor D/x \rfloor$ -orthogonal space.*

Proof. Given $x \in \{1, \dots, D\}$, by Prop. 1, the intersection of $\lfloor D/x \rfloor$ (or less) x -pinned sets is either empty or a pinned set with at most D pins, hence it has even weight for a pin code relation. \square

Interestingly it is also not too difficult to find pin code relations for which the 1-pin sets are D -even. For example, using a chain complex whose boundary map have even row and column weights and is regular enough will typically suffice.

One could also hope for the second part of proposition 5 to always hold. Unfortunately it holds only partially in general.

Proposition 9 (X -logical intersection with X -stabilizers). *Let F be a $(D + 1)$ -ary pin code relation, and consider the associated (x, z) -pin code for $x \in \{1, \dots, D\}$ and $z = D - x$. Then for any one X -logical operator, \mathbf{L} , and k X -stabilizer generators, \mathbf{G}^j , with $k \leq \lfloor D/x \rfloor - 1$,*

$$\left| \mathbf{L} \wedge \mathbf{G}^1 \wedge \dots \wedge \mathbf{G}^k \right| = 0 \pmod{2}.$$

Proof. Indeed, using Prop. 1, the overlap between $\lfloor D/x \rfloor - 1$ (or less) different x -pinned sets is either empty or a pinned set with at most $D - x = z$ pins. Hence by Prop. 2, it can be decomposed into z -pinned sets, i.e. Z -stabilizers which have even overlap with X -logicals by definition. \square

Overlaps involving more than one X -logical operator do not have such guarantees in general.

Focusing on the case $\ell = 3$, given the two propositions above the only problematic conditions are the ones of type

$$\left| \mathbf{L}^j \wedge \mathbf{L}^k \wedge \mathbf{G}^\ell \right| = 0 \pmod{2}. \quad (14)$$

In order for these terms to hold, one has to have that the intersection of two X -logical operators is always a Z -logical operator. This is the case for example for euclidean color codes.

E. Boundaries and free pins

The geometrical notion of colored boundaries existing for color codes can also be generalized to pin codes. The way to do this is to introduce a specific type of pins which will be called *free pins*.

Consider the chain complex approach to building pin code relations presented in Sec. II-D3. In this construction, it is sometimes necessary to add a rank-0 pin b_0 (in the level C_0) or a rank- D pin b_D (in the level C_D) in order to ensure that the relation F is a pin code relation. The new pin b_0 is linked to all the rank-1 pins which previously where linked to an odd number of rank-0 pins. So even if the initial boundary relation is sparse, the number of connections to b_0 may be large. As such the 1-pinned set pinned by this new pin b_0 potentially contains a large number of flags. To keep the size of the 1-pinned sets under control it is then preferable to not allow to pin b_0 alone. That is why we then call b_0 a free pin. Any of the $D + 1$ levels can contain free pins, the chain complex construction potentially put one in C_0 and one in C_D . The rule for a larger collection of pins is that if it contains at least one non-free pin then it can define a valid pinned set, but if it is composed of only free pins then it is disregarded. Finally consider when a flag is only composed of free pins, in that case this flag will not enter any valid pinned sets. Hence such flags must also be discarded. This is summarized in the following definition.

Definition 7 (Pin code with free pins). *Let F be a pin code relation defined on $D + 1$ levels of pins. Let some of the pins be labeled as free pins. Let x and z be two natural integers such that $x + z \leq D$. The associated (x, z) -pin code is defined as follows: The elements of F containing at least one non-free pin are associated with qubits. All the x -pinned sets defined by a collection of pins containing at least one non-free pin are associated with X -stabilizer generators. All the z -pinned sets defined by a collection of pins containing at least one non-free pin are associated with Z -stabilizer generators.*

As examples we give a representation of Steane's $[[7, 1, 3]]$ code and the $[[4, 2, 2]]$ code as a $(1, 1)$ -pin codes with free pins in Figure 6.

One idea to introduce free pins in every level could be to consider boundary map matrices which are almost sparse except for a small number of row or columns which could be dense. The basis element corresponding to these would then be labeled as free pins in the construction of the pin code relation.

Note that in the presence of free pins, the proof of Prop. 6 can only be reproduced when at least one level selected by

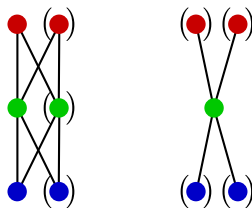


Fig. 6. (Left) Representation of Steane’s $[[7,1,3]]$ code as a $(1,1)$ -pin code from a chain complex with free pins. There are three levels represented by the colors red green and blue. The free pins are represented between parenthesis. There are 8 flags but one is composed only of free pins hence only 7 qubits. There are three non-free pins defining three 1-pinned sets for both X - and Z -stabilizers. (Right) Representation of the $[[4,2,2]]$ code as a $(1,1)$ -pin code with free pins. There are four flags, and a single non-free pin defining the X - and Z -stabilizer both containing the four flags.

