

HYPERMAP-HOMOLOGY QUANTUM CODES

by
Martin Leslie

A Dissertation Submitted to the Faculty of the
DEPARTMENT OF MATHEMATICS
In Partial Fulfillment of the Requirements
For the Degree of
DOCTOR OF PHILOSOPHY
In the Graduate College
THE UNIVERSITY OF ARIZONA

2013

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Martin Leslie entitled
Hypermap-Homology Quantum Codes
and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

_____ DATE: MAY 10, 2013
MAREK RYCHLIK

_____ DATE: MAY 10, 2013
KLAUS LUX

_____ DATE: MAY 10, 2013
PHAM HUU TIEP

_____ DATE: MAY 10, 2013
JANEK WEHR

FINAL APPROVAL AND ACCEPTANCE OF THIS DISSERTATION IS CONTINGENT UPON THE CANDIDATE'S SUBMISSION OF THE FINAL COPIES OF THE DISSERTATION TO THE GRADUATE COLLEGE.

I HEREBY CERTIFY THAT I HAVE READ THIS DISSERTATION PREPARED UNDER MY DIRECTION AND RECOMMEND THAT IT BE ACCEPTED AS FULFILLING THE DISSERTATION REQUIREMENT.

_____ DATE: MAY 10, 2013
MAREK RYCHLIK

STATEMENT BY AUTHOR

THIS DISSERTATION HAS BEEN SUBMITTED IN PARTIAL FULFILLMENT OF REQUIREMENTS FOR AN ADVANCED DEGREE AT THE UNIVERSITY OF ARIZONA AND IS DEPOSITED IN THE UNIVERSITY LIBRARY TO BE MADE AVAILABLE TO BORROWERS UNDER RULES OF THE LIBRARY.

BRIEF QUOTATIONS FROM THIS DISSERTATION ARE ALLOWABLE WITHOUT SPECIAL PERMISSION, PROVIDED THAT ACCURATE ACKNOWLEDGMENT OF SOURCE IS MADE. REQUESTS FOR PERMISSION FOR EXTENDED QUOTATION FROM OR REPRODUCTION OF THIS MANUSCRIPT IN WHOLE OR IN PART MAY BE GRANTED BY THE HEAD OF THE MAJOR DEPARTMENT OR THE DEAN OF THE GRADUATE COLLEGE WHEN IN HIS OR HER JUDGMENT THE PROPOSED USE OF THE MATERIAL IS IN THE INTERESTS OF SCHOLARSHIP. IN ALL OTHER INSTANCES, HOWEVER, PERMISSION MUST BE OBTAINED FROM THE AUTHOR.

SIGNED: _____ MARTIN LESLIE _____

ACKNOWLEDGMENTS

Thank you to my advisor Marek Rychlik for all his valuable effort and time and to my committee members Klaus Lux, Pham Huu Tiep and Janek Wehr. Thank you to my girlfriend Deborah Shelton, my parents Pat and Michael and my sisters Jillian and Ellen for their support over so many years. Thank you to my friends and colleagues among the graduate students in the mathematics department, especially Enrique Acosta, Michael Bishop, Michael Gilbert, Yaron Hadad, Shane Passon and Sarah Mann.

DEDICATION

For my mum and dad.

TABLE OF CONTENTS

LIST OF FIGURES	7
LIST OF TABLES	8
ABSTRACT	9
CHAPTER 1. INTRODUCTION	10
CHAPTER 2. BACKGROUND	14
2.1. Classical codes	14
2.2. Quantum mechanics on qubits	16
2.3. Stabilizer codes	18
2.4. Check matrix representation	22
2.5. CSS codes	25
2.6. An example of a stabilizer code	26
CHAPTER 3. MAP-HOMOLOGY CODES	27
3.1. Codes from \mathbb{F}_2 -chain complexes	27
3.2. Codes from graphs on surfaces	29
3.3. Planar codes	33
CHAPTER 4. HYPERMAP-HOMOLOGY CODES	38
4.1. Hypermaps	38
4.2. The dual hypermap	41
4.3. Hypermap homology	45
4.4. Hypermap-homology codes	51
4.5. Finding weights given a special basis	57
4.6. Future work	60
APPENDIX A. SOFTWARE	62
A.1. Matlab software for hypermap-homology codes	62
INDEX	64
REFERENCES	65

LIST OF FIGURES

FIGURE 3.1.	A toric code with $m = 4$	32
FIGURE 3.2.	A toric code and its dual with $m = 4$	32
FIGURE 3.3.	A planar code with one hole	35
FIGURE 3.4.	A planar code with a grid of holes	36
FIGURE 3.5.	A planar code with a grid of 2×2 holes	37
FIGURE 4.1.	First example of a hypermap	41
FIGURE 4.2.	Octagon-square hypermap	44
FIGURE 4.3.	Octagon-square hypermap with dual	46
FIGURE 4.4.	Hexagon hypermap	54
FIGURE 4.5.	4×4 square hypermap	54
FIGURE 4.6.	4×4 square hypermap with dual	55
FIGURE 4.7.	Dual(L(Cube)) hypermap	56

LIST OF TABLES

TABLE 1.1.	Some code constructions	11
TABLE 4.1.	Table of conventions for hypermaps	40

ABSTRACT

We introduce a new type of sparse CSS quantum error correcting code based on the homology of hypermaps. Sparse quantum error correcting codes are of interest in the building of quantum computers due to their ease of implementation and the possibility of developing fast decoders for them. Codes based on the homology of embeddings of graphs, such as Kitaev's toric code, have been discussed widely in the literature and our class of codes generalize these. We use embedded hypergraphs, which are a generalization of graphs that can have edges connected to more than two vertices.

We develop theorems and examples of our hypermap-homology codes, especially in the case that we choose a special type of basis in our homology chain complex. In particular the most straightforward generalization of the $m \times m$ toric code to hypermap-homology codes gives us a $[(3/2)m^2, 2, m]$ code as compared to the toric code which is a $[2m^2, 2, m]$ code. Thus we can protect the same amount of quantum information, with the same error-correcting capability, using less physical qubits.

CHAPTER 1

INTRODUCTION

This dissertation introduces a new type of quantum error correcting code which we will call hypermap-homology quantum codes. We also provide examples and analysis of said codes.

Quantum computers may be able to provide significant speedups over classical computers in certain problems. The most famous example is factoring integers using Shor's algorithm, where the quantum algorithm is exponentially faster than any known classical algorithm. They may also be able to simulate quantum systems faster than classical computers. However quantum noise (such as undesired unitary evolution of a quantum state or undesired measurements) makes building quantum computers difficult. Quantum error correction may be able to provide an answer to these problems. The quantum fault tolerance theorem roughly says that if we can implement quantum gates with error probability per unit of time below some threshold then by concatenating quantum codes (i.e. building sufficiently many layers of quantum error correction into our gates) we can get the error rate arbitrarily low. We will not go into the details of such practicalities, instead looking at quantum stabilizer codes as mathematical objects. See [NC10] for more information on quantum computing and error correction.

We consider binary CSS (Calderbank-Shor-Steane) codes, a special case of stabilizer codes (see Chapter 2 for definitions and details). CSS codes are defined by two \mathbb{F}_2 matrices H_X, H_Z such that $H_X H_Z^T = 0$. These codes have parameters $[N, K, D]$ where K information qubits are encoded into N physical qubits with minimum distance D . A minimum distance D code can correct any errors on $\lfloor (D-1)/2 \rfloor$ qubits.

It was shown non-constructively in [CS96] that there exist CSS codes with K/N

fixed and $D \sim cN$ for a constant c . Codes with this property are called *good*. However it is not known if *sparse* good CSS codes (codes with H_X and H_Z sparse) exist. Sparse quantum codes are interesting for at least two reasons. The first is that sparse classical codes such as Low Density Parity Check (LDPC) codes are known to have performance as good as any codes while also having efficient suboptimal decoders. Being able to decode quickly is important in applications such as quantum fault tolerance. The second reason is that in physical implementations of quantum codes, the matrices can correspond to connections between the physical qubits. So a sparse quantum code can mean that the qubits are connected to few neighboring qubits which can lead to easier implementation.

In [Kit03] a family of sparse codes called toric codes were introduced which have parameters $[2m^2, 2, m]$, thus K is fixed and $D \sim (1/\sqrt{2})\sqrt{N}$. Despite their relatively poor performance toric and toric-like codes have promise in that they may be easy to implement and decode as discussed above. In [MMM04] sparse graph quantum codes inspired by classical LDPC codes were introduced. However the most practically successful class of such codes, the ‘bicycle codes’, are expected to have D bounded above by a constant as N increases.

Codes with growing distance have been suggested in the years since. We summarize some attempts with Table 1.1. In [Zém09] it was suggested that looking at KD^2 may be a way to compare toric-like codes so we include this information also.

Code	K	D	KD^2
Good codes [CS96]	$\sim c_1N$	$\sim c_2N$	$\sim c_3N^3$
Bicycle codes [MMM04]	$\sim c_1N$	expect $\leq c_2$	$\sim c_3N$
Toric code [Kit03]	2	$(1/\sqrt{2})\sqrt{N}$	N
Systolic freedom [FML02], [Fet11]	c_1	$\sim c_2\sqrt{N \log N}$	$\sim c_3N \log N$
Hypergraph-product [TZ09], [KP12]	$\sim c_1N$	$\sim c_2\sqrt{N}$	$\sim c_3N^2$
Cayley graph [Zém09]	$\sim c_1N$	$\sim c_2 \log N$	$\sim c_3N(\log N)^2$
Cayley graph repetition [CDZ11]	$\sqrt{2}\sqrt{N}$	$(1/\sqrt{2})\sqrt{N}$	$(1/\sqrt{2})N^{3/2}$

TABLE 1.1. Some code constructions.

