Revista Colombiana de Matemáticas
Volumen 50(2016)2, páginas 295-345

Algebraic Methods for Quantum Codes on Lattices

Jeongwan Haah[1,✉]

[1]Massachusetts Institute of Technology, Cambridge,
Massachusetts, USA

Abstract. This is a note from a series of lectures at Encuentro Colombiano de Computación Cuántica, Universidad de los Andes, Bogotá, Colombia, 2015. The purpose is to introduce additive quantum error correcting codes, with emphasis on the use of binary representation of Pauli matrices and modules over a translation group algebra. The topics include symplectic vector spaces, Clifford group, cleaning lemma, an error correcting criterion, entanglement spectrum, implications of the locality of stabilizer group generators, and the classification of translation-invariant one-dimensional additive codes and two-dimensional CSS codes with large code distances. In particular, we describe an algorithm to find a Clifford quantum circuit (CNOTs) to transform any two-dimensional translation-invariant CSS code on qudits of a prime dimension with code distance being the linear system size, into a tensor product of finitely many copies of the qudit toric code and a product state. Thus, the number of embedded toric codes is the complete invariant of these CSS codes under local Clifford circuits.

Key words and phrases. quantum stabilizer codes, additive codes, symplectic codes, Laurent polynomial ring, toric code, Clifford circuit.

2010 Mathematics Subject Classification. 81P70, 81R05, 12Y05.

1. Introduction

Quantum error correcting codes were invented by Shor who showed that a quantum computer can in principle be built out of faulty components [36, 37]. The basic idea is that despite quantum mechanical state vectors and operators form continuous spaces, errors can be effectively treated as if they were discrete.

Before long, a simple and systematic method of slicing the state vector space into discretely labeled subspaces with help of classical error correcting codes was discovered by Calderbank and Shor [10] and Steane [38], and generalized to what is now known as stabilizer codes by Gottesman [15] or symplectic/additive codes by Calderbank, Rains, Shor and Sloane [8, 9].

Another conceptual and practical use of quantum error correcting codes is provided by Kitaev [25], who presented a class of exactly solvable local Hamiltonians associated with quantum error correcting codes, exhibiting so-called topological order. Although the phenomenology of a certain topological order was known by Sachdev and Read [35], Kitaev's models facilitated understanding significantly, and demonstrated how a topologically ordered medium can be used as a naturally fault-tolerant quantum information processing platform.

In this note, we present a mathematically coherent and mostly self-contained treatment of stabilizer/additive/symplectic codes with applications to quantum spin systems governed by a translation-invariant local Hamiltonian associated with codes. We emphasize binary symplectic vector spaces over groups of Pauli matrices. This makes the appearance of translation-group algebra with coefficients in the binary field very natural, which we will mainly study in the later half of the present note.

We discuss neither a particular way of designing a quantum error correcting code nor decoding algorithms thereof. Also, we do not attempt to develop insight about topological order in general. Rather, this note is to introduce and review notions from commutative algebra that the author has found useful and interesting in the understanding of the cubic code model [18] and its cousins [11, 6, 24, 41, 39, 40]. Most results in this note is hardly new, but the derivation of the results will be sometimes different from existing literature. The classification theorem 5.13 for two-dimensional translation-invariant CSS codes has not previously appeared, and is intimately related to a result of Bombín [5]. Detailed comparison is given in 5.14.

The author thanks Cesar Galindo-Martinez and Julia Plavnik for their hospitality during the workshop in Bogotá, Colombia, and Héctor Bombín for guiding along his paper [5]. The author was supported by Pappalardo Fellowship in Physics while at MIT.

2. Additive/Stabilizer/Symplectic codes

The set of all 2×2 matrices acting on $\mathbb{C}^2 = \text{span}\,\{|0\rangle\,, |1\rangle\}$ has a linear basis consisting of

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{1}$$

called **Pauli matrices**. They square to be the identity, are hermitian, and satisfy

$$XY = iZ, \quad YZ = iX, \quad ZX = iY. \tag{2}$$

Thus, any pair of non-identity Pauli matrices anti-commute. As a complex algebra, we can further reduce the generating set to $\{X, Z\}$ since $Y = -iZX$. The matrix algebra on $(\mathbb{C}^2)^{\otimes n}$ is the n-fold tensor product of the 2×2 matrix algebra, of which the generating set can be chosen as the set of all n-fold tensor products of the Pauli matrices. To avoid lengthy phrasing, let us say just **Pauli operators** to mean n-fold tensor products of Pauli matrices.

An **additive code** [9, 8] or **stabilizer code** [15] is a subspace of n-qubit Hilbert space[1] that is defined as the common eigenspace of a set of commuting Pauli operators of eigenvalue $+1$. The common eigenspace is referred to as **code space** to distinguish it from other related entities. Any vector in the code space is a **code vector**. The defining Pauli operators or any product of them are called **stabilizers**. The stabilizers form the multiplicative **stabilizer group**.

Excercise 2.1. Show that the stabilizer group of nonzero additive code space does not contain -1. ◇

How do we test whether a pair of Pauli operators commute? Since X and Z generate the matrix algebra, we see that any Pauli operator P can be written, for example, as

$$P = \eta \underbrace{(X \otimes I \otimes X \otimes \cdots \otimes X)}_{x=101\cdots1} \underbrace{(Z \otimes Z \otimes I \otimes \cdots \otimes I)}_{z=110\cdots0} = \eta X(x)Z(z) \tag{3}$$

where $\eta = \pm 1, \pm i$, a fourth root of unity. We have associated bit strings x and z to keep record of the positions of nontrivial tensor factors. It is now clear that any Pauli operator is uniquely specified by the overall phase factor η and two bit strings of length n. The commutation relation between a pair of Pauli matrices is then calculated as

[1] In functional analysis, a Hilbert space refers to an inner product space that is complete in the induced metric topology. For finite dimensional vector spaces, the completeness, meaning that every Cauchy sequence converges, follows trivially from the completeness of the real numbers. So, the wording "Hilbert space" is superfluous in our setting where we only consider finite dimensional spaces. Nonetheless, we will keep using this terminology whenever we are referring to the complex vector space of all state vectors.

$$\begin{aligned}
&\eta_1 X(x_1) Z(z_1) \eta_2 X(x_2) Z(z_2) \\
&= \eta_1 \eta_2 X(x_1) \left[(-1)^{x_2 \cdot z_1} X(x_2) Z(z_1)\right] Z(z_2) \\
&= \eta_1 \eta_2 (-1)^{x_2 \cdot z_1} X(x_1 + x_2 \mod 2) Z(z_1 + z_2 \mod 2) && (4)\\
&= \eta_1 \eta_2 (-1)^{x_2 \cdot z_1} X(x_2) X(x_1) Z(z_2) Z(z_1) \\
&= \eta_1 \eta_2 (-1)^{x_2 \cdot z_1} X(x_2) \left[(-1)^{x_1 \cdot z_2} Z(z_2) X(x_1)\right] Z(z_1) \\
&= (-1)^{x_2 \cdot z_1 - x_1 \cdot z_2} \eta_2 X(x_2) Z(z_2) \eta_1 X(x_1) Z(z_1). && (5)
\end{aligned}$$

Hence, any pair of Pauli operators commute or anti-commute, and the two cases are distinguished by

$$-x_2 \cdot z_1 + x_1 \cdot z_2 \mod 2. \tag{6}$$

More generally, we can consider $d \times d$ matrix algebra acting on $\mathbb{C}^d$ with generators

$$X_d = \begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ \vdots & & 0 & 1 & \\ 0 & & & & \ddots \\ 1 & 0 & \cdots & & \end{pmatrix}, \quad Z_d = \begin{pmatrix} 1 & & & & \\ & \omega & & & \\ & & \omega^2 & & \\ & & & \ddots & \\ & & & & \omega^{d-1} \end{pmatrix} \tag{7}$$

where $\omega = \exp(2\pi i/d)$. Due to the commutation relation

$$X_d Z_d = \omega Z_d X_d, \tag{8}$$

any pair of these generalized Pauli operators $X_d(x) Z_d(z)$ and $X_d(x') Z_d(z')$ commutes up to ω^m where

$$m = -x' \cdot z + x \cdot z' \pmod{d} = \begin{pmatrix} x & z \end{pmatrix} \begin{pmatrix} 0 & \mathrm{id} \\ -\mathrm{id} & 0 \end{pmatrix} \begin{pmatrix} x' \\ z' \end{pmatrix} \pmod{d}. \tag{9}$$

When $x = x'$ and $z = z'$, we trivially have $m = 0$.

Excercise 2.2. Verify (9) by deriving the analogue of (5). ◇

Excercise 2.3. Let G be the group of 3×3 matrices of form

$$h_{a,b,c} = \begin{pmatrix} 1 & a & c \\ 0 & 1 & b \\ 0 & 0 & 1 \end{pmatrix} \tag{10}$$

where $a, b, c \in \mathbb{Z}/d\mathbb{Z}$ and the group operation is the matrix multiplication. Show that the matrices in (7) together with $\omega = \exp(2\pi i/d)$ form a representation of the group G. (The group G has a name, **Heisenberg group** over $\mathbb{Z}/d\mathbb{Z}$.) *Hint*: What is the commutator of $h_{1,0,0}$ and $h_{0,1,0}$? What does $h_{0,0,1}$ correspond to? ◇

We may regard the *dit* string x, z as one dit string of length $2n$ [16, 34]. If d is a prime number, then we may further regard the dit string of length $2n$ as a $2n$-dimensional vector over the finite field $\mathbb{F}_d$.[2] Upon multiplication of two Pauli operators, the corresponding dit string is added modulo d, which can be interpreted as the vector addition. In addition, we see that Eq. (9) introduces a symplectic form on this vector space. This symplectic form will be central to further development, and it is thus necessary to understand the symplectic structure thoroughly.

2.1. Symplectic vector spaces

Let $\mathbb{F}$ be any field and V denote a vector space over $\mathbb{F}$. A bilinear form $\lambda : V \times V \to \mathbb{F}$ is called **symplectic** or **alternating** if

$$\lambda(v, v) = 0 \quad \forall v \in V. \tag{11}$$

If the association

$$V \ni v \mapsto \lambda(v, \cdot) \in V^* \tag{12}$$

from V to its dual vector space V^* is bijective, then we say λ is **non-degenerate**. When V is finite dimensional, we can express the λ as a matrix Λ given a basis of V, and the bilinear form λ is non-degenerate if and only if the matrix Λ has a nonzero determinant. Below we will not distinguish the form λ from its matrix representation Λ whenever the basis choice is clear.

Excercise 2.4. Show that the matrix representation of the symplectic form is skew-symmetric ($\Lambda^T = -\Lambda$), but a skew-symmetric matrix may not yield a symplectic form if the field is of characteristic 2. ◇

In order to understand the symplectic space better, we will find a canonical basis. To this end, we consider a variant of Gram-Schmidt orthogonalization for inner product spaces. Let V be a n-dimensional symplectic space, not necessarily non-degenerate. With respect to λ, we can consider the symplectic complement

$$W^\perp = \{v \in V \;:\; \lambda(v, w) = 0 \quad \forall w \in W\} \tag{13}$$

of any subspace W. (Some authors call it as the "orthogonal" complement.)

Lemma 2.5. *If $v, w \in V$ satisfy $\lambda(v, w) \neq 0$, then*

$$V = \operatorname{span}\{v, w\} \oplus (\operatorname{span}\{v, w\})^\perp. \tag{14}$$

[2] At this stage, it is less clear why we need the field rather than just additive group $\mathbb{Z}/d\mathbb{Z}$. See (4). It is actually more of a technical convenience rather than an essential ingredient. However, some of our claims we will make below depend on the fact that $\mathbb{Z}/d\mathbb{Z} = \mathbb{F}_d$ is a field. See Ref. [13] for discussions regarding composite numbers d.

Proof. "$\supseteq$" is trivial by definition. Let $v' = v$ and $w' = w/\lambda(v,w)$ form a basis for $W = \operatorname{span}\{v,w\}$ so that $\lambda(v',w') = 1$. The decomposition

$$x = -\lambda(x,v')w' + \lambda(x,w')v' + \underbrace{x + \lambda(x,v')w' - \lambda(x,w')v'}_{y} \tag{15}$$

for an arbitrary $x \in V$ proves "$\subseteq$" since

$$\begin{aligned} \lambda(y,v') &= \lambda(x,v') + \lambda(x,v')\lambda(w',v') - \lambda(x,w')\lambda(v',v') = 0, &(16)\\ \lambda(y,w') &= \lambda(x,w') - \lambda(x,v')\lambda(w',w') - \lambda(x,w')\lambda(v',w') = 0. &(17)\end{aligned}$$

If $z \in W \cap W^\perp$, then $z = av' + bw'$ for some $a, b \in \mathbb{F}$ and $\lambda(z,v') = \lambda(z,w') = 0$, which imply $a = 0$ and $b = 0$. Therefore, the sum $W + W^\perp$ is direct. ☑

A two-dimensional subspace on which the symplectic form is non-degenerate, as in the lemma, is called a **hyperbolic plane**. Note that the matrix representation of the symplectic form for the hyperbolic plane is

$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}. \tag{18}$$

Suppose we have an unstructured basis $\{v_1, \ldots, v_n\}$ where λ is not always zero. By examining all values $\lambda(v_i, v_j)$, we can find a hyperbolic plane. By the lemma, the symplectic complement of the span of $\{v_i, v_j\}$ has smaller dimension, and we can inductively proceed to decompose the space. At some point the decomposition may encounter a subspace on which the symplectic form vanishes, which is called an **isotropic** subspace. The proof of the lemma gives an algorithm to find a canonical basis, which is essentially the same as the Gram-Schmidt orthogonalization for inner product spaces. We arrive at a structure theorem of finite dimensional symplectic spaces.

Proposition 2.6. *Any finite dimensional symplectic vector space is a direct sum of hyperbolic planes and an isotropic subspace. In particular, a non-degenerate symplectic vector space is even dimensional.*

Excercise 2.7. Show that the dimension of an isotropic subspace of a non-degenerate symplectic space of dimension $2n$ is at most n. ◇

2.2. Automorphisms of symplectic spaces

By definition, an automorphism A of a symplectic vector space V is an invertible linear map from V to itself such that

$$\lambda(Av, Aw) = \lambda(v,w) \quad \forall v, w \in V. \tag{19}$$

With respect to a basis of V the condition for the matrix A to be an automorphism is

$$A^T \lambda A = \lambda, \quad \det A \neq 0 \tag{20}$$

where the superscript T denotes the transpose. If λ is non-degenerate, the second condition $\det A \neq 0$ is redundant.

Let us choose a canonical basis of $2n$-dimensional non-degenerate symplectic vectors space where the matrix representation of the symplectic form is

$$\lambda_n = \begin{pmatrix} 0 & \mathrm{id} \\ -\mathrm{id} & 0 \end{pmatrix} \tag{21}$$

where id stands for the $n \times n$-identity matrix. Thus, for the unit column vectors e_i, where the sole nonzero 1 appears at i-th component $(i = 1, \ldots, n)$, we have $\lambda(e_i, e_{i+n}) = 1 = -\lambda(e_{i+n}, e_i)$, and all the other symplectic pairings vanish.

A few elementary automorphisms can be found by solving an equation,

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^T \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \Leftrightarrow ad - bc = 1. \tag{22}$$

Thus, the automorphism group for a hyperbolic plane is $SL(2, \mathbb{F})$, which is generated by three types of elements

$$S = \begin{pmatrix} 1 & 0 \\ a & 1 \end{pmatrix} \text{ where } a \in \mathbb{F}, \; H = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \text{ and } R = \begin{pmatrix} a & 0 \\ 0 & a^{-1} \end{pmatrix} \text{ where } a \in \mathbb{F}^\times. \tag{23}$$

If $n \geq 2$, we see that there is another automorphism C

$$C|_{\langle e_i, e_j, e_{i+n}, e_{j+n} \rangle} = \begin{pmatrix} 1 & 0 & & \\ a & 1 & & \\ & & 1 & -a \\ & & 0 & 1 \end{pmatrix} \text{ where } a \in \mathbb{F}, \quad 1 \leq i \neq j \leq n \tag{24}$$

where we only displayed the action of C on the four dimensional subspace, on the complement of which C acts by the identity. It is often useful to think of the four elementary **symplectic transformations** (automorphisms) as row operations on a column vector. S adds a-times i-th component to $(i+n)$-th component. H interchanges i-th and $(i+n)$-th with an extra sign. C adds a-times i-th component to j-th component $(j \neq i \leq n)$ while $(-a)$-times $(n+j)$-th to $(n+i)$-th.