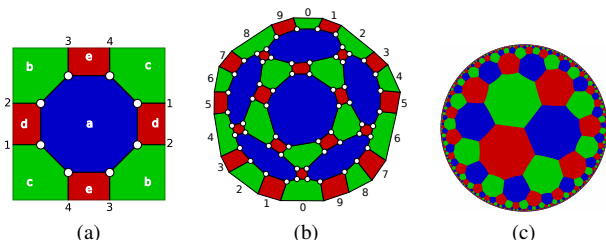


Fig. 7. (a) $[[8,2,2]]$ color code on the projective plane based on octahedral symmetry. The octahedron is topologically a sphere which can be turned into a projective plane by identifying opposite sides. (b) $[[60,2,6]]$ color code on the projective plane based on dodecahedral symmetry. (c) 3-colored hyperbolic tiling with edges and vertices forming a 3-valent graph.

the chosen type t does not contain any free pin. So as long as at least one level does not contain any free pin, it still holds that all logical operators have even weight. When all levels contain at least one free pin then the code may contain odd weight logical operators.

The notion of free pins carries over straightforwardly to gauge pin codes.

V. EXAMPLES AND APPLICATIONS

A. Coxeter groups, hyperbolic color codes

In Section II-D1 we discussed the construction of pin codes from tilings and Coxeter groups. Well-known examples of such code families are color codes on euclidean tilings such as the hexagonal tiling in 2D and the bitruncated cubic honeycomb in 3D. Using the Wythoff construction we can construct tilings which fulfill right pin code condition and therefore have the correct colorability for defining a color code.

Besides the known euclidean examples we can consider tilings of more exotic spaces. For the projective plane (cf. [7]) there exist two tilings based on the Wythoff construction: The first is based on the symmetry group of an octahedron. It is an $[[8,2,2]]$ -code where the check generators correspond to one octagon, two red squares and two green squares, see Figure 7a. Note that this code does not quite fit the pin code definition because it contains distinct qubits which would be described by the same flag, for example (d, c, a) on edge 1. This degeneracy explains why it escapes Prop. 7. The second is based on the icosahedral symmetry group, which gives a

a $[[60,2,6]]$ -code with checks given by 6 decagons (blue), 10 hexagons (green) and 15 squares (red), see Figure 7b.

Color codes based on two-dimensional hyperbolic tilings were first considered in [40] where 3-colorability and 3-valence was postulated (see Figure 7c for an example). The Wythoff construction of Section II-D1 allows us to obtain color codes from arbitrary regular tilings of closed hyperbolic surfaces. To define a family of closed surfaces one needs to compactify the infinite lattice as explained in [41]. There are infinitely many regular tilings of 2D hyperbolic space. The lowest weight achievable with our construction is 4.8.10, meaning that checks are squares, octagons and dodecagons. The smallest code in this family is $[[120,10,6]]$ based on a non-orientable hyperbolic surface (cf. Table 3.1 in [42]). Another small example is a $[[160,20,8]]$ code with stabilizer checks of weight 4 and 10 based on a 4.10.10 tiling of an orientable hyperbolic surface of genus 10.

Using the construction outlined in Section II-D1 we can consider any D -dimensional hyperbolic reflection group and obtain a tiling which is $D+1$ -colorable and which has a $D+1$ -valent graph. In particular, we can consider hyperbolic tilings in 3D which are 4-colorable. There exist four regular hyperbolic tilings in 3D of which two are self-dual tilings and two related by duality. The self-dual ones are a tiling by dodecahedra, denoted $\{5,3,5\}$, and one by icosahedra, denoted $\{3,5,3\}$. The other are a tiling by cubes $\{4,3,5\}$ and its dual $\{5,3,4\}$. All of these give rise to codes with maximum stabilizer weight 120. Here we will focus on the $\{5,3,5\}$ -tiling, which is the unique self-dual tiling of space by dodecahedra where five dodecahedra are placed around an edge. Performing the Wythoff construction on a family of closed manifolds, all equipped with a $\{5,3,5\}$ -tiling yields a code family where checks are of weight 20 and 120. The weight of the stabilizer is given by the order of the subgroup of the full reflection group which is generated by all except for one of the generators. The smallest example is a $[[7200,5526,4]]$ code.

B. Pin codes from chain complexes

In Sec. II-D3 we showed how from any \mathbb{F}_2 chain complex one can construct a pin code relation. In this section we explore some specific examples of chain complexes and the corresponding pin codes.

One way to obtain arbitrary length chain complexes is to use repeatedly the hypergraph product with a classical code. The hypergraph product was introduced in [15] as a way to turn any two classical codes into a quantum code. This product can be viewed as the tensor product of chain complexes, which takes two length-2 chain complexes to a length-3 chain complex. More generally the product of a length- k_1 and length- k_2 chain complexes yields a length- $(k_1 + k_2 - 1)$ chain complex. This generalization and its characteristics has been studied in the context of homological codes [16], [43], [44]. We consider here the approach of [44] but look at the resulting chain complexes from the point of view of pin codes.

