

An Introduction to Homological Codes with an emphasis on Fourier and Poincaré Dualities

Sanchayan Dutta

2 November, 2022

Table of Contents

- 1 Part I: Homology and Cohomology
- 2 Part II: Homological Codes and the Special Case of Toric Codes
- 3 Part III: A Brief Digression on Fourier Dualities
- 4 Part IV: Revisiting Homological Codes

Part I: Homology and Cohomology

Simplicial Complex

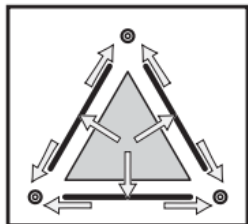
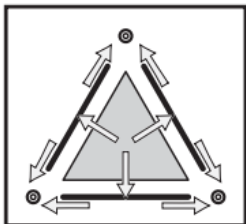


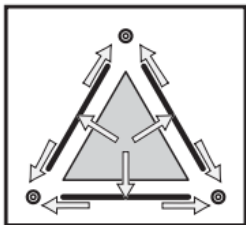
Figure: Robert Ghrist's *Elementary Applied Topology*, Chapter 4.

A simplex is a generalization of the notion of triangles to arbitrary dimensions. In the above image, think of the vertices of the triangles as "0-simplices" or "0-cells". The edges of the triangles as "1-simplices" or "1-cells". And the body of the solid triangle is a "2-simplex" or a "2-cell".



The idea is that we have a vector space called a 0-chain C_0 whose basis elements are the 0-cells. A vector called 1-chain C_1 whose basis elements are the 1-cells. And a vector space called 2-chain C_2 whose basis elements are 2-cells. You can generalize this to higher dimensions.

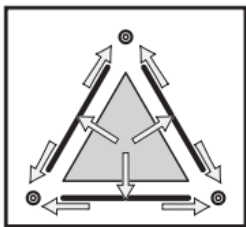
Let's restrict the scalars of our vector spaces to be $\mathbb{F}_2 = \mathbb{Z}/2 = \{0, 1\}$ for now, but we can also take any general field.



With this kind of construction we have a chain complex (C_\bullet, d_\bullet) of vector spaces and linear maps over some field F :

$$\cdots \xrightarrow{d_{k+1}} C_k \xrightarrow{d_k} C_{k-1} \xrightarrow{d_{k-1}} \cdots \xrightarrow{d_2} C_1 \xrightarrow{d_0} 0$$

What do the maps d_i mean? Well, they are called boundary maps. In the previous example, d_0 maps any 1-cell to its two boundary 0-cells, i.e., an edge to the sum of its two vertices. d_1 maps any 2-cell to its three boundary 1-cells, i.e., the solid triangle to the sum of its three edges.



You can check that the boundary of a boundary is zero, meaning that if you take the solid triangle and map it to the sum of its three edges $e_1 + e_2 + e_3$ and then map each edge to the sum of its vertices, then you get $(v_1 + v_2) + (v_2 + v_3) + (v_3 + v_1)$ which is zero since each vertex occurs twice in the sum and we know that in \mathbb{F}_2 that $2 \equiv 0$.

An important definition arises at this point: The k -th homology group of (C_\bullet, d_\bullet) is defined to be the quotient vector space $H_k(C_\bullet, d_\bullet) = \frac{\ker(d_k)}{\text{im}(d_{k+1})}$. The mnemonic here is “homology is cycles modulo boundaries”.

This definition might look scary at first but it's really not. Let's quickly dive into an example.

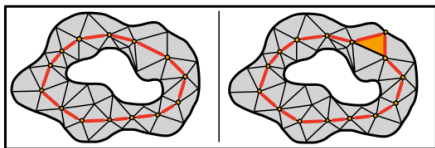
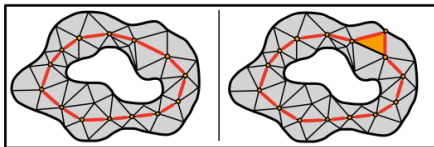


Figure: Images are from Vidit Nanda's lecture note on Homology.

The above picture shows the triangulation of a surface. You should be able to identify the 0-cells, 1-cells and the 2-cells. In the left picture, we can have a sum of edges that form a cycle γ , in the sense that if you add up all their vertices you'll get 0 since each vertex occurs twice.

In the right picture, we also have a sum of edges that form a cycle γ' , but we notice that the two cycles γ and γ' only differ by the boundary of a 2-cell (solid triangle) which has been shaded in orange.