Some results constraining the parameters of toric-like codes have been proved. In [Fet11] it is shown that codes based on the homology of a fixed surface have a bound $D^2 \leq cN$. Similarly in [BPT10] it is shown that for ‘geometrically local’ codes on a 2D lattice we have $KD^2 \leq N$. Another relevant result is Gallager’s proof in [Gal62] that classical LDPC codes of column weight two cannot have minimum distance d linear with blocklength n . This result does not imply that CSS codes with column weight two matrices cannot achieve $D \sim cN$ but it does make it quite unlikely. It is our hope (as yet unrealized) that our codes may be able to avoid these limits.

Hypergraph based quantum codes have been proposed in [TZ09] (based on products of hypergraphs) and [SB12]. However to our knowledge no codes based on homology of hypermaps have been suggested before.

Our interest in hypermap-homology codes initially came from considering the existing construction of codes from homology of embeddings of graphs. In these constructions the fact that an edge is connected to only two vertices in classical graphs leads to one of the matrices in the CSS construction having column weight two. Hypermap-homology codes are based on embeddings of hypergraphs - a generalization of graphs where an edge can be connected to more than two vertices. Unfortunately, in our examples we make a choice of basis that does lead to the relevant matrix having column weight two. However our construction is still a generalization of codes based on graph homology and in Example 4.24 we can see that the straightforward generalization of the toric code to hypermap-homology codes has better parameters (it can store the same amount of information, with the same error correction capability, in less qubits). We see this as a proof of concept that hypermap-homology codes can be useful.

We will not discuss decoding of our codes. They are quantum LDPC codes so can be decoded by standard belief propagation techniques as in [MMM04]. However the example of decoding toric codes show that this may not work particularly well. See for example [DCP10] for specialized techniques that may be able to be extended to

hypermap-homology codes.

In Chapter 2 we give the background knowledge required in classical coding theory, quantum mechanics on qubits, and stabilizer codes. In Chapter 3 we give an exposition of how we can create CSS codes from \mathbb{F}_2 -chain complexes, including from graphs embedded in surfaces and *planar codes* which come from grids in the plane with certain holes removed. In Chapter 4 we give the required background in hypermaps and hypermap homology before introducing hypermap-homology codes and their analysis. Appendix A is a short discussion of computer software that we developed to compute parameters of hypermap-homology codes.

The work in this dissertation that to our knowledge is original is:

1. an explanation that CSS codes can be constructed from any \mathbb{F}_2 -chain complex in Section 3.1,
2. an exposition of planar codes that fills in some of the more intuitive arguments from the literature in Section 3.3,
3. propositions expressing different ways to understand hypermap-homology in Section 4.3, and
4. definitions and methods to determine the weight of hypermap-homology codes with a certain type of basis in Sections 4.4 and 4.5. We also include examples.

CHAPTER 2

BACKGROUND

2.1 Classical codes

Classical information theory and coding theory were initiated by the works of Shannon and Hamming in 1948 and 1950 respectively. This section draws from a number of sources including [NC10], [Mac03], [RU08] and [Gur10] to give an introduction to binary linear codes.

Consider the vector space \mathbb{F}_2^n with the non-degenerate symmetric bilinear form

$$(x, y) \mapsto x \cdot y = \sum_i x_i y_i.$$

We write elements of \mathbb{F}_2^n as column vectors. Define an $[n, k]$ *binary linear code* C to be a k -dimensional subspace of \mathbb{F}_2^n , with elements of C called *codewords*. We say that $G \in M_{n \times k}(\mathbb{F}_2)$ is a *generator matrix* for C if $C = G(\mathbb{F}_2^k)$. A matrix $H \in M_{m \times n}(\mathbb{F}_2)$ is called a *parity check matrix* for C if $C = \ker(H)$. Since G is injective we know that G is full rank i.e. $\text{rank}(G) = k$. However, since $\mathbb{F}_2^n / \ker(H) \cong \text{im}(H)$ we have $\dim(\text{im}(H)) = n - k$ so with our definition H does not have to be full rank. We allow the parity check matrix to include $m \geq n - k$ conditions of which $m - (n - k)$ must be redundant. This is purely for convenience; many of our code constructions will be via specifying parity check matrices which may not necessarily be full rank.

Furthermore we have $HG = 0$ because columns of G are elements of C and H times an element of C is 0.

To form a generating matrix for a code C choose a basis for C and place these as columns of G . Then we can generate all the codewords by adding basis codewords i.e. multiplying G by some x .

To form a full rank parity check matrix for C we consider the orthogonal comple-

ment

$$C^\perp = \{y \in \mathbb{F}_2^n : y \cdot x = 0 \text{ for all } x \in C\},$$

called the *dual code* of C . One thing to note is that we do not necessarily have $C \cap C^\perp = \{0\}$ (for example even weight codewords are orthogonal to themselves). However our bilinear form is non-degenerate so we do still have

$$\dim(C) + \dim(C^\perp) = \dim(\mathbb{F}_2^n)$$

and thus $\dim(C^\perp) = n - k$. Now choose a basis for C^\perp and use this as the rows of H .

To show that H is a parity check matrix for C we need

$$C = \{x \in \mathbb{F}_2^n : Hx = 0\}.$$

If $x \in C$ then $Hx = 0$ because each row of H is orthogonal to all elements of C . If $Hx = 0$ then we know that x is orthogonal to a basis for C^\perp and is thus orthogonal to C^\perp . Thus $x \in (C^\perp)^\perp = C$ (to see this last equality note that $C \subseteq (C^\perp)^\perp$ and that $\dim(C^\perp) + \dim((C^\perp)^\perp) = n$).

We now claim that C^\perp is a $[n, n - k]$ code with generator matrix H^T and parity check matrix G^T . To see that H^T is a generator matrix notice that the rows of H are a basis for C^\perp and thus any column vector in C^\perp can be written as a sum of columns of H^T . For the parity check matrix: if $x \in C^\perp$ then x is orthogonal to all elements of C so is orthogonal to columns of G so $G^T x = 0$. Finally if $G^T x = 0$ then x is orthogonal to a basis of C so is orthogonal to C so $x \in C^\perp$.

We now explain error correction with an $[n, k]$ code. To transmit $u \in \mathbb{F}_2^k$ we send $x = Gu$. Then some error $r \in \mathbb{F}_2^n$ (it may be the trivial error $r = 0$) occurs and we receive $y = x + r$. The decoding algorithm is to decide that the sent codeword is a codeword $x^* \in C$ which is closest to y . Here closest is with respect to the Hamming distance

$$d_H(a, b) = \text{number of elements in which } a \text{ and } b \text{ differ.}$$

Then to recover the information we decode to the unique $u^* \in \mathbb{F}_2^k$ which corresponds to x^* .

With this in mind, when designing a linear code we would like codewords to be far apart. The *minimum distance* d of a code is the minimum of $d_H(a, b)$ for all $a \neq b \in C$. For linear codes this minimum distance is also the minimum weight of nonzero codewords where the weight of a codeword is

$$\text{wt}(a) = d_H(a, 0) = \text{number of 1's in } a.$$

To see this, we have

$$d = \min_{a \neq b \in C} d_H(a, b) = \min_{a \neq b \in C} \text{wt}(a + b) = \min_{c \in C \setminus \{0\}} \text{wt}(c).$$

We will refer to such a code as an $[n, k, d]$ -code.

2.2 Quantum mechanics on qubits

This introduction to quantum mechanics for quantum computing follows [NC10] with some simplifications. Our quantum computing model is based on *qubits* (named for quantum bits) although more general qudits (quantum digits) or other systems are possible. Choose a basis of \mathbb{C}^2 (called the *computational basis*) to be

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

We now give some postulates for the quantum mechanics of qubits.

1. The state of n qubits is an element of $\mathcal{H}_n = (\mathbb{C}^2)^{\otimes n}$. We work with vectors which are normalized to have $\langle \psi | \psi \rangle = 1$ and consider states differing multiplicatively by $e^{i\theta}$ with $\theta \in \mathbb{R}$ to be equal. More concretely, $|\psi\rangle \in \mathcal{H}_n$ can be written as

$$|\psi\rangle = \sum_{i \in \mathbb{F}_2^n} a_i |i\rangle$$

where $|i\rangle = |i_1 i_2 \dots i_n\rangle = |i_1\rangle |i_2\rangle \dots |i_n\rangle = |i_1\rangle \otimes |i_2\rangle \otimes \dots \otimes |i_n\rangle$ are all just different notations for the same thing. The normalization condition means $\sum |a_i|^2 = 1$.

2. The evolution of the system is described by a unitary transformation. That is, the state $|\psi\rangle$ of the system at time t_1 is related to the state $|\psi'\rangle$ of the system at time t_2 by $U = U(t_1, t_2) \in U(\mathcal{H}_n)$ via $|\psi'\rangle = U|\psi\rangle$.
3. Given an *observable* M (a Hermitian operator on \mathcal{H}_n) with spectral decomposition $M = \sum_m m P_m$ we can measure a state $|\psi\rangle$ with respect to M . The result of the measurement is m with probability $p(m) = \langle \psi | P_m | \psi \rangle$ and if the result is m then the new state is $P_m |\psi\rangle / \sqrt{p(m)}$.

The following proposition justifies the commonly used statement that commuting observables can be measured ‘simultaneously’.