The idea goes as follows: consider \mathcal{A} , a \mathbb{F}_2 chain complex of length k , characterized by \mathbb{F}_2 -vector spaces $(\mathcal{A}_j)_{0 \leq j \leq k-1}$

and $(k-1)$ boundary maps $\partial_j^A : \mathcal{A}_j \rightarrow \mathcal{A}_{j-1}$, obeying Eq. (3). We now take the product with a chain complex of length 2. Note that any two vector spaces, \mathcal{B}_1 and \mathcal{B}_0 and any linear map between them $\partial^B : \mathcal{B}_1 \rightarrow \mathcal{B}_0$ defines a length-2 chain complex. The product, $\mathcal{C} = \mathcal{A} \otimes \mathcal{B}$, is defined by $(k+1)$ vector spaces \mathcal{C}_j for $0 \leq j \leq k$,

$$\mathcal{C}_j = (\mathcal{B}_1 \otimes \mathcal{A}_{j-1}) \oplus (\mathcal{B}_0 \otimes \mathcal{A}_j),$$

with the convention that \mathcal{A}_{-1} and \mathcal{A}_k are both the zero vector spaces. And the k boundary maps, $\partial_j^C : \mathcal{C}_j \rightarrow \mathcal{C}_{j-1}$, are defined as

$$\forall u = v \oplus w \in (\mathcal{B}_1 \otimes \mathcal{A}_{j-1}) \oplus (\mathcal{B}_0 \otimes \mathcal{A}_j)$$

$$\partial_j^C(u) = (\mathbb{1}_{\mathcal{B}_1} \otimes \partial_{j-1}^A + \partial^B \otimes \mathbb{1}_{\mathcal{A}_{j-1}})(v) + (\mathbb{1}_{\mathcal{B}_0} \otimes \partial_j^A)(w).$$

One straightforwardly checks that the ∂_j^C are valid boundary maps, i.e. obeying Eq. (3).

Repeatedly taking the product with a length-2 chain complex therefore increase the length of the resulting chain complex each time by one. Moreover any binary matrix defines a valid \mathbb{F}_2 chain complex of length 2 so this approach allows to explore numerically many pin codes.

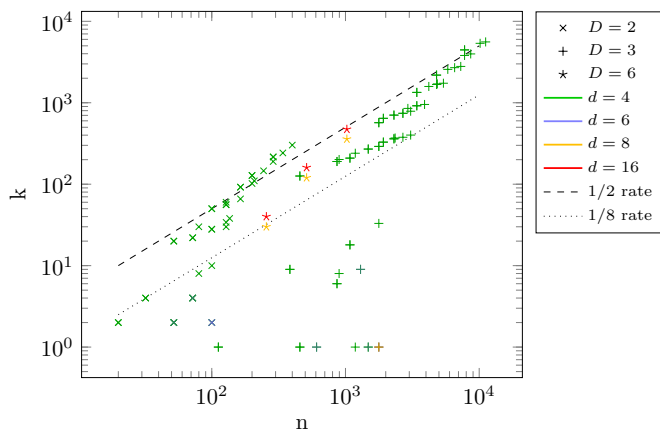


Fig. 8. Plot of the $[[n, k, d]]$ parameters of pin codes from chain complexes described in this section. One application of the hypergraph product on 3×4 binary matrices yields the $D = 2$ pin codes represented by ‘x’ and two applications the $D = 3$ pin codes represented by ‘+’. The $D = 6$ pin codes described in Table. II are represented by stars. The colors indicate the distance of the codes. The dashed and dotted lines represent the rates $k/n = 1/2$ and $k/n = 1/8$, respectively.

We have looked at small binary matrices, up to 3×4 , and their self product to form pin code relations with $D = 2$ and $D = 3$. We plot in Fig. 8 the code parameters obtained $[[n, k, d]]$. Strikingly these codes seem to show a general trend of high encoding rate k/n for a small distance. Indeed most of them are around $1/2$ rate but just distance 4 which is the lower bound guaranteed by Prop. 7. A few of them reach distance 6 or 8 but for significantly smaller rates. The codes yielding no logical qubits are not displayed in this plot. Note that this procedure is far from generating all chain complex of a given length.

We have also looked at a few pin code relations for $D = 6$ using small even size levels and the complete relation for F . Three notable examples are presented in Table. II. When writing $2^{6 \times 6} \times 4$ we mean that 6 of the levels contain each 2

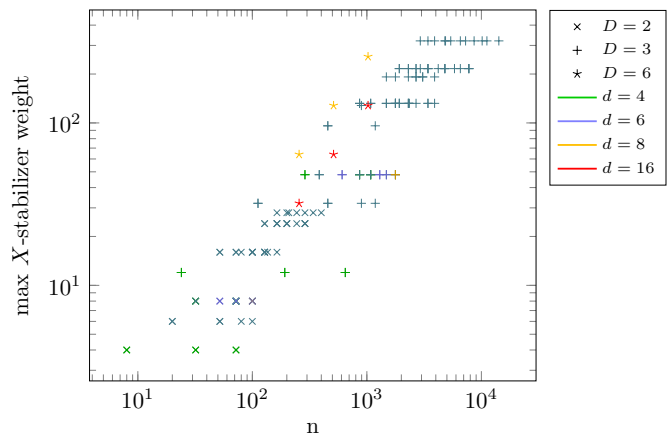


Fig. 9. Plot of the maximum X -stabilizer weight of the pin codes from chain complexes described in this section.

pins and the last one contains 4. Since we use the complete relation, the number of flags and the size of the pinned sets are easily computed as a product of the size of some levels. The number of logical qubits is computed numerically and the distance is upper bounded and we believe is tight.