So both the cycles γ and γ' were elements of $\ker(d_1)$ and differ by an element of $\text{im}(d_2)$. That's the point of homology: γ and γ' lie in the same homology class $H_1(C_\bullet, d_\bullet)$, because they are elements of $\ker(d_1)$ that differ merely by an element of $\text{im}(d_2)$. See, that was easier than expected!

Let's move on to the other scary-sounding creature **cohomology**. What's that? The idea is that we can dualize our vector spaces in C_\bullet and also dualize all the linear maps d_\bullet in between them, to get the cochain complex (C^\bullet, d^\bullet) . The arrows will certainly be reversed.

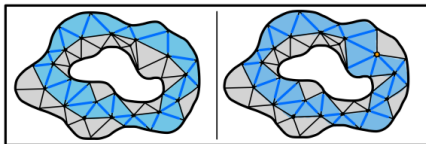
$$0 \rightarrow C^0 \xrightarrow{d^0} C^1 \xrightarrow{d^1} C^2 \xrightarrow{d^2} \dots \xrightarrow{d^{k-1}} C^k \xrightarrow{d^k} C^{k+1} \xrightarrow{d^{k+1}} \dots$$

where each $C^k = C_k^*$ and $d^k = d_{k+1}^*$. d^k is the k th coboundary map which is the (conjugate) transpose of d_{k+1} . You should recall the notions of dual vector spaces and dual linear maps at this point.

The formal definition of the k -th cohomology group (C^\bullet, d^\bullet) is the quotient vector space $H^k(C^\bullet, d^\bullet) = \frac{\ker(d^k)}{\text{im}(d^{k-1})}$. The new mnemonic for this is “cohomology is cocycles module coboundaries”.

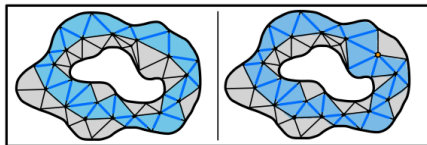
The important point here is that to every basis element of the vector space C_k there is a uniquely associated basis element of the dual vector space C^k that eats it and produces $1 \in \mathbb{F}_2$. We know this from elementary linear algebra!

So you imagine the basis elements of C^k as being “dual” k -cells or in fact, if you’re trying to visualize them then you can think of them as our usual k -cells. Let’s jump to an example again.



We need to be vigilant here. In the left diagram, you can notice a set of edges shaded with dark blue. You can consider these to be your "dual" 1-cells. The linear map d^2 sends each "dual" 1-cell to its neighboring "dual" 2-cells, i.e., the light blue shaded triangles that share the "dual" 1-cell as an edge.

If you add up the neighboring "dual" 2-cells for each of the dark blue edges you'll get a sum total of 0, because each light blue triangle will occur twice in such a sum. Thus, the sum of these dark blue edges γ is a 1-cocycle as its coboundary is 0.



The vertices here are your "dual" 0-cells and the map d^1 sends each vertex to its neighboring edges, i.e., the "dual" 1-cells. In the right diagram, notice the vertex in red, with 6 "dual" 1-cells incident at it. This is the coboundary of that vertex and the sum of those edges is an element of $\text{im}(d^0)$.

Like previously, the sum of the dark blue edges is an element of $\ker(d^1)$, i.e., a 1-cocycle γ' . The 1-cocycles γ and γ' clearly only differ by an element of $\text{im}(d^0)$, which is a 1-coboundary, and hence these 1-cocycles are elements of the same cohomology class $H^1(C^\bullet, d^\bullet)$. That's all there is to this example.

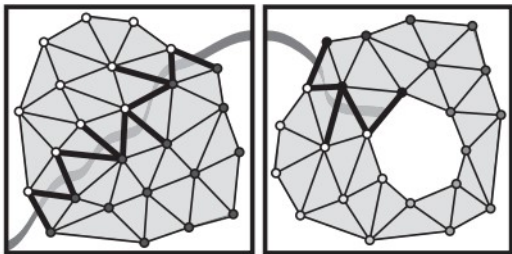
You might wonder why the coboundary map takes an i -cell to the $i + 1$ -cells incident at it. This can be easily shown if we think about i -cochains as functionals that "eat" and assign values to i -chains.

This deserves a good and elaborate explanation, but for now, I'll leave you with this brief excerpt from Robert Ghrist's amazing book, which if you unwrap will give you the full mental picture. In short, recall that given a linear map $\varphi : V \rightarrow W$, the dual map $\varphi^* : W^* \rightarrow V^*$ maps $\gamma \mapsto \varphi \circ \gamma$, which on each $i + 1$ -cell assigns a value based on the formal linear sum of the boundary i -cells (with sign).