Proposition 2.1. *If observables M_1, \dots, M_k commute then we can measure $|\psi\rangle$ with respect to these in any order and have the same probability of measurement and state outcome.*

Proof. Since the M_i are Hermitian they are diagonalizable and since they commute they are mutually diagonalizable. Write $M_i = \sum_{m_i} m_i P_{i,m_i}$. Notice that $P_{i,m_i} P_{i,m'_i} = \delta_{m_i m'_i} P_{i,m_i}$ and $P_{i,m_i} P_{j,m_j} = P_{j,m_j} P_{i,m_i}$ if $i \neq j$.

If we first measure with respect to M_i then we get result m_i with probability $p(m_i) = \langle \psi | P_{i,m_i} | \psi \rangle$ and new state $|\psi'\rangle = P_{i,m_i} |\psi\rangle / \sqrt{p(m_i)}$. Measuring this new state with respect to M_j gives result m_j with probability

$$p(m_j) = \langle \psi' | P_{j,m_j} | \psi' \rangle = \frac{\langle \psi | P_{i,m_i}^\dagger P_{j,m_j} P_{i,m_i} | \psi \rangle}{\sqrt{p(m_i)}} = \frac{\langle \psi | P_{j,m_j} P_{i,m_i} | \psi \rangle}{p(m_i)}$$

and new state

$$\frac{P_{j,m_j} |\psi'\rangle}{\sqrt{p(m_j)}} = \frac{P_{j,m_j} P_{i,m_i} |\psi\rangle}{\sqrt{\langle \psi | P_{j,m_j} P_{i,m_i} | \psi \rangle}}.$$

Thus measuring with respect to M_i and M_j in either order is equivalent to measuring with respect to $M_i M_j = \sum_{m_i, m_j} P_{j, m_j} P_{i, m_i}$. So measuring M_1, \dots, M_k in any order is equivalent to measuring $M_1 \dots M_k$. \square

2.3 Stabilizer codes

The stabilizer code formalism, first introduced by Gottesman in [Got97], is a way of describing quantum codes somewhat analogous to linear codes in the classical setting. Our discussion in this section mainly follows [NC10] and, in parts, [MMM04].

Define the Pauli matrices

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

They can be easily seen to be Hermitian and unitary and checked to satisfy the following relations:

$$\begin{aligned} X^2 &= Y^2 = Z^2 = I, \\ XY &= iZ \quad ZX = iY \quad YZ = iX, \\ YX &= -iZ \quad XZ = -iY \quad ZY = -iX. \end{aligned}$$

Let $U(\mathcal{H}_n)$ be the group of unitary operators on the space of n qubits. Define the *Pauli group* to be the group G_n inside $U(\mathcal{H}_n)$ generated by operators of the form $A_1 \otimes \dots \otimes A_n$ where each $A_i \in \{I, X, Y, Z\}$. We will sometimes use notation where we omit the tensor signs or include only the non-identity operators. For example $XIY = X_1 Y_3 \in G_3$ is shorthand for $X \otimes I \otimes Y$.

Using this notation we have $X_1 Y_1 Z_1 = iI \in G_n$. Thus the group G_n must contain $\{\pm I, \pm iI\}$. But this is enough to ensure G_n is closed under products and inverses: if $c, d \in \{\pm 1, \pm i\}$ and $A_i, B_i \in \{I, X, Y, Z\}$ then we have

$$\left(c \bigotimes_{i=1}^n A_i \right) \left(d \bigotimes_{i=1}^n B_i \right) = (cd) \bigotimes_{i=1}^n (A_i B_i)$$

and

$$\left(c \bigotimes_{i=1}^n A_i \right)^{-1} = c^* \bigotimes_{i=1}^n A_i$$

and thus

$$G_n = \left\{ c \bigotimes_{i=1}^n A_i : c \in \{\pm 1, \pm i\}, A_i \in \{I, X, Y, Z\} \right\}.$$

Proposition 2.2. *If we call elements of the Pauli group Pauli operators we have the following facts.*

1. *Pauli operators commute if and only if they have an even number of places with different non-identity matrices. If they do not commute then they anti-commute.*
2. *Squaring a Pauli operator gives $\pm I$.*
3. *A Pauli operator $c \bigotimes_{i=1}^n A_i$ is Hermitian if and only if $c = \pm 1$.*

Proof. These follow easily from our multiplication rule above and the relations among the Pauli matrices. □

Now for $S \leq G_n$ define $V_S \subseteq \mathcal{H}_n$ to be the set of vectors stabilized by S i.e.

$$V_S = \{|\psi\rangle : s|\psi\rangle = |\psi\rangle \text{ for all } s \in S\}.$$

It is easy to check that V_S is a subspace of \mathcal{H}_n and that we can write

$$V_S = \bigcap_{s \in S} V_{\{s\}}.$$

Proposition 2.3. *The subspace $V_S \neq 0$ only if $-I \notin S$. In this case we have S abelian, $\pm iI \notin S$ and the relations $g^2 = I$ and $g^\dagger = g$ for all $g \in S$.*

Proof. If $-I \in S$ then $|\psi\rangle = -|\psi\rangle$ so $|\psi\rangle = 0$ for all $|\psi\rangle \in V_S$.

If S is not abelian then there exists $M, N \in S$ such that $MN = -NM$, then $MNM^\dagger N^\dagger = -I \in S$. If $\pm iI \in S$ then $(\pm iI)^2 = -I \in S$. Also for any $g \in S$ we have $g^2 = \pm I$. So $-I \notin S$ implies $\pm iI \notin S$ and $g^2 = I$ for all $g \in S$. Then $g^2 = I$ implies $g^\dagger = g$ because g is unitary. □

Define a *stabilizer group* to be an abelian subgroup $S \leq G_n$ with $-I \notin S$. The *stabilizer code* given by S is V_S .

Recall the definitions of the normalizers and centralizer of a subgroup: the normalizer of the stabilizer group S in G_n is

$$N(S) = \{E \in G_n : EgE^\dagger \in S \text{ for all } g \in S\}$$

and the centralizer is

$$C(S) = \{E \in G_n : EgE^\dagger = g \text{ for all } g \in S\}.$$

Clearly $C(S) \subseteq N(S)$ but in this case the inclusion is true in the opposite direction also: if $E \in N$ then $EgE^\dagger = \pm gEE^\dagger = \pm g$. So since $EgE^\dagger \in S$ we must have plus not minus and $EgE^\dagger = g$.

We will often work with generators for S such that $S = \langle g_1, \dots, g_m \rangle$. Note that E commutes with all elements of S if and only if it commutes with all the generators. To prove this: using the fact that the g_l commute and square to one, a general $g \in S$ can be written as $g = g_1^{\epsilon_1} \cdots g_m^{\epsilon_m}$ with $\epsilon_l \in \{0, 1\}$. Then if E commutes with each g_l it commutes with g .

We now describe the error correction process. Our errors are elements of G_n . We start with a state $|\psi\rangle \in V_S$ then after error E occurs the system is in state $E|\psi\rangle$. If $S = \langle g_1, \dots, g_m \rangle$ then the *syndrome* of an error operator E is

$$\beta = \beta(E) = (\beta_1, \dots, \beta_m)$$

where $\beta_l \in \{0, 1\}$ is defined by the equation

$$Eg_l = (-1)^{\beta_l} g_l E.$$

Now the stabilizer generators g_l are commuting Hermitian operators so are observables that can be measured simultaneously. Each observable g_l has eigenvalues

± 1 (because $g_l^2 = I$) and the projectors onto the $+1$ and -1 eigenspaces are $(I + g_l)/2$ and $(I - g_l)/2$ respectively. Thus

$$g_l = (+1)\frac{I + g_l}{2} + (-1)\frac{I - g_l}{2}.$$

If β_l is the syndrome of g_l then measuring g_l gives result $+$ with probability

$$\begin{aligned} p(+) &= \langle \psi | E^\dagger \left(\frac{I + g_l}{2} \right) E | \psi \rangle \\ &= \frac{1}{2} \langle \psi | \psi \rangle + \frac{1}{2} \langle \psi | E^\dagger g_l E | \psi \rangle \\ &= \frac{1}{2} + \frac{1}{2} (-1)^{\beta_l} \langle \psi | g_l E^\dagger E | \psi \rangle \\ &= \frac{1}{2} + \frac{1}{2} (-1)^{\beta_l} \langle \psi | \psi \rangle \\ &= \frac{1}{2} + \frac{1}{2} (-1)^{\beta_l} \\ &= 1 - \beta_l. \end{aligned}$$

Thus the outcome of the measurement is deterministic and depends only on the syndrome of the error, not the state $|\psi\rangle$.

If we have a collection of errors with distinct syndromes then we can correct the errors. In fact more is true as we will now see.

Theorem 2.4. *Suppose $\{E_j\}$ is a set of error operators such that $E_j^\dagger E_k \notin C(S) \setminus S$ for all j, k . Then $\{E_j\}$ is correctable.*

Proof. First note that $E g_l = (-1)^{\beta_l} g_l E$ can be rewritten as $E^\dagger g_l = (-1)^{\beta_l} g_l E^\dagger$ so E^\dagger has the same syndrome as E . Let $\beta(E_j) = (\beta_{j,1}, \dots, \beta_{j,m})$ and similarly for $\beta(E_k)$. Then

$$E_j^\dagger E_k g_l = (-1)^{\beta_{j,l}} (-1)^{\beta_{k,l}} g_l E_j^\dagger E_k$$

so $E_j E_k^\dagger \in C(S)$ if and only if $(-1)^{\beta_{j,l}} (-1)^{\beta_{k,l}} = 1$ for all l if and only if E_j and E_k have the same syndrome. Thus the theorem to be proved can be restated as: a set of errors is correctable if errors with the same syndrome differ by an element of the stabilizer.