TABLE II
PARAMETERS OF SOME $D = 6$ PIN CODES USING THE COMPLETE RELATION DESCRIBED BY THE SIZE OF THE $D + 1 = 7$ LEVELS.

(x, z)	$2^{6 \times 6} \times 4$	$2^{5 \times 5} \times 4 \times 2$	$2^{4 \times 4} \times 4 \times 3$
(2, 4)	[[256, 30, 8]]	[[512, 120, 8]]	[[1024, 358, 8]]
(3, 3)	[[256, 40, 16]]	[[512, 160, 16]]	[[1024, 472, 16]]

We also represent the maximum weight of the X -stabilizers for these codes in Fig. 9. When checking for transversal phase gates for $\ell = 3$, most of the codes examined above do not satisfy (14).

C. Puncturing triply-even spaces

If pin codes in general are not guaranteed to fulfill all the requirements of Prop. 4 or Prop. 5, their stabilizers always form multi-orthogonal spaces, see Prop. 8. This is directly useful as multi-orthogonal spaces together with puncturing techniques can be used to construct codes fulfilling Prop. 5 (or Prop. 4 if the space is multi-even), as explained for example in [26]. We focus here on triply-even spaces and tri-orthogonal spaces. The idea goes as follows: take a binary matrix, G , whose rows generate a tri-orthogonal space, using Gaussian elimination it is always possible to put the matrix in the following form:

$$G = \begin{matrix} \leftarrow k \rightarrow & \leftarrow n \rightarrow \\ \begin{matrix} k \updownarrow \\ r \updownarrow \end{matrix} & \left(\begin{array}{cc} \mathbb{1} & G_1 \\ 0 & G_0 \end{array} \right). \end{matrix} \quad (15)$$

To obtain this form one just performs row operations as well as column permutations and different column permutations will yield different G_0 and G_1 . Then choosing the rows of G_0 as X -stabilizer generators and the rows of G_1 as X -logical operators (this fully specifies the Z -stabilizers and Z -logicals)

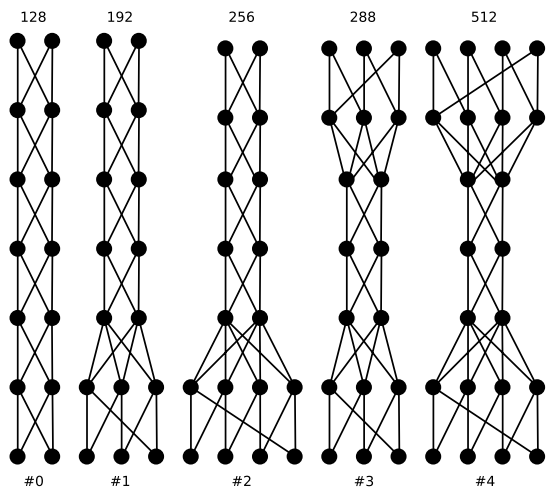


Fig. 10. Some possible variations on Reed-Muller codes seen as pin codes on a chain complexes, by modifying the ends of the chain complex, the left most chain complex is used to define Reed-Muller codes. Example of dimension 6 are shown here, making variations on $\mathcal{RM}(2, 7)$. The number of flags is written above each chain complex and an identification number is given below.

yields a code fulfilling Prop. 5. Moreover the logical gate obtained is the transversal T^\dagger so directly usable in a T -gate distillation protocol. It distills n magic states into k ones of better quality which depends on the distance of the code that has to be computed independently.

In [26], the authors use Reed-Muller codes, $\mathcal{RM}(r, m)$, to obtain initial tri-orthogonal spaces (even triply-even). Viewed as a pin code, $\mathcal{RM}(r, m)$ is a very simple chain complex, represented on the left of figure 10, see also appendix A. This chain complex can be modified in several ways to obtain different pin codes. We tried different modifications in the case $D = 6$, they are represented in figure 10. For all of them the 2-pinned sets generate a triply-even space.

We tried to randomly puncture the pin codes obtained from these chain complexes; similarly to [26] but without deploying the more advanced techniques. We were able to find a few interesting codes this way, see Table III, which can be used to distill T magic states. The obtained parameters γ , see Eq. (13), are similar but do not improve on the small examples found in [26].