Example 6.2 (Simplicial cochains)



Examples in the simplicial category are illustrative. Consider a triangulated disc with a 1-cocycle on edges using \mathbb{F}_2 coefficients.



Any such 1-cocycle is the coboundary of a 0-cochain which labels vertices with 0 and 1 *on the left* and *on the right* of the 1-cocycle, so to speak: this is what a trivial class in $H^1(\mathbb{D}^2)$ looks like. On the other hand, if one considers a surface with some nontrivial H_1 – say, an annulus – then one can construct a similar 1-cocycle that is nonvanishing in H^1 . The astute

reader will notice the implicit relationship between such cocycles and gradients of a local *potential* over the vertices, with cohomology class in H^1 differentiating between those which are or are not globally expressible as a gradient of a potential. 

Part II: Homological Codes and the Special Case of Toric Codes

Homological Codes: The Special Case of Toric Codes

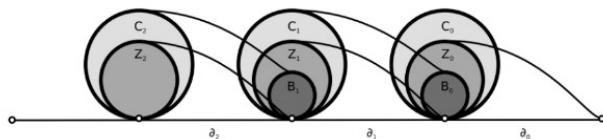
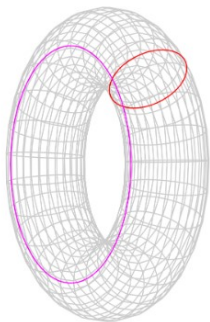


Figure: Image from Nikolas Breuckmann's 2018 Ph.D. thesis.

We recall the definitions of the boundary map $\delta_i : C_i \rightarrow C_{i-1}$ and the coboundary map $\partial_i : C_i \rightarrow C_{i+1}$. Writing δ_i and ∂_i down as matrices in a few small examples should also make it clear that $\partial_i = \delta_{i+1}^T$. The i -cycles are $Z_i = \ker \partial_i$ and the i -boundaries are $B_i = \text{im } \partial_{i+1}$. Since $\partial_i \circ \partial_{i+1} = 0$ it follows that $B_i \subseteq Z_i$, meaning all i -boundaries are necessarily i -cycles.



There may however be i -cycles that are not boundaries of $i+1$ -chains, called essential cycles. For instance, consider any 1-cycle that surrounds any one of the two 1-dimensional holes of a torus. Essential cycles are considered equivalent if they differ by a boundary.

The first homology group of a torus \mathbb{T}^2 over $\mathbb{Z}/2$ is given by $H_1(\mathbb{T}^2) = \mathbb{Z}/2 \oplus \mathbb{Z}/2$. You might wonder what the third non-trivial homology class represents, besides the two essential cycles shown above. It's actually equivalent to an essential 1-cycle along the diagonal direction of a rectangle whose opposite edges were identified to form the torus. It looks like this:

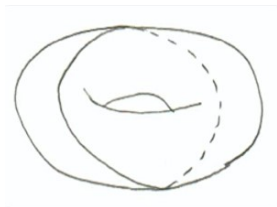
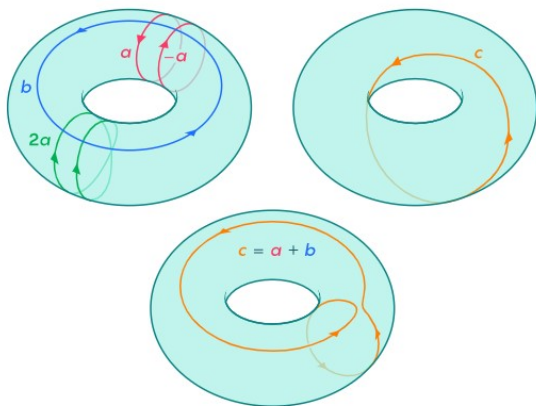


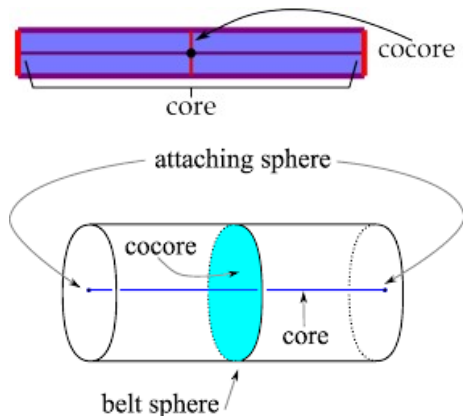
Figure: **The One That Got Away** from *What is the Homology of a Torus?*



The torus has infinitely many different loops on its surface. The oriented loops a , b , and c are all different, but c can be deformed to obtain the union of loops a and b .