If the syndrome corresponding to E_j is unique then we can correct the error by applying the operation E_j^\dagger , so the state becomes $E_j^\dagger E_j |\psi\rangle = |\psi\rangle$. If we have two errors E_j and E_k with the same syndrome then, by assumption in the theorem, $E_j^\dagger E_k \in S$. Thus even if we use the ‘wrong’ operator, E_j^\dagger instead of E_k^\dagger , to correct we still have $E_j^\dagger E_k |\psi\rangle = |\psi\rangle$. \square

Define the *distance* of a stabilizer code to be the minimum weight (number of non-identity components) of a Pauli operator in $C(S) \setminus S$. Then by the theorem above a distance d stabilizer code can fix errors on any $\lfloor (d-1)/2 \rfloor$ qubits. We will say V_S is an $[n, k, d]$ quantum code if V_S is a 2^k dimensional subspace of \mathcal{H}_n with distance d .

2.4 Check matrix representation

Define a function $r: G_n \rightarrow \mathbb{F}_2^{2n}$ by writing g as a string of X ’s times a string of Z ’s, forgetting the constant out the front, and then writing those strings in binary. For example $g = XIY = i(XIX)(IIZ)$ and $r(g) = 101001$.

Notice $r(gg') = r(g) + r(g')$ and that $\ker r = \{\pm I, \pm iI\}$. Also for any string in \mathbb{F}_2^{2n} we can find an operator that maps to it. Thus $G_n / \{\pm I, \pm iI\} \cong \mathbb{F}_2^{2n}$ (an isomorphism of a multiplicative group with the additive group of \mathbb{F}_2^{2n}). Define the *effective Pauli group* to be $\overline{G}_n = G_n / \{\pm I, \pm iI\}$. We will choose coset representatives of \overline{G}_n of the form

$$\bigotimes_{i=1}^n A_i$$

where each $A_i \in \{I, X, Y, Z\}$,

Now if S is a stabilizer group then since $-I \notin S$ we have no pairs of elements differing by $-I$ or $\pm iI$ so $S \cong r(S)$.

Define

$$\Lambda = \begin{bmatrix} 0 & I_n \\ I_n & 0 \end{bmatrix}.$$

Now write $A = [x(g) \ z(g)]$ and define the *twisted product*

$$r(g) \odot r(g') = r(g)\Lambda r(g')^T = x(g) \cdot z(g') + z(g) \cdot x(g').$$

Then we claim that g and g' commute if and only if $r(g) \odot r(g') = 0$. To see this, note that wherever g and g' have different non-identity matrices in a position, we get exactly one one in the sum for the twisted product.

The generators for $S = \langle g_1, \dots, g_m \rangle$ are said to be *irredundant* if removing any generator makes the group smaller.

Proposition 2.5. *The generators g_i of a stabilizer group are irredundant if and only if the collection of $r(g_i)$ are linearly independent.*

Proof. The rows are linearly independent over \mathbb{F}_2 if and only if $\sum a_i r(g_i) = 0$ with $a_j = 1$ for some j . But $\sum a_i r(g_i) = 0$ if and only if $\prod g_i^{a_i} \in \{\pm I, \pm iI\}$. But this can only be I because it is in S and $-I \notin S$. So some $a_j = 1$ if and only if $g_j = \prod_{i \neq j} g_i^{a_i}$ i.e. g_j is dependent on the other generators. \square

So, if we create a matrix of rows which have twisted product zero and which are linearly independent then that defines a stabilizer group (choose the g_i corresponding to the r_i to be the one with $c = 1$). If we write the matrix as

$$A = [A_1 \ A_2]$$

then the condition that all pairs of rows have twisted product zero becomes $A_1 A_2^T + A_2 A_1^T = 0$.

To decode in the check matrix representation we need to find the syndrome of an error E . As discussed earlier this depends on whether E commutes with the generators g_i . Thus we need to take the twisted product of rows of the check matrix A with $r(E)$. In practice if we write $[z(E) \ x(E)]$ instead of $r(E) = [x(E) \ z(E)]$ then we can instead use the normal \mathbb{F}_2 matrix product. Once again, if the error has

unique syndrome then it can be decoded and also if errors with the same syndrome differ by elements of the stabilizer they can be corrected.

We now develop a formula for the dimension of a stabilizer code.

Lemma 2.6. *If $S = \langle g_1, \dots, g_m \rangle$ with irredundant generators and $-I \notin S$ then for each i there exists $h_i \in G_n$ such that $h_i g_j = (-1)^{\delta_{ij}} g_j h_i$ (i.e. h_i anti commutes with g_i and commutes with all other g_j).*

Proof. This is equivalent to solving $A[z(h_i) \quad x(h_i)] = e_i$ which can be done because A has linearly independent rows. \square

Now for $x \in \mathbb{F}_2^m$ define

$$h_x = \prod_{i=1}^m h_i^{x_i} \text{ and } E_x = \{|\psi\rangle : g_j |\psi\rangle = (-1)^{x_j} |\psi\rangle\}.$$

Notice that $h_x g_j = (-1)^{x_j} g_j h_x$.

Proposition 2.7. *If $S = \langle g_1, \dots, g_m \rangle$ with irredundant generators and $-I \notin S$ then $\dim V_S = 2^{n-m}$.*

Proof. The g_j are commuting and diagonalizable so we can simultaneously diagonalize them. This gives

$$\mathcal{H}_n = \bigoplus_{x \in \mathbb{F}_2^m} E_x.$$

Now we claim that $E_0 \cong E_x$ for any x . To see this we have the linear map $h_x : E_0 \rightarrow E_x$. This has the correct codomain because if $|\psi\rangle \in E_0$ then $g_j h_x |\psi\rangle = (-1)^{x_j} h_x |\psi\rangle$ so $h_x |\psi\rangle \in E_x$. There is also the linear map $h_x^\dagger : E_x \rightarrow E_0$ for which $|\psi\rangle \in E_x$ has $g_j h_x^\dagger |\psi\rangle = (-1)^{x_j} h_x^\dagger g_j |\psi\rangle = h_x^\dagger |\psi\rangle$ so $h_x^\dagger |\psi\rangle \in E_0$. These two maps are inverses of each other and thus we have $\dim(\mathcal{H}_n) = 2^m \dim E_0$. But $V_S = E_0$ so we conclude that $\dim V_S = 2^{n-m}$. \square

2.5 CSS codes

A Calderbank-Shor-Steane (CSS) code is a stabilizer code built out of two classical codes. We give a definition allowing not necessarily full rank matrices similar to the one in [TZ09].

Proposition 2.8. *Assume that parity check matrices H_X and H_Z define classical binary linear codes C_X and C_Z of length n and that $H_X H_Z^T = 0$ (this is equivalent to $C_Z^\perp \subseteq C_X$ which is equivalent to $C_X^\perp \subseteq C_Z$). Then the stabilizer code with binary check matrix*

$$A = \begin{bmatrix} H_X & 0 \\ 0 & H_Z \end{bmatrix},$$

is a quantum $[n, k, d]$ code where

$$k = n - \dim(C_X^\perp) - \dim(C_Z^\perp)$$

and

$$d = \min\{\text{wt}(c) : c \in (C_Z \setminus C_X^\perp) \cup (C_X \setminus C_Z^\perp)\}.$$

Proof. We have $C_X = \ker(H_X)$ and $C_Z = \ker(H_Z)$ and thus $C_X^\perp = \text{im}(H_X^T)$ and $C_Z^\perp = \text{im}(H_Z^T)$. For the dimension of the code we have

$$\begin{aligned} k &= n - \# \text{ independent rows of } A \\ &= n - \dim(\text{im } H_Z^T) - \dim(\text{im } H_X^T) \\ &= n - \dim(C_X^\perp) - \dim(C_Z^\perp). \end{aligned}$$

Next, recall that the distance of a quantum stabilizer code is

$$d = \min\{\text{wt}(E) : E \in C(S) \setminus S\}.$$

Notice that

$$S = \langle X^a, Z^b : a \text{ is a row of } H_X, b \text{ is a row of } H_Z \rangle = \{X^a Z^b : a \in C_X^\perp, b \in C_Z^\perp\}$$

where X^a is notation for $X_1^{a_1} \cdots X_n^{a_n}$. Now for $X^a Z^b \in C(S)$ we need $X^a Z^b$ to commute with $X^{a'}$ for $a' \in C_X^\perp$. Notice $X^a Z^b X^{a'} = (-1)^{a' \cdot b} X^{a'} X^a Z^b$ so we must have $a' \cdot b = 0$ for all $a' \in C_X^\perp$ so $b \in C_X$. Similar arguments show $a \in C_Z$ so, ignoring global phase factors,

$$C(S) = \{X^a Z^b : a \in C_Z, b \in C_X\}.$$

For each $X^a Z^b$ at least one of X^a and Z^b has lesser weight. Thus there exists a minimum weight vector in $C(S) \setminus S$ of the form X^a or Z^b so the formula in the proposition follows. \square

2.6 An example of a stabilizer code

We describe Steane's 7 qubit code. Begin with the classical $[7, 4]$ Hamming code C specified by the parity check matrix

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

This code can correct one error because the syndromes of the 8 weight-zero and weight-one errors are distinct. The code also has $C^\perp \subseteq C$: the codewords in C^\perp are linear combinations of rows of H but the rows of H are all orthogonal to all rows of H . Define a CSS code by $H_X = H_Z = H$. Then this is an $[n, k, d]$ code with $n = 7$ and $k = 7 - 3 - 3 = 1$. To find $d = \min\{\text{wt}(c) : c \in C \setminus C^\perp\}$, note that there are no elements of weight less than or equal 2 because no column or sum of two columns of H is equal to 0. Furthermore the sum of the first three columns is 0 so $1110000 \in C$ and this codeword is not in C^\perp because no sum of rows of H is equal to it. Thus $d = 3$ and we have a $[7, 1, 3]$ CSS code.