TABLE III

SOME TRIORTHOGONAL CODES FOUND BY RANDOMLY PUNCTURING THE PIN CODES REPRESENTED IN FIGURE 10.

code #	initial n	punctured code: $[[n, k, d]]$	$\gamma = \frac{\ln(n/k)}{\ln d}$
0	128	$[[116, 12, 4]]$	1.64
1	192	$[[175, 17, 4]]$	1.68
2	256	$[[236, 20, 4]]$	1.78
3	288	$[[261, 27, 4]]$	1.64
4	512	$[[466, 46, 4]]$	1.67

D. Logical circuits of CCZs

It is also possible to use the property of multi-orthogonality of pinned sets on a given pin code relation in a slightly

TABLE IV
ALTERNATIVE CONSTRUCTION OF CSS CODES WITH TRANSVERSAL T
IMPLEMENTING SOME CIRCUIT OF CCZ GATES ON THE LOGICAL LEVEL.

$D = 5, x = 2$	$D = 8, x = 3$
$[[64, 15, 4]]$	$[[512, 84, 8]]$
$[[96, 23, 4]]$	$[[768, 126, 8]]$
$[[128, 31, 4]]$	$[[1024, 168, 8]]$
$[[144, 35, 4]]$	$[[1152, 188, 8]]$
$[[256, 63, 4]]$	$[[2048, 332, 8]]$

different way. The construction proposed for Reed-Muller codes in [45] can be directly adapted to general pin code relations. This construction is the following: given a pin code relation on $D + 1$ sets and a positive integer x , choose the $(x - 1)$ -pinned sets as X -stabilizer generators and impose the x -pinned sets to be the X -logical operators. This is enough to completely characterize a CSS code. Then by choosing carefully the parameters D and x , one can obtain a code guaranteed to satisfy Prop. 5 where the logical operation realized belongs to some level ℓ of the Clifford hierarchy. Roughly x has to be small enough so that the conditions in Prop. 5 hold but large enough for the logical operation described by the weighted polynomial in Eq. (9) to be in the ℓ th level of the Clifford hierarchy. When taken together these constraints become

$$x = \frac{D + 1}{\ell}. \quad (16)$$

We can for example adapt the pin code relations presented in Fig. 10 to have the correct dimension D by inserting or removing levels of size 2 in the middle of the chain complexes and look at what code parameters they give. These parameters are compiled in Table IV, for $D = 5$ we remove the middle level and for $D = 8$ we add two levels of size 2 compared to Fig. 10. All these codes support the transversal T and up to a Clifford correction the logical operation implemented is some circuit of CCZs characterized by which triple of X -logical operators have an odd overlap.

VI. DISCUSSION

Quantum pin codes form a large family of CSS codes which we have just begun to explore. These codes can be viewed as a vast generalization of quantum color codes and the notions of boundaries, colored logical operators and shrunk lattices all generalize to pin codes. Pin codes also have a gauge code version with potentially similar advantage as the gauge color codes. The main property of pin codes is that their X - and Z -stabilizers form multi-orthogonal spaces. We have presented two concrete ways of constructing pin codes and numerically explored some examples. Several aspects of pin codes merit further studying.

First is finding restricted families with good parameters and LDPC property. Exploring other finite groups with even order generators, other families of sparse chain complexes or finding other constructions of pin code relations altogether would help figuring out the achievable parameters for pin codes.

Second one concerns logical operators. Understanding if some conditions on the pin code relation F can make the

logical operators fulfill the second condition of Prop. 4 or Prop. 5 would help in the design of codes with transversal gates. Also, logical operators and boundaries of 2D color codes have a richer structure than the colored logicals and boundaries that we have explored, it would be interesting to generalize to pin codes with $D = 2$ all the ones presented in [46], as well as for larger D . Moreover, the structure of colored logicals plays a key role in decoding color codes [34]–[37]. Understanding if it can help in finding efficient decoders for more general pin codes is a natural question.

Finally more extensively exploring tri-orthogonal spaces obtained from pin code relations and puncturing them to obtain good T distillation protocols as well as using them as the basis for T -to-CCZ or other protocols seems worth trying as distilling magic state will constitute a sizable fraction of any fault-tolerant quantum computation.

APPENDIX A

REED-MULLER CODE-WORDS AS PINNED SETS

One can define the Reed-Muller code, $\mathcal{RM}(r, m) \subset \mathbb{F}_2^m$, as follows. Define

$$k = \sum_{j=0}^r \binom{m}{j},$$

and given coefficients, $c \in \mathbb{F}_2^k$, define the multivariate polynomial $p_c \in \mathbb{F}_2[X_1, \dots, X_m]$

$$p_c = \sum_{\substack{S \subset \{1, \dots, m\} \\ |S| \leq r}} c_S \prod_{j \in S} X_j.$$

Then the code $\mathcal{RM}(r, m)$ is defined as

$$\mathcal{RM}(r, m) = \left\{ (p_c(x))_{x \in \mathbb{F}_2^m} : c \in \mathbb{F}_2^k \right\}.$$