Figure: *The Hole Truth* from **Quanta Magazine**

Handle Decomposition: Cores and Co-Cores



Topology is preserved when switching between core and co-core. We can think of the rectangle as either a fattened core or a fattened co-core. Likewise, we can think of the cylinder either as a fattened disc or as a fattened line.

Dual Cellulation and Poincaré Duality

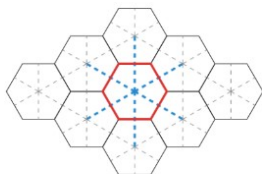


Figure 2.5: Part of a 2-dimensional tessellation by hexagons. The dual tessellation is drawn with dashed lines. The boundary of a face (red) corresponds to the coboundary of a vertex in the dual structure (blue).

This intuition is that the i -cells of a cell complex X correspond to the $D - i$ cells of the dual cell complex X^* . By linear extension, this gives rise to an isomorphism $*$: $C_i(X) \rightarrow C_{D-1}(X^*)$. It's also at least intuitively clear that transitioning to the dual chain leaves the inner product (evenness or oddness) invariant, although does require formal proof.

$$\langle a, b \rangle_{C_i} = \langle *a, *b \rangle_{C_{D-i}^*}$$

If we work through the definitions then we will also have that applying the coboundary operator to a chain of a cell complex is equivalent to going to the dual complex and applying the boundary operator of the complementary dimension.

$$\begin{array}{ccc}
 C_i & \xrightarrow{\delta_i} & C_{i+1} \\
 \downarrow * & & \downarrow * \\
 C_{D-i}^* & \xrightarrow{\partial_{D-i}} & C_{D-i-1}^*
 \end{array}$$

$$\delta_i = *^{-1} \circ \partial_{D-i} \circ *$$

Toric Code from Cell Complex

Disclaimer: *I claim that this construction will fall out of the sky. But I also claim that in hindsight the construction will be “obvious”.*

To turn a cell complex X into a “stabilizer code” we can pick a dimension $i \in \{1, \dots, D - 1\}$ and identify all i -cells with qubits. The boundaries of the $i + 1$ -cells are used to define Z -checks and the coboundaries of the $i - 1$ -cells are used to define X -checks.

Like in the following picture for the toric code:

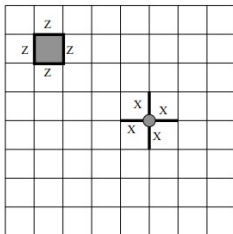


FIG. 1. Check operators of the toric code. Each plaquette operator is a tensor product of Z 's acting on the four links contained in the plaquette. Each site operator is a tensor product of X 's acting on the four links that meet at the site.

Actually, let's take a short break from here and discuss Fourier transforms. We'll be right back.

Part III: A Brief Digression on Fourier Dualities

Fourier Transforms in Terms of Basis Expansions

I'm sure all of you have seen the following in one of its many incarnations:

Let R be the line parameterized by x . Let f be a complex function on R that is integrable. The Fourier transform $\hat{f} = Ff$ is

$$\hat{f}(k) = \int_{-\infty}^{+\infty} e^{-ikx} f(x) dx.$$

Which I interpret as the projection of $f(x)$ onto the basis of exponential functions e^{-ikx} , i.e., (f, e_k) and $f = (f, e_k)e_k$.

One thing to look is what happens when you scale x by a factor. You'll find that you have to scale the Fourier variable k by the inverse factor. That means k lives in a vector space that is dual to the real line x lives in.

Multi-Dimensional Fourier Transform

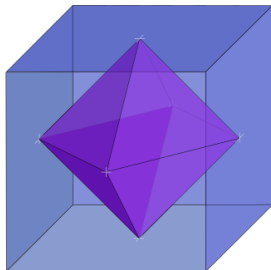
Now how would we define the Fourier transform for a function on a finite-dimensional vector space V ? What replaces that kx in the e^{ikx} factor in the Fourier transform?

The answer is that in order to write a meaningful operation down without choosing any additional structure we must take $e^{i\langle k, x \rangle}$ where $\langle k, x \rangle$ is the dual pairing between $k \in V^*$ and $x \in V$. We'll end up getting a generalization of the single-dimensional Fourier transform and we can also see the effect on k upon applying a linear transformation to x .