CHAPTER 3

MAP-HOMOLOGY CODES

In this chapter we give an exposition of some code construction techniques from the point of view of homology. We begin with the most general case.

3.1 Codes from \mathbb{F}_2 -chain complexes

Let $(C_\bullet, \partial_\bullet)$ be a chain complex of finite dimensional \mathbb{F}_2 -vector spaces

$$\cdots \rightarrow C_{i+1} \xrightarrow{\partial_{i+1}} C_i \xrightarrow{\partial_i} C_{i-1} \rightarrow \cdots$$

(of course this means that $\partial_i \circ \partial_{i+1} = 0$) and as usual define the homology vector spaces $H_i = \ker(\partial_i) / \text{im}(\partial_{i+1})$.

We fix bases of the C_i and use $[\cdot]$ to denote matrices with respect to these bases. Let $H_X = [\partial_i]$, a $\dim C_{i-1} \times \dim C_i$ matrix, and $H_Z = [\partial_{i+1}]^T$, so H_Z is a $\dim C_{i+1} \times \dim C_i$ matrix. Then $H_X H_Z^T = [\partial_i \circ \partial_{i+1}] = 0$ so we can form the $[n, k, d]$ CSS code with binary check matrix

$$A = \begin{bmatrix} H_X & 0 \\ 0 & H_Z \end{bmatrix}.$$

We have $n = \dim C_i$ and also notice $H_i = \ker(H_X) / \text{im}(H_X^T) = C_X / C_X^\perp$. From our discussion of classical dual codes we have $\dim(C_X) + \dim(C_X^\perp) = \dim(C_i)$. This gives

$$\begin{aligned} k &= \dim(C_i) - \dim(C_X^\perp) - \dim(C_Z^\perp) \\ &= \dim(C_X) - \dim(C_Z^\perp) \\ &= \dim(C_X / C_Z^\perp) \\ &= \dim(H_i). \end{aligned}$$

To understand the set $C_Z \setminus C_X^\perp$ we can consider the dual complex. We now justify the intuitively reasonable idea that taking transposes of H_X and H_Z^T leads to cohomology.

Dualizing the chain complex

$$\cdots \rightarrow C_{i+1} \xrightarrow{\partial_{i+1}} C_i \xrightarrow{\partial_i} C_{i-1} \rightarrow \cdots$$

gives

$$\cdots \leftarrow C_{i+1}^* \xleftarrow{\partial_{i+1}^*} C_i^* \xleftarrow{\partial_i^*} C_{i-1}^* \leftarrow \cdots$$

where $C_i^* = \text{Hom}(C_i, \mathbb{F}_2)$ and $\partial_i^*(\phi)$ is defined by $\partial_i^*(\phi)(v) = \phi(\partial_i(v))$. To check that the dual complex really is a complex we have

$$(\partial_{i+1}^* \circ \partial_i^*)(\phi)(v) = \partial_{i+1}^*(\partial_i^*(\phi))(v) = \partial_{i+1}^*(\phi)(\partial_{i+1}(v)) = \phi(\partial_i(\partial_{i+1}(v))) = 0.$$

Having fixed a basis for each C_i we have a non-degenerate bilinear form $\langle \cdot, \cdot \rangle: C_i \times C_i \rightarrow \mathbb{F}_2$ given by $\langle a, b \rangle = [a] \cdot [b]$. Then $C_i \cong C_i^*$ by $\psi_i: a \mapsto \langle a, \cdot \rangle$. Now define $\delta_i = \psi_i^{-1} \circ \partial_i^* \circ \psi_{i-1}$ and then since

$$\delta_{i+1} \circ \delta_i = \psi_{i+1}^{-1} \circ \partial_{i+1}^* \circ \psi_i \circ \psi_i^{-1} \circ \partial_i^* \circ \psi_{i-1} = \psi_{i+1}^{-1} \circ \partial_{i+1}^* \circ \partial_i^* \circ \psi_{i-1} = 0$$

we have a chain complex

$$\cdots \leftarrow C_{i+1} \xleftarrow{\delta_{i+1}} C_i \xleftarrow{\delta_i} C_{i-1} \leftarrow \cdots$$

The definition of δ_i gives us $\psi_i(\delta_i(v)) = \partial_i^*(\psi_{i-1}(v))$ for all v which implies $\langle \delta_i(v), w \rangle = \langle v, \partial_i(w) \rangle$ for all v, w . In particular if C_i has basis $v_1, \dots, v_{\dim C_i}$ and C_{i+1} has basis $w_1, \dots, w_{\dim C_{i+1}}$ then we have $\langle \delta_i(v_j), w_k \rangle = \langle v_j, \partial_i(w_k) \rangle$. For the right hand side of this equality we have

$$\langle v_j, \partial_i(w_k) \rangle = [v_j] \cdot [\partial_i(w_k)] = e_j \cdot [\partial_i]e_k = [\partial_i]_{j,k}$$

and similarly $\langle \delta_i(v_j), w_k \rangle = [\delta_i]_{k,j}$ which gives us $[\delta_i] = [\partial_i]^T$. With our earlier definitions $H_X = [\partial_i]$ and $H_Z = [\partial_{i+1}]^T$ we then have $[\delta_i] = H_X^T$ and $[\delta_{i+1}] = H_Z$.

Define the cohomology vector space H^i by

$$H^i = \ker(\delta_{i+1}) / \text{im}(\delta_i) = \ker(H_Z) / \text{im}(H_X^T) = C_Z / C_X^\perp.$$

Once again we have

$$\begin{aligned}
 k &= \dim(C_i) - \dim(C_X^\perp) - \dim(C_Z^\perp) \\
 &= \dim(C_Z) - \dim(C_X^\perp) \\
 &= \dim(C_Z/C_X^\perp) \\
 &= \dim(H^i).
 \end{aligned}$$

Recall that $d = \min\{\text{wt}(c) : c \in (C_X \setminus C_Z^\perp) \cup (C_Z \setminus C_X^\perp)\}$ so the minimum weight can be found by looking at elements of C_X and C_Z whose classes are not zero in H_i and H^i respectively.

3.2 Codes from graphs on surfaces

The idea of creating CSS codes from graphs embedded on surfaces has been discussed in a number of papers. See for example [Zém09] for an introduction to this and see [LZZ04] for an introduction to graphs embedded in surfaces. Let Σ be a compact, connected, oriented surface (i.e. 2-manifold) with genus g . We consider cellular homology with coefficients in \mathbb{F}_2 (see [Hat02] for an introduction to algebraic topology). It is well known that

$$H_k(\Sigma) \cong \begin{cases} \mathbb{F}_2 & \text{if } k = 0 \text{ or } 2 \\ (\mathbb{F}_2)^{2g} & \text{if } k = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Define an *embedding* of an undirected (simple) graph G in Σ to be a function $G \rightarrow \Sigma$ that takes vertices of G to distinct points in Σ and edges in G to simple paths in Σ (i.e. images of injective continuous functions $[0, 1] \rightarrow \Sigma$) that intersect only at common vertices. We will denote such an embedding by (Σ, G) . Define a *face* of the embedding to be a maximal connected subset of Σ that does not intersect G . A *2-cell embedding* is one where all the faces are homeomorphic to open disks; we shall only consider 2-cell embeddings. A 2-cell embedding is also called a *map*.

Notice that a 2-cell embedding is actually a 2-dimensional CW complex and thus we have the standard cellular homology with \mathbb{F}_2 coefficients as follows. Let V, E, F be sets of vertices, edges and faces of (Σ, G) and $\mathcal{V}, \mathcal{E}, \mathcal{F}$ be \mathbb{F}_2 -vector spaces with bases V, E, F respectively. Then we have a chain complex

$$0 \rightarrow \mathcal{F} \xrightarrow{\partial_2} \mathcal{E} \xrightarrow{\partial_1} \mathcal{V} \rightarrow 0$$

where ∂_2 takes a face to the sum of the edges around that face and ∂_1 takes an edge to the sum of the vertices adjacent to that edge. We can then check that $\partial_1 \circ \partial_2$ takes a face to twice the sum of the vertices around the face i.e. zero.

So we have $H_1((\Sigma, G)) = \ker \partial_1 / \text{im } \partial_2$ and by homotopy invariance we know that $H_1((\Sigma, G)) \cong H_1(\Sigma) \cong (\mathbb{F}_2)^{2g}$. Create a code from the chain complex as discussed in Section 3.1. This gives $H_X = [\partial_1]$, a $|V| \times |E|$ matrix that we can think of as the (unsigned) vertex-edge incidence matrix. Similarly $H_Z = [\partial_2]^T$ is an $|F| \times |E|$ matrix, the face-edge incidence matrix.

We now discuss the way we will understand $C_X \setminus C_Z^\perp$ in this context. Define the (Poincaré) dual of an embedded graph (Σ, G) to be the embedded graph (Σ, G^*) with:

- One vertex of G^* inside each face of G ,
- For each edge e of G there is an edge e^* of G^* between the two vertices of G^* corresponding to the two faces of G adjacent to e .

Then the faces of G^* correspond to the vertices of G . To see this, notice that a vertex of G is adjacent to a set of edges of G and the corresponding edges of G^* form the boundary of a face.