Proposition 2.8. *Let Σ be an s- dimensional isotropic subspace of a $2n$-dimensional non-degenerate symplectic space $\mathbb{F}^{2n}$. There exists a symplectic transformation that maps Σ onto the span of $\{e_1, \ldots, e_s\}$, a subset of the canonical basis vectors of $\mathbb{F}^{2n}$. Moreover, the symplectic complement $\Sigma^\perp$ is isomorphic to $\Sigma \oplus W$ for some non-degenerate symplectic subspace W.*

Proof. We provide an algorithm to find the symplectic transformation (automorphism) using S, H, R, and C. Let us identify the subspace Σ with a matrix of basis vectors written in the columns. This matrix is $2n \times s$. Any column operation on the matrix Σ is just a different choice of basis vectors for the space Σ. If we could find a combination of the four elementary symplectic transformations and column operations such that the matrix Σ is transformed to

$$\left(\begin{array}{cc} \mathrm{id} & 0 \\ 0 & 0 \\ \hline 0 & 0 \\ 0 & 0 \end{array}\right),$$

then the composition of the elementary symplectic transformations is the desired transformation. We inserted a horizontal line to distinguish upper and lower half blocks.

Using C, we see that any row operation on the upper-half block can be made symplectic by a suitable row operation on the bottom-half block. Hence, for the upper half-block, we can freely employ row and column operations to the matrix Σ to obtain

$$\Sigma' = \left(\begin{array}{cc} 1 & 0_{1\times(s-1)} \\ 0_{(n-1)\times 1} & \star_{(n-1)\times(s-1)} \\ \hline a & \star \\ b_{(n-1)\times 1} & \star \end{array}\right).$$

Then, S on the 1st and $(n+1)$st row can make the entry a to zero. H's on the all but 1st and $(n+1)$st row bring the submatrix b to the upper half block, and a subsequent row operation can annihilate it. As a result, we obtain

$$\Sigma'' = \left(\begin{array}{cc} 1 & 0_{1\times(s-1)} \\ 0_{(n-1)\times 1} & \star_{(n-1)\times(s-1)} \\ \hline 0 & c_{1\times(s-1)} \\ 0_{(n-1)\times 1} & \star \end{array}\right).$$

Being isotropic, it satisfies $(\Sigma'')^T \lambda_n \Sigma'' = 0$ where λ_n is in (21). This equation tells us that the submatrix c has to be zero. We have reduced the dimension in the problem: $s \to s-1, n \to n-1$. The desired transformation is found by recursion.

The second statement is a corollary of the first since the symplectic complement of $\{e_1, \dots, e_s\}$ is the span of $\{e_1, \dots, e_s\}$ plus the span of $\{e_{s+1}, \dots, e_n, e_{n+s+1}, \dots, e_{2n}\}$, where the latter is non-degenerate. ☑

Proposition 2.9. *S, H, R, and C generate the full automorphism group of a finite dimensional non-degenerate symplectic space.*

Excercise 2.10. Prove 2.9. *Hint*: Transform the matrix of an automorphism into the identity matrix by S, H, R, and C, using the similar strategy as in the proof of 2.8. ◇

2.3. Logical operators

We return to the discussion of Pauli operators. We have seen that the multiplicative group of Pauli operators is rather simple, since, if we ignore the overall phase factors $\pm 1, \pm i$, the group is actually abelian. More precisely,

Proposition 2.11. *If $\mathcal{P}$ denotes the multiplicative group of all Pauli operators including $\pm 1, \pm i$ acting on n-qubit Hilbert space $(\mathbb{C}^2)^{\otimes n}$, then $\mathcal{P}/\langle i \rangle$ is an abelian group, which is isomorphic to the additive group $(\mathbb{Z}/2\mathbb{Z})^{2n}$.*

It is a useful coincidence that this additive group can be regarded as a vector space $\mathbb{F}_2^{2n}$ over $\mathbb{F}_2$. Moreover, the commutation relation naturally endows this vector space with a symplectic form. The form is non-degenerate because X and Z on i-th qubit defines a hyperbolic plane, the direct sum of which is the whole space. We use by convention an ordered canonical basis $e_1, \ldots, e_{2n}$ on the symplectic space $\mathbb{F}_2^{2n}$ in which the symplectic matrix is as in (21). So, the Pauli operator $P(e_i)$ corresponding to e_i is equal up to a phase factor to

$$P(e_i) = \begin{cases} X_i & \text{if } 1 \le i \le n, \\ Z_i & \text{if } n+1 \le i \le 2n \end{cases} \tag{25}$$

where the subscript i denote the sole qubit that is acted on nontrivially by the designated Pauli matrix.

We now introduce the simplest additive code on n qubits. Let $\{Z_i : i = 1, \ldots, n-k\}$ be a commuting set of Pauli operators. What is the code space, the common $(+1)$-eigenspace of the stabilizers? It is obvious that any such eigenvector (code vector) must be of form

$$|0\rangle \otimes \cdots \otimes |0\rangle \otimes |\psi\rangle \tag{26}$$

for some vector $|\psi\rangle \in (\mathbb{C}^2)^{\otimes k}$, and we can identify the code space with this $(\mathbb{C}^2)^{\otimes k}$. There exist Pauli operators $X_{n-k+1}, Z_{n-k+1}, \ldots, X_n, Z_n$ that generate the operator algebra on the code space. These Pauli operators are called logical operators.[3] The logical operators may appear in various guises. For example, $Z_1 Z_n$ has the same action on the code space as Z_n since $(Z_1 Z_n)(Z_n^{-1}) = Z_1$ acts by the scalar $+1$ on the code space by construction. Formally, any Pauli operator that maps the code vector into the code space is called a **logical**

[3] The term "logical" comes from the intended use of the code in an error correction scheme, where information is redundantly encoded into a physical system shielding the "logical level" from errors, and the logical operators are those that transform the encoded information. Of course, in this oversimplified example, there is no protection.

operator. The logical operator P is said to be **equivalent** to another logical operator P' if $P'P^{-1}$ is a stabilizer up to a phase factor. Sometimes, the logical operator is termed as a *nontrivial logical operator* in order to distinguish it from a *trivial logical operator*, where the latter is nothing but a stabilizer.

Excercise 2.12. Show that a Pauli operator is a (trivial or nontrivial) logical operator if and only if it commutes with every stabilizer. ◇

Let us translate the discussion around the simplest example into the language of symplectic spaces. The stabilizers generate an abelian multiplicative group, called **stabilizer group** $\mathcal{S}$. Since it is abelian, the corresponding symplectic space is isotropic. By 2.12, we see that the set $\mathcal{L}$ of all trivial and nontrivial logical operators corresponds to the symplectic complement of this isotropic subspace: $\mathcal{L} = \mathcal{S}^\perp$. By 2.8, $\mathcal{S}^\perp$ contains $\mathcal{S}$ and a non-degenerate subspace which is isomorphic to $\mathcal{S}^\perp/\mathcal{S}$. Any nonzero element in this quotient space corresponds to a nontrivial logical operator. Two different logical operators may have the same action on the code space, which is precisely captured by the quotient space $\mathcal{S}^\perp/\mathcal{S}$. All these statements on the symplectic space does not rely on a specific basis. What does the basis change (automorphism) in the symplectic space over $\mathbb{F}_2$ corresponds to in the space of operators on the Hilbert space over $\mathbb{C}$?

2.4. Clifford group

A unitary operator on n-qubit Hilbert space (over $\mathbb{C}$) is called **Clifford** if it maps any Pauli operator to a Pauli operator. Any Clifford operator U induces a linear map A in the corresponding symplectic space. This is easily proved as

$$\begin{aligned}
&UP(v+w)U^\dagger \propto UP(v)U^\dagger UP(w)U^\dagger, \\
&\pm P(A(v+w)) \propto P(A(v))P(A(w)) \propto P(A(v)+A(w))
\end{aligned}$$

where $P(v)$ denotes a Pauli operator on n qubits specified by the bit string of length $2n$. Since U is invertible, the induces linear map A is also invertible. Recall that the symplectic form λ is defined by

$$P(v)P(w) = (-1)^{\lambda(v,w)} P(w)P(v). \tag{27}$$

Conjugating this equation by U, we see that A preserves the symplectic form in the sense of (19). Therefore, a Clifford unitary induces an automorphism of the symplectic space.

We have found a generating set of the symplectic group on $\mathbb{F}_2^{2n}$. Are the generators induced from Clifford unitary operators? Consider the following unitaries.

$$U_H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, \quad U_S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}, \quad U_{CNOT} = \begin{pmatrix}1 & 0 & & \\ 0 & 1 & & \\ & & 0 & 1\\ & & 1 & 0\end{pmatrix}. \tag{28}$$

They are called Hadamard gate, phase gate, and controlled-not gate, respectively. It is straightforward to verify that U_H interchanges X and Z by conjugation. Therefore, U_H induces the elementary symplectic transformation H of 2.9.

Excercise 2.13. Show that U_S and U_{CNOT} are Clifford, and induce S and C of 2.9, respectively. *Hint*: A linear operator is determined by the image of basis vectors. ◇

By 2.9, it follows that any symplectic automorphism is induced by some Clifford unitary operator. This is summarized by stating

Proposition 2.14. *There exists a surjective group homomorphism from the Clifford group on n qubits to the symplectic automorphism group on $\mathbb{F}_2^{2n}$.*

Excercise 2.15. Generalize this to qudits of a prime dimension d. *Hint*: The symplectic transformation R of 2.9 needs to be included. Find the unitary that induces R. The dimension d being prime means that $\mathbb{F}_d \xrightarrow{\times a} \mathbb{F}_d$ for $a \in \mathbb{F}_d^\times$ is a permutation. ◇

A natural question is then what the kernel of this homomorphism φ is. To find the kernel, suppose $\varphi(U) = \mathrm{id}$, which is to say that

$$UP(v)U^\dagger = \eta P(v) \tag{29}$$

for some phase factor $\eta = \pm 1, \pm i$ that may depend on v. If the action of U on the generators of the Pauli group, then U is uniquely determined. The generators of the Pauli group are the Pauli operators X_i and Z_i for qubit $i = 1, \ldots, n$. Since X_i and Z_i are hermitian, we must have $UX_iU^\dagger = \pm X_i$ and $UZ_iU^\dagger = \pm Z_i$. Suppose $n = 1$, $UXU^\dagger = -X$, and $UZU^\dagger = Z$. One solution to these equations is $U = Z$, and we knew that there is a unique solution. For general n, one can assume U is a tensor product of single qubit operators, and for each factor one finds a Pauli matrix component of U. Thanks to the uniqueness, a solution is the answer. Therefore, we conclude that the kernel of φ is equal to the Pauli group (up to an arbitrary phase factor, which does not alter the conjugation action and hence we ignore). The Clifford group is generated by Pauli group and three types of elements of (28). Actually, the Pauli group is generated by the elements of (28). The phase gate U_S squares to become Z. The Hadamard conjugates it to X. X and Z generates the Pauli group. Hence, we have

Proposition 2.16. *The three types of Clifford unitary operators of* (28) *generates the full Clifford group.*

Now, suppose we have an additive code defined by a set of stabilizers on n qubits. We have learned that this stabilizer group $\mathcal{S}$ corresponds to an isotropic subspace Σ in the symplectic space $\mathbb{F}_2^{2n}$. Let $s = \dim_{\mathbb{F}_2} \Sigma \leq n$. By 2.8, there exists a symplectic transformation that maps Σ onto the span of $e_1, \ldots, e_s \in \mathbb{F}_2^{2n}$. Since φ is surjective, there exists a Clifford unitary that turns the stabilizer group to $\langle X_1, \ldots, X_s \rangle$ up to signs. The set of all logical operators after this Clifford unitary is precisely $\langle X_{s+1}, Z_{s+1}, \ldots, X_n, Z_n \rangle$. This leads to an important conclusion. Let us say that a set of Pauli operators are **independent** if the corresponding binary vectors are linearly independent over $\mathbb{F}_2$.

Theorem 2.17. *Any stabilizer code defined by s independent stabilizers on n qubits has code space dimension 2^{n-s}. There exist Pauli logical operators that generate the full operator algebra acting on the code space.*

Excercise 2.18. Generalize 2.17 to qudits $\mathbb{C}^d$ of prime dimension d. ◇

Since the code space dimension is always a power of 2, it is convenient to work with the exponent

$$k = n - s, \tag{30}$$

which is called **the number of encoded or logical qubits**.

2.5. Cleaning lemma

We have studied the Pauli group $\mathcal{P}$, focusing on its abelianization $\mathcal{P}/\langle i \rangle$, which happens to be a vector space, and the symplectic structure provided by commutation relations. We converted the code space into the simplest one that we clearly understand, like the trivial example of additive code defined in (26). We did so by considering the largest set of transformations, the Clifford group, mapping Pauli operators to Pauli operators.

We are now going to discuss error correction. For this purpose, the notion of locality is of central importance, and we should not make a transformation that breaks the notion of locality. In terms of the vector space associated with the Pauli group, the locality demands us to use a particular basis.

We say a **region** to mean a subset of qubits. An operator is said to be **supported** on a region M, if it acts by identity on the complement of M. The **support** of an operator is the minimal region on which the operator is supported. We learned that given an additive code, there exists a set of logical Pauli operators. Where are they supported? The logical operators are the interface of the code space to the external world, so it is important to know locate them precisely. For a region M, let ℓ_M be the largest number of independent logical operators supported on M. Here, the notion of independence is more restrictive than what we used to define independent stabilizers in 2.17. Given a stabilizer group $\mathcal{S}$ and the corresponding isotropic subspace $\Sigma \subset \mathbb{F}_2^{2n}$, a set of **independent logical operators** is one that maps to a linearly independent set in the quotient vector space $\mathbb{F}_2^{2n}/\Sigma$.

Proposition 2.19.

$$\ell_M + \ell_{M^c} = 2k.$$

The case $\ell_M = 0$ is covered in [7], and more general case is shown in [42]. The proof below follows [21].

Proof. The set of all Pauli operators supported on M corresponds to a vector space $\mathbb{F}_2^{2m}$ spanned by $\{e_i : i \in M\}$, where m is the number of qubits in M. The set of all logical operators on M corresponds to $\Sigma^\perp \cap \mathbb{F}_2^{2m}$. By definition of independent logical operators,

$$\begin{aligned}\ell_M &= \dim_{\mathbb{F}_2}(\Sigma^\perp \cap \mathbb{F}_2^{2m})/(\Sigma \cap \mathbb{F}_2^{2m})\\ &= \dim_{\mathbb{F}_2}(\Sigma^\perp \cap \mathbb{F}_2^{2m}) - \dim_{\mathbb{F}_2}(\Sigma \cap \mathbb{F}_2^{2m}).\end{aligned}$$

Let us consider calculating the first term algorithmically. If we write down the basis vectors for Σ in the rows of a matrix A, then $\Sigma^\perp$ amounts to calculating the kernel of the matrix and transform it with the symplectic matrix λ^{-1}. Since we are only interested in the dimension, the invertible map λ is immaterial. The restriction "$\cap\mathbb{F}_2^{2m}$" means that we have to find the kernel with zero components in entries for M^c. This is to say that we calculate the kernel of the submatrix of A obtained by deleting all columns for M^c. The dimension we seek for is then $2m$ minus the rank of this submatrix. This is a straightforward algorithm, and we translate it back to linear algebra. The rank of this submatrix is the dimension of $\pi_M(\Sigma)$ where π_M is the linear map that sets the components of M^c to zero. The vectors in $\Sigma_M := \Sigma \cap \mathbb{F}_2^{2m}$ will remain untouched by π_M, those in $\Sigma_{M^c} := \Sigma \cap (\mathbb{F}_2^{2m})^\perp$ will be annihilated, and the other vectors will be somehow modified. Consider the decomposition

$$\Sigma = \Sigma_M \oplus \Sigma_{M^c} \oplus \Sigma' \tag{31}$$

where Σ' includes whatever remains beyond $\Sigma_M \oplus \Sigma_{M^c}$. The choice of Σ' is not canonical, but it is easy to check that $\pi_M|_{\Sigma'}$ is injective, so $\dim_{\mathbb{F}_2} \pi_M(\Sigma') = \dim_{\mathbb{F}_2} \Sigma'$. Thus,

$$\begin{aligned}\ell_M &= 2m - \dim_{\mathbb{F}_2} \pi_M(\Sigma) - \dim_{\mathbb{F}_2} \Sigma_M\\ &= 2m - 2\dim_{\mathbb{F}_2} \Sigma_M - \dim_{\mathbb{F}_2} \Sigma'.\end{aligned}$$

By symmetric argument,

$$\begin{aligned}\ell_{M^c} + \ell_M &= 2m + 2(n-m) - 2\dim_{\mathbb{F}_2} \Sigma_M - 2\dim_{\mathbb{F}_2} \Sigma_{M^c} - 2\dim_{\mathbb{F}_2} \Sigma' \\ &= 2n - 2\dim_{\mathbb{F}_2} \Sigma \\ &= 2k\end{aligned}$$

where the last line is by 2.17. ☑

The proposition has an important corollary:

Theorem 2.20 (Cleaning Lemma)**.** *If a region M does not support any nontrivial logical operator* **(correctable)***,*[4] *then for any logical operator there exists an equivalent logical operator supported on the complement region M^c.*

In other words, a correctable region can be *cleaned* of any logical operators.