Examples of Fourier-style dualities

Polar Dual of a Convex Body:

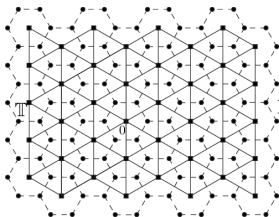
If K is a convex body, then $K^* = \{\mathbf{y} : \langle \mathbf{y}, \mathbf{x} \rangle \leq 1, \forall \mathbf{x} \in K\}$



Examples of Fourier-style dualities

Reciprocal Lattice:

If L is a lattice, then $L^* = \{\mathbf{u}, \forall \mathbf{v} \in L, \langle \mathbf{u}, \mathbf{v} \rangle \in \mathbb{Z}\}$. If B is a basis for L , then $(B^{-1})^T$ is a basis for L^* .



Pontryagin duality: Co-equal status for all Fourier transforms

There are several distinct but related notions of Fourier transforms. You might have heard of them as continuous Fourier transforms, discrete Fourier transforms, discrete-time Fourier transforms and so on.

These disparate notions are all unified under **Pontryagin duality** and given *co-equal* status. If G is a locally compact Hausdorff group, the Pontryagin dual is the group \hat{G} of continuous group homomorphisms from G to the circle group T . That is, $\hat{G} = \text{Hom}(G, T)$. Then $L^2(G) \cong L^2(\hat{G})$, where \cong is a generalized Fourier transform.

In the context of quantum computing, the discrete Fourier transform is the one that is usually most relevant and we will focus on that for now.

The discrete Fourier transform transforms a sequence of N complex numbers $\{\mathbf{x}_n\} := x_0, x_1, \dots, x_{N-1}$ into another sequence of complex numbers, $\{\mathbf{X}_k\} := X_0, X_1, \dots, X_{N-1}$, which is defined by

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{\frac{-i2\pi}{N} kn}$$

The inverse Fourier transform is given by

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{i\frac{2\pi}{N} kn}$$

A circular shift of the input x_n corresponds to multiplying the output X_k by a linear phase. Similarly, multiplying x_n by a linear phase $e^{\frac{i2\pi(n-1)}{n}m}$ for some integer m corresponds to a *circular shift* of the output X_k : X_k is replaced by X_{k-m} where the subscript is interpreted modulo N (i.e., periodically).

If $\mathcal{F}(\{x_n\})_k = X_k$ then

$$\mathcal{F}(\{x_n \cdot e^{i2\pi nm/N}\}) = X_{k-m}$$

and

$$\mathcal{F}(\{x_{n-m}\})_k = X_k \cdot e^{-\frac{i2\pi}{N}km}$$

But note that this is exactly what our “quantum computing” equation $HXH = Z$ and $HZH = X$ is saying!

Shift and Clock matrices

This becomes even clearer if we use the Sylvester shift and clock matrices.

$$\Sigma_1 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

$$\Sigma_3 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & \omega & 0 & \cdots & 0 \\ 0 & 0 & \omega^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \omega^{d-1} \end{bmatrix}$$

The Walsh-Hadamard Transform

The Sylvester matrix for the Walsh-Hadamard Transform is:

$$W = \frac{1}{\sqrt{d}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^{d-1} & \omega^{2(d-1)} & \cdots & \omega^{(d-1)^2} \\ 1 & \omega^{d-2} & \omega^{2(d-2)} & \cdots & \omega^{(d-1)(d-2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega & \omega^2 & \cdots & \omega^{d-1} \end{bmatrix}$$

Direct calculation yields:

$$\Sigma_1 = W \Sigma_3 W$$

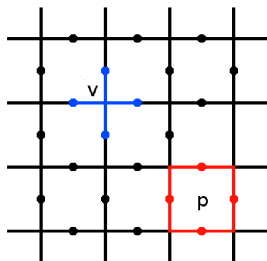
and

$$\Sigma_3 = W \Sigma_1 W.$$

We specifically note that the Walsh-Hadamard transform can be regarded as being built out of size-2 discrete Fourier transforms (DFTs), and is in fact equivalent to a multidimensional DFT of size $2 \times 2 \times \cdots \times 2$.

Part IV: Revisiting Homological Codes

Part IV: Revisiting Homological Codes

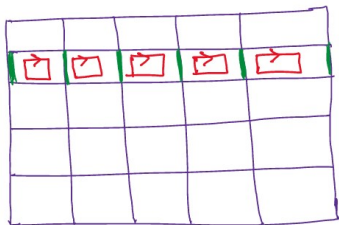


Let \mathcal{H} denote the Hilbert space that is the state space of the toric lattice with periodic boundary conditions. Then we can write $V_S \subseteq \mathcal{H}$ as

$$V_S = \{|\psi\rangle \in \mathcal{H} : A_v|\psi\rangle = |\psi\rangle, B_p|\psi\rangle = |\psi\rangle\}$$

where $A_v = \bigotimes_{v \in j} \sigma_j^X$ and $B_p = \bigotimes_{j \in \partial P} \sigma_j^Z$.