So identifying the set of edges in G^* with the edges of G we can consider the homology chain complex for (Σ, G^*) :

$$0 \rightarrow \mathcal{V} \xrightarrow{\delta_1} \mathcal{E} \xrightarrow{\delta_2} \mathcal{F} \rightarrow 0$$

where $\delta_1(v)$ is the sum of edges of G adjacent to v and $\delta_2(e)$ is the sum of faces of G adjacent to e . This shows that $[\delta_1] = [\partial_1]^T = H_X^T$ and $[\delta_2] = [\partial_2]^T = H_Z$ so

$$H_1((\Sigma, G^*)) \cong \ker H_Z / \text{im } H_X^T = C_Z / C_X^\perp \cong H^1((\Sigma, G)).$$

Also (Σ, G^*) is a CW complex structure for Σ so $H_1((\Sigma, G^*)) \cong \mathbb{F}_2^{2g}$.

So from (Σ, G) we have constructed a CSS code with parameters $[n, k, d]$ where n is the number of edges of G , $k = 2g$ and

$$d = \min\{\text{wt}(c) : c \in (C_X \setminus C_Z^\perp) \cup (C_Z \setminus C_X^\perp)\}.$$

Elements of C_X are cycles of the graph G (equivalently subgraphs where each vertex is adjacent to an even number of edges) and similarly elements of C_Z are cycles of G^* . Elements of $C_X \setminus C_Z^\perp$ are cycles whose homology class is not zero i.e. non-boundary cycles. So d is the lowest weight (i.e. shortest) non-boundary cycle in G or G^* .

There is a way to understand the lengths of cycles of G^* just looking at G . A cycle in G^* corresponds to a collection of faces of G where each face is edge-adjacent to an even number of faces in the collection. The length of the cycle is exactly the number of edges of G that we cross as we traverse the cycle. Call this the *ladder distance* of the cycle.

Example 3.1 (Toric codes). Toric codes were suggested by Kitaev (see for example [DKLP02]). Fix a positive integer m and embed an $m \times m$ square grid G in the torus (see Figure 3.1 where $m = 4$ and we represent the torus by a square with left and right edges identified and top and bottom edges identified). Then G^* is isomorphic to G (see Figure 3.2 for the $m = 4$ case). We have $n = 2m^2$, the number of edges and $k = 2g = 2$. The minimum weight non-boundary cycles are the straight vertical and horizontal cycles of length m so $d = m$. Thus the toric code from the $m \times m$ grid is a $[n, k, d] = [2m^2, 2, m]$ code. From this we see $k = 2$ is constant, $d = \sqrt{\frac{1}{2}n}$ and thus $kd^2 = n$.

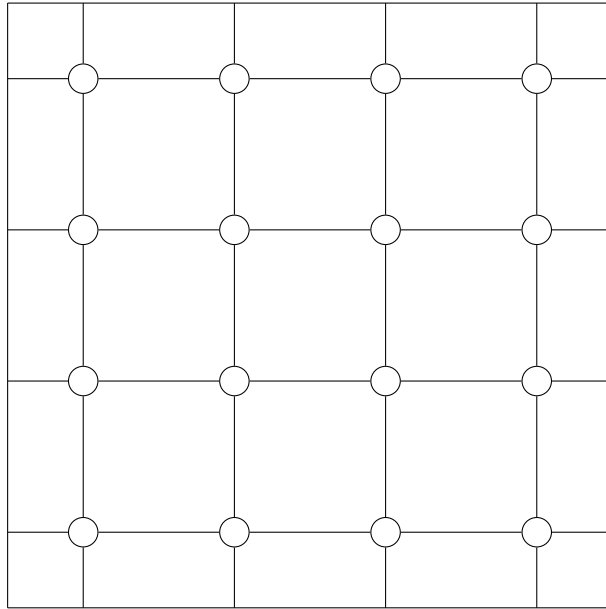


FIGURE 3.1. A toric code with $m = 4$.

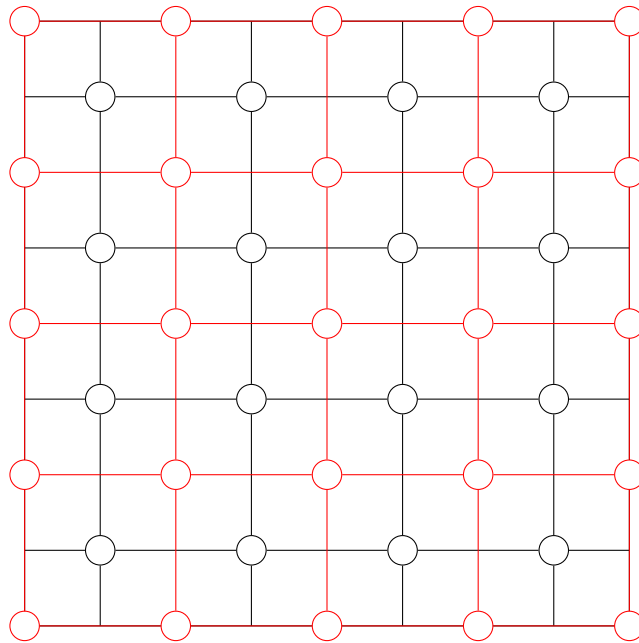


FIGURE 3.2. A toric code and its dual with $m = 4$. The dual is shown in red.

3.3 Planar codes

The following is an attempt to make more explicit discussion in [DKLP02] and [BMD07]. Planar codes have similar properties to the toric code but allow some choice of parameters and so we can use them to show the tradeoffs in code construction from homology of graphs. They also have the practically desirable property that they can be implemented in a plane rather than a torus.

Begin with a $m \times n$ grid of vertices with h ‘holes’ (edge-contiguous collections of faces together with interior edges and vertices) removed. Think of this as a CW complex Γ with 1-skeleton G and 2-cells for each of the non-removed squares in the grid attached to G by homeomorphic maps from S^1 to the boundary of the square. Each of these attaching maps have degree ± 1 . We will also define an embedding of $\tilde{G} = G$ into the sphere S^2 where we further attach a 2-cell to each of the removed holes and to the outer boundary of the grid.

This graph G has at most mn vertices and at most $(m-1)n + (n-1)m$ edges while Γ has at most $(m-1)(n-1)$ faces and \tilde{G} has at most $(m-1)(n-1) + 1$ faces. The graph \tilde{G} has a (Poincaré) dual graph \tilde{G}^* on the sphere. As discussed in Section 3.2 this gives us matrices \tilde{H}_X and \tilde{H}_Z and since $0 = H_1(S^2) = \tilde{C}_X / \tilde{C}_Z^\perp$ we have $\tilde{C}_X = \tilde{C}_Z^\perp$ and thus of course $\tilde{C}_X^\perp = \tilde{C}_Z$. In words this says all cycles of \tilde{G} and \tilde{G}^* are boundaries.

From the CW complex Γ we have an \mathbb{F}_2 -homology chain complex

$$0 \rightarrow \mathcal{F} \rightarrow \mathcal{E} \rightarrow \mathcal{V} \rightarrow 0$$

and as in Section 3.1 we construct matrices H_X and H_Z .

Note H_X is the vertex-edge incidence matrix of G so $H_X = \tilde{H}_X$ and H_Z is the face-edge incidence matrix of Γ so it can be formed from \tilde{H}_Z , the face-edge incidence matrix of \tilde{G} , by removing the rows corresponding to the removed faces. These observations imply that $\tilde{C}_X = C_X$ and $\tilde{C}_Z \subseteq C_Z$.

The planar grid with h holes removed is homotopy equivalent to a bouquet of h circles and thus has $H_1(\Gamma) = (\mathbb{F}_2)^h$ and so we have an $[N, k, d]$ code with $N \leq (m-1)n + (n-1)m$ and $k = h$.

To calculate d we notice that

$$\begin{aligned} C_X \setminus C_Z^\perp &= \tilde{C}_X \setminus C_Z^\perp \\ &= \tilde{C}_Z^\perp \setminus C_Z^\perp \\ &= \{\text{boundaries of } \tilde{G}\} \setminus \{\text{boundaries of } G\} \end{aligned}$$

and

$$\begin{aligned} C_Z \setminus C_X^\perp &= C_Z \setminus \tilde{C}_X^\perp \\ &= C_Z \setminus \tilde{C}_Z \\ &= \{\text{cycles of } G^*\} \setminus \{\text{cycles of } \tilde{G}^*\}. \end{aligned}$$

Thus a minimum weight element of $C_X \setminus C_Z^\perp$ is a boundary of a removed hole and a minimum weight element of $C_Z \setminus C_X^\perp$ is a cycle of G^* , which we can think of as a cycle of \tilde{G}^* except we don't have to check that there an even number of edges adjacent to the vertices corresponding to removed faces. So

$$d = \min\{\text{length of boundaries of removed faces, paths between holes or paths from holes to edge of grid}\}.$$

Example 3.2. We remove an $l \times l$ hole (including removing $(l-1)^2$ vertices and $2l(l-1)$ edges) from the middle of a square grid. See Figure 3.3 for an example with $l = 2$. The boundary of the hole is distance $4l$ so place the hole at a ladder distance $4l$ away from the edge. Then the grid has $m = n = 9l - 1$ so we have a code with $N = 2(9l-1)(9l-2) - 2l(l-1) \sim 160l^2$, $k = 1$ and $d = 4l$. So in this case $k = 1$ is constant and $d \sim \sqrt{0.1N}$ with $kd^2 \sim 0.1N$.