Proof. The assumption is that $\ell_M = 0$. By 2.19, $\ell_{M^c} = 2k$, which means that every logical operator's action can be achieved by a logical operator on M^c. ☑

This simple fact will incur a number of interesting applications below.

2.6. Error correction criterion

Imagine one has embedded a state vector into an additive code in order to send it over a noisy channel or to store it safely. A physics qubit may be damaged or lost during the process. When and how can we recover from the damage? Let us examine the trivial example in (26) first. If the error occurs in one of the first $n-k$ qubits, and $|0\rangle$ will be mapped to some other state. Our message is not damaged at all, and the formal restoration is achieved simply by replacing the first $n-k$ qubits with fresh qubits in the known $|0\rangle$ state. On the contrary, if error occurs in one of the last k qubits, then there is no way to recover it; the damage is permanent.

In fact, the error correction for general additive codes is not too different. Suppose a physical qubit i in a correctable region is damaged. By the cleaning lemma 2.20, there exists a complete set $\mathcal{L}$ of logical operators supported outside of the damaged qubit i. In particular, every member of $\mathcal{L}$ commutes with any operator on the damaged qubit i. We learned from the discussion leading to 2.17 that there exists a Clifford unitary U that maps the given additive code space into that of the trivial code of (26). U necessarily maps the complete set $\mathcal{L}$ of logical operators to a complete set $U\mathcal{L}U^\dagger$ of logical operators on the trivial

[4] Although in this lecture note we treat the two notions, the correctability and the absence of logical operator, equally, but they are not in general equivalent. The existence of a correcting map is stronger than the absence of operators that act nontrivially within the code space.

code. In the trivial code, given an operator E that acts nontrivially on the last k (logical) qubits, any complete set of logical operators must have one member that does not commute with E. Therefore, if the operator D caused the damage on the qubit i in the correctable region, $UDU^\dagger$ must be supported on the first $n-k$ qubits. The recovery operation R is then easy, and the composition $U^\dagger RU$ is our desired error recovery operation. [5] Summarizing,

Theorem 2.21. *For any error that has occurred in a correctable region of an additive code, there exists a recovery map that corrects it.* [6]

This idea goes back to the very first quantum code by Shor [37]. A more general criterion was discovered shortly after by Knill and Laflamme [28].

The above error correction motivates us to introduce a quantitative attribute to an additive code. The size of the smallest correctable region is one less than that of *the smallest support of any nontrivial logical operator*, where the latter is called **code distance** or **minimal distance**,[7] denoted usually by d. So, whenever error occurs on $d-1$ or less qubits, there exists a recovery map. Obviously, the large d is preferred.

Note that in the above we assumed that we knew the region where the error had occurred. This is not a very realistic assumption, and it is necessary to devise a method to locate the error. This task in general cannot be deterministic, because one has to measure the system in such a way that it does not modify the encoded quantum state where the measurement outcome is probabilistic in nature, and two different errors might result in the identical measurement outcomes. (Often the procedure of locating the error is the hardest step in an *error correcting algorithm* since if the locations are known the recovery map is provided by the code itself.) Therefore, it is important for an error correcting code not only to have large correctable region, but also to admit a reliable (and efficient) error correcting algorithm.

Excercise 2.22. Using the cleaning lemma 2.20, show that the code distance d should obey $2(d-1) \le n-1$ if $k \ge 1$ for any additive code on n qubits with k encoded qubits. (The statement actually follows from *quantum Singleton bound* $2(d-1) \le n-k$ [28], but the weaker version can be derived from the cleaning lemma.) $\diamond$

[5] The standard notion of the operation is a quantum channel, a completely positive and trace preserving linear map on the space of density operators. Everything we said here can be phrased using channels.

[6] This is not a tautology; we have defined the correctable region because of this result.

[7] It is a "distance" when we consider the Hamming distance on the symplectic binary vector space. This jargon is influenced by classical coding theory.

2.7. Entanglement spectrum

A bipartite state vector $|\psi_{AB}\rangle$ can always be written as

$$|\psi_{AB}\rangle = \sum_i \sqrt{p_i} \left|\phi_A^{(i)}\right\rangle \left|\phi_B^{(i)}\right\rangle \tag{32}$$

for positive numbers p_i and some orthonormal vectors $\left|\phi_A^{(i)}\right\rangle$ and $\left|\phi_B^{(i)}\right\rangle$ where $\sum_i p_i = 1$ by the normalization $\langle\psi_{AB}|\psi_{AB}\rangle = 1$. The numbers p_i are called the Schmidt coefficients of the state $|\psi_{AB}\rangle$, and also called the **entanglement spectrum**. The entanglement spectrum is actually a complete set of *invariants* under unitary transformations on either partition; it is invariant under unitaries, and conversely the entanglement spectrum determines the pure state up to unitary transformations on each partition.

A code state vector is defined on a Hilbert space consisting of n tensor factors. Every choice of a subset of qubits (region) defines a bipartition, and one can ask what the entanglement spectrum of a code state is. In general, the entanglement spectrum depends on the particular code vector, but if the code vector is an eigenvector of a maximal set of commuting logical Pauli operators, then the entanglement spectrum turns out to be very simple. The commuting set of logical operators can be regarded as Pauli stabilizers, so the state vector is uniquely determined by a set of n commuting Pauli stabilizers. The eigenvalues of the stabilizers need not be all $+1$.

Proposition 2.23. *Let $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$ be a nonzero common eigenvector of n independent Pauli operators P_i. With respect to any bipartition, the entanglement spectrum is independent of the eigenvalues. In other words, the entanglement spectrum is determined by the binary vectors of the Pauli operators P_i.*

Proof. First the n independent Pauli operator must all commute with one another; otherwise, if P and Q are any two anti-commuting Pauli operators with eigenvalues p, q, respectively, then

$$pq\,|\psi\rangle = pQ\,|\psi\rangle = Qp\,|\psi\rangle = QP\,|\psi\rangle = -PQ\,|\psi\rangle = -qP\,|\psi\rangle = -pq\,|\psi\rangle \tag{33}$$

so $|\psi\rangle = 0$. By 2.8, there exists a Clifford unitary U that conjugates the stabilizer group $\mathcal{S}$ of the given commuting Pauli operators to that of X_i ($i = 1, \dots, n$). The state $U\,|\psi\rangle$ is a common eigenstate of X_i with some eigenvalues. Two states with different eigenvalues ± 1 of X_i can be mapped to each other by some Z_i. Since U is Clifford, $UZ_iU^\dagger$ is also a Pauli operator, and maps $|\psi\rangle$ to another state that is a common eigenvector of $\mathcal{S}$. Since a Pauli operator $UZ_iU^\dagger$ is a tensor product unitary operator, it cannot change the entanglement spectrum with respect to any bipartition. ☑

Theorem 2.24. *Given any bipartition $M \sqcup M^c$ of n qubits, and a nonzero common eigenvector $|\psi\rangle$ of n independent Pauli operators, there exists a tensor product Clifford operator that transforms $|\psi\rangle$ into Bell pairs. In particular, the entanglement spectrum is flat. The number of nonzero Schmidt coefficients is equal to 2^s where*

$$s = |M| - \dim_{\mathbb{F}_2} \Sigma_M. \tag{34}$$

Here, $|M|$ is the number of qubits in M and $\dim_{\mathbb{F}_2} \Sigma_M$ is the number of independent stabilizers supported on M.

The last formula goes back at least to [14]. (See also [22].) It can also be easily derived from [27]; see [31] and references therein.

***Proof*.** From 2.23, we may assume that n independent Pauli operators P_i are commuting and have eigenvalue $+1$. Order the qubits so that those in M are the first $m = |M|$ qubits.

Suppose some P_i is supported on M. The single operator P_i defines an isotropic $\mathbb{F}_2$-subspace of $\mathbb{F}_2^M$, and by 2.8, we see that there exists a Clifford unitary U supported on M such that P_i is mapped to X_1. The eigenstate of X_1 is always of form $|+\rangle \otimes \star$, i.e., the first qubit becomes disentangled by U. This Clifford does not affect the entanglement spectrum between M and M^c, and hence we can remove the first qubit and we are left with one less qubit and one less stabilizer X_1. The formula (34) retains its form since $|M|$ is reduced by 1, and simultaneously $\dim_{\mathbb{F}_2} \Sigma_M$ is reduced by 1. Therefore, without loss of generality we may assume no stabilizer P_i is supported only on M or M^c.

Consider the matrix Σ that has the binary vectors for P_i in the columns. The rows $1, \ldots, m$ and $n+1, \ldots, n+m$ are for M.

We first claim that $n = 2m$. To show this, consider the decomposition of Σ in (31). There we showed that the restriction map π_M that sets the component of M^c to zero is injective on Σ'. By our assumption, $\Sigma = \Sigma'$, and hence π_M is injective. This means that the $2m \times n$ submatrix $\Sigma^{(M)}$ of Σ consisting of rows for M has the same rank as the full matrix Σ. The rank of Σ is n by assumption. This demands that $2m \geq n$. Repeating the argument for M^c in place of M, we have $2(n-m) \geq n$. This proves $n = 2m$.

The column operation of Σ is nothing but a different choice of independent stabilizers of $|\psi\rangle$. Run column operations on Σ such that the $2m \times 2m$ submatrix $\Sigma^{(M)}$ becomes the identity matrix.

$$\Sigma' = \left(\begin{array}{cccc}
1 & 0_{1\times(m-1)} & & \\
0_{(m-1)\times 1} & \mathrm{id}_{(m-1)} & & \\
\star & \star & \star & \star \\
\star & \star & \star & \star \\
\hline
 & & 1 & 0_{1\times(m-1)} \\
 & & 0_{(m-1)\times 1} & \mathrm{id}_{(m-1)} \\
\star & \star & \star & \star \\
\star & \star & \star & \star
\end{array}\right)$$

Next, we apply symplectic transformation on M or M^c without changing the entanglement spectrum, so that the transformed matrix of Σ will be simple. The $(4m-2)\times 1$ submatrix in the first column corresponding to M^c entries cannot be zero, since otherwise the first column will be supported on M. As in the proof of 2.8, we use the elementary symplectic transformations on M^c to obtain

$$\Sigma'' = \left(\begin{array}{cccc}
1 & 0_{1\times(m-1)} & & \\
0_{(m-1)\times 1} & \mathrm{id}_{(m-1)} & & \\
1 & \star & \star & \star \\
0_{(m-1)\times 1} & \star & \star & \star \\
\hline
 & & 1 & 0_{1\times(m-1)} \\
 & & 0_{(m-1)\times 1} & \mathrm{id}_{(m-1)} \\
0 & E_{1\times(m-1)} & f & G_{1\times(m-1)} \\
0_{(m-1)\times 1} & \star & \star & \star
\end{array}\right).$$

We now employ the equation $(\Sigma'')^T\lambda(\Sigma'') = 0$. Due to the first column, the equation enforces $E = 0$, $f = 1$, $G = 0$.

$$\Sigma''' = \left(\begin{array}{cccc}
1 & 0_{1\times(m-1)} & & \\
0_{(m-1)\times 1} & \mathrm{id}_{(m-1)} & & \\
1 & \star & \star & \star \\
0_{(m-1)\times 1} & \star & \star & \star \\
\hline
 & & 1 & 0_{1\times(m-1)} \\
 & & 0_{(m-1)\times 1} & \mathrm{id}_{(m-1)} \\
0 & 0_{1\times(m-1)} & 1 & 0_{1\times(m-1)} \\
0_{(m-1)\times 1} & \star & \star & \star
\end{array}\right).$$

Again by C operations and S operations on M^c, we obtain

$$\Sigma'''' = \left(\begin{array}{cccc}
1 & 0_{1\times(m-1)} & & \\
0_{(m-1)\times 1} & \mathrm{id}_{(m-1)} & & \\
1 & \star & 0 & \star \\
0_{(m-1)\times 1} & \star & 0_{(m-1)\times 1} & \star \\
\hline
 & & 1 & 0_{1\times(m-1)} \\
 & & 0_{(m-1)\times 1} & \mathrm{id}_{(m-1)} \\
0 & 0_{1\times(m-1)} & 1 & 0_{1\times(m-1)} \\
0_{(m-1)\times 1} & \star & 0_{(m-1)\times 1} & \star
\end{array}\right).$$

We see that the first column and $(m+1)$-st column is block diagonal with the rest. These isolated columns represent the state that is stabilized by $X_1 X_m$ and $Z_1 Z_m$. There is a unique such state $(|00\rangle + |11\rangle)/\sqrt{2}$, the **Bell pair**. The entanglement spectrum is $\{1/2, 1/2\}$, and there are two nonzero Schmidt coefficients. The formula (34) is clearly valid. ☑

For generalizations to prime d-dimensional qudits, the entry f in Σ'' should be -1. The number of nonzero Schmidt coefficients is a power of d.

Corollary 2.25. *For any state in an additive code space, the entanglement spectrum of any correctable region M is flat. The entanglement entropy of M is given by* (34)*.*

Proof. By the cleaning lemma 2.20, any logical operator can be pushed away from M. This implies that any observable supported on M has the same expectation value regardless of the actual encoded state. In other words, the reduced density matrix for M is independent of the code state. Therefore, we can conveniently choose the code state to be an eigenstate of a maximal set of commuting logical operators, and we can use 2.24. The entanglement entropy is $-\sum_i p_i \log p_i$, which is equal to log of the number of nonzero Schmidt coefficients because they are the same. ☑

3. Geometric locality

From now on, we consider additive codes whose stabilizer group is generated by Pauli operators supported on small balls in some metric space. We will call an upper bound w on the ball's diameter as the the **geometric locality** of the code.

The error correcting capability of the additive codes is, crudely speaking, due to the entanglement structure of the code states. If the code distance is larger than, say 3, any pair of qubits have the same reduced density matrix for any code state. In other words, the complete data for all pairs of qubits cannot determine the code state at all; the entanglement made the global information hidden from local degrees of freedom. In 3.2 below we will see that this interesting property is severely restricted by the geometric locality.

This signals implications in many-body physics. Physical degrees of freedom, qubits, interact with each other whose strength is only strong for those that are nearby. We cannot in general fully determine the microscopic interaction, and, even if we did, it is almost always hopeless to calculate consequences of the interaction exactly. Instead, physicists model the system and identify important aspects, which can be compared with experiments. The geometrically local additive codes provides a class of physically relevant (i.e., local) models which we can analyze relatively easily. Since the additive codes is designed to produce highly entangled states, the intuition gained from this class of models will be valuable to enhance our understanding of physical systems where entanglement is presumably essential.

Another practical motivation to study geometrically local additive codes is to use them as "firmware" in quantum information processing architectures. A raw physical qubit is likely to be noisy, so it is expected that an error correction layer will be added on top of a system of physical qubits. The error correction scheme would be implemented easier if the necessary operation is on a small local cluster of qubits. One of the most important operations in any error correcting code implementation is to check whether a state is in the code space. In geometrically local codes, this membership test can be done on local clusters of qubits.

The first result under the geometric locality is the following.

Lemma 3.1 (Union Lemma [7, 21])**.** *Let M and N be correctable regions for a geometrically local additive code of locality w. If M and N are separated by distance $> w$, then $M \cup N$ is also correctable.*

Proof. We have to show that any logical operator O supported on $M \cup N$ is trivial. Since O is a tensor product operator, we can write $O = O_M \otimes O_N$ where O_M is supported on M and O_N on N. The locality implies that there is no stabilizer group generator that acts nontrivially on both M and N. By 2.12, each stabilizer group generator must commute with each tensor factor O_M and O_N. This means that each of O_M and O_N is a logical operator. Since each region is correctable, O_M is trivial, so is O_N. The product O is also trivial. ☑

Below we give important applications of the cleaning lemma 2.20 and the union lemma 3.1.

3.1. Code distance is bounded in one dimension

Suppose we have an array of n qubits along a line. We consider an additive code whose stabilizer group is generated by Pauli operators supported on intervals of length at most w. How large can the code distance d be?

Theorem 3.2. *$d \leq 3w$ if the additive code encodes at least one qubit.*

The result in Ref. [7] is essentially this, though they have phrased it primarily with two-dimensional systems.

Proof 1 [21]. Suppose on the contrary that $d > 3w$. Then any interval of length $\leq 3w$ is correctable. For integers $r \geq 0$, let $M_r = [-2rw, 3w+2rw]$ be intervals. By the cleaning lemma, M_0 is correctable.

We claim by induction that M_r for any $r \geq 0$ is correctable. $M_{r+1} \setminus M_r$ consists of two intervals of length $2w$ separated by distance $> w$. By the union lemma, the union of the two intervals is correctable. If O is any logical operator, by the cleaning lemma an equivalent logical operator O' exists on the complement of $M_{r+1} \setminus M_r$, which is $M_r \sqcup M_{r+1}^c$. Since O' is a tensor product operator, we may consider its tensor factor O'_{M_r} on M_r, which by itself must be logical due to the locality. Since M_r is correctable by induction hypothesis, O'_{M_r} is a stabilizer, and the tensor factor $O'_{M_{r+1}^c}$ is equivalent to O. Since O was arbitrary, this implies that on M_{r+1}^c a complete set of logical operators can be found, i.e., $\ell_{M_{r+1}^c} = 2k$, and by 2.19 we conclude that M_{r+1} is correctable. This completes the induction.