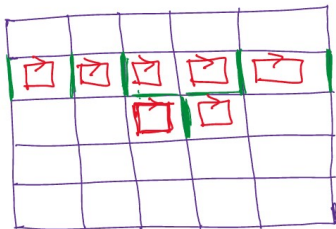
Encoded Subspace Calculation in Original Space



The red square circulations represent Z -plaquettes in my image. The green edges represent 1s in the computational basis and the non-green edges represent $|0\rangle$ s in the computational basis.

In this scenario, the Z -plaquette operators affect the phases trivially because each plaquette has a $0 \pmod 2$ overlap with the $|1\rangle$ edges. This occurs precisely because such a state is a 1-cocycle, i.e., an element of $\ker d^1$.

So elements of \mathcal{H} that satisfy the $B_p|\psi\rangle = |\psi\rangle$ constraint are elements of $\ker d^1$. But there's also the vertex constraints that we need to take care of.



The vertex constraints basically set up an equivalence relation on the states which can intuitively be interpreted as a permutation on the state bitstrings. We note that upon application of a B_p an element of $\ker d^1$ is still taken to an element of $\ker d^1$.

We also note that B_p is a 1-coboundary. So the legitimate states of our encoded subspace V_S is spanned by the 1-cocycles modulo the 1-coboundaries which correspond to the cohomology classes of $H^1(\mathbb{T}_2, \mathbb{Z}/2)$.

Then we can take basis vectors for V_S related to the cohomology classes as follows:

$$|\psi_{00}\rangle := \sum_{\alpha} a_{\alpha} |\alpha\rangle$$

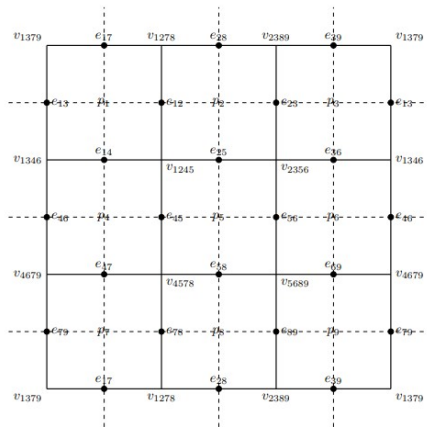
$$|\psi_{01}\rangle := \sum_{\beta} b_{\beta} |\beta\rangle$$

$$|\psi_{10}\rangle := \sum_{\gamma} c_{\gamma} |\gamma\rangle$$

$$|\psi_{11}\rangle := \sum_{\delta} d_{\delta} |\delta\rangle$$

where α are the elements of the $(0,0)$ cohomology class, β are the elements of the $(0,1)$ cohomology class, and so on.

The constraints arising from the A_V ensure that all the a_{α} are equal, b_{β} are equal, c_{γ} are equal and d_{δ} are equal.



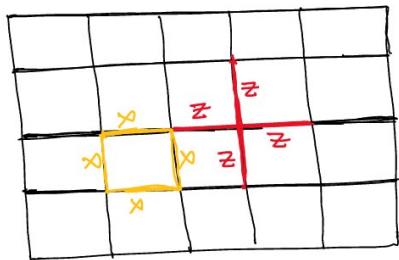
$$|00\rangle = \frac{1}{\sqrt{|A|}} \sum_{g \in A} g|0\dots 0\rangle$$

$$|01\rangle = \frac{1}{\sqrt{|A|}} \sum_{g \in A} g|e_{46}, e_{45}, e_{56} = 1, 0 \text{ else}\rangle$$

$$|10\rangle = \frac{1}{\sqrt{|A|}} \sum_{g \in A} g|e_{17}, e_{14}, e_{47} = 1, 0 \text{ else}\rangle$$

$$|11\rangle = \frac{1}{\sqrt{|A|}} \sum_{g \in A} g|e_{46}, e_{45}, e_{56}, e_{17}, e_{14}, e_{47} = 1, 0 \text{ else}\rangle$$