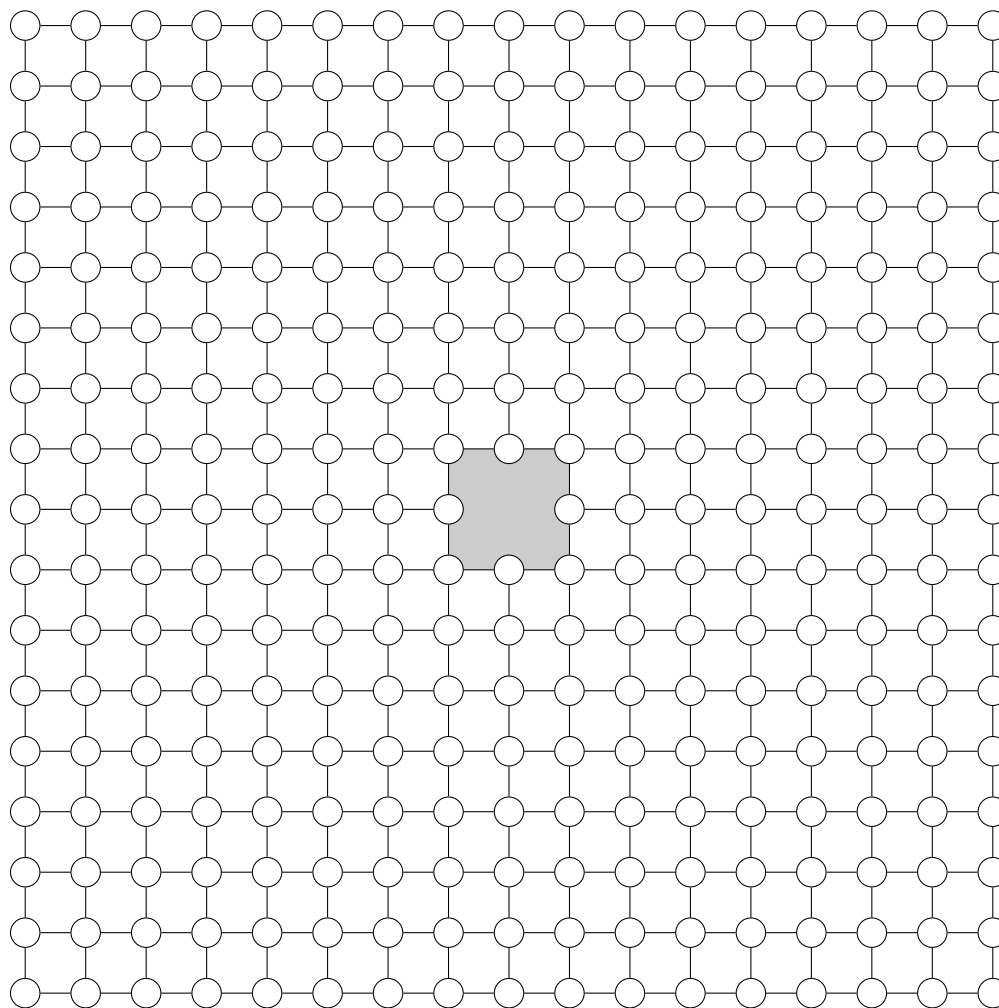


FIGURE 3.3. A planar code where we have removed one 2×2 square hole.

Example 3.3. At the opposite extreme, remove 1×1 holes arranged in a $l \times l$ square grid from a larger square grid, each at distance 4 away from each other. See Figure 3.4 for an example with $l = 2$. Now $m = n = 4(l + 1)$ so $N = 2(4l + 4)(4l + 3)$, $k = l^2$ and $d = 4$. So we get $k \sim \frac{1}{32}N$ with $d = 4$ constant and $kd^2 \sim 0.5N$.

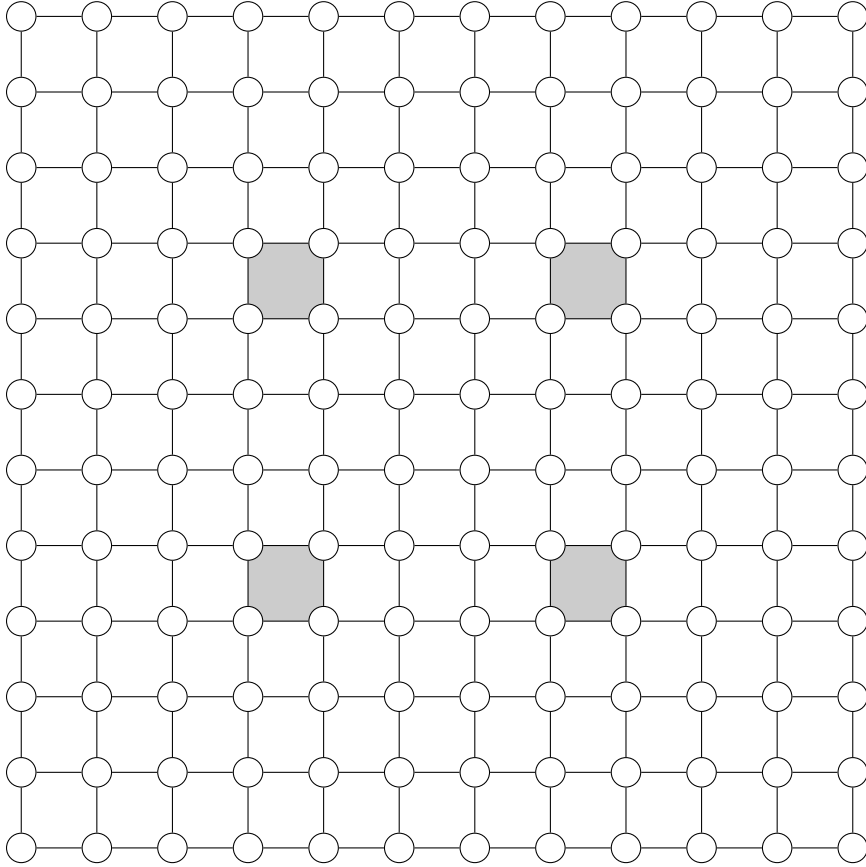


FIGURE 3.4. A planar code where we have removed a 2×2 grid of 1×1 square holes.

Example 3.4. Remove $l \times l$ holes arranged in $l \times l$ grid, each hole $4l$ ladder distance away from other holes and edges. See Figure 3.5 for an example with $l = 2$. Then we must have $m = n = (4l)(l + 1) + l = 4l^2 + 5l$. Then $N = 2(4l^2 + 5l)(4l^2 + 5l - 1) - l^2(2l(l - 1)) \sim 30l^4$, $k = l^2$ and $d = 4l$. So in this case $k \sim \sqrt{\frac{1}{30}N}$, $d \sim \sqrt[4]{\frac{128}{15}N}$ and $kd^2 \sim \frac{8}{15}N$.

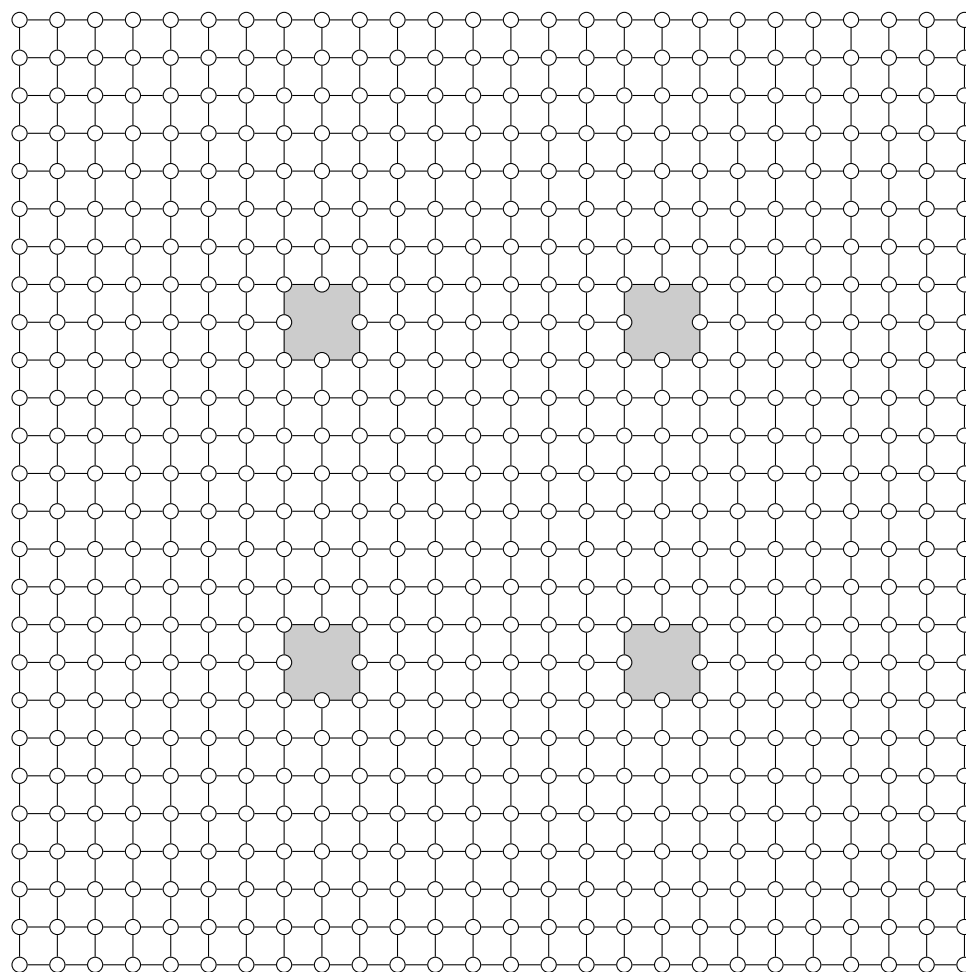


FIGURE 3.5. A planar code where we have removed a 2×2 grid of 2×2 square holes.