For a sufficiently large r, M_r includes all n qubits, and the whole system is correctable. This contradicts the assumption that there exists a nontrivial logical operator. ☑

Proof 2 [7]. For integer $r \in \mathbb{Z}$, let $N_r = [2rw, 2(r+1)w]$. Consider $N_{\text{even}} = \bigcup_{r \in 2\mathbb{Z}} N_r$ and $N_{\text{odd}} = \bigcup_{r \in 2\mathbb{Z}+1} N_r$. If $d > 3w$, then N_{even} is correctable by the union lemma. A nontrivial logical operator can then be found in N_{odd} by the cleaning lemma, which is a contradiction since N_{odd} is correctable as well by the union lemma. ☑

Excercise 3.3. Sharpen the statement of 3.2 using the argument in Proof 2.
◇

Excercise 3.4. Apply the conclusion of 3.2 for codes in higher dimensional lattices, to show that there always exists a nontrivial logical operator supported on a thin slab [7, 21]. ◇

4. Translation invariance

We specialize to the additive codes that obeys translation invariance. The material of this section and the next is mostly from Ref. [19]. An exception is the proof of 5.13. Specifically, we consider an array of qubits on a lattice. The lattice is modeled by the additive group $\mathbb{Z}^D$, where D is called the **spatial dimension**.[8] The lattice is a collection of **sites**, the elements of $\mathbb{Z}^D$. We assume that a finite number q of qubits are located on each site. A local operator on this infinite array of qubits is an operator supported on a ball of finite size. The set of all local operators naturally admits an action from $\mathbb{Z}^D$ by permuting (translating) the tensor factors. A **translation invariant code** is one for which the generating set of the stabilizer group consists of local Pauli operators and is translation-invariant.

This generalizes our previous discussion, since if we set $D = 0$ the notion of translation becomes vacuous. In other words, our study of finite dimensional symplectic vector spaces and automorphism groups was zero-dimensional.

For a positive D, one might be worried about the infinitely large lattice and the corresponding infinite dimensional Hilbert space. We are not going to discuss about this infinity. Instead, we contend ourselves by implicitly considering a family of codes by factoring out sublattices (subgroup) of $\mathbb{Z}^D$ of finite subgroup indices. This index, the order of the quotient group, is more often called **system size**. The procedure of factoring out the subgroup of $\mathbb{Z}^D$ is to impose **periodic boundary conditions** on the lattice.

The translation invariance allows us unambiguously define the stabilizer group over the family of finite systems. It should be noted that the absence of -1 from the stabilizer group (See 2.1) is not always guaranteed. For instance, in $D = 1$ consider the stabilizer group generated by $-Z_i Z_{i+1}$. On the ring of odd length, a product of these generators is equal to -1. We intentionally avoid such a situation, and focus on those where -1 does not appear.[9]

Concretely, the translation-invariance helps us to deal with the formal infinite dimensionality. The bit string that encodes Pauli operator up to a phase factor is really a specification of the support of the Pauli operator. For a Pauli operator on lattice, we have the coordinate system $\mathbb{Z}^D$, so a list of integer D-tuples is all we need. Furthermore, a Pauli operator is a finite product of single qubit operator, which we can write as (assuming $D = 3$)

$$P(a, b, c; e_i)$$

where e_i denotes qubit i within the site (a unit cell) $(a, b, c) \in \mathbb{Z}^3$. By convention, e_i for $1 \leq i \leq q$ means the Pauli X, and e_{q+i} for $1 \leq i \leq q$ means the

[8] This is yet another "dimension," which is different from the Hilbert space dimension, or the binary vector space dimension.

[9] It can be shown that for any finite abelian group of Pauli operators, there exists a finite abelian group of Pauli operators that does not contain -1, but has the same binary vector representation.

Pauli Z. The Pauli Y is represented as $e_i + e_{i+q}$. Let us employ formal variables x, y, z associated with the generators of $\mathbb{Z}^3$, and write

$$(a, b, c) \Leftrightarrow x^a y^b z^c.$$

In previous section, for multiplication of Pauli operators, we added the corresponding bit strings over $\mathbb{F}_2$. Similarly we write, for example,

$$P(1,2,3;e_1)P(-1,2,4;e_{q+2}) \Leftrightarrow xy^2z^3e_1 + x^{-1}y^2z^4e_{q+2} \tag{35}$$

for a general Pauli operator. These observations can be summarized as

Proposition 4.1. *The following two groups are isomorphic.*

- *The multiplicative group of* finitely supported Pauli operators *modulo phase factors on the lattice of dimension D with q qubits per site*
- *The additive group of all* Laurent polynomial column vectors[10] *in D formal variables of length $2q$*

Recall that a vector space is an additive group with an action from a field. Likewise, a **module** M is an additive group with an action ("$\cdot$") from a more general ring R: $r \cdot m \in M$ for any $r \in R$ and $m \in M$. The distributive law is assumed by the same formula: $(r_1 + r_2) \cdot m = r_1 \cdot m + r_2 \cdot m$ and $r \cdot (m_1 + m_2) = r \cdot m_1 + r \cdot m_2$. We will generally omit the dot ("$\cdot$") of the ring action.

The translation group acts on the set of Pauli operators, and hence on the set of Laurent polynomial column vectors. The translation along x-direction by a unit distance increases the x-coordinate by 1, which is equivalent to multiplication by x on the Laurent polynomial column vector. This action naturally defines an action from the group ring $R = \mathbb{F}_2[\mathbb{Z}^D] \cong \mathbb{F}_2[x_1^{\pm 1}, \ldots, x_D^{\pm}]$. Therefore, we speak of **Pauli module** over the the translation group algebra R for the additive group of all Laurent polynomial column vectors of length $2q$.

By the definition of the translation invariant codes, the stabilizer group is closed under the translation group action. This is to say that the set of Laurent polynomial column vectors corresponding to the stabilizer group is an R-submodule, which we call **stabilizer module**, of the Pauli module.

In the zero-dimensional study, the symplectic form emerged from the commutation relation. We interchanged the vector components corresponding to X and Z parts, and using the dot product we counted the number of overlap. The interchange of X and Z components is readily defined for the Pauli module elements, but dot product is not immediately applicable. For Laurent polynomial

[10] We have overloaded the term "vectors." It is a $2q \times 1$ matrix over the Laurent polynomial ring, but this is too wordy. As a vector over the base field $\mathbb{F}_2$, the number of components is infinite. Here, we just say column vectors to mean anything that has several components arranged in a column.

column vectors, what we want is to count the number of overlapping terms. A trick is to consider the following antipode, denoted by the bar over the element.

4.1. Symplectic form

The **antipode map**[11] is an involutive[12] $\mathbb{F}$-linear map from R to R defined by

$$x_1^{a_1} \cdots x_D^{a_D} \mapsto \overline{x_1^{a_1} \cdots x_D^{a_D}} := x_1^{-a_1} \cdots x_D^{-a_D}. \tag{36}$$

It is then clear that for any $f, g \in R$,

$$\text{Sum of the coefficients of overlapping terms of } f \text{ and } g = \text{Coefficient of 1 in } \bar{f}g. \tag{37}$$

The latter quantity actually defines a $\mathbb{F}$-bilinear form on R (valued in $\mathbb{F}$), and naturally generalizes to R^n, which we call **dot product**:

$$\text{For any } v, w \in R^n, \text{ define } (v \cdot w) = \left(\text{Coefficient of 1 in } \sum_{i=1}^{n} \bar{v}_i w_i \right) \in \mathbb{F}. \tag{38}$$

Clearly, this dot product generalizes the zero-dimensional ($D = 0$) dot product. Note that

$$(gv \cdot w) = (v \cdot \bar{g}w) \text{ for any } g \in R, \; v, w \in R^n. \tag{39}$$

The generalization of (9), the **symplectic form**, on R^{2n} is given by

$$\lambda(v, w) = (v \cdot \lambda_n w) \tag{40}$$

where λ_n is the $2n \times 2n$ symplectic matrix (21). We used the same symbol λ as it is a generalization of the zero-dimensional case. With respect to this symplectic form, we continue to say that an R-submodule Σ of R^n is **isotropic** if

$$\forall v, w \in \Sigma, \quad \lambda(v, w) = (v \cdot \lambda_n w) = 0. \tag{41}$$

Proposition 4.2. *A submodule Σ of R^{2n} generated by the columns of a Laurent polynomial matrix σ is isotropic if and only if*

$$\bar{\sigma}^T \lambda_n \sigma = 0 \tag{42}$$

as a matrix. We will denote $\bar{\sigma}^T$ by $\sigma^\dagger$.

[11] It is the antipode map of the group algebra $R = \mathbb{F}[\mathbb{Z}^D]$ taken as a Hopf algebra.

[12] The inverse is itself.

Proof. Say σ is $2n \times t$. ($\Leftarrow$) An arbitrary element of Σ is σh for some column vector $h \in R^t$. Thus,

$$\lambda(\sigma h, \sigma h') = \text{Coeff. of } 1 \text{ in } (h^\dagger \sigma^\dagger \lambda_n \sigma h') = 0.$$

($\Rightarrow$) We must have

$$\text{Coeff. of } 1 \text{ in } (e_i^\dagger \sigma^\dagger \lambda_n \sigma g^{-1} e_j) = 0$$

for any basis unit vector e_i, e_j, and any monomial g. e_i and e_j selects (i, j)-entry of the matrix $\sigma^\dagger \lambda_n \sigma$. The coefficient of 1 in a Laurent polynomial f that is shifted by g^{-1} is exactly the coefficient of g in f. Since g addresses any term, this implies that the (i, j)-entry is identically zero. ☑

***Remark* 4.3.** The dot product is a non-degenerate symmetric bilinear form on R. That is, if $(v \cdot w) = 0$ for all $w \in R$, then we must have $v = 0$. This means that the $\mathbb{F}$-linear map $v \mapsto f_v \in R^*$, where $f_v : w \mapsto (v \cdot w)$ is a linear functional on R, is injective. This association is not surjective whenever $D > 0$, but becomes surjective if we consider "finite systems" by imposing conditions such as $x_i^{L_i} = 1$ since R becomes finite dimensional $\mathbb{F}$-vector space.

The dual space R^* can be endowed with an R-module structure by defining $rf(\star) = f(\bar{r}\star)$ for any $r \in R$ and $f \in R^*$. If $f = (v \cdot \star)$, then $rf = f(\bar{r}\star) = (v \cdot \bar{r}\star) = (rv \cdot \star)$ by (39).

Given a R-module map $\varphi : R^n \to R^m$, i.e., an $m \times n$ matrix with entries in R, we can consider its dual $\varphi^* : (R^m)^* \to (R^n)^*$ by the rule $(\varphi^* f)(\star) = f(\varphi(\star))$. If the association $v \mapsto (v \cdot \star)$ is bijective, then we can consider

$$R^m \xrightarrow{\cong} (R^m)^* \xrightarrow{\varphi^*} (R^n)^* \xrightarrow{\cong} R^n$$

and ask the matrix representation of this R-module map. The answer is simply $\varphi^\dagger$. This is in fact how the transpose for real matrices and the hermitian conjugate for complex matrices are defined with respect to the usual inner product. ◇

***Remark* 4.4.** The base ring $R = \mathbb{F}[\mathbb{Z}^D]$ can be obtained from a polynomial ring in D variables by inverting a single element $x_1 \cdots x_D$. In particular, it is Noetherian. Hence, every submodule of R^n is finitely generated, which means that there always exists a (rectangular) finite matrix σ whose columns generate the given submodule over R. Almost all statements below require that the stabilizer module is finitely generated, which is guaranteed by the finite dimensional lattice $\mathbb{Z}^D$. For thorough treatment of Noetherian rings and modules, consult Atiyah-MacDonald [3]. ◇

4.2. Stabilizer and Excitation map

When there are q qubits per site on the lattice $\mathbb{Z}^D$, the matrix equation of 4.2 reads

$$\sigma^\dagger \lambda_q \sigma = 0. \tag{43}$$

Note that this is a genuine generalization of the zero-dimensional case $D = 0$. The number q is the total number of qubits in the system, and † reduces to the usual transpose. The matrix σ is usually called a **generating matrix** of the code on q qubits. In our convention, the columns of σ are the stabilizer group generators.[13] We call σ a **stabilizer map** in order to emphasize the importance of the image of this R-linear map rather than the particular matrix representation.

It will prove useful to think of the $t \times 2q$ matrix

$$\epsilon = \sigma^\dagger \lambda_q \tag{44}$$

separately from σ and define a chain complex

$$(E = R^t) \xleftarrow{\epsilon} (P = R^{2q}) \xleftarrow{\sigma} (G = R^t) \tag{45}$$

of length 2. (A **chain complex** is an array of maps such that the composition of any consecutive maps is zero. It has nothing to do with complex numbers.)

Previously in Section 3, we briefly noted that the geometrically local additive codes model physical systems. This is because local codes define a Hamiltonian

$$H = -\sum_g P_g \tag{46}$$

where P_g are the hermitian local stabilizer generators. The generators may be redundant. The Hamiltonian actually depends on the particular choice of generators, and is not uniquely determined by the code space. The lowest energy eigenspace (*ground* space) of H, however, is by definition the code space regardless of the choice of P_g.

What about higher energy states? These *excited* eigenspace decomposes into the eigenspaces of individual generators P_g; the ground space has eigenvalue $P_g = +1$. For each eigenspaces, we may visualize the distribution of the eigenvalues of P_g's in the lattice, as P_g is supported on a small region of the lattice. These eigenvalues are precisely what we can measure without worrying about corrupting the encoded states in the ground (code!) space. Any eigenvalue -1 of P_g is called a **defect** or **excitation**.

[13] Many other references make convention where rows of a binary matrix represent stabilizer group generators.

Excercise 4.5. Show that for any common eigenspace of P_g's, there exists a Pauli operator that maps the ground space onto it. ◇

In the translation invariant codes, the Hamiltonian can also be chosen to be translation invariant, and the choice of the Hamiltonian is conveniently expressed by a particular matrix σ of the stabilizer map. The columns represent Pauli operators supported near the origin, and all the other terms in the Hamiltonian is obtained by translations. In this case, the excitations are less sensitive to the choice of the matrix σ. Consider a state $|\psi_e\rangle$ that has an excitation $P_i = -1$ around a site i with respect to the σ. If we choose another Hamiltonian, $-\sum_{g'} P'_{g'}$, for the same code space, represented by σ', then the operator P_i as a Laurent polynomial column vector v should be given by some R-linear combination of the columns of σ':

$$v = \sum_j a_j \sigma'_j, \quad a_j \in R. \tag{47}$$

Since a_j consists of finitely many terms, this means that P_i is a product of some finitely many local terms $P'_{g'}$ around i, among which there must be an operator $P'_{g'}$ that has eigenvalue -1 on $|\psi_e\rangle$. The location of the excitation $P'_{g'} = -1$ is not too different from that of $P_i = -1$.

The matrix ϵ yields a convenient way to determine the locations of the defects (excitations) when a Pauli operator acts on the ground state. Let $|\psi\rangle$ be a ground state (code state), and let P be arbitrary Pauli operator that is finitely supported. To locate the excitations of $|\psi'\rangle = P|\psi\rangle$, we consider

$$P_g|\psi'\rangle = P_g P|\psi\rangle = \pm P P_g|\psi\rangle = \pm P|\psi\rangle = \pm|\psi'\rangle \tag{48}$$

where the sign ± is determined by the commutation relation between P_g and P. We know how to express this sign by the symplectic form. The Hamiltonian term P_g is a translation by $g \in \mathbb{Z}^D$ of one of the columns of σ, say i-th column σ_i. The Pauli operator P is expressed by some column vector v. The anticommutation happens precisely when

$$1 = (g\sigma_i \cdot \lambda_q v) = (g \cdot \sigma_i^\dagger \lambda_q v). \tag{49}$$

If we vary $g \in \mathbb{Z}^D$, then we will collect all positions of the excitation associated with σ_i, and may convert these position data into a Laurent polynomial. But, this Laurent polynomial is exactly $\sigma_i^\dagger \lambda_q v$. If we collect these polynomials, one for each column i of σ, we obtain a column vector of Laurent polynomials of length t, which is equal to $\sigma^\dagger \lambda_q v$. This shows how $\epsilon = \sigma^\dagger \lambda_q$ defines a map from the Pauli operators to the excitations. The map ϵ is the **excitation map**.

Excercise 4.6. Recall 2.12. Assume periodic boundary conditions for the lattice to work with finite systems, and show that a Pauli operator is logical if and only if its Laurent polynomial representation v satisfies $\epsilon(v) = 0$. Show also that it is nontrivial if $v \notin \operatorname{im} \sigma$. ◇

4.3. Symplectic/Locality-preserving Clifford transformations

In the zero-dimensional study above, we identified symplectic transformations that preserves the symplectic form in the abstract vector space, and found a generating set for the symplectic group. We then showed that, on the $\mathbb{F}_2$-symplectic space derived from the commutation relation among Pauli operators on qubits, the generators of the symplectic group are induced by Clifford unitary transformations on qubits. Here we parallel the discussion with the translation invariance. Recall that q is the number of qubits per site, and $R = \mathbb{F}_2[x_1^\pm, \ldots, x_D^\pm]$ is our base ring, the translation group algebra.