Encoded Subspace Calculation in Conjugate Space



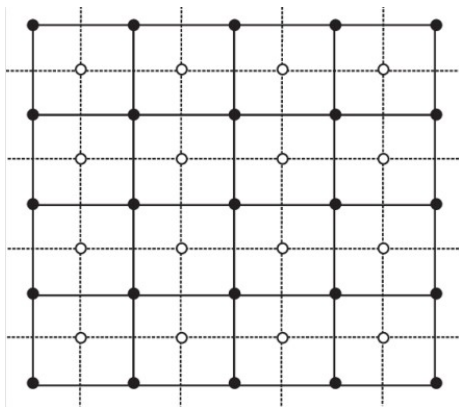
We noted in our earlier discussion that a bit flip error shows up as a phase flip error in the conjugate $\{|+\rangle, |-\rangle\}$ basis and the Hadamard is the basis change matrix. If we apply $H^{\otimes n}$ to the n physical qubits we transfer to the dual vector space where the edges are now 1-cochains rather than 1-chains (because the Hadamard transform is a Fourier transform).

The vertex operators in conjugate space are Z -type operators and the plaquette operators are X -type operators.

Due to the vertex checks in this conjugate space, the valid states should be elements of $\ker d_1$. That is, at each vertex there must be an even number of $|1\rangle$ s. But they're equivalent up to boundaries (or products of applications of the plaquette checks). So the encoded subspace corresponds to the homology classes in $H_1(\mathbb{T}_2, \mathbb{Z}/2)$ which has the same dimension as the cohomology group $H^1(\mathbb{T}_2, \mathbb{Z}/2)$.

Therefore, although there is no canonical isomorphism between the bases of H_1 and H^1 , there is a canonical isomorphism between $\ell_2(H_1(\mathbb{T}_2, \mathbb{Z}/2))$ and $\ell_2(H^2(\mathbb{T}_2, \mathbb{Z}/2))$, which was essentially the point of Pontryagin duality. $\ell_2(H_1(\mathbb{T}_2, \mathbb{Z}/2))$ and $\ell_2(H^1(\mathbb{T}_2, \mathbb{Z}/2))$ both represent the states of the encoded qubits.

Dual Cellulation and the Utility of Poincaré Duality



At times, it is *convenient* to represent the conjugate space on a Poincaré dual cellulation, but that is by no means necessary!

It is unfortunate that texts on homological codes do not separate these two notions of Fourier duality and Poincaré duality. Yes, we can compose them, but it's not crucial to the subject.

The above image shows a primal (solid) and its dual (dashed) square lattice. The i -cells in the primal graph correspond to $(2-i)$ -cells in the dual graph.

Let \mathcal{H} denote the Hilbert space which is the state space of the lattice. Then we can write the logical codespace as $V_S \subseteq \mathcal{H}$ as

$$V_S = \{|\psi\rangle \in \mathcal{H} | A_v |\psi\rangle = |\psi\rangle, B_p |\psi\rangle = |\psi\rangle\}$$

for all v, p . A_v and B_p are the vertex and plaquette operators as we had discussed in our earlier slides.

We will now relate the logical codespace to the (co)homology of the torus. Let C_0, C_1 and C_2 denote the free \mathbb{Z}_2 -vector space over vertices (0-cells), edges (1-cells) and plaquettes (2-cells) in the primal graph respectively. Similarly, we denote C'_0, C'_1 and C'_2 as the corresponding vector spaces for vertices, edges, and plaquettes in the dual graph. We then have the following chain complexes

$$\cdots \rightarrow 0 \rightarrow C_2 \xrightarrow{\delta_2} C_1 \xrightarrow{\delta_1} C_0 \rightarrow 0$$

$$\cdots \rightarrow 0 \rightarrow C'_2 \xrightarrow{\delta_2} C'_1 \xrightarrow{\delta_1} C'_0 \rightarrow 0$$

and we denote the homology groups for the primal graph and the dual graph by $H(\mathbb{T}^2)$ and $H'(\mathbb{T}^2)$. They're both $\mathbb{Z}/2 \oplus \mathbb{Z}/2$ because both the primal and dual graphs come from a cellulation of a torus.



The 1-cell $|$ corresponds to the 1-cell $---$ in the dual graph.

Let us choose a basis for \mathcal{H} by assigning a label $t_j = 0, 1$ to every edge j of the primal graph (which also assigns the same label to the corresponding edge in the dual graph). Denote this basis by \mathcal{B} . An element $|\psi\rangle \in \mathcal{B}$ then corresponds to some element in C_1 (and C'_1).

So if the edge qubits in the primal lattice are in the computational basis state (say) $|000 \cdots 1000 \cdots\rangle$ then that corresponds to a certain (not necessarily connected) chain in the dual lattice, as we can see in the picture above.