CHAPTER 4

HYPERMAP-HOMOLOGY CODES

4.1 Hypermaps

A hypergraph can be thought of as a generalization of a graph, where edges can be connected to more than two vertices. A hypermap is an embedding of a hypergraph in a surface. We will give definitions below of topological hypermaps that follow this intuition and of combinatorial hypermaps that are described instead by a pair of permutations. We will explain that they really are the same object and show how to go back and forth between them. Our discussion of hypermaps follows [CM92] with some differences, particularly in our choice of graphical representation. Note that, following this source, we multiply permutations left to right. We will however write the action of σ on i as $\sigma(i)$ so that with our convention we have $(\alpha\sigma)(i) = \sigma(\alpha(i))$.

Definition 4.1. A *hypergraph* with n darts is a pair of partitions V and E of $B = \{1, \dots, n\}$. We call the elements of V *vertices* and the elements of E *edges*. Noting that each *dart* (element of B) is in one vertex and one edge, we say that it is *incident* to that vertex and that edge. A hypergraph is *connected* if whenever a union of elements of V equals a union of elements of E then this union is empty or the whole of B .

For a graphical representation of a hypergraph we will use the bipartite graph representation (in the context of hypermaps this is usually called the Walsh representation, after [Wal75]). Represent vertices as circles, edges as squares and darts as line segments between the edge and vertex they are incident to. We will label darts by their number, written counterclockwise of the dart with respect to edges.

For us hypergraphs and hypermaps will always be considered to be labeled (darts,

vertices, edges and faces each have a label). We will usually not label vertices, edges and faces explicitly, instead labeling them by the set of labels of adjacent darts.

Notice that a hypergraph is connected if and only if its bipartite graph representation is connected (the definition for a hypergraph to be connected is equivalent to the bipartite graph having only one connected component).

Definition 4.2. A *topological (oriented) hypermap* is a 2-cell embedding of the bipartite graph representation of a connected hypergraph in a compact, connected, oriented surface.

Now this surface has a genus g and the bipartite graph embedding satisfies Euler's formula:

$$\# \text{ vertices} - \# \text{ edges} + \# \text{ faces} = 2 - 2g.$$

Let $|F|$ be the number of faces of the embedding of the bipartite graph. Then noting that the bipartite graph has $|V| + |E|$ vertices, n edges and $|F|$ faces we can derive an Euler formula for topological hypermaps:

$$|V| + |E| + |F| = n + 2 - 2g.$$

Definition 4.3. A *combinatorial (oriented) hypermap* is a pair (σ, α) with $\sigma, \alpha \in S_n$ such that $\langle \sigma, \alpha \rangle$ is transitive on B .

We now describe the process to go back and forth between topological and combinatorial hypermaps. Given a topological hypermap, we define σ and α by $\sigma(i)$ being the dart which is counterclockwise of dart i with respect to the vertex incident to i and $\alpha(i)$ being the dart which is clockwise of dart i with respect to the edge incident to i . To show $\langle \sigma, \alpha \rangle$ is transitive we note that since the hypergraph in the definition of a topological hypermap is connected there is a path from one dart to any other which we can take by using σ and α .

Now note that the orbits of σ are V and the orbits of α are E . Also notice $\alpha^{-1}\sigma$ goes clockwise around the interior of a face: start at a dart whose label is inside the

face, α^{-1} takes it counterclockwise around an edge, then σ takes $\alpha^{-1}(i)$ clockwise around a vertex. Thus the orbits of $\alpha^{-1}\sigma$ are the faces of the topological hypermap.

Permutation	Orbits are	Ordering
σ	vertices	counterclockwise
α	edges	clockwise
$\alpha^{-1}\sigma$	faces	clockwise
Labels are counterclockwise of darts w.r.t. rotation about edges.		

TABLE 4.1. Table of conventions for hypermaps.

To go from a combinatorial hypermap to a topological hypermap we form a polygon for each cycle of $\alpha^{-1}\sigma$. If the cycle is (i_1, \dots, i_m) then place the labels i_1, \dots, i_m inside the polygon on darts that go from vertex to edge as we travel clockwise. Outside the polygon place the label $\alpha^{-1}(i_j)$ on the dart counterclockwise of i_j with respect to rotation about edges.

Notice each label in B is on the inside of precisely one polygon (because $\alpha^{-1}\sigma$ is a partition of B) and on the outside of precisely one polygon (because α^{-1} is a permutation so takes the inside partition to a partition). So we can now specify a new topological space to be the polygons glued according to their labelings (i.e. the disjoint union of the polygons modulo identifying the corresponding edges).

Proposition 4.4. *The space so constructed is a compact connected oriented surface and the subspace given by the now-identified edges can be thought of as the bipartite graph representation of a connected hypergraph. This construction of a topological hypermap is an inverse process to the construction of combinatorial hypermaps given above.*

Proof. To see that the space is a surface (i.e. a 2-manifold) we check that each point has a neighborhood homeomorphic to \mathbb{R}^2 . This is certainly true inside each polygon, it is true on darts since we are gluing exactly two darts together and it is true at vertices and edges because at each one we have a cyclic order of darts around it. The

surface is compact because there is a finite number of polygons. The hypergraph (i.e. the 1-skeleton) is connected because $\langle \sigma, \alpha \rangle$ is transitive and this together with the interior of the polygons being connected gives us that the surface is connected. The surface is oriented because each polygon has the clockwise orientation and so when we glue together polygons at darts, those darts have opposite orientations. The second sentence follows by construction. \square

Example 4.5. Consider the square below to be a torus (identify boundary: top with bottom and left with right) then interpret it as the Walsh representation of a topological hypermap.

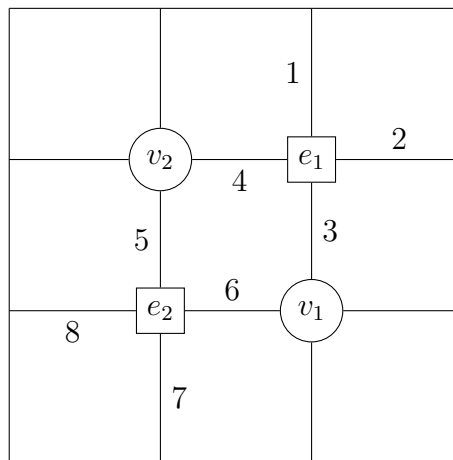


FIGURE 4.1. A topological hypermap.

The combinatorial hypermap coming from it is (σ, α) with

$$\sigma = (1\ 8\ 3\ 6)(2\ 5\ 4\ 7), \quad \alpha = (1\ 2\ 3\ 4)(5\ 6\ 7\ 8)$$

and from this we can calculate

$$\alpha^{-1}\sigma = (1\ 7)(2\ 8)(3\ 5)(4\ 6).$$

4.2 The dual hypermap

The following definition can be found in [CP80].

Definition 4.6. The *dual* of a combinatorial hypermap (σ, α) is

$$(\sigma', \alpha') = (\alpha^{-1}\sigma, \alpha^{-1}).$$

We can see that $(\alpha')^{-1}\sigma' = (\alpha^{-1})^{-1}\alpha^{-1}\sigma = \sigma$ so duality switches vertices and faces while fixing edges (the orbits of α^{-1} are the same as the orbits of α). Also $\sigma'' = (\alpha')^{-1}\sigma' = \sigma$ and $\alpha'' = (\alpha^{-1})^{-1} = \alpha$ so the dual of the dual is the original hypermap.

Our definition of the dual of a topological hypermap is similar to the one in [Maz11].

Definition 4.7. A *dual* of a topological hypermap $H = (\Sigma, \Gamma)$ is a topological hypermap $H^* = (\Sigma_{\text{op}}, \Gamma^*)$ which we will now describe.

1. Σ_{op} is the surface Σ with the opposite orientation.
2. The edges of H^* are the edges of H .
3. There is precisely one vertex of H^* for each face of H , inside that face.
4. The darts of H^* go from vertices of H^* to edges around the corresponding face of H .
5. To label the darts of H^* we draw the primal hypergraph in black and the dual hypergraph in red. Also draw a dotted line between red vertices and black vertices (this gives the *canonical triangulation* of [LZZ04]). Then each black label is inside one triangle: copy that label to the solid red line without leaving the triangle.

We will not discuss isomorphisms of combinatorial or topological hypermaps (see for example [LZZ04]). But we do need a stronger notion of isomorphism of topological hypermaps that corresponds to equality of combinatorial hypermaps.

Definition 4.8. We say topological hypermaps $H = (\Sigma, \Gamma)$ and $H' = (\Sigma', \Gamma')$ are *strongly isomorphic*, and write $H = H'$, if there exists an orientation-preserving homeomorphism $u: \Sigma \rightarrow \Sigma'$ with $u|_{\Gamma}$ giving an equality of hypergraphs.

Proposition 4.9. *The dual of a topological hypermap is unique up to strong isomorphism and $(H^*)^* = H$.*

Proof. If $H^* = (\Sigma_{\text{op}}, \Gamma^*)$ and $H^{*'} = (\Sigma_{\text{op}}, \Gamma^{*'})$ are both dual hypermaps of H then we can see pictorially that there exists an orientation-preserving homeomorphism $u: \Sigma_{\text{op}} \rightarrow \Sigma_{\text{op}}$ with vertices and darts of H^* taken to vertices and darts of $H^{*'}$. Since the labels are specified by the definition of a dual this is enough to give $H^* = H^{*'}$.

Now we check that H is a dual of H^* and thus this uniqueness gives us $(