A $2q \times 2q$ matrix T on R^{2q} is symplectic if it satisfies the matrix equation

$$T^\dagger \lambda_q T = \lambda_q. \tag{50}$$

Restricting the entries of T to be in $\mathbb{F}_2$ we recover the symplectic group in the zero-dimensional case. This subgroup of the symplectic group is induced by the application of the Hadamard, the phase, and the controlled-not gate on every unit cell, uniformly over the lattice.

There are other symplectic transformations, of which we enumerate a few. When $q = 1$, for any monomial $g = x_1^{a_1} \cdots x_D^{a_D}$, the matrix $\begin{pmatrix} g & 0 \\ 0 & g \end{pmatrix}$ is symplectic. Interpreting in terms of action on qubits, this amounts to translating qubits into $(a_1, \ldots, a_D)$-direction. This certainly maps a Pauli operator to a Pauli operator and preserves the size of the support; it is a locality-preserving Clifford transformation.

Assuming $q = 1$ still, we consider transformation of form $S_f = \begin{pmatrix} 1 & 0 \\ f & 1 \end{pmatrix}$. Plugging it into (50), we obtain an equation $f = \bar{f}$. The matrix S of (23) corresponds to S_f for $f \in \mathbb{F}_2 \subseteq R$. S_f with a general f can be split into a product of finitely many $S_{m+\bar{m}}$ for a monomial m and at most one S_a for $a \in \mathbb{F}_2$. The transformation $S_{m+\bar{m}}$ is new arising from the translation structure, and is induced by a *controlled-Z* gate

$$U_{CZ} = \mathrm{diag}(1, 1, 1, -1) \tag{51}$$

on a pair of qubits. Since it is diagonal, it commutes with any other U_{CZ} acting on other qubits. This commutes with Pauli Z, so only nontrivial action is on Pauli X. It is simple calculation to verify that $U_{CZ}(X \otimes I)U_{CZ}^\dagger = X \otimes Z$ and $U_{CZ}(I \otimes x)U_{CZ} = Z \otimes X$. This implies that if U_{CZ} acts on every pair of qubits separated by the displacement $m \in \mathbb{Z}^D$, then the induced symplectic transformation is precisely $S_{m+\bar{m}}$.

When $q \geq 2$, we can generalize the controlled-NOT (24). If we apply the controlled-NOT translation invariantly where the target qubit is at $m \in \mathbb{Z}^D$ *relative* to the control qubit, then the presence of X at the control qubit will bring a new X to the target qubit, and the presence of Z at the target qubit will bring a new Z to the control qubit. Since the controlled-NOT's on two different but overlapping pairs in general do not commute, the control and target qubit should be the different qubit within the unit cell, in order to define the controlled-NOT uniformly over the lattice unambiguously. Such a translation-invariant controlled-NOT induces a symplectic transformation

$$C|_{\langle e_i, e_j, e_{i+n}, e_{j+n}\rangle} = C(m) := \begin{pmatrix} 1 & 0 & & \\ m & 1 & & \\ & & 1 & -m \\ & & 0 & 1 \end{pmatrix} \quad \text{where} \quad 1 \le i \ne j \le n \tag{52}$$

where m is a monomial of R. Since $C(m)C(m') = C(m+m')$, the entry of m in the matrix (52) can be occupied by any Laurent polynomial of R.

***Remark* 4.7.** A natural question in analogy with 2.9 is whether the symplectic transformations that are found so far generate the full symplectic group. The author does not know the answer when $D \ge 2$. The case $D = 0$ is covered in 2.9, and the case $D = 1$ will be solved in the next section. ◇

5. Low dimensions

5.1. Smith normal form

For a moment, we digress from the translation invariant additive codes, and consider matrices over a "nice" ring. The conclusion will have immediate applications in one-dimensional additive codes.

A ring is an abelian group where two elements may be multiplied. Integers, complex numbers, polynomials, etc. form rings. An **ideal** I of a ring R is a subset of the ring that is a subgroup of R under addition such that $rm \in I$ for all $r \in R$ and $m \in I$. So, an ideal is a collection of multiples of its members, and sums thereof. Most of the time, we only care about the generators, the multiples of which form the ideal. When the generators g_i are known, we write $I = (g_i, \ldots)$. A commutative ring with 1 is a **principal ideal domain** if no nonzero elements multiply to become zero and every ideal is generated by a single element. Important examples are the ring of integers $\mathbb{Z}$, and the (Laurent) polynomial ring in one variable over a field $\mathbb{F}[x]$ ($\mathbb{F}[x, x^{-1}]$). These are examples of **Euclidean domains**.

Why is $\mathbb{Z}$ a principal ideal domain? Suppose $I = (a, b)$ is an ideal of $\mathbb{Z}$. If $b > a > 0$, then we can consider $(a, b-a)$, and see that it is the same ideal as before because an ideal is closed under addition. We can proceed similarly to find smaller and smaller generators, but this must stop at some point because positive integers cannot decrease forever. In the end, we must be left with a single number (principal generator), which is actually the greatest common divisor of a and b. This is the Euclid's algorithm, and proves that $(a, b) =$

$(\gcd(a,b))$. This argument will always give the single generator of an ideal of $\mathbb{Z}$ generated by a finitely many elements. For an arbitrary ideal I where the number of generators is unknown, one can consider the smallest positive member d of the ideal, and prove that every member of I has to be a multiple of d. Important is the division algorithm that allows us to find a *smaller* element as a linear combination of two given elements. This argument applies with a slight modification to the univariate polynomial and Laurent polynomial ring. The "size" of a polynomial is measured by the degree of the polynomial, and the size of a Laurent polynomial is measured by the difference of the greatest exponent and the least exponent. We can rephrase the implication of the algorithm as follows.

Proposition 5.1. *For any column vector v over an Euclidean domain, there exists a finite product M of elementary row operation matrices such that the column vector Mv consists of a single nonzero entry.*

5.1.1. Classification of finitely generated abelian groups

A group G is said to be **finitely generated** if there is a finite set of members (generators) $g_1, \dots, g_n$ of the group such that all other members can be written as a finite product of the generators and their inverses. Following the convention for abelian groups, we should denote the group operation as a sum instead of a product. So, any group element can be written as

$$\sum_{i=1}^{n} c_i g_i \tag{53}$$

where $c_i \in \mathbb{Z}$. This means that the n-tuple of integers $(c_1, \dots, c_n)$ can express any element of the group, and we may say that the map

$$\mathbb{Z}^n \to G \tag{54}$$

is surjective. This map is a group homomorphism as one can easily check from (53). Whenever we see a surjective homomorphism we should consider the kernel K. The kernel itself is finitely generated[14] by $k_1, \dots, k_m \in \mathbb{Z}^n$. We again think of the kernel as the image of the map

$$\varphi : \mathbb{Z}^m \to \mathbb{Z}^n \tag{55}$$

[14] This is not too trivial. One of the best ways to show this is through the notion of **Noetherian** rings and modules. Here is a sketch. A Noetherian module is one of which any submodule is finitely generated. An equivalent definition is that every increasing chain of submodules saturates. Similarly, a Noetherian ring is Noetherian if it is Noetherian module over itself. After showing the equivalence of the definitions, one can further show that finitely generated modules over a Noetherian ring is Noetherian. Since $\mathbb{Z}$ is a principal ideal domain, it is Noetherian. The kernel is a submodule of $\mathbb{Z}^n$ and therefore is finitely generated.

sending the unit vectors of $\mathbb{Z}^m$ to k_i. In this way we *present* the group G as the cokernel of φ:

$$\mathbb{Z}^m \xrightarrow{\varphi} \mathbb{Z}^n \to G \to 0. \tag{56}$$

If we express the map φ as a matrix, it will contain the generators $k_1, \ldots, k_m$ in the columns of the $n \times m$ integer matrix M.

Now, what do the row and column operations on M correspond to? Instead of choosing the generating set of G as $g_1, \ldots, g_n$, we could choose $g_1 + g_2, g_2, g_3, \ldots, g_n$. In the latter case, we would write (53) as

$$g = c_1(g_1 + g_2) + c_2 g_2 + \cdots c_n g_n = c_1 g_1 + (c_1 + c_2) g_2 + \cdots c_n g_n \tag{57}$$

We see that different choices of the generators lead to row operations on M. Similarly, the choice of generators of K is of course arbitrary, and since the columns of M are the generators of K, this corresponds to column operations on M.

Therefore, any row and column operations on M does not change the isomorphism class of G; they are just differences how we describe the group G. Let us use the Euclid's algorithm in order to simplify M. Pick any nonzero column and run the Euclid's algorithm to eliminate all but one entries in the upper-left corner. Run the algorithm on the first row, to single out a nonzero entry on the upper-left corner. If any nonzero element appears, repeat. Since a positive integer cannot decrease forever, this procedure must end after finitely many iterations. A nonzero element will reside at the upper-left corner and all other entries in the first column and first row will be zero.

$$\begin{pmatrix} \star & \star & \star \\ \star & \star & \star \\ \star & \star & \star \end{pmatrix} \mapsto \begin{pmatrix} d_1 & 0 & 0 \\ 0 & \star & \star \\ 0 & \star & \star \end{pmatrix} \tag{58}$$

If the bottom-right block contains an integer that is not divisible by d_1, then we bring that integer to the first row, and repeat the above. This will decrease the number in the upper-left corner, and after finitely many iterations, d_1 will divide all the other entries.

We inductively proceed to obtain

$$\begin{pmatrix} d_1 & 0 & 0 \\ 0 & \star & \star \\ 0 & \star & \star \end{pmatrix} \mapsto \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & \ddots \end{pmatrix} \tag{59}$$

where

$$d_1 \mid d_2 \mid \cdots \mid d_r. \tag{60}$$

where $d_r > 0$ and r is the rank of the matrix M, i.e., r is maximal such that some $r \times r$ submatrix of M has nonzero determinant. This diagonal form is called the **Smith normal form** of M. The diagonal elements are called **elementary divisors** of M. We have defined the elementary divisors as a result of the Smith algorithm. There is some arbitrariness in the details of the algorithm. So, a priori, we do not know whether the elementary divisors are unique regardless how we obtain them. However, there is another characterization of elementary divisors, which will prove their uniqueness. For any rectangular matrix M, let $I_s(M)$ be the ideal generated by the determinants of $s \times s$ submatrices of M, called **s-th determinantal ideal** of M.

Proposition 5.2. *It holds that $I_s(M) = I_s(AM) = I_s(MB)$ for any invertible matrices A and B. The elementary divisors of M are determined by the determinantal ideals of M, and hence are uniquely determined by M.*

Proof. The second claim follows from the first because $d_1 \cdots d_s$ is the (principal) generator of $I_s(M)$.

To show the first claim, suppose three matrices satisfy $AM = C$. An $(s \times s)$-submatrix C' of C is the product of some $s \times s'$ submatrix A' of A and some $s' \times s$ submatrix M' of M. Each row v_i is a linear combination of rows b_j of M' as $v_i = \sum_{j_i} A'_{ij_i} b_{j_i}$. The minor $\det C'$ is a multilinear function of rows v_i of C';

$$\det C' = \det(v_1, \ldots, v_s) \tag{61}$$

$$= \sum_{j_1, j_2, \ldots, j_s} A'_{1j_1} A'_{2j_2} \cdots A'_{sj_s} \det(b_{j_1}, \ldots, b_{j_s}). \tag{62}$$

This implies that $\det C'$ is a linear combination of minors of M'. Since C' was arbitrary, we see that $I_s(C) \subseteq I_s(M)$. If A is invertible, then the opposite inclusion holds, implying an equality. This proves $I_s(AM) = I_s(M)$. To show $I_s(MB) = I_s(B)$, transpose everything above. ☑

Since G is the cokernel of M, we see that

$$G \cong \mathbb{Z}/d_{t+1}\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}/d_r\mathbb{Z} \oplus \mathbb{Z}^{n-r}. \tag{63}$$

where $d_t = 1 < d_{t+1}$.[15] We have decomposed the group G into familiar abelian groups, effectively by row and column operations on the presentation of G.

***Remark* 5.3.** Over any principal ideal domain the Smith normal form is defined: For any matrix M there exists invertible matrices A and B such that AMB is diagonal such that upper-left elements divide lower-right elements.

[15] To show that this is a unique expression, observe the following. The number of the direct summands $\mathbb{Z}$ is the vector space dimension upon tensoring $\mathbb{Q}$ over $\mathbb{Z}$. In addition, for the minimal n, either $d_1 = 0$ or $d_1 > 1$.

The difference is that over a non-Euclidean domain, the algorithm to find A and B may need to do more than just adding one row (column) to another row (column). The conclusion that the Smith normal form is unique (i.e., the elementary divisors are well-defined) remains true. $\diamond$

5.2. Classification in one dimension

We turn back to the lattice codes with translation structure, here with the simplest possible translation. Consider qubits arranged on a straight line where q qubits are clustered at each site $i \in \mathbb{Z}$. The translation group is $\mathbb{Z}$ and we identify the group algebra as $R = \mathbb{F}_2[x, x^{-1}] \cong \mathbb{F}_2[\mathbb{Z}]$. We have shown that a translation invariant additive code on this array of qubits is defined by a stabilizer module Σ over R. The stabilizer module has finitely many generators as an R-module, and if we express the generators in the columns of a $2q \times t$ matrix σ then it satisfies

$$\sigma^\dagger \lambda_q \sigma = 0 \tag{64}$$

by 4.2. The entries of σ are Laurent polynomials with one variable x. The ring R happens to be a Euclidean domain, and we can try to convert the matrix σ into a simpler form by the elementary symplectic transformations that we found in Section 4.3. We will show that the Smith normal form of σ can be obtained.

Proposition 5.4. *For any translation-invariant one-dimensional additive code, there exists a locality-preserving and translation-invariant Clifford transformation such that the stabilizer map is diagonal with zero matrix in the bottom half. The diagonal elements completely determines the equivalence class of the translation-invariant additive code up to Clifford operations.*

This has appeared in [17] and also in [19].

Proof. We employ the technique used in the proof of 2.8. Using the controlled-NOT, Hadamard, and column operations, we can bring σ into

$$\sigma' = \begin{pmatrix} f & 0 \\ 0 & \star \\ g & \star \\ 0 & \star \end{pmatrix}. \tag{65}$$

The equation $\sigma'^\dagger \lambda_q \sigma' = 0$ demands that $f\bar{g} = \bar{f}g = \overline{f\bar{g}}$. Suppose $f = \alpha x^a + \cdots + \beta x^b$ and $g = \gamma x^c + \cdots + \delta x^d$ with the exponents increasing and $\alpha, \beta, \gamma, \delta$ are all nonzero. Assume harmlessly that the degree of f is smaller than that of g: $b - a < d - c$. (If not, apply Hadamard to interchange them). Then the equation $f\bar{g} = \overline{f\bar{g}}$ implies that $\gamma/\alpha = \delta/\beta$ and $a + b = c + d$. Setting

$h = (\gamma/\alpha)(x^{c-a} + x^{d-b})$, we see that $h = \bar{h}$ and controlled-Z by h reduces the degree at the position of g by at least 2. Since the degree cannot decrease forever, after finitely many iterations we obtain a column with sole nonzero entry. Then, the top row of the bottom half block must be zero, due to the equation $\sigma\lambda_q\sigma = 0$.

$$\sigma'' = \begin{pmatrix} d_1 & 0 \\ 0 & \star \\ 0 & 0 \\ 0 & \star \end{pmatrix}. \tag{66}$$

If there is any entry that is not divisible by d_1, then we can bring it to the first row by Hadamard and controlled-NOT, and repeat the above. The degree cannot decrease forever, and we must be left with a stabilizer map in the form of σ'' where d_1 divides every entry. We finish by induction in q. ☑

5.2.1. Coarse-graining

If we are lenient about the translation structure, then stronger classification can be obtained. The translation group $\mathbb{Z}$ has subgroups $b\mathbb{Z}$ for any positive integer b. Any stabilizer module is a module over this smaller translation group, and we can consider Clifford operations that conforms with this smaller translation group. This can be viewed as taking a larger unit cell in the lattice. Instead of saying that the unit cell consists of q qubits, we now take the unit cell to consist of bq qubits.

To be clear, by **coarse-graining** we mean taking a smaller base ring $R' = \mathbb{F}_2[x^b, x^{-b}]$ of $R = \mathbb{F}_2[x, x^{-1}]$, and regarding all R-modules to R'-modules.