Once again, we can take the basis vectors of V_S related to the four homology classes of the dual graph as follows:

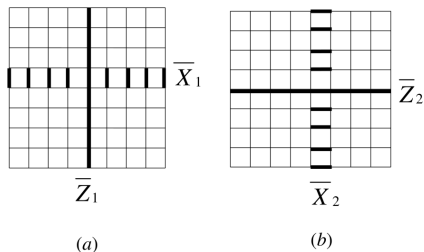
$$|\psi_{00}\rangle := \sum_{\alpha} a_{\alpha} |\alpha\rangle$$

$$|\psi_{01}\rangle := \sum_{\beta} b_{\beta} |\beta\rangle$$

$$|\psi_{10}\rangle := \sum_{\gamma} c_{\gamma} |\gamma\rangle$$

$$|\psi_{11}\rangle := \sum_{\epsilon} d_{\epsilon} |\epsilon\rangle$$

where the α are the elements of C'_1 that correspond to the homology class $(0, 0)$, the β correspond to the class $(0, 1)$, and so on. The constraints arising from the A_v ensure that all the a_{α} are equal, b_{β} are equal, c_{γ} are equal and d_{ϵ} are equal. This is basically the same calculation which we already saw for cohomology H^1 .

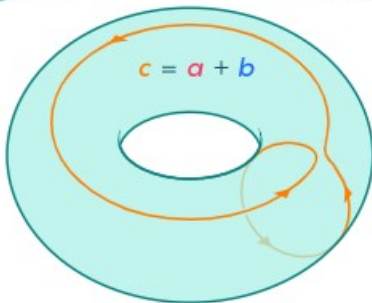
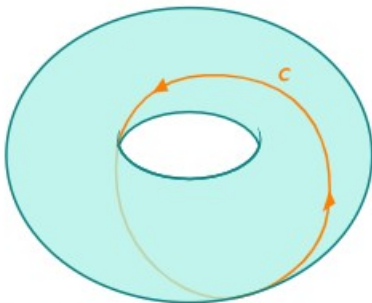
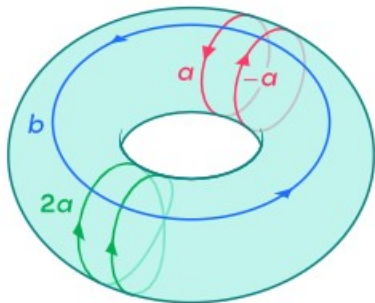


On a chain belonging to the $(0, 0)$ homology class if we apply the \bar{X}_1 operator which corresponds to a horizontal essential cycle in the dual graph then we would have converted the $(0, 0)$ homology class to the $(1, 0)$ homology class! And that is exactly how the logical operator \bar{X}_1 is supposed to act on the state $|\psi_{00}\rangle$ and take it to $|\psi_{10}\rangle$.

Likewise, we can think about the action of the logical operators \bar{X}_1 and \bar{X}_2 on the rest of the logical basis states.

Two useful hints are:

1. Two essential cycles in the same direction (horizontal or vertical) are basically a boundary and they together cancel out homologically (because they together form a boundary).
2. The $(1, 1)$ homology class $a + b$ can be thought of as noodles along the diagonal of the rectangle whose opposite edges were identified to form a torus.



One remaining confusion one might have is how to think about the action of the Z -type logical operators \overline{Z}_1 and \overline{Z}_2 . A quick explanation is that the homology class $(0, 0)$ in the dual lattice looks like non-essential cycles and in the primal lattice that will correspond to cycles that intersect with the \overline{Z}_1 and \overline{Z}_2 operators an even number of times.

Whereas the homology class $(1, 0)$ and $(0, 1)$ in the dual lattice are homologically equivalent to an application of the \overline{X}_1 or \overline{X}_2 operators on a $(0, 0)$ homology state in the dual lattice (think!). We can also think about the $(1, 1)$ homology class.

Now even for a single qubit, we know that

$$Z|1\rangle = ZX|0\rangle = -XZ|0\rangle = -|1\rangle.$$

From this simple logic, it follows that the \overline{Z}_1 operator flips the sign of the $|\psi_{10}\rangle$ and $|\psi_{11}\rangle$ basis states due to an odd number of overlaps. Likewise, \overline{Z}_1 flips the sign of the $|\psi_{10}\rangle$ and $|\psi_{11}\rangle$ logical basis states.

So everything works out as expected!

The moral of our story was that any cellulation X of a D -dimensional manifold gives a quantum CSS code. Using \mathbb{Z}_2 -homology we can relate the properties of the code, namely the number of physical qubits n , the number of encoded qubits k , and the code distance d to properties of X . The Poincaré dual cellulation was convenient but not crucial.

The End!