

Relation Between Surface Codes and Hypermap-Homology Quantum Codes

Pradeep Sarvepalli*

Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai 600 036, India

(Dated: March 14, 2014)

Recently, a new class of quantum codes based on hypermaps were proposed. These codes are obtained from embeddings of hypergraphs as opposed to surface codes which are obtained from the embeddings of graphs. It is natural to compare these two classes of codes and their relation to each other. In this context two related questions are addressed in this paper: Can the parameters of hypermap-homology codes be superior to those of surface codes and what is precisely the relation between these two classes of quantum codes? We show that a canonical hypermap code is identical to a surface code while a noncanonical hypermap code can be transformed to a surface code by CNOT gates alone. Our approach is constructive; we construct the related surface code and the transformation involving CNOT gates.

Keywords: quantum codes, topological codes, hypermap-homology codes, surface codes, hypergraphs, hypermap homology

I. INTRODUCTION

Surface codes, proposed by Kitaev [11], are an extremely appealing class of codes for fault tolerant quantum computation [8, 15]. They have been generalized in various directions [1, 2, 4–6, 18, 19]. Recently, a new generalization of surface codes was proposed in [13]. In this approach quantum codes are constructed based on the homology of hypergraphs rather than the homology of graphs, which is the case in surface codes. Hypergraphs are a generalization of graphs; the edges of a hypergraph can be incident on two or more vertices. Just as the embedding of graphs on surfaces gives rise to codes which are topological in nature [11], embeddings of hypergraphs also give rise to topological quantum codes. These codes have been termed hypermap-homology codes in [13]; we also refer to them as hypermap codes.

Arguably, the most popular surface code is Kitaev's toric code defined on a square lattice of size $n \times n$ with periodic boundary conditions [11]. This code is a $[[2n^2, 2, n]]$ quantum code. On the other hand, Ref. [13] proposed a hypermap homology code defined on the $n \times n$ square lattice. This is a $[[3n^2/2, 2, n]]$ code and it is more efficient than Kitaev's toric code with respect to the number of physical qubits required to protect the same number of logical qubits while maintaining the same level of error correcting capability. This seemed to suggest that better quantum codes maybe obtained by using hypergraphs. But there are other surface codes with better parameters than the $[[2n^2, 2, n]]$ toric code. There exist surface codes whose parameters are $[[n^2 + 1, 2, n]]$, see [3, 12].

This begs the question if hypermap codes improve upon the parameters of best surface codes? A related question is the exact relation between hypermap codes and surface codes. It was also not apparent which codes could be realized using hypermaps but not by embedding graphs on surfaces.

In this paper we address these questions. The construction of hypermap codes requires us to make a choice as to whether we will represent certain boundary maps using the standard basis or not. We call those codes with standard basis as canonical hypermap codes and those with a nonstandard basis as

noncanonical hypermap codes. We show that for every canonical hypermap code there is a surface code which is identical to the hypermap code. This implies that the $[[3n^2/2, 2, n]]$ hypermap code, although it improves upon the $[[2n^2, 2, n]]$ toric code, it is identical to another surface code. We also show that every noncanonical hypermap code can be transformed to a surface code; we only need CNOT gates for this transformation. In some cases, a noncanonical hypermap code can be identical to a surface code. A hypermap code that cannot be realized by a surface code must be a noncanonical code.

Our results imply that hypermap codes that improve upon surface codes or which cannot be realized as surface codes must be noncanonical hypermap codes. Through our results, many questions related to hypermap codes can be posed as questions about surface codes and we can study hypermap codes using surface codes. For instance, decoding of these codes can be studied in terms of related surface codes.

This paper is structured as follows. In Section II after briefly reviewing surface codes, we give a self-contained introduction to hypermap codes. Then in Section III we present our main results which clarify the relation between canonical and noncanonical hypermap codes on one hand and the relation between hypermap codes and surface codes on the other. We then conclude with a brief discussion on the significance and consequences of our results.

II. BACKGROUND

A. Surface codes

We assume that the reader is familiar with the basics of quantum codes and stabilizer formalism [7, 9]. Let G be a graph with vertex set $V(G)$ and edge set $E(G)$; when there is no confusion we drop the dependence on G . Consider an embedding of the graph on a surface Σ i.e. a drawing of the graph on the surface so that no two edges cross each other. Denote the faces of the embedding as $F(G)$. We restrict our attention only to those embeddings in which the faces are homeomorphic to open discs. Such an embedding is also called a 2-cell embedding or a map. A (stabilizer) quantum code is obtained from the embedding as follows. On each edge we place a

* pradeep@ee.iitm.ac.in

qubit. We associate to each vertex an operator called vertex operator, denoted by A_v and to each face an operator, denoted by B_f . The face operators are also sometimes referred to as plaquette operators. These operators are defined as follows:

$$A_v = \prod_{u \in \delta(v)} X_u, \text{ where } \delta(v) = \{u \mid (u, v) \in E(G)\} \quad (1)$$

$$B_f = \prod_{e \in \partial(f)} Z_e, \quad (2)$$

where $\partial(f) = \{e \mid e \text{ is in the boundary of } f\}$; while X and Z are the Pauli matrices. The surface code is defined as the joint +1-eigenspace stabilized by the operators A_v and B_f . In other words, it is the quantum code with the stabilizer

$$S = \langle A_v, B_f \mid v \in V(G), f \in F(G) \rangle. \quad (3)$$

The stabilizer matrix of the surface code can be represented by $\begin{bmatrix} I_V & 0 \\ 0 & I_F \end{bmatrix}$, where I_V is the vertex-edge incidence matrix of G and I_F is the face-edge incidence matrix of G . The surface code is an $[[|E|, 2g]]$ quantum code, where g is the genus of the surface on which G is embedded.