As a standalone ring, R' is isomorphic to R, but now R is a module of rank b over R' with a basis $1, x, \ldots, x^{b-1}$. The Pauli module R^{2q} is now R'^{2bq} as $R = R'^b$. Since the stabilizer map is from R^t to R^{2q}, under the coarse translation group, the new stabilizer map is from R'^{bt} to R'^{2bq}, and the corresponding matrix gets bigger by a factor of b. To figure out the bigger matrix, observe that each entry in the stabilizer map can be regarded as a map $R \to R$. Over the smaller ring R', this map has to be represented by $R'^b \to R'^b$. Since the multiplication in R is compatible with the composition of maps $R \to R$ (That is what module is about), it is enough to find the matrix representation of the generator x of the ring over $\mathbb{F}_2$. The multiplication by x sends the basis elements $1, x, \ldots, x^{b-1}$ to $x, x^2, \ldots, x^b = x' \cdot 1$. Hence,

$$(x : R'^b \to R'^b) = \begin{pmatrix} 0 & & & x' \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{pmatrix}. \tag{67}$$

Excercise 5.5. Consider the **Ising model** whose stabilizer map is given by $\sigma_{\text{Ising}} = \begin{pmatrix} 1+x \\ 0 \end{pmatrix}$ on the one-dimensional lattice. Write down the corresponding Hamiltonian according to (46). Take a smaller translation group $2\mathbb{Z}$ instead of $\mathbb{Z}$, and rewrite a corresponding stabilizer map, which should be 4×2. Verify that the Hamiltonian is not changed. ◇

Using this passive coarse-grain procedure, the Clifford group actually becomes larger. From 5.4, the only potentially interesting stabilizer map is (f) where we omit the lower half block and f is a Laurent polynomial. Since multiplying a monomial to f does not change $\operatorname{im}(f)$ at all, we may assume that all exponents of f is nonnegative and $f(x = 0) \neq 0$. Upon coarse graining, this 1×1 matrix becomes $b \times b$ matrix, and a controlled-NOT provides any row operation, and redefinition of stabilizer generators provides any column operation. Hence, this $b \times b$ matrix can be brought into the Smith normal form, and we would hopefully simplify f into a smaller degree polynomial in x'. The following tells us how to choose b. It is convenient to introduce the **annihilator** of a module M over a ring S:

$$\operatorname{ann}_S M = \{r \in S : rm = 0 \ \forall m \in M\} \tag{68}$$

An annihilator is an ideal of S.

Proposition 5.6. *Let $f(x) \in \mathbb{F}[x]$ be a polynomial with $f(0) \neq 0$ over a field $\mathbb{F}$. Suppose $f(x)$ divides $x^n - 1$. Then, the annihilator of the module $M = \mathbb{F}[x]/(f(x))$ over $R' = \mathbb{F}[x^n]$ is precisely the ideal $(x^n - 1) \subseteq R'$.*

Proof. Since R' is a subring of $R = \mathbb{F}[x]$, we immediately have $\operatorname{ann}_{R'} M = R' \cap \operatorname{ann}_R M = R' \cap (f(x))$. The latter includes $x^n - 1$ by the supposition. Hence, $(x^n - 1) \subseteq \operatorname{ann}_{R'} M \subsetneq R'$, but $(x^n - 1)$ is maximal in R' since $R'/(x^n - 1) \cong \mathbb{F}$ is a field. ☑

Let us see how this implies the simplification of f into a smaller degree polynomial in $x' = x^n$. By coarse-graining, we obtain the matrix representation M of the map $f(x) : R'^n \to R'^n$. The R'-module $R/(f(x))$ is equal to R'-module $\operatorname{coker} M$, and hence $\operatorname{ann}_{R'} \operatorname{coker} M = \operatorname{ann}_{R'} R/(f(x)) = (x^n - 1) = (x' - 1)$. By inspection of the Smith normal form, every elementary divisors must divide the annihilator $x' - 1$. There are only two possible ways to divide $x' - 1$: either by 1 or by $x' - 1$. This implies that the elementary divisors are either 1 or $x' - 1$. Since the module M has $\mathbb{F}$-vector space dimension $\deg f$, there are precisely $\deg f$ elementary divisors that are equal to $x' - 1$ and $n - \deg f$ elementary divisors that are equal to 1, up to scalars in $\mathbb{F}$.

We work out an example explicitly. Let $f(x) = 1 + x + x^2$, and then $x^3 - 1 = (x - 1)f(x)$, so let $R' = \mathbb{F}[x^3]$. As a R'-linear map $R'^3 \to R'^3$, $f(x)$ becomes

$$M = \begin{pmatrix} 1 & x' & x' \\ 1 & 1 & x' \\ 1 & 1 & 1 \end{pmatrix}. \tag{69}$$

Applying row and column operations, we have

$$M \mapsto \begin{pmatrix} 1 & x' & x' \\ 0 & 1-x' & 0 \\ 0 & 1-x' & 1-x' \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1-x' & 0 \\ 0 & 1-x' & 1-x' \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1-x' & 0 \\ 0 & 0 & 1-x' \end{pmatrix}. \tag{70}$$

In fact, the supposition in 5.6 is always satisfied whenever the field $\mathbb{F}$ is finite, which is the case for the additive codes on lattices.

Proposition 5.7. *For any polynomial $f(x)$ with coefficients in a finite field $\mathbb{F}$ such that $f(0) \neq 0$, there exists a positive integer n such that $f(x)$ divides $x^n - 1$.*

Proof. We have to use some facts about finite fields; namely, any finite field of characteristic[16] p consists of solutions of $x^{p^m} - x = 0$.[17] This means that the roots of the polynomial $f(x)$ are roots of $x^{p^m} - x$ for some m. Since 0 is not a root of $f(x)$, we see that the roots of $f(x)$ are among those of $x^{n'} - 1$ for some n'. If there is any multiplicity of the roots of $f(x)$, take the smallest m' such that $p^{m'}$ is at least the largest multiplicity. Then, $(x^{n'} - 1)^{p^{m'}} = x^{n'p^{m'}} - 1$ contains all factors of $f(x)$ and hence is a multiple of $f(x)$. ☑

Combining 5.4, 5.6, 5.7, we arrive at the classification:

Theorem 5.8. *Any one-dimensional translation-invariant additive code can be converted into several copies of Ising models and some trivial codes, by Clifford operators that obey coarse translation invariance.*

Note the consistency with our result 3.2. The Ising model has code distance 1 independent of system size. In 3.2, we did not assume the translation-invariance, and concluded that the code distance is bounded. Here we assumed the translation-invariance and obtained a complete classification under Clifford operations. The Clifford operations in 5.8 preserve locality, so any logical operator in the one-dimensional translation-invariant code is a conjugation of a logical operator of the Ising model, which acts on a geometrically local set of qubits.

[16] The minimal positive integer p such that $px = 0$ for any element x of the field. It is necessarily a prime number.

[17] The proof using the result on finitely generated abelian groups can be found in 5.9 below.

***Remark* 5.9.** The set $\mathbb{F}^\times$ of all nonzero elements of a finite field is a (multiplicative) group of finite order. In particular, it is finitely generated and finite. Therefore, it is isomorphic to direct sum of finite cyclic groups. Choose the largest period N, which is a factor of $|\mathbb{F}^\times|$. Then, every nonzero field element satisfies $x^N = 1$. The polynomial $x^N - 1$ has at most N distinct roots, but we know it has $|\mathbb{F}^\times|$ distinct roots. Hence, $|\mathbb{F}^\times| = N$ and the $\mathbb{F}^\times$ is a cyclic group of order N. Finally, $\mathbb{F}$ is a vector space over $\mathbb{F}_p$ of some finite dimension m, implying $N = p^m - 1$. Therefore, $x^{p^m} - x = 0$ holds for any element of the field $\mathbb{F}$. ◇

5.3. Translation-invariant two-dimensional CSS codes

In one-dimensional classification, it was crucially used in the initial stage that the base ring $\mathbb{F}_2[x, x^{-1}]$ is a Euclidean domain. In two-dimensions, the base ring is $R = \mathbb{F}_2[x^\pm, y^\pm]$, and the problem becomes more complicated.

First, we need to distinguish infinite lattice $\mathbb{Z}^2$ versus finite lattice $\mathbb{Z}^2/\Lambda$ obtained by periodic boundary conditions (factoring out a subgroup $\Lambda \le \mathbb{Z}^2$ of finite index). This distinction was unnecessary in the one-dimensional case because the ring $\mathbb{F}_2[x^\pm]$ was so simple that we didn't have to talk about logical operators. Roughly speaking, we saw that a one-dimensional translation-invariant code is not going to be useful for error correcting purposes, which makes the discussion of logical operators unimportant. On the contrary, two-dimensional codes may have error correcting capability with the code distance comparable with the system size, and it is crucial to understand logical operators. Unfortunately, the logical operators in the infinite system are out of our scope since we have only studied finite dimensional symplectic spaces.

The complication is already lurking in the previous one-dimensional case. Consider the Ising model (see 5.5) with excitation map $\epsilon = (0, 1 + x)$. Applying 4.6, we would say that the logical operators correspond to $\ker \epsilon$. Since in $\mathbb{F}_2[x^\pm]$ no two nonzero elements multiply to become zero, the kernel has the zero second component. However, if we had considered $\ker \epsilon$ over the factor ring $\mathbb{F}_2[x^\pm]/(x^L - 1) = \mathbb{F}_2[x]/(x^L - 1)$, then the kernel would have nonzero second component since $(1 + x)\sum_{i=1}^L x^i = 0$. Remark here that the ideal $(x^L - 1)$ imposes the *periodic boundary condition* that translation by L units is equivalent to no translation. Thinking of "infinite L" to recover the infinite lattice, we would say that an infinite series $\sum_{i\in\mathbb{Z}} x^i (0, 1)^T$ lies in $\ker \epsilon$, but the infinite series is *not* a member of the Pauli module $(\mathbb{F}_2[x^\pm])^2$. It is not too difficult to extend the module to include the infinite series, but we are not going to do it.

The most important distinction for the two- or higher dimensional cases from the one-dimensional ones is that there exists a stabilizer map σ such that

$$\ker \epsilon = \operatorname{im} \sigma \quad \text{over } R = \mathbb{F}_2[x^\pm, y^\pm], \tag{71}$$

$$\ker \epsilon \supsetneq \operatorname{im} \sigma \quad \text{over } \mathbb{F}_2[x, y]/(x^L - 1, y^L - 1) \tag{72}$$

for some L, where $\epsilon = \sigma^\dagger \lambda_q$ is the excitation map. In 4.6 one has shown that $\ker \sigma^\dagger / \operatorname{im} \sigma$ is the set of all independent logical operators. (72) means that there are nontrivial logical operators, and they cannot be expressed by a Laurent polynomial vector in the infinite system. The logical operators have to be *global*.

If (71) does not hold, then there exists a nontrivial logical operator supported on a finite region of the lattice.[18] The size of the finite region is independent of the system size, and an error on that finite region will not be corrected. This is a situation we want to avoid. Thus, we assume (71) from now on.

Here is an example of (72), which we refer to as **toric code** [25].

$$\sigma = \begin{pmatrix} x-1 & 0 \\ y-1 & 0 \\ 0 & \bar{y}-1 \\ 0 & -\bar{x}+1 \end{pmatrix}. \tag{73}$$

Excercise 5.10. Verify that $\sigma^\dagger \lambda_2 \sigma = 0$, and that $\ker \sigma^\dagger \lambda_2 = \operatorname{im} \sigma$. ◇

One could directly compute $\ker \sigma^\dagger \lambda_2 / \operatorname{im} \sigma$ over $R/(x^L - 1, y^L - 1)$, but we are going to make connection from this quotient module to the cellular homology in the next subsection. In the rest of this section we show that this is essentially the only example of translation-invariant additive (CSS) codes in two dimensions.

5.3.1. Canonical form of stabilizer maps

The following fact is our starting point of the further discussion. Unfortunately the proof is beyond the scope of this lecture note. The proof can be found in Ref. [19].

Proposition 5.11. *For any two-dimensional translation-invariant additive code, if the stabilizer map σ satisfies $\ker \sigma^\dagger \lambda = \operatorname{im} \sigma$ over $R = \mathbb{F}_2[x^\pm, y^\pm]$, then there is a choice of another stabilizer map σ' such that*

$$\operatorname{im} \sigma = \operatorname{im} \sigma' = \ker \sigma'^\dagger \lambda_q, \quad \ker \sigma' = 0, \tag{74}$$

Moreover, any such σ' has size $2t \times t$ for some t, and there exists a positive integer b such that

$$\operatorname{ann}_{R'} \operatorname{coker} \sigma'^\dagger = (x^b - 1, y^b - 1) \tag{75}$$

where $R' = \mathbb{F}_2[x^{\pm b}, y^{\pm b}]$ is the coarser translation group algebra.

[18] We had better be more cautious here. We only have defined logical operators in the finite systems, and here we are saying that $\ker \epsilon / \operatorname{im} \sigma \neq 0$ over R implies that there is a nontrivial logical operator for finite systems. We elaborate on this in 5.15 below.

Due to this, we may assume that our stabilizer map is given such that $b = 1$ and $\sigma' = \sigma$. The last condition (75) imposes stringent restrictions on σ. To see this, let us recall the definition of the annihilator. The cokernel is $R^t / \operatorname{im} \sigma^\dagger$. If $e_1, \dots, e_t$ are the unit basis vectors of R^t, the annihilator condition says that $(x-1)e_i$ and $(y-1)e_i$ must be in the image of $\sigma^\dagger$ whenever e_i is nonzero modulo $\operatorname{im} \sigma^\dagger$. In other words, a linear combination of the columns of $\sigma^\dagger$ must yield $(x-1)e_i$, and another yield $(y-1)e_i$ if it cannot generate e_i. In particular, the row i of $\sigma^\dagger$ must generate either the maximal ideal $(x-1, y-1)$ or the unit ideal (the ring R itself).

The notion of torsion submodules is useful to characterize the $\operatorname{coker} \sigma^\dagger$. A **torsion submodule** $\mathcal{T}(M)$ of a module M over R is defined[19] as

$$\mathcal{T}(M) = \{\ m \in M \mid \exists r \in R \setminus \{0\} \text{ such that } rm = 0\ \}. \tag{76}$$

The condition (75) says any nonzero element of $\operatorname{coker} \sigma'^\dagger$ is a torsion element; $\mathcal{T} \operatorname{coker} \sigma'^\dagger = \operatorname{coker} \sigma'^\dagger$ is a torsion module. Depending on the choice of σ, $\operatorname{coker} \sigma^\dagger$ may not be a torsion module. However,

Proposition 5.12. *If $\operatorname{im} \sigma = \operatorname{im} \tau$, then $\mathcal{T} \operatorname{coker} \sigma^\dagger$ and $\mathcal{T} \operatorname{coker} \tau^\dagger$ are isomorphic as R-modules.*

Proof. Regard τ and σ as matrices. We may combine two matrices as $\mu = (\sigma\ \tau)$. By assumption, $\operatorname{im} \mu = \operatorname{im} \sigma = \operatorname{im} \tau$. Since every column of τ is in the span of σ, we can find a column operation matrix C such that $\mu C = (\sigma\ 0)$. Similarly, there is a column operation matrix C' such that $\mu C' = (0\ \tau)$. Now, $\operatorname{coker}(\mu C)^\dagger = \operatorname{coker}(C^\dagger \mu^\dagger)$, where the invertible $C^\dagger$ induces an isomorphism between $\operatorname{coker} \mu^\dagger$ and $\operatorname{coker}(C^\dagger \mu^\dagger)$. It follows that $\operatorname{coker}(\sigma\ 0)^\dagger \cong \operatorname{coker}(0\ \tau)^\dagger$, and $\mathcal{T} \operatorname{coker}(\sigma\ 0)^\dagger \cong \mathcal{T} \operatorname{coker}(0\ \tau)^\dagger$. On the other hand, a torsion submodule is oblivious to a free summand: For any module M, we see $\mathcal{T}(M \oplus R) = \mathcal{T}(M)$ since R is a domain. To finish the proof, observe that $\operatorname{coker}(\sigma\ 0)^\dagger = (\operatorname{coker} \sigma^\dagger) \oplus R^m$ where m is the number of columns of τ, so $\mathcal{T} \operatorname{coker}(\sigma\ 0)^\dagger = \mathcal{T} \operatorname{coker} \sigma^\dagger$, and likewise $\mathcal{T} \operatorname{coker}(0\ \tau)^\dagger = \mathcal{T} \operatorname{coker} \tau^\dagger$. We conclude that $\mathcal{T} \operatorname{coker} \tau^\dagger \cong \mathcal{T} \operatorname{coker} \sigma^\dagger$. ☑

5.3.2. Structure theorem

The strong constraint (75) leads to a structure theorem, at least for CSS codes. Recall that a **CSS code** [10, 38] is a code where stabilizer generators can be chosen to be either X- or Z-type. Hence, a CSS code has a block diagonal

[19] This definition assumes R has no two nonzero elements that multiply to become zero. That is, we are using the fact that R is a **domain**. Note that over a field a torsion submodule is always zero.

stabilizer map σ and excitation map ϵ.

$$\epsilon = \sigma^\dagger \lambda = \begin{pmatrix} 0 & \sigma_X^\dagger \\ -\sigma_Z^\dagger & 0 \end{pmatrix}, \quad \sigma = \begin{pmatrix} \sigma_X & 0 \\ 0 & \sigma_Z \end{pmatrix}. \tag{77}$$

Theorem 5.13. *For any two-dimensional translation-invariant CSS code, if the stabilizer map σ satisfies $\ker \sigma^\dagger \lambda = \operatorname{im} \sigma$ over $R = \mathbb{F}_2[x^\pm, y^\pm]$, then the code becomes a tensor product of finitely many copies of the toric code and a product state by (a finite number of layers of) Clifford operations. The number of copies of the toric code in the CSS code is equal to $\frac{1}{2} \dim_{\mathbb{F}_2} \mathcal{T} \operatorname{coker} \sigma^\dagger$.*

Note that by 5.12, the number $\dim_{\mathbb{F}_2} \mathcal{T} \operatorname{coker} \sigma^\dagger$ depends only on $\operatorname{im} \sigma$. A similar result is in Ref. [5]. See 5.14 below for comparison.