We now show that $\mathcal{RM}(r, m)$ is generated by the pinned sets with r pins of a certain pin code relation F . Consider m levels, each containing two pins,

$$\forall j \in \{0, \dots, m-1\}, C_j = \{0, 1\},$$

and the complete m -ary relation, $F = C_0 \times \dots \times C_{m-1}$, see also the left of figure 10. Consider now a r -pinned set defined, with type $t = \{j_1, \dots, j_r\}$, and pins $b = (b_{j_1}, \dots, b_{j_r})$. One can check that a flag, $f \in F = \mathbb{F}_2^m$, belongs to $P_t(b)$ if and only if the following degree r polynomial, $p_{t,b}$, evaluates to 1,

$$p_{t,b} = \prod_{\substack{j \in t \\ b_j=1}} X_j \prod_{\substack{k \in t \\ b_k=0}} (1 - X_k).$$

So the pinned sets with r pins generate the following code,

$$\mathcal{P}(r, m) = \left\langle (p_{t,b}(x))_{x \in \mathbb{F}_2^m} : t \subset \{0, \dots, m-1\}, |t| = r, b \in \mathbb{F}_2^m \right\rangle.$$

Then we just have to check that these generate all polynomial of degree at most r . By definition they generate polynomials constituted of a product of r elements being either X_j or $(1 - X_j)$. Let's suppose, for some $\ell \leq r$, they can generate all such

product with only ℓ terms. Then we can construct all product with only $\ell - 1$ terms, q , as follows

$$q = q \cdot (1 - X_j) + q \cdot X_j,$$

where X_j is a variable that does not appear in q . It follows by induction that they generate all degree at most r polynomials and so

$$\mathcal{RM}(r, m) = \mathcal{P}(r, m).$$

Another way to see this is to use the decomposition property of pinned sets (proposition 2) and generate the lower degree monomial directly with pinned sets with less pins and decompose these pinned sets into disjoint union of r -pinned sets.

APPENDIX B

QUASI-TRANSVERSALITY

In this appendix we detail the three weighted polynomial in Eqs. (9), (10) and (11) which determine the transversal action of R_ℓ on code states. Using identity (5) and denoting L^m as the m^{th} row of matrix L and G^n as the n^{th} row of matrix G we can write

$$\begin{aligned} F_\ell(\mathbf{x}) &= |\mathbf{x}L| \\ &= \sum_{s=1}^{\ell} (-2)^{s-1} \sum_{1 \leq m_i \leq k} \left| \bigwedge_{i=1}^s L^{m_i} \right| \prod_{i=1}^s x_{m_i}, \end{aligned} \quad (17)$$

$$\begin{aligned} F'_\ell(\mathbf{y}) &= |\mathbf{y}G| \\ &= \sum_{t=1}^{\ell} (-2)^{t-1} \sum_{1 \leq n_j \leq r} \left| \bigwedge_{j=1}^t G^{n_j} \right| \prod_{j=1}^t y_{n_j}, \end{aligned} \quad (18)$$

$$\begin{aligned} F''_\ell(\mathbf{x}, \mathbf{y}) &= -2 |\mathbf{x}L \wedge \mathbf{y}G| = \\ &= \sum_{\substack{s+t=2 \\ s \geq 1, t \geq 1}}^{\ell} (-2)^{s+t-1} \sum_{\substack{1 \leq m_i \leq k \\ 1 \leq n_j \leq r}} \left| \bigwedge_{i=1}^s L^{m_i} \bigwedge_{j=1}^t G^{n_j} \right| \prod_{i=1}^s x_{m_i} \prod_{j=1}^t y_{n_j}. \end{aligned} \quad (19)$$

One can readily see that these are all correctly weighted polynomial, i.e. with a prefactor of 2^{s-1} in front of monomials of degree s , and their coefficients are given by the size of the overlaps between rows of the matrices L or G .

We can check that Eq. (12) follows from Proposition 5. Indeed, assuming Prop. 5 holds, then (i) enforces that all coefficients $\left| \bigwedge_{j=1}^t G^{n_j} \right|$ are divisible by 2 and (ii) that all coefficients $\left| \bigwedge_{i=1}^s L^{m_i} \bigwedge_{j=1}^t G^{n_j} \right|$ also are divisible by 2. Hence we can pull out a factor 2 in front of everything while keeping the correct prefactor in front of each monomial.

ACKNOWLEDGMENT

C.V. would like to thank B. Audoux, E.T. Campbell and L.P. Pryadko for fruitful discussions at different stages of this project. C.V. acknowledge support by the European Research Council (EQEC, ERC Consolidator Grant No: 682726) as well as a QuantERA grant for the QCDA consortium. NPB is supported by the UCLQ fellowship.