B. Hypermap homological codes

We now review the hypermap-homological codes proposed in [13]. The reader can find more details on these codes therein. Let Γ be a hypergraph with vertex set $V(\Gamma)$ and hyperedge set $E(\Gamma)$. A hyperedge is any nonempty subset of the vertex set. If $E(\Gamma)$ has only subsets of size two then Γ reduces to a standard graph. (We use Greek alphabet for hypergraphs only.) Hypergraphs are often studied by means of a bipartite graph representation. This bipartite graph is formed as follows: Form a bipartite graph G_Γ with vertex set $V(G_\Gamma) = V(\Gamma) \cup E(\Gamma)$. Place an edge between $v \in V(\Gamma)$ and $e \in E(\Gamma)$ if and only if e is incident on v . We refer to the edges of the bipartite graph as darts or half-edges and denote the collection of darts by $E(G_\Gamma)$. The darts will also be denoted as $W(\Gamma)$. For any dart i , we denote the unique vertex in $V(\Gamma)$ on which i is incident by $v_{\ni i}$ and the unique hyperedge on which i is incident by $e_{\ni i}$.

Consider the embedding of G_Γ on a surface Σ . Denote the faces of G_Γ by $F(G_\Gamma)$. An embedding of G_Γ is called a hypermap. The labeling of the darts in the hypermap is performed as follows: we place a label to the left of the dart as we move along the dart from a hyperedge to an adjacent vertex in G_Γ . Alternatively, we place the label counterclockwise of the dart with respect to hyperedge on which it is incident. To distinguish between vertices and hyperedges of Γ , elements in $E(\Gamma)$ are shown as squares whereas elements in $V(\Gamma)$ are shown as circles, see Fig. 1. If the hypergraph is connected then the bipartite graph is also connected and it is possible to associate a pair of permutations $\sigma, \tau \in S_n$ acting on the darts of the hypermap such that the group $\langle \sigma, \tau \rangle$ is transitive on the set of darts. Here S_n is the symmetric group of permutations on n elements.

Let us consider a simple example. Consider a hypergraph with 2 vertices and 2 hyperedges, where $V = \{v_1, v_2\}$ and $E = \{(v_1, v_2, v_1, v_2), (v_1, v_2, v_1, v_2)\} = \{e_1, e_2\}$. The associated bipartite graph, see Fig. 1, has 4 vertices and 8 darts. Vertices can be repeated in a hyperedge. An embedding of this hypergraph, i.e. the embedding of its bipartite graph representation, on a torus has 4 faces, 4 vertices, 8 darts.

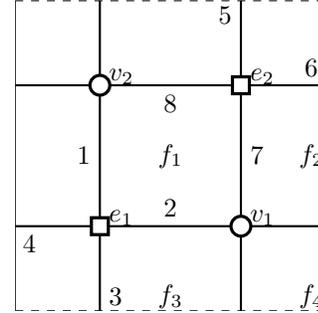


FIG. 1. A hypergraph embedded on a torus; opposite ends are to be identified. The circles are the vertices of the hypergraph. The square vertices are the hyperedges of the hypergraph. The labels are to the left as one moves from squares to the circles.

In Fig. 1 the permutations σ and τ are defined as follows:

$$\begin{aligned} \sigma &= (1 \ 8 \ 3 \ 6)(2 \ 5 \ 4 \ 7) \\ \tau &= (1 \ 2 \ 3 \ 4)(5 \ 6 \ 7 \ 8) \\ \sigma\tau^{-1} &= (1 \ 7)(2 \ 8)(3 \ 5)(4 \ 6), \end{aligned}$$

where $\sigma\tau^{-1}(i) = \sigma(\tau^{-1}(i))$.

The number of orbits of σ is the number of vertices of original hypergraph while the number of orbits of τ is the number of hyperedges of the hypergraph. The number of orbits of $\sigma\tau^{-1}$ is the number of faces in the embedding of the hypergraph. A dart belongs to a face if and only if the label is in the interior of the face. We use the following shorthand for simplicity:

$$F = F(G_\Gamma) = \{f_1, f_2, \dots, f_{|F|}\}, \quad (4)$$

$$W = E(G_\Gamma) = \{w_1, w_2, \dots, w_{|W|}\} = W(\Gamma), \quad (5)$$

$$E = E(\Gamma) = \{e_1, e_2, \dots, e_{|E|}\} \quad (6)$$

$$V = V(\Gamma) = \{v_1, v_2, \dots, v_{|V|}\}. \quad (7)$$

Denote the binary vector spaces formed by taking $F(G_\Gamma)$, $E(G_\Gamma)$, $E(\Gamma)$ and $V(\Gamma)$ as bases by \mathcal{F} , \mathcal{W} , \mathcal{E} , and \mathcal{V} respectively. Topological codes are usually defined with respect to a series of boundary maps. We now proceed to define similar maps for the hypermaps with respect to the spaces just introduced. Define a ‘‘boundary’’ map for each face, dart, and hyperedge as follows:

$$d_2(f) = \sum_{i \in f} w_i \quad (8)$$

$$d_1(w_i) = v_{\ni i} + v_{\ni \tau^{-1}(i)}, \quad (9)$$

$$\iota(e) = \sum_{i \in \delta(e)} w_i \quad (10)$$

are given as

$$H_x = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}; H_z = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad (29)$$

Irrespective of the basis chosen for $\mathcal{W}/\iota(\mathcal{E})$, the total number of encoded qubits is a function of the genus of the surface. In this sense we encode into the topological degrees of freedom afforded by the surface. The distance on the other hand is not basis invariant. For instance, the hypermap code obtained from Fig. 1 with special basis has distance two while the code with nonspecial basis has distance one. The distance of hypermap codes was analyzed when the basis was special [13]. The distances of hypermap codes with nonspecial bases are a little more difficult to compute and it is not necessary they all have the same distance. For a special basis, the distance can be related to the cycles of the hypermap and its dual, and thus given a topological interpretation, see for example [13, Corollary 4.23]. In case of a nonspecial basis such a topological interpretation of the distance has not yet been given.

The following result was proved in [13] although it is not stated as such therein.

Theorem 1 (Hypermap-homology codes[13]). *Let Γ be a hypergraph with $|\mathcal{E}|$ hyperedges and G_Γ its bipartite graph representation with $|\mathcal{W}|$ darts. The hypermap obtained by embedding G_Γ on a surface of genus g leads to a $[|\mathcal{W}| - |\mathcal{E}|, 2g]$ quantum code.*

In Theorem 1, neither the choice of special darts nor the basis for $\mathcal{W}/\iota(\mathcal{E})$ is explicit, but these choices must be made before constructing the quantum code. As was clear from the example, different choices of bases could lead to different codes with potentially different distances.

III. HYPERMAP CODES AND SURFACE CODES

In this section, we address some of the questions raised in [13]. We prove some new results about hypermap codes and then relate them to surface codes. We show that the hypermap codes with special and nonspecial bases are related by transformations involving just CNOT gates. Along the way we establish a correspondence between the qubits of the code and the hypermap [14]; this was only implicit in [13]. Finally we relate the hypermap codes to the surface codes showing an equivalence between canonical hypermap codes and surface codes.

A. Relation between hypermap codes of different bases

Consider the bipartite representation of the hypergraph or its embedding. Let S be the collection of special darts. These special darts are $|E|$ in number and no two of them are incident

on the same hyperedge. Choose a special basis for $\mathcal{W}/\iota(\mathcal{E})$. A special basis for $\mathcal{W}/\iota(\mathcal{E})$ is of the form

$$\{w_{i_1}, w_{i_2}, \dots, w_{i_n}\} = \mathcal{W} \setminus S \quad (30)$$

where $n = |\mathcal{W}| - |E|$; n is also the length of the code. Then we can place n qubits, one on each of the nonspecial darts labeled $i_j \in \{i_1, i_2, \dots, i_n\}$. The special darts carry no qubits. (Note that there are more darts than there are qubits. So the labels of the qubits need not be the same as labels of the darts on which the qubits are placed. It is possible to relabel the hypermap so that both qubit and dart labels coincide.) Now define the stabilizer generators using the matrices H_x and H_z . The matrix H_z^t is simply the face-dart incidence matrix of the hypermap modulo $\iota(\mathcal{E})$ i.e. relations of the form given in Eq. (11). In other words, it is constrained to have no special darts. We can view matrix H_x as the vertex incidence matrix of nonspecial darts of the hypermap but the incidence vector is found after extending the half-edge i to a full edge by combining i and $\tau^{-1}(i)$.

We would like to give a similar correspondence for the non-canonical codes and make precise the connection between canonical and noncanonical hypermap codes. But first we need the following lemma.

Lemma 2. *Let T be an invertible $n \times n$ binary matrix. Then T and T^{-1} can be decomposed as*

$$T = R_{j_1}^{i_1} R_{j_2}^{i_2} \dots R_{j_m}^{i_m}, \quad (31)$$

$$T^{-1} = R_{j_m}^{i_m} \dots R_{j_2}^{i_2} R_{j_1}^{i_1}, \quad (32)$$

where $R_j^i = I + e_i e_j^t$ and $m \leq n^2$.

Proof. Multiplying T by R_j^i from right adds the i th column to the j th column of T . Denote by $(T)_{i,j}$, the entry in i th row and j th column of T . Suppose that $(T)_{i,i} = 1$, then we can eliminate the nonzero entries $(T)_{i,j}$ in the i th row, for $1 \leq j \neq i \leq n$ by adding the i th column to the j th column. If $(T)_{i,i} \neq 1$, then we can find some column j such that $(T)_{i,j} = 1$. Such a column must exist because T is full rank. We can first add this column to i th column before eliminating the remaining nonzero entries $(T)_{i,j}$. If an entry $(T)_{i,j}$ is already zero we do not need to multiply by R_j^i . Assume now that we have made all entries in the i th row zero except the entry $(T)_{i,i}$. Since T is full rank, the $(i+1)$ st row is not identical to the i th row. So we can assume that there is a nonzero entry $(T)_{i+1,j}$ in some column $j \neq i$. Let us eliminate all the non-diagonal entries in $(i+1)$ st row, i.e. we eliminate all the nonzero entries except $(T)_{i+1,i+1}$. In this process the i th row will not be affected because all its non-diagonal entries are zero. Starting from the first row we can reduce T to the identity matrix by multiplying by matrices of the form R_j^i . The product of these matrices must equal T^{-1} . So we have

$$T^{-1} = \prod_{k=1}^m R_{j_k}^{i_k}$$

$$T = \prod_{k=m}^1 (R_{j_k}^{i_k})^{-1} = \prod_{k=m}^1 R_{j_k}^{i_k},$$

where the last equality follows from the fact that R_j^i is its own inverse when $i \neq j$; $R_j^i R_j^i = I + e_i e_j^t + e_i e_j^t + e_i e_j^t e_i e_j^t = I$. Relabeling the i_k and j_k , we obtain Eqs. (31) and (32).

Each row i requires no more than $n - 1$ entries to be made nonzero; accounting for one additional multiplication when $(T)_{i,i} = 0$, we need at most n multiplications per row. Thus $m \leq n^2$. \square

Theorem 3. *Suppose that an $[[n, k]]$ canonical hypermap code has basis $B = \mathcal{W} \setminus \mathcal{S}$ and a noncanonical code has basis $B' = TB$. The canonical code can be transformed to the noncanonical code by the application of $m \leq n^2$ CNOT gates where the l th CNOT gate is applied from qubit i_l to j_l . The number and location of CNOT gates is determined by the decomposition $T = R_{j_1}^{i_1} R_{j_2}^{i_2} \cdots R_{j_m}^{i_m}$, where $R_j^i = I + e_i e_j^t$.*

Proof. The special basis B for the canonical hypermap code can be given as in Eq. (30). Then we can write the nonspecial basis B' for the noncanonical hypermap code as

$$B' = \{T w_{i_1}, T w_{i_2}, \dots, T w_{i_n}\} = T(\mathcal{W} \setminus \mathcal{S}), \quad (33)$$

where T is an invertible (binary) matrix that transforms the special basis to the nonspecial basis. Now the columns of H_x and H_z are indexed by the basis vectors of $\mathcal{W}/\iota(\mathcal{E})$. The relations between the representations of ∂_i for different bases are as follows:

$$[\partial_2]_{B'} = T^{-1}[\partial_2]_B \text{ and } [\partial_1]_{B'} = [\partial_1]_{BT}. \quad (34)$$

This ensures that H_x and H_z are orthogonal because $[\partial_1]_{BT} T^{-1} [\partial_2]_B = 0$.

By Lemma 2, we see that

$$[\partial_1]_{B'} = [\partial_1]_{BT} = [\partial_1]_B R_{j_1}^{i_1} R_{j_2}^{i_2} \cdots R_{j_m}^{i_m} \quad (35)$$

$$\begin{aligned} [\partial_2]_{B'} &= [\partial_2]_B (T^{-1})^t = [\partial_2]_B (R_{j_m}^{i_m} \cdots R_{j_2}^{i_2} R_{j_1}^{i_1})^t \\ &= [\partial_2]_B (R_{j_1}^{i_1})^t (R_{j_2}^{i_2})^t \cdots (R_{j_m}^{i_m})^t \\ &= [\partial_2]_B R_{i_1}^{j_1} R_{i_2}^{j_2} \cdots R_{i_m}^{j_m}. \end{aligned} \quad (36)$$

Let us parse Eqs. (35) and (36) equations closely. The first says that we add the column i_l to the column j_l for $[\partial_1]_B$ while we add the column j_l to the column i_l for $[\partial_2]_B$. This is precisely the action of a CNOT gate acting on qubits i_l and j_l with i_l as control qubit, see for instance [10, Lemma 2]. This proves that the transformation from the canonical hypermap code to a noncanonical hypermap code can be achieved by the application of a sequence of CNOT gates given by the decomposition of T . \square

We make a few observations regarding the relation between canonical and noncanonical codes. First the transformation assumes that the codes have been defined with respect to the same set of special darts. Different set of special darts could lead to different codes. With regard to the parameters of the canonical code we can relate the dimension and distance to topological properties of the surface on which the hypergraph is embedded. For a noncanonical code while the dimension is related to the genus of the surface, the distance does not appear to have such a straightforward relation in general. More

importantly, the distance of the canonical code and the noncanonical code need not be the same. The transformation in Theorem 3 may not preserve distance. It is also possible that the stabilizer generators of the noncanonical code are not local, because there is no restriction on the nonspecial basis. It is obvious that a noncanonical code can be converted to a canonical code by application of CNOT gates in the reverse order.

If T is simply a permutation matrix, then the code remains canonical. Since T is composed of transformations of the form $R_{i_2}^{i_1}$, it is instructive to study the effect of one such transformation on the canonical code. Assume that we are applying $R_{i_2}^{i_1}$, in other words we are applying a CNOT between qubits i_1 and i_2 , with i_1 as the control qubit. (We assume without loss of generality that the nonspecial darts have the labels 1 to n so that we have the same labels for qubits and nonspecial darts.) Suppose the qubits i_1 and i_2 are such that they are on adjacent darts in the hypermap as shown in Fig. 3, then the effect of CNOT on the hypermap is shown in Fig. 4. The boundary of the face to which i_1 and i_2 belong is modified so that it no longer contains i_1 . The resulting code is still canonical with respect to the modified hypermap.

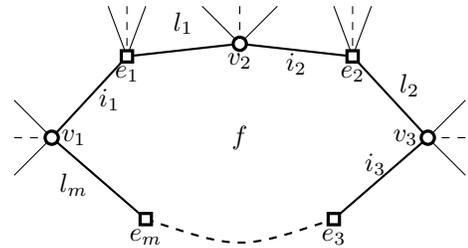


FIG. 3. A typical face in the embedding of the hypergraph. It has an even number of darts and exactly $|f|/2$ darts have their labels inside the face. The darts l_α and i_α are related as $i_\alpha = \tau(l_\alpha)$.

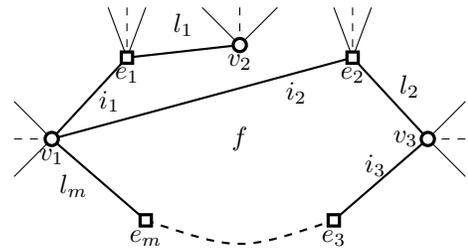


FIG. 4. The code with the basis $R_{i_2}^{i_1} B$ can be obtained by applying a CNOT gate on qubits i_1 and i_2 with i_2 as the target qubit. For the special case when these qubits are adjacent, the effect can be understood graphically. The resulting code is still canonical with respect to a slightly modified hypermap. A CNOT gate between non-adjacent qubits leads to a nontrivial modification of the hypermap in general.

If the qubits are not adjacent, then $R_{i_2}^{i_1} B$ does not seem to effect such a simple modification to the hypermap. Understanding this transformation and relating it to the hypermap would be a very useful contribution in this context. A combination of the transformations of R_j^i could lead to a simple modification though. For instance swapping qubits i_1 and i_2 would lead to a relabeling of the hypermap and we would still end up with a canonical code. It would be interesting to find out which transformations T can be described in terms of simple operations on the hypermap.

B. Relation between hypermap codes and surface codes

Our central result is that a canonical hypermap-homology code can be reduced to a surface code. The graph associated to this surface code can be derived from the hypergraph. Our approach is constructive and Algorithm 1 gives this transformation. We only consider hypergraphs whose bipartite representations are connected. If we are dealing with noncanonical hypermap code, then it can be associated to a canonical hypermap code. From Theorem 3 these two codes are related by a transformation involving CNOT gates alone.

Theorem 4. *Every canonical hypermap-homological code is equivalent to a surface code. Given a hypergraph Γ (embedded on a surface), Algorithm 1 outputs a standard graph \overline{G}_Γ (embedded on the same surface) such that the surface code associated to \overline{G}_Γ has the same stabilizer as the hypermap-homology code associated to Γ .*

Algorithm 1 Surface code from a canonical hypermap code

Input: A hypergraph Γ , and a set of special darts $S \subset W(\Gamma)$, such that $|S| = |E(\Gamma)|$ and every dart in S is incident on a distinct hyperedge.

Output: A graph \overline{G}_Γ such that the surface code of \overline{G}_Γ is same as the canonical hypermap-homology code.

- 1: Form G , the bipartite representation of the hypergraph. Note that $E(G_\Gamma) = W(\Gamma)$.
 - 2: For each hyperedge e of the hypergraph choose one special dart $s_e \in S$ such that $s_e \in \delta(e)$.
 - 3: Embed G on a suitable surface; let the genus of the surface be g .
 - 4: In each face f of the embedding, draw new edges connecting vertex nodes of the hypermap. In other words, for each dart $i \in f$ join $v_{\ni i}$ to $v_{\ni \tau^{-1}(i)}$. Denote this graph by G'_Γ .
 - 5: Each face of G'_Γ now contains $|f|/2$ triangles; each triangle consists of two darts and one newly added edge. Exactly one label is present in each triangle. Label the new edge by that label.
 - 6: Modify G'_Γ by deleting all the darts and the vertices associated to each hyperedge. Denote this graph by G''_Γ .
 - 7: For each hyperedge e there is a special dart s_e . Delete the edge in G''_Γ which has this label. Denote the resulting graph as \overline{G}_Γ .
 - 8: The surface code defined by embedding of \overline{G}_Γ gives the same quantum code as the hypermap. The stabilizer of the surface code is defined using Eqs. (1)–(3).
-

Proof. We show that Algorithm 1 leads to the same stabilizer code as the hypermap-homology code. More precisely,

we show that that the matrices H_x and H_z arising in the hypermap-homology construction are exactly the vertex-edge and face-edge incidence matrices of the graph \overline{G}_Γ .

Consider first the stabilizer of the hypermap-homology code. Each face leads to a Z -only stabilizer generator. Note that H_z is a $(|W| - |E|) \times |F|$ matrix. Although there are only $|F| - 1$ independent generators though, see [13], we include the generator from each face in H_z . Let f be a face of the hypermap. The labeling of the darts in the hypermap is such that only half the darts that constitute the boundary of f have their labels inside f . This is illustrated in Fig. 3.

We can choose the set of nonspecial darts as a basis for $W/\iota(\mathcal{E})$. If w_i is not special then we can write $p(w_i) = w_i$ otherwise we can write $p(w_i) = \sum_{j \neq i \in \delta(e_{\ni i})} w_j$. The latter follows from the relation $\sum_{j \in \delta(e)} w_j = 0$ for every hyperedge e . Then we obtain

$$\begin{aligned} \partial_2(f) &= p(d_2(f)) = p\left(\sum_{i \in f} w_i\right) \\ &= \sum_{\substack{i \in f \\ i \notin S}} p(w_i) + \sum_{\substack{i \in f \\ i \in S}} p(w_i) = \sum_{\substack{i \in f \\ i \notin S}} w_i + \sum_{\substack{i \in f \\ i \in S}} \sum_{\substack{j \in \delta(e_{\ni i}) \\ j \neq i}} w_j \end{aligned} \quad (37)$$

The row associated to $\partial_2(f)$ in H_z is simply the characteristic vector of $\partial_2(f)$.

Now let us consider the Z -type stabilizer generators of \overline{G}_Γ . Let us begin with the graph G'_Γ . With respect to the hypermap it has additional edges. The addition of edges between $v_{\ni i}$ and $v_{\ni \tau^{-1}(i)}$ transforms the face f in Fig. 3 as shown in Fig. 5. This creates $|f|/2$ new (triangular) faces within f , each of which is bounded by two darts of the hypermap and one new edge. Furthermore, exactly one of the labels is contained in each of these new faces and every label is contained in some triangle. Therefore, we can label each new edge by a unique label; furthermore note that these new edges are exactly $|W|$ in number. The face f is modified so that it only contains the new edges and the vertices of the hypergraph in its boundary. We label this derived face also by f' . The derived face f' has only the newly added edges in its boundary. In fact we have $\partial(f') = \sum_{i \in f'} w_i$, where the boundary is with respect to G'_Γ . The boundary of f' in G'_Γ is same as the boundary of f in the hypermap.

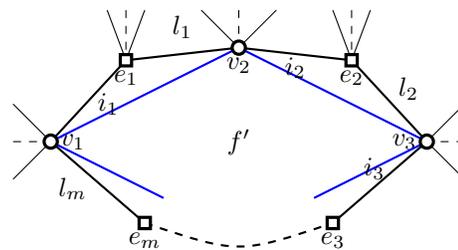


FIG. 5. (Color online) Illustrating the addition of edges between $v_{\ni i}$ and $v_{\ni \tau^{-1}(i)}$. The additional edges are shown in color. We call them edges to distinguish them from the darts (half-edges) of the hypermap. Observe that $\partial(f') = \partial(f)$.

Let us consider the transformation of the hyperedges due to the addition of the new edges. Because of the edges added between the vertices $v_{\ni i}$ and $v_{\ni \tau^{-1}(i)}$, exactly one label is adjacent to any new edge. So we can label the newly added edges by the label in the triangle, this is illustrated in Fig. 6.

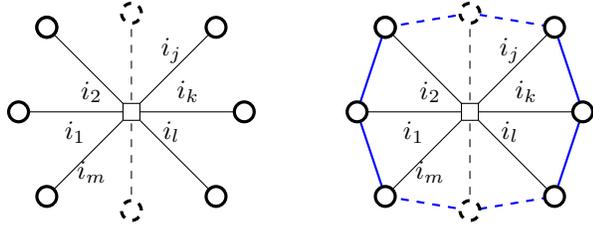


FIG. 6. A hyperedge in the embedding of the hypergraph Γ . The addition of edges leads to a creation of triangular faces each containing exactly one label.

The deletion of hyperedges and the darts incident on them leaves each face f' unchanged. So f' is also a face of G''_{Γ} . So we can denote without ambiguity the face derived from f' as f'' . Furthermore, deletion of the hyperedges and the darts transforms each hyperedge to a face in G''_{Γ} , see Fig. 7. Thus exactly $|\mathcal{E}|$ new faces are added to G''_{Γ} with respect to the hypermap; and they can be indexed by the hyperedges.

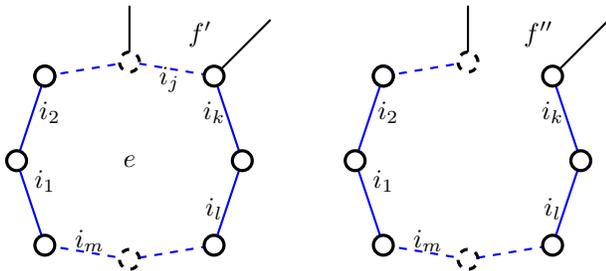


FIG. 7. The deletion of the darts and the hyperedge creates a new face with exactly one special edge. On the deletion of the special edge, say i_j , it gets merged with exactly one face f . The boundary of f now includes the rest of the edges of e i.e. excepting the special edge i_j .

Consider any such face e . Only one of the edges in the boundary of e has the same label as some special dart. We call such edges special edges. Now when such a special edge in the boundary of e is deleted, then e is merged with exactly one face of f'' of G''_{Γ} , since an edge is shared only by two faces, see Fig. 7. While f'' can be merged with many faces e_i derived from the hyperedges, it is not merged with any other face derived from the faces of the hypermap, because such derived faces do not share any edges. Thus \bar{G}_{Γ} has exactly as many faces as the hypermap and the face derived from the merging

of f'' and e (and possibly other faces which share a special edge with f'') can be labeled uniquely as \bar{f} . Let us look at the boundary \bar{f} in \bar{G}_{Γ} . The boundary of f'' in G''_{Γ} is the same as the boundary of f in G_{Γ} and is equal to $\sum_{i \in f} w_i$. When all the special edges are deleted, f'' may be merged with other faces $e \in F(G''_{\Gamma})$ which share a special edge with f'' . The resulting face \bar{f} has a boundary that is given by union of their boundaries, (the sum is take modulo 2). The boundary of f'' is $\sum_{i \in f} w_i$. Each special edge i_j that is removed causes the boundary to include the remaining edges bounding $e_{\ni i_j}$. The boundary of e is $\sum_{i \in \delta(e)} w_i$. Thus the boundary of \bar{f} in \bar{G}_{Γ} is $\sum_{i \in f, i \notin S} w_i + \sum_{i \in f, i \in S} \sum_{j \in \delta(e_{\ni i}), j \neq i} w_j$, which is precisely $\partial_2(f)$, the boundary of f in the hypermap, as computed in Eq. (37). Thus the stabilizer generator associated with a face in \bar{G}_{Γ} is same as the generator associated to its parent face in the hypermap. This shows that H_z is the same as the face-edge incidence matrix of G_{Γ} .

It remains now to show that the matrix H_x is the same as the vertex-edge incidence matrix of \bar{G}_{Γ} . The matrix H_x is determined by the map ∂_1 and acts on the space $\mathcal{W}/\iota(\mathcal{E})$. As mentioned earlier, we can take the nonspecial darts as a basis for $\mathcal{W}/\iota(\mathcal{E})$. Then the columns of H_x are given by the characteristic vector of $\partial_1(w_i)$, where $i \notin S$. We have $\partial_1(w_i) = v_{\ni i} + v_{\ni \tau^{-1}(i)}$. But the vertices $v_{\ni i}$ and $v_{\ni \tau^{-1}(i)}$ are connected by an edge in \bar{G}_{Γ} . This edge is also labeled i in \bar{G}_{Γ} . Hence, we can obtain the i th column of H_x by considering the incidence vector of every edge in \bar{G}_{Γ} . The columns put together then give the vertex-edge incidence matrix of \bar{G}_{Γ} . Thus the surface code generated by embedding of G_{Γ} has the same stabilizer as the hypermap-homology code on Γ . This proves that every hypermap-homology code can be realized by an equivalent surface code. \square

Although Theorem 4 does not mention, the choice of special darts is made explicit in Algorithm 1. We give a simple example that illustrates the application of Theorem 4. Consider the hypermap given in Fig. 1. Draw additional edges in each face connecting two adjacent circles as one goes around the face. The hypermap in Fig. 1 is modified as show in Fig. 8.

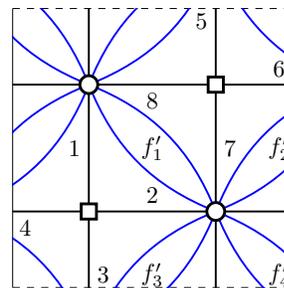


FIG. 8. (Color online) The graph G'_{Γ} for the hypermap in Fig. 1. It is obtained by the addition of new edges to the hypermap. Any two adjacent darts and a newly added edge form a triangle which contains exactly one label; therefore the newly added edge can be uniquely identified by a label.

Next we modify the graph in Fig. 8 as follows. We remove all the original darts and edges of the hypergraph. In addition we also remove the special darts. These transformations are shown in Figs. 9(a) and 9(b) respectively. The hypermap-

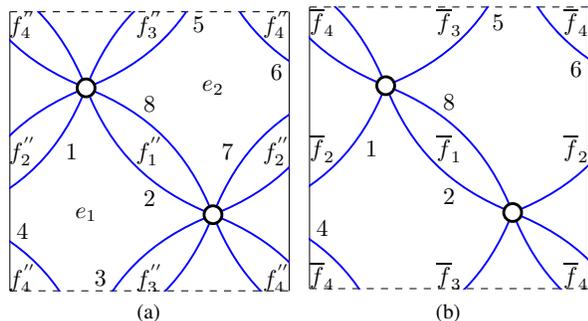


FIG. 9. (Color online) (a) The graph G''_Γ obtained by the removal of the darts and hyperedge-vertices in G'_Γ ; the hyperedges are transformed to faces in G''_Γ . (b) The graph \bar{G}_Γ obtained by removing the special edges i.e. $\{3,7\}$; each hyperedge-face e_i is merged with exactly one face. \bar{G}_Γ has the same stabilizer as the hypermap-homology code of Fig. 1.

homology code obtained from Fig. 1 is identical to the surface code obtained from Fig. 9(b). Consider l_V and l_F , the vertex-edge and face-edge incidence matrices of \bar{G}_Γ in Fig. 9(b):

$$l_V = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad l_F = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad (38)$$

The stabilizer matrix of the surface code is $\begin{bmatrix} l_V & 0 \\ 0 & l_F \end{bmatrix}$. We can see that this is the same as the stabilizer matrix of the hypermap-homology code given in Eq. (24).

The substance of Theorem 4 is that the procedure illustrated works in general and reduces a canonical hypermap-homology code to a surface code. The converse of Theorem 4, i.e. every surface code is also a (canonical) hypermap-homology code, is straightforward.

Corollary 5. *Every canonical hypermap-homology code is equivalent to a surface code and vice versa.*

Proof. We only sketch the proof. Every graph G can also be viewed as a hypergraph, denote it also by Γ . The bipartite graph representation of this hypergraph is obtained by taking the original graph and splitting every edge in G into two edges and adding a new vertex for each edge. Then we can proceed with the construction proposed in [13] to obtain a hypermap-homology code. At this point we have two codes: a hypermap code and a surface both derived from the same graph G . But it is by no means obvious that they are identical. We show that they are the same code. Note that every hyperedge in the hypermap has only two darts incident on it. One of these can

be chosen as a special dart. Now if we apply Algorithm 1, and trace through the various transformations, we find that every hyperedge is transformed to a face with two edges in G''_Γ . This graph is identical to the graph obtained from G where every edge is replaced by two edges. Hence, \bar{G}_Γ obtained after the deletion of all the special edges from G''_Γ would be same as the original graph G . Thus every surface code is equivalent to a hypermap-homology code. This together with Theorem 4 implies the corollary. \square

Combining Corollary 5 and Theorem 3 we obtain the following result:

Corollary 6. *Any $[[n, k]]$ hypermap code is either identical to a surface code or it can be transformed to a surface code with the application of $m \leq n^2$ CNOT gates.*

The CNOT gates are required only if the hypermap code is noncanonical. The sequence in which the CNOT gates are applied in the transformation in Corollary 6 is reverse of the sequence in Theorem 3. In other words it is the sequence of CNOT gates required to transform the noncanonical hypermap code into a canonical code. While a noncanonical code may be transformed to a surface code, it is not necessary that it has the same parameters as the resulting surface code. In particular, the distance can change. Therefore some noncanonical codes may be realized only from the embedding of the hypergraphs and not by the embedding of graphs.

IV. CONCLUSION AND DISCUSSION

Our results imply that canonical hypermap-homology codes cannot improve upon the parameters of surface codes. Noncanonical hypermap codes may have better parameters than surface codes. An interesting problem in this context is to construct noncanonical hypermap codes that have better parameters than surface codes. In such a construction, we also want to be able to preserve the locality of the stabilizer generators. Understanding the distance and decoding of the noncanonical codes merits further study. Hypermap codes provide a different perspective on topological codes which might yield new insights into their properties and potentially lead to better quantum codes and alternate decoding algorithms for topological codes.

Hypermap-homology codes introduce in a very concrete fashion the use of standard hypermap homology into the study of quantum codes. The use of hypermap homology in the construction of quantum codes is an important development and its applications are yet to be fully explored in the context of quantum codes. Other results such those in [12, 18] suggest that hypergraphs offer a fertile ground for construction of new quantum codes. The use of hypergraphs has been fruitful in subsystem codes as well and it would be interesting to study homology of hypermaps in that context also [17]. Homology of hypergraphs is considered with a different perspective in [16, 17]. It seems that the notion of homology used therein does not entirely coincide with that used in [13]. It would be an interesting problem to relate the two.

-
- [1] B. Bombin and M. A. Martin-Delgado. Topological quantum distillation. *Phys. Rev. Lett.*, 97(180501), 2006.
- [2] B. Bombin and M. A. Martin-Delgado. Exact topological quantum order in $D = 3$ and beyond: Branyons and brane-net condensates. *Phys. Rev. B*, 75(075103), 2007.
- [3] B. Bombin and M. A. Martin-Delgado. Optimal resources for topological 2d stabilizer codes: Comparative study. *Phys. Rev. A*, 76(012305), 2007.
- [4] B. Bombin and M. A. Martin-Delgado. Topological computation without braiding. *Phys. Rev. Lett.*, 98(160502), 2007.
- [5] S. B. Bravyi and A. Yu Kitaev. Quantum codes on a lattice with boundary. [quant-ph/9811052](https://arxiv.org/abs/quant-ph/9811052), 1998.
- [6] S. S. Bullock and G. K. Brennen. Qudit surface codes and gauge theory with finite cyclic groups. *J. Phys. A: Math. Theor.*, 40(3481), 2007.
- [7] A.R. Calderbank, E.M. Rains, P.W. Shor, and N.J.A. Sloane. Quantum error correction via codes over $GF(4)$. *IEEE Trans. Inform. Theory*, 44:1369–1387, 1998.
- [8] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill. Topological quantum memory. *J. Math. Phys.*, 43:4452–4505, 2002.
- [9] D. Gottesman. Stabilizer codes and quantum error correction. Caltech Ph. D. Thesis, [eprint: quant-ph/9705052](https://arxiv.org/abs/quant-ph/9705052), 1997.
- [10] M. Grassl, M. Rötteler, and T. Beth. Efficient quantum circuits for non-qubit quantum error-correcting codes. *Internat. J. Found. Comput. Sci.*, 14(5):757–775, 2003.
- [11] A.Y. Kitaev. Fault-tolerant quantum computation by anyons. *Annals. of Physics*, 303:2–30, 2003.
- [12] A. A. Kovalev and L. P. Pryadko. Improved quantum hypergraph-product LDPC codes. In *Proceedings of the IEEE International Symposium on Information Theory, ISIT '12*, pages 348–352, 2012.
- [13] M. Leslie. Hypermap-homology quantum codes. Ph. D. Thesis, [eprint: arXiv:1310.5376](https://arxiv.org/abs/eprint:arXiv:1310.5376), 2013.
- [14] Recall that qubits are placed on the edges of the graph in case of surface codes. If we try to associate a qubit with each dart we find that there are more darts than there are qubits. An additional problem is that $\mathcal{W}/\iota(\mathcal{E})$ has many bases in general. In such a situation each column of H_x and H_z correspond to linear combinations of the labels of the darts. So at first sight it appears that we cannot have a direct correspondence between the qubits and the darts. But fortunately this is not the case.
- [15] R. Raussendorf and H. J. Harrington. Fault-tolerant quantum computation with high threshold in two dimensions. *Phys. Rev. Lett.*, 98(150504), 2001.
- [16] P. Sarvepalli and K. R. Brown. Topological subsystem codes from graphs and hypergraphs. *Phys. Rev. A*, 86(042336), 2012.
- [17] M. Suchara, S. Bravyi, and B. Terhal. Constructions and noise threshold of topological subsystem codes. *Journal of Physics A*, 3, 2011.
- [18] J.-P. Tillich and G. Zemor. Quantum LDPC codes with minimum distance proportional to $n^{1/2}$. In *Proc. IEEE International Symposium on Information Theory*, pages 799–804, 2009.
- [19] G. Zemor. On Cayley graphs, surface codes, and the limits of homological coding for quantum error correction. In *Coding and Cryptology, second international workshop IWCC 2009, LNCS 5557*, pages 259–273. Springer, 2009.