***Proof*.** 5.11 provides us with a stabilizer map that satisfies (74). Thus, we may assume that our stabilizer map σ satisfies (74) and (75) with $b = 1$.

We wish to convert a row of the excitation map $\epsilon = \sigma^\dagger \lambda_q$ that generates the maximal ideal

$$\mathfrak{m} = (x-1, y-1) \tag{78}$$

into one that has only two components as in (73). We know that there exists a vector p of Laurent polynomials such that ϵp is a vector with sole nonzero entry $x-1$, $y-1$, or 1. We wish to turn p into a unit vector. The transformation has to be induced from Clifford operation, and in particular should preserve the symplectic form. We are not going to follow this line as we do not understand the symplectic group over R well enough.[20] But, the complication can be reduced by going to a coarser lattice.

Since we can always multiply monomials on the rows of ϵ, we may assume every entry has positive exponent. If n is the maximum exponent of the terms in ϵ, take $R' = \mathbb{F}_2[x^{\pm n}, y^{\pm n}]$ as our new base ring (Coarse-graing I). Then every entry of the new excitation map ϵ' has exponent at most 1. That is, every entry is a $\mathbb{F}_2$-linear combination of $1, x', y', x'y'$. Let us say that such ϵ is *of degree one*. If a row i of ϵ generates $\mathfrak{m}$, then its entries are members of $\mathfrak{m}$, and hence they are $\mathbb{F}_2$-linear combinations of $x-1, y-1, xy-1$. Using symplectic transformations that does not involve any variables x, y, we can eliminate all but at most three entries in the row i. The number of survived entries must be either three or two, because they should generate two elements $x'-1$ and $y'-1$ over R. If there are three entries survived, they can be rearranged to be

[20] Even the following problem, which is presumably simpler, is fairly complicated. Let v be a vector over $S = \mathbb{F}[x_1, \ldots, x_n]$, a polynomial ring with the coefficients in a field. Suppose the entries of v generate the unit ideal. Is there an invertible matrix M with entries in S where v is a column of M? The answer is affirmative, known as Quillen-Suslin theorem. See XXI.3.5 of [29].

$x' - 1$, $y' - 1$, and $x'y' - 1$.[21] Now, we can *use the CSS assumption* to erase the entry $xy - 1$ by some controlled-NOT operation. (Note that a symplectic transformation on ϵ acts on its columns.) This controlled-NOT operation may destroy the fact that ϵ' has been of degree one, in which case we take an even coarser base ring to reduce the maximum exponent back to 1 Coarse-graining II).

Therefore, we may now assume without loss of generality that if $\mathcal{T} \operatorname{coker} \epsilon \neq 0$, there exists a row of ϵ such that it consists of two entries of degree 1. There are three possibilities up to $\mathbb{F}_2$-symplectic transformations. The two entries can be (i) $x - 1, y - 1$, (ii) $x - 1, xy - 1$, or (iii) $y - 1, xy - 1$. In either of (ii) and (iii), we can redefine the translation variable so that $xy \mapsto y$ or $xy \mapsto x$. They corresponds to automorphisms of R. To summarize,

(•) If $\mathcal{T} \operatorname{coker} \epsilon \neq 0$, then by symplectic transformations and a redefinition of the base ring, we can find an excitation map ϵ with a row i that has only two nonzero entries $x - 1$ and $y - 1$.[22]

Let us say that $x - 1$ is at column a and $y - 1$ at b. Assume that $a, b \leq q$; otherwise, interchange left-half and right-half blocks of ϵ:

$$\epsilon = \left(\begin{array}{ccc|ccc} \star & \star & \star & & & \\ 0 & x-1 & y-1 & & & \\ \hline & & & \star & \star & \star \end{array}\right). \tag{79}$$

Now we examine what can be done to ϵ towards the form of (73). It is going to be elementary and careful examination of the conditions (74) and (75). We state intermediate goals and then give the proofs for them. Let's roll our sleeves up!

(I) After some $\mathbb{F}_2$-symplectic transformation, we can obtain ϵ' such that all the entries on columns a and b of ϵ' belong to $\mathfrak{m}$:

$$\epsilon' = \left(\begin{array}{ccc|ccc} \star & \circ & \circ & & & \\ 0 & x-1 & y-1 & & & \\ \hline & & & \star & \star & \star \end{array}\right) \quad \text{where any } \circ \in \mathfrak{m}. \tag{80}$$

By the condition (75) there exists a vector p_x and p_y such that $\epsilon p_x = (x-1)e_i$ and $\epsilon p_y = (y-1)e_i$, where e_i is a unit vector with i-th component 1. Setting $x = y = 1$, p_x, p_y become a vector over $\mathbb{F}_2$, and their symplectic product is zero due to the CSS condition. Then, by 2.8 there exists a symplectic transformation

[21] If the base field is $\mathbb{F}_p$ for some prime p, then one should multiply some scalar of $\mathbb{F}_p$ to get the coefficient 1.

[22] This does not involve any additional qubits; however, a similar statement can be shown much more simply if one allows to insert additional qubits in the trivial product state.

S (controlled-NOT's) that do not involve variables x, y such that $Sp_x(x = y = 1) = e_a$ and $Sp_y(x = y = 1) = e_b$ while the row i of ϵS^{-1} is still the same as the row i of ϵ. That is, Sp_x has a-th component outside $\mathfrak{m}$, but everything else in $\mathfrak{m}$. Similarly, Sp_y has b-th component outside $\mathfrak{m}$, but everything else in $\mathfrak{m}$. Now, the new excitation map $\epsilon' = \epsilon S^{-1}$ has the claimed property: The equation $\epsilon'(Sp_x) = (x-1)e_i$ demands that

$$\epsilon'_{k,a}(Sp_x)_a + (\text{terms in } \mathfrak{m}) = 0 \text{ for } k \neq i, \tag{81}$$

implying $\epsilon'_{k,a} \in \mathfrak{m}$ for all $k \neq i$. Similarly, $\epsilon'(Sp_y) = (y-1)e_i$ demands that $\epsilon'_{k,b}(Sp_y) + (\text{terms in } \mathfrak{m}) = 0$ whenever $k \neq i$, implying $\epsilon'_{k,b} \in \mathfrak{m}$ for all $k \neq i$.

(II) Both the columns $a + q$ and $b + q$ of ϵ' are nonzero.

If either of them was zero, then the condition $\ker \epsilon = \operatorname{im} \lambda_q \epsilon^\dagger$ would imply that there must be a row with sole nonzero element 1 at column a or b. That is not possible as we have shown in (I) that ϵ' has entries in $\mathfrak{m}$ for the columns a and b.

(III) For any row j each tuple of entries $(\epsilon'_{j,a+q}, \epsilon'_{j,b+q})$ is a $\mathbb{F}_2$-multiple of the tuple $(x - xy, xy - y)$.

Since the row i of ($\bullet$) has only two nonzero components, the condition $\epsilon' \lambda_q \epsilon'^\dagger = 0$ implies that $\epsilon'_{j,a+q}(\bar{x}-1) + \epsilon'_{j,b+q}(\bar{y}-1) = 0$ for all j. The solution of degree-one is a $\mathbb{F}_2$-multiple of $x - xy,\ xy - y$.

(IV) By Gauss elimination that does not involve any variables x and y, the excitation map ϵ' can be turned into ϵ'' such that each of the left and right blocks of $\epsilon''(x = y = 1)$ is in the reduced row echelon form.

This is obvious.

(V) ϵ'' has a row j such that $\epsilon''_{j,a+q}, \epsilon''_{j,b+q} \neq 0$ but $\epsilon''_{j,k}(x = y = 1) = 0$ for all k:

$$\epsilon'' = \left(\begin{array}{ccc|ccc} \star & \circ & \circ & & & \\ 0 & x-1 & y-1 & & & \\ \hline & & & \star & \circ & \circ \\ & & & \circ & x-xy & xy-y \end{array}\right). \tag{82}$$

Let J_1 be the collection of all row indices j such that the row j is nonzero upon setting $x = y = 1$. The reduced row echelon form of ϵ'' tells us that $j \in J_1$ if and only if $e_j \in \operatorname{im} \epsilon''$, which can be verified by setting $x = y = 1$. For $j \in J_1$, choose vectors $f^{(j)}$ such that $e_j = \epsilon'' f^{(j)}$. ($f^{(j)}$ are determined up to $\operatorname{im} \lambda_q \epsilon''^\dagger$.) On the contrary to the claim (V), suppose that $\epsilon''_{j,a+q}, \epsilon''_{j,b+q} \neq 0$ happens only for $j \in J_1$. Then, for the unit vector e_{a+q} we have $\epsilon'' e_{a+q} = \sum_{j \in J_1} c_j e_j$ for some $c_j \in R$. This means that $e_{a+q} - \sum_{j \in J_1} c_j f^{(j)} \in \ker \epsilon''$. On the other hand, setting $x = y = 1$, we see that $\epsilon''(x = y = 1) e_{a+q} = 0$ by (III). This implies that $c_j(x = y = 1) = 0$ for all $j \in J_1$. Since $\ker \epsilon'' = \operatorname{im} \lambda_q \epsilon''^\dagger$, it follows that $\lambda_q e_{a+q} - \sum_{j \in J_1} c_j \lambda_q f^{(j)} \in \operatorname{im} \epsilon''^\dagger$, and therefore $\lambda_q e_{a+q} = e_a \in \operatorname{im} \epsilon''^\dagger(x = y = 1)$. But, the column a of $\epsilon''(x = y = 1)$ is zero by (I). This is a contradiction. Since (II) implies that the column $a + q$ and $b + q$ are nonzero, (V) is proved.

(VI) By symplectic transformations and $\mathbb{F}_2$-row operations on ϵ'', we can obtain ϵ''' such that $\epsilon'''_{j,a+q}$ and $\epsilon'''_{j,b+q}$ are the only nonzero entries in the row j ($\neq i$) and columns $a + q$, $b + q$:

$$\epsilon''' = \left(\begin{array}{ccc|ccc} \star & \star & \star & & & \\ 0 & x-1 & y-1 & & & \\ \hline & & & \star & 0 & 0 \\ & & & 0 & x-xy & xy-y \end{array}\right). \tag{83}$$

By (III), row operations on ϵ'' can eliminate all the entries but j-th row in the columns $a + q$ and $b + q$. The row i is left intact as it has zero entries in the columns $a + q$ and $b + q$ anyway. After this, since $x - xy, xy - y$ generate the ideal $\mathfrak{m}$ and all the entries of row j are in $\mathfrak{m}$ by (V), appropriate controlled-NOTs clean the row j by adding some multiple of $\epsilon''_{j,a+q}$ and $\epsilon''_{j,b+q}$ to other places in the row j in the right-half block of ϵ''. These controlled-NOTs do not change the row i because on the left-half block of ϵ'' the controlled-NOT adds multiples of 0 to $\epsilon''_{i,a}$ and $\epsilon''_{i,b}$.

(VII) For any row r the tuple $(\epsilon'''_{r,a}, \epsilon'''_{r,b})$ is a R-multiple of the tuple $(x-1, y-1)$. Therefore, some row operation on ϵ''' clears up the column a, b, leaving $\epsilon'''_{i,a} = x - 1$ and $\epsilon'''_{i,b} = y - 1$ intact:

$$\epsilon'''' = \left(\begin{array}{ccc|ccc} \star & 0 & 0 & & & \\ 0 & x-1 & y-1 & & & \\ \hline & & & \star & 0 & 0 \\ & & & 0 & x-xy & xy-y \end{array}\right). \tag{84}$$

This follows from the proof of (III) using the cleaned row j of ϵ'''.

Now the excitation map ϵ'''' obtained from (VII) has a distinguished submatrix consisting of rows i and j, columns $a, b, a + q, b + q$, which is the same the excitation map of the toric code. We have shown that $\mathcal{T} \operatorname{coker} \epsilon \neq 0$ implies $k = \dim_{\mathbb{F}_2} \mathcal{T} \operatorname{coker} \epsilon \geq 2$. By induction on k, we see that k has to be even, and that there is a Clifford operation obeying a coarse translation-invariance which factors out $k/2$ copies of the toric code. The proof so far is algorithmic.

It remains to turn the code state into a product state when $k = 0$. This part appears to be no easier than the Quillen-Suslin theorem on the unimodular completion problem. Since the excitation map of a CSS code is block diagonal, and any elementary column operation on the left-block can be compensated by another column operation in the right-block to become a symplectic transformation, we only need to show that the left-half block of ϵ can be transformed into the identity matrix, representing the trivial state. The condition (75) says that each row generates the unit ideal (each row is unimodular), and therefore our problem is to convert each row that generates the unit ideal into the unit vector using elementary column operations (unimodular completion) over the Laurent polynomial ring. This is known to be possible with explicit algorithms. Park [33] gives an algorithm that reduces the problem into that over a polynomial ring, so that the algorithm of Park and Woodburn [32] can then be applied. (Amidou and Yengui [1] find a more efficient algorithm when the coefficient field is infinite, which is not applicable for our purpose.) This completes our proof of the classification theorem. ☑

5.3.3. Remarks

***Remark* 5.14.** Bombín [5] reports a classification theorem, which is similar to 5.13. Here we sketch his approach and clarify the difference from 5.13. In Ref. [5], the content of 5.11 is stated, albeit in different terminology. Namely, it is argued that there exists a translation-invariant set of local generators of the stabilizer group such that there is no "local relation" among the generators. This amounts to $\ker \sigma = 0$ in our language. In addition, there are finitely many topological charges which form an abelian group, and one can coarse-grain the lattice such that each new unit cell can support any topological charge. The former amounts to $\operatorname{coker} \epsilon$ being a finite additive group, and the latter amounts to (75).

Given this intermediate conclusion, Bombín studies the topological spins and mutual statistics using the commutation relations among string operators associated with the topological charges, using an idea by Levin and Wen [30]. His finding can be summarized as follows. The mutual statistics can be captured by a bilinear symmetric form κ with the diagonal elements being zero. Over the binary field $\mathbb{F}_2$, the symmetric form κ happens to be symplectic, and therefore there exists a canonical basis as in 2.6 for the set of topological charges. Under this canonical basis, there are two complementary subsets C, D of topological charges on which κ vanishes. The topological spin $e^{i\pi\theta(a)}$ obeys $\theta(a - b) = \kappa(a, b) + \theta(a) + \theta(b)$ for any charges a and b [4, Eq. (3.1.2)], [26, Eq. (223)]. On a subspace C or D where κ vanishes, the angle θ becomes a linear functional θ_C or θ_D. (We used the property that $b = -b$.) Then, there is a basis change on the space of topological charges to simplify the linear functionals θ_C and

θ_D. If these linear functionals cannot be transformed to zero on C and D, then Bombín calls the code "chiral."

The main conclusion is that any non-chiral translation-invariant code is equivalent to a finitely many copies of the toric code. Here, the equivalence is defined through locality-preserving unitary transformations that map any Pauli operator to a Pauli operator. There is a difference between this equivalence and the equivalence under controlled-NOT, Hadamard, and phase gates. The latter equivalence (under local quantum circuits) trivially implies the former notion, while the former does not imply the latter unless one includes auxiliary qubits.

Ref. [2] shows that for any locality-preserving unitary U there exists another locality-preserving unitary V such that $U \otimes V$ is a local quantum circuit. The proof is very simple: Let $\mathcal{S}$ denote the swap operation between two identical many-qubit systems. This is a local quantum circuit as the swap can be implemented by swapping every pair of qubits, i.e., $\mathcal{S} = \bigotimes_i \mathcal{S}_i$. Then, $(U^\dagger \otimes I)\mathcal{S}_i(U \otimes I)$ is a local unitary, and therefore $(U^\dagger \otimes I)\mathcal{S}(U \otimes I)$ is a local quantum circuit, and $U \otimes U^\dagger = \mathcal{S}(U^\dagger \otimes I)\mathcal{S}(U \otimes I)$ is a local quantum circuit, too. However, this result together with Bombín's does not imply that one can turn a 2D translation-invariant code state with no topological charge into a product state by a local quantum circuit, since the auxiliary qubits may be transformed into an entangled state. Our treatment under CSS assumption does not need the auxiliary qubits.

Note that 5.13 remains true for qudit stabilizer codes with prime dimensions where X-type stabilizer group generators are separated from Z-type ones (CSS). On the other hand, Bombín's result does not assume the CSS condition, but appears to rely on the binary field $\mathbb{F}_2$ over which a skew-symmetric matrix is symmetric. This make it obscure to apply his approach to qudit prime-dimensional codes. ◇

***Remark* 5.15.** Here we elaborate on the relation between $\ker \epsilon / \operatorname{im} \sigma$ being nonzero and logical operators on finite systems. Suppose $v \in \ker \epsilon$ is a vector such that $v \notin \operatorname{im} \sigma$, so when passed to $\ker \epsilon / \operatorname{im} \sigma$ it represents a nonzero element. The equation $\epsilon v = 0$ is still valid even if we mod out $\mathfrak{b}_L = (x^L - 1, y^L - 1)$, meaning that v is a logical operator. Here, we need to prove that v is nontrivial for some L. That is, $v \notin \operatorname{im} \sigma + \mathfrak{b}_L R^{2q}$ for some L if $v \notin \operatorname{im} \sigma$. Phrased differently, we have to show that $(R/\mathfrak{b}_L) \otimes_R K \neq 0$ where $K := (Rv + \operatorname{im} \sigma)/\operatorname{im} \sigma \neq 0$.

Here is a proof for readers who are familiar with localization. (See Atiyah-MacDonald [3] for concise treatment.) Let us lift the coefficient field to the algebraic closure $\mathbb{F}^a$ of $\mathbb{F}_2$. ($\mathbb{F}^a$ is a flat module over $\mathbb{F}_2$.) $K \neq 0$ if and only if $K_\mathfrak{m} \neq 0$ for some maximal ideal $\mathfrak{m}$. By Nullstellensatz, $\mathfrak{m} = (x - a, y - b)$ for some $a, b \in \mathbb{F}^a$, and hence there exists a positive odd integer L such that $a^L = 1 = b^L$. (There are infinitely many such L.) Since K is finitely generated R-module, it admits a finite presentation i.e., $K \cong \operatorname{coker} \phi$ for some matrix ϕ over R. Upon localization at $\mathfrak{m}$, we can think of a minimal ϕ where every entry is a member of $\mathfrak{m}$. Then, this minimal ϕ becomes a zero matrix upon factoring $(\mathfrak{b}_L)_\mathfrak{m}$ because $(\mathfrak{b}_L)_\mathfrak{m} = \mathfrak{m}_\mathfrak{m}$. Since the minimal ϕ had nonzero number of rows, it follows that $(R/\mathfrak{b}_L \otimes K)_\mathfrak{m}$ is nonzero. Therefore, $(R/\mathfrak{b}_L) \otimes K$ is nonzero for infinitely many L. ◇

5.4. Connection to cellular homology

We have learned that any translation-invariant CSS code (X stabilizers and Z stabilizers are separated) in two-dimensions with large code distance has to be a finitely many copies of the toric code. The stabilizer map of the toric code consists of two blocks which are related by exchanging two rows and $x \leftrightarrow \bar{x}, y \leftrightarrow \bar{y}$. The excitation map inherits the duality as well, and thus we can focus on one block. The equation $\ker \epsilon = \operatorname{im} \sigma$ decomposes into two equivalent equations, one of which reads

$$R^1 \xleftarrow{\partial_1 = \begin{pmatrix} x-1 & y-1 \end{pmatrix}} R^2 \xleftarrow{\partial_2 = \begin{pmatrix} y-1 \\ -x+1 \end{pmatrix}} R^1. \tag{85}$$

It is readily verified that $\partial_1 \partial_2 = 0$ as matrices. We promised earlier that we will calculate $\ker \epsilon / \operatorname{im} \sigma$ subject to the periodic boundary condition that is expressed by an ideal $\mathfrak{b}_L = (x^L - 1, y^L - 1)$. Here we make a connection to the cellular homology of the two-dimensional torus.

Consider a torus obtained by gluing small squares. One can imagine a big square with vertical sides identified with each other, and the horizontal sides identified with each other, and then slice the big square into smaller ones. Suppose the number of small squares are L in each direction so there are L^2 squares in total.

Introduce an abelian group $C_2 = \mathbb{F}_2^n$ where $n = L^2$ is equal to the number of squares. We may visualize an element of C_2 as a configuration of bits written on the centers of the small squares. Similarly, introduce another abelian group $C_1 = \mathbb{F}_2^{2n}$ where $2n$ is the number of edges. Each vector in C_1 is identified with an array of bits written on the edges. Now, define a group homomorphism $\nabla_2 : C_2 \to C_1$ by the rule that the bit 1 on a square is mapped to four bits written on the four edges that surrounds the square. If a collection of squares is given as an input to ∇_2, then the output from ∇_2 is the boundary of the union of the squares. Thus, it is legitimate to call ∇_2 a boundary map.

Let us introduce a coordinate system for the squares and the edges. Pick a square and declare it as the origin. The coordinates for the small squares are defined modulo L. To give the coordinates for the edges, we associate the vertical edge to the square on the right of the edge, and the horizontal edge to the square above the edge. For example, the boundary edges of the square at $(0,0)$ are $(0,0)_v$, $(0,1)_h$, $(1,0)_v$, and $(0,0)_h$, where the subscripts distinguishes the horizontal or vertical edges. Following the trick to write the collection of tuples as a polynomial, we can write the collection of edges as $1_v + x_v$ and $1_h + y_h$. Even more compactly, we have $\begin{pmatrix} 1+y \\ 1+x \end{pmatrix}$. We see that this matrix is equal to ∂_2 over $\mathbb{F}_2$.

Excercise 5.16. Verify that ∂_1 represents the boundary map from the group of edges to the group of vertices. To which square do we have to associate a vertex to have a consistent coordinate system? ◇

The kernel of the boundary map modulo the group of boundaries of one-higher dimensional cells is called a **homology** group. There is one homology group at each "dimension": Formally we set $\partial_3 = 0$ and $\partial_0 = 0$, and define $H_i = \ker \partial_i / \operatorname{im} \partial_{i+1}$ for $i = 0, 1, 2$.

We could define these homology groups without ever involving the Laurent polynomial ring, which was rather a convenient extra feature allowed by the Cartesian array of small squares. Although it is a another problem how one can identify the homology group, there is no problem of defining the homology group given any tessellation of the space into small polygons. To little (or big) surprise, it is very important that the resulting homology group is independent of the tessellation (cell decomposition). More general homology theory [23] is beyond the scope of this note, but one can think of a toy version where one square from our tessellation is divided into two triangles. In this case one can show that the homology groups remain the same. On top of this, one can show that any two tessellations can be refined by a process of such subdivisions to become the same tessellation, which leads the independence of the homology group on tessellations. The simplest tessellation would be to have only a single square to cover the entire space (the two-dimensional torus), with vertices and edges properly identified. This is to set $L = 1$ in our earlier discussion, in which case the homology groups are clearly seen as $H_0 = \mathbb{F}_2$, $H_1 = \mathbb{F}_2^2$, and $H_2 = \mathbb{F}_2$.

Excercise 5.17. What is the code space dimension of the toric code under periodic boundary conditions? What is the code distance? ◇

5.5. Dimension three and beyond

As of this writing, there is no known structure theorem for stabilizer modules in three or higher dimensions. In terms of pure commutative algebra this problem appears not too well guided, since under a fairly mild condition any chain complex obtained by continued calculation of kernels (free resolution) gives rise to stabilizer module with large code distances. General one-dimensional and CSS two-dimensional cases were fortunately simple enough that we could handle directly.

From a physical viewpoint, one can focus on the behavior of excitations. We briefly defined the excitations as a flipped term of a Hamiltonian, but we did not give any further attributes to the excitations. The most important notion to begin with is that of topological excitations — an excitation that cannot be created alone from a ground state by any local operator. The topological excitation is insensitive to local basis change (Clifford transformations), relabeling of terms of the Hamiltonian, or the translation structure. The set of all point-like topological excitations is identified with the torsion submodule of the cokernel of the excitation map [19], and it is no coincidence that we introduced the torsion submodule in the structure theorem for 2D CSS codes. It is a characteristic of a code that is invariant under various basis changes.

There is conceptual progress from our consideration of additive codes and corresponding Hamiltonians, which is surprising from a conventional perspective on topological order. It has been for long taken granted that a point-like topological excitations has well-defined "statistics," a consistent number that the quantum mechanical state vector acquires upon exchanging two such excitations. Following our formalism, it is easy to show that this is not always the case. In three or higher dimensions, there are many systems of which point-like topological excitations cannot have well-defined "statistics" since their motion is ill-defined [11, 6, 18, 39].

To illustrate the point more clearly, consider the toric code excitation map: $\epsilon = (x-1,\ y-1)$, the cokernel of which is $\mathbb{F}_2[x^{\pm}, y^{\pm}]/(x-1, y-1) \cong \mathbb{F}_2$. An element of the cokernel is an equivalence class of flipped terms in the Hamiltonian modulo those that can be flipped by local Pauli operators. Thus, the element of $\operatorname{coker}\epsilon$ is precisely what we call a topological excitation. The action from the translation group on $\operatorname{coker}\epsilon \cong \mathbb{F}_2$ is trivial. This means that a topological excitation at one location is connected to that at other location by some local operator, implying that the "motion" is allowed by some other interaction. The algebraic origin to this conclusion is that the excitation map have binomial elements $x-1$ and $y-1$ in the image $\operatorname{im}\epsilon$. In higher dimensions, there is no reason for the excitation map to have a binomial term in the image, in which case the notion of motion for topological excitations becomes ambiguous. An example in three dimensions is given by $\epsilon_{\text{cubic}} = (1+x+y+z,\ 1+xy+yz+zx)$, known as the cubic code [18]. The reader is encouraged to show that ϵ_{cubic} cannot generate a binomial term in the image. In fact, $\operatorname{coker}\epsilon$ does not have any binomial zero-divisor.

The behavior of topological excitations can be even richer. In four or higher dimensions, there is a system in which a single Hamiltonian term cannot be flipped alone [12], but they have to appear as a line or in some other more complicated shape. There is a way of defining line-like topological excitations if the Hamiltonian consists of commuting terms [20], but a concise algebraic characterization for additive code systems is immature.

References

[1] Morou Amidou and Ihsen Yengui, *An algorithm for unimodular completion over laurent polynomial rings*, Linear Algebra and its Applications **429** (2008), no. 7, 1687–1698.

[2] Pablo Arrighi, Vincent Nesme, and Reinhard Werner, *Unitarity plus causality implies localizability*, 2007.

[3] M. F. Atiyah and I. G. MacDonald, *Introduction to commutative algebra*, 1969.

[4] Bojko Bakalov and Alexander Kirillov Jr., *Lectures on tensor categories and modular functors*, University Lecture Series **21** (2001).

[5] Héctor Bombín, *Structure of 2D topological stabilizer codes*, Commun. Math. Phys. **327** (2014), 387–432.

[6] Sergey Bravyi, Bernhard Leemhuis, and Barbara M. Terhal, *Topological order in an exactly solvable 3D spin model*, Annals of Physics **326** (2011), no. 4, 839–866.

[7] Sergey Bravyi and Barbara Terhal, *A no-go theorem for a two-dimensional self-correcting quantum memory based on stabilizer codes*, New J. Phys. **11** (2009), 043029.

[8] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, *Quantum error correction and orthogonal geometry*, Phys. Rev. Lett. **78** (1997), 405–408.

[9] ______, *Quantum error correction via codes over gf(4)*, Information Theory, IEEE Transactions on **44** (1998), no. 4, 1369–1387.

[10] A. R. Calderbank and Peter W. Shor, *Good quantum error-correcting codes exist*, Phys. Rev. A **54** (1996), no. 2, 1098–1105.

[11] Claudio Chamon, *Quantum glassiness*, Phys. Rev. Lett. **94** (2005), 040402.

[12] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill, *Topological quantum memory*, J. Math. Phys. **43** (2002), 4452–4505.

[13] J. M. Farinholt, *An ideal characterization of the clifford operators*, Phys. A: Math. Theor. **47** (2014), 305303.

[14] David Fattal, Toby S. Cubitt, Yoshihisa Yamamoto, Sergey Bravyi, and Isaac L. Chuang, *Entanglement in the stabilizer formalism*, (2004).

[15] Daniel Gottesman, *A class of quantum error-correcting codes saturating the quantum hamming bound*, Phys. Rev. A **54** (1996), 1862.

[16] ______, *Fault-tolerant quantum computation with higher-dimensional systems*, Quantum Computing and Quantum Communications (ColinP. Williams, ed.), Lecture Notes in Computer Science **1509** (1999), 302–313, Springer Berlin Heidelberg.

[17] M. Grassl and M. Roetteler, *Non-catastrophic encoders and encoder inverses for quantum convolutional codes*, vol. 21, Information Theory, 2006 IEEE International Symposium on, 2016, pp. 1109-1113.

[18] Jeongwan Haah, *Local stabilizer codes in three dimensions without string logical operators*, Phys. Rev. A **83** (2011), no. 4, 042330.

[19] ______, *Commuting Pauli Hamiltonians as maps between free modules*, Commun. Math. Phys. **324** (2013), 351–399.

[20] ______, *An invariant of topologically ordered states under local unitary transformations*, Communications in Mathematical Physics **342** (2016), no. 3, 771–801.

[21] Jeongwan Haah and John Preskill, *Logical operator tradeoff for local quantum codes*, Phys. Rev. A **86** (2012), 032308.

[22] Alioscia Hamma, Radu Ionicioiu, and Paolo Zanardi, *Bipartite entanglement and entropic boundary law in lattice spin systems*, Phys. Rev. A **71** (2005), 022315.

[23] Allen Hatcher, *Algebraic topology*, Cambridge University Press, 2002.

[24] Isaac H. Kim, *3d local qupit quantum code without string logical operator*, (2012).

[25] Alexei Kitaev, *Fault-tolerant quantum computation by anyons*, Annals of Physics **303** (2003), 2–30.

[26] ______, *Anyons in an exactly solved model and beyond*, Annals of Physics **321** (2006), 2–111.

[27] A. Klappenecker and M. Roetteler, *Beyond stabilizer codes ii: Clifford codes*, IEEE Transactions on Information Theory **48** (2002), no. 8, 2396–2399.

[28] Emanuel Knill and Raymond Laflamme, *Theory of quantum error-correcting codes*, Phys. Rev. A **55** (1997), 900–911.

[29] Serge Lang, *Algebra*, revised 3rd ed., Springer, 2002.

[30] Michael Levin and Xiao-Gang Wen, *Fermions, strings, and gauge fields in lattice spin models*, Phys. Rev. B **67** (2003), 245316.

[31] Noah Linden, abd Mary Beth Ruskai Frantisek Matus, and Andreas Winter, *The Quantum Entropy Cone of Stabiliser States*, 8th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2013) (Dagstuhl, Germany) (Simone Severini and Fernando Brandao, eds.), Leibniz International Proceedings in Informatics (LIPIcs) **22** (2013), 270–284, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[32] H. Park and C. Woodburn, *An algorithmic proof of Suslin's stability theorem over polynomial rings*, Journal of Algebra **178** (1995), 277–298.

[33] Hyungju Park, *Symbolic computation and signal processing*, Journal of Symbolic Computation **37** (2004), no. 2, 209–226.

[34] E. M. Rains, *Nonbinary quantum codes*, Information Theory, IEEE Transactions on **45** (1999), no. 6, 1827–1832.

[35] N. Read and Subir Sachdev, *Large-n expansion for frustrated quantum antiferromagnets*, Phys. Rev. Lett. **66** (1991), 1773–1776.

[36] Peter W. Shor, *Scheme for reducing decoherence in quantum computer memory*, Phys. Rev. A **52** (1995), R2493–R2496.

[37] ———, *Fault-tolerant quantum computation*, Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on (1996), 56–65.

[38] Andrew Steane, *Multiple particle interference and quantum error correction*, Proc. Roy. Soc. Lond. A **452** (1996), 2551.

[39] Sagar Vijay, Jeongwan Haah, and Liang Fu, *A new kind of topological quantum order: A dimensional hierarchy of quasiparticles built from stationary excitations*, Phys. Rev. B **92** (2015), 235136.

[40] ———, *Fracton topological order, generalized lattice gauge theory and duality*, (2016).

[41] Beni Yoshida, *Exotic topological order in fractal spin liquids*, Phys. Rev. B **88** (2013), 125122.

[42] Beni Yoshida and Isaac L. Chuang, *Framework for classifying logical operators in stabilizer codes*, Phys. Rev. A **81** (2010), 052302.

(Recibido en julio de 2016. Aceptado en noviembre de 2016)

Station Q Quantum Architectures and Computation Group
One Microsoft Way, Redmond, Washington, USA
e-mail: jwhaah@microsoft.com