REFERENCES

- [1] E. T. Campbell, B. M. Terhal, and C. Vuillot, "Roads towards fault-tolerant universal quantum computation," *Nature*, vol. 549, no. 7671, pp. 172–179, Sep. 2017. [Online]. Available: <https://www.nature.com/articles/nature23460>
- [2] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Physical Review A*, vol. 54, no. 2, pp. 1098–1105, Aug. 1996. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.54.1098>
- [3] Steane Andrew, "Multiple-particle interference and quantum error correction," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 452, no. 1954, pp. 2551–2577, Nov. 1996. [Online]. Available: <https://royalsocietypublishing.org/doi/10.1098/rspa.1996.0136>
- [4] D. Gottesman, "Stabilizer codes and quantum error correction," PhD Thesis, California Institute of Technology, Jan. 1997.
- [5] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, "Quantum error correction via codes over GF(4)," in *Proceedings of IEEE International Symposium on Information Theory*, Jun. 1997, pp. 292–.
- [6] M. Grassl, T. Beth, and T. Pellizzari, "Codes for the quantum erasure channel," *Physical Review A*, vol. 56, no. 1, pp. 33–38, Jul. 1997. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.56.33>
- [7] M. H. Freedman and D. A. Meyer, "Projective Plane and Planar Quantum Codes," *Foundations of Computational Mathematics*, vol. 1, no. 3, pp. 325–332, Jul. 2001. [Online]. Available: <https://doi.org/10.1007/s102080010013>
- [8] M. H. Freedman, D. A. Meyer, and F. Luo, "\$Z_2\$-systolic freedom and quantum codes," in *Mathematics of quantum computation*, ser. Computational Mathematics, R. K. Brylinski and G. Chen, Eds. Boca Raton, FL: Chapman & Hall/CRC, 2002, no. 3, pp. 287–320.
- [9] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," *Annals of Physics*, vol. 303, no. 1, pp. 2–30, Jan. 2003. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0003491602000180>
- [10] D. Bacon, "Operator quantum error-correcting subsystems for self-correcting quantum memories," *Physical Review A*, vol. 73, no. 1, p. 012340, Jan. 2006. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.73.012340>
- [11] H. Bombin and M. A. Martin-Delgado, "Topological Quantum Distillation," *Physical Review Letters*, vol. 97, no. 18, p. 180501, Oct. 2006. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.97.180501>
- [12] A. A. Kovalev and L. P. Pryadko, "Quantum Kronecker sum-product low-density parity-check codes with finite rate," *Physical Review A*, vol. 88, no. 1, p. 012311, Jul. 2013. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.88.012311>
- [13] A. Couvreur, N. Delfosse, and G. Zmor, "A Construction of Quantum LDPC Codes From Cayley Graphs," *IEEE Transactions on Information Theory*, vol. 59, no. 9, pp. 6087–6098, Sep. 2013.
- [14] L. Guth and A. Lubotzky, "Quantum error correcting codes and 4-dimensional arithmetic hyperbolic manifolds," *Journal of Mathematical Physics*, vol. 55, no. 8, p. 082202, Aug. 2014. [Online]. Available: <https://aip.scitation.org/doi/10.1063/1.4891487>
- [15] J. Tillich and G. Zmor, "Quantum LDPC Codes With Positive Rate and Minimum Distance Proportional to the Square Root of the Blocklength," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1193–1202, Feb. 2014.
- [16] B. Audoux and A. Couvreur, "On tensor products of CSS codes," *Annales de l'Institut Henri Poincaré D*, vol. 6, no. 2, pp. 239–287, Mar. 2019. [Online]. Available: https://www.emis-ph.org/journals/show_abstract.php?issn=2308-5827&vol=6&iss=2&rank=4
- [17] A. Leverrier, J. Tillich, and G. Zmor, "Quantum Expander Codes," in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, Oct. 2015, pp. 810–824.
- [18] S. Bravyi and A. Kitaev, "Universal quantum computation with ideal Clifford gates and noisy ancillas," *Physical Review A*, vol. 71, no. 2, p. 022316, Feb. 2005. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.71.022316>
- [19] H. Bombin, "Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes," *New Journal of Physics*, vol. 17, no. 8, p. 083002, Aug. 2015. [Online]. Available: <https://doi.org/10.1088%2F1367-2630%2F17%2F8%2F083002>
- [20] H. Bombin and M. A. Martin-Delgado, "Exact topological quantum order in $\{d=3\}$ and beyond: Branyons and brane-net condensates," *Physical Review B*, vol. 75, no. 7, p. 075103, Feb. 2007. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevB.75.075103>
- [21] A. Kubica and M. E. Beverland, "Universal transversal gates with color codes: A simplified approach," *Phys. Rev. A*, vol. 91, no. 3, p. 032330, Mar. 2015. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.91.032330>
- [22] H. S. M. Coxeter, *Regular polytopes*, 3rd ed. New York: Dover Publications, 1973, open Library ID: OL5436086M.
- [23] M. W. Davis, *The Geometry and Topology of Coxeter Groups. (LMS-32) (London Mathematical Society Monographs)*. Princeton University Press, Oct. 2007, open Library ID: OL11182953M.
- [24] P. McMullen and E. Schulte, *Abstract Regular Polytopes*, ser. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2002.
- [25] E. T. Campbell and M. Howard, "Unified framework for magic state distillation and multiqubit gate synthesis with reduced resource cost," *Physical Review A*, vol. 95, no. 2, p. 022316, Feb. 2017. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.95.022316>
- [26] J. Haah and M. B. Hastings, "Codes and Protocols for Distilling $\$T\$, controlled- $\$S\$, and Toffoli Gates," *Quantum*, vol. 2, p. 71, Jun. 2018. [Online]. Available: <https://quantum-journal.org/papers/q-2018-06-07-71/>$$
- [27] S. X. Cui, D. Gottesman, and A. Krishna, "Diagonal gates in the Clifford hierarchy," *Physical Review A*, vol. 95, no. 1, Jan. 2017. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.95.012329>
- [28] S. Bravyi and J. Haah, "Magic-state distillation with low overhead," *Physical Review A*, vol. 86, no. 5, p. 052329, Nov. 2012. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.86.052329>
- [29] C. Jones, "Multilevel distillation of magic states for quantum computing," *Physical Review A*, vol. 87, no. 4, p. 042305, Apr. 2013. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.87.042305>
- [30] E. T. Campbell and M. Howard, "Unifying Gate Synthesis and Magic State Distillation," *Physical Review Letters*, vol. 118, no. 6, p. 060501, Feb. 2017. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.118.060501>
- [31] M. B. Hastings and J. Haah, "Distillation with Sublogarithmic Overhead," *Physical Review Letters*, vol. 120, no. 5, p. 050504, Jan. 2018. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.120.050504>
- [32] A. Kubica, B. Yoshida, and F. Pastawski, "Unfolding the color code," *New Journal of Physics*, vol. 17, no. 8, p. 083026, Aug. 2015. [Online]. Available: <https://doi.org/10.1088%2F1367-2630%2F17%2F8%2F083026>
- [33] A. Bhagoji and P. Sarvepalli, "Equivalence of 2d color codes (without translational symmetry) to surface codes," in *2015 IEEE International Symposium on Information Theory (ISIT)*, Jun. 2015, pp. 1109–1113.
- [34] N. Delfosse, "Decoding color codes by projection onto surface codes," *Physical Review A*, vol. 89, no. 1, p. 012317, Jan. 2014. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.89.012317>
- [35] A. B. Alosious and P. K. Sarvepalli, "Projecting three-dimensional color codes onto three-dimensional toric codes," *Physical Review A*, vol. 98, no. 1, p. 012302, Jul. 2018. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.98.012302>
- [36] A. Kubica, M. E. Beverland, F. Brandt, J. Preskill, and K. M. Svore, "Three-Dimensional Color Code Thresholds via Statistical-Mechanical Mapping," *Physical Review Letters*, vol. 120, no. 18, p. 180501, May 2018. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.120.180501>
- [37] A. Kubica and N. Delfosse, "Efficient color code decoders in $\$d\geq 2\$$ dimensions from toric code decoders," *arXiv:1905.07393 [quant-ph]*, May 2019, arXiv: 1905.07393. [Online]. Available: <http://arxiv.org/abs/1905.07393>
- [38] D. Poulin, "Stabilizer Formalism for Operator Quantum Error Correction," *Physical Review Letters*, vol. 95, no. 23, p. 230504, Dec. 2005. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.95.230504>
- [39] H. Bombin, "Single-Shot Fault-Tolerant Quantum Error Correction," *Physical Review X*, vol. 5, no. 3, p. 031043, Sep. 2015. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevX.5.031043>
- [40] N. Delfosse, "Tradeoffs for reliable quantum information storage in surface codes and color codes," in *2013 IEEE International Symposium on Information Theory*, Jul. 2013, pp. 917–921.
- [41] N. P. Breuckmann and B. M. Terhal, "Constructions and Noise Threshold of Hyperbolic Surface Codes," *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3731–3744, Jun. 2016.
- [42] N. P. Breuckmann, "Homological quantum codes beyond the toric code," PhD Thesis, RWTH Aachen University, 2017. [Online]. Available: <https://arxiv.org/abs/1802.01520>

- [43] E. T. Campbell, “A theory of single-shot error correction for adversarial noise,” *Quantum Science and Technology*, vol. 4, no. 2, p. 025006, Feb. 2019. [Online]. Available: <https://doi.org/10.1088%2F2058-9565%2F4%2F2%2F025006>
- [44] W. Zeng and L. P. Pryadko, “Higher-dimensional quantum hypergraph-product codes,” *arXiv:1810.01519 [math-ph, physics:quant-ph]*, Oct. 2018, arXiv: 1810.01519. [Online]. Available: <http://arxiv.org/abs/1810.01519>
- [45] N. Rengaswamy, R. Calderbank, and H. D. Pfister, “Synthesizing stabilizer codes that support physical T and T^\dagger gates,” *in preparation*, 2019.
- [46] M. S. Kesselring, F. Pastawski, J. Eisert, and B. J. Brown, “The boundaries and twist defects of the color code and their applications to topological quantum computation,” *Quantum*, vol. 2, p. 101, Oct. 2018. [Online]. Available: <https://quantum-journal.org/papers/q-2018-10-19-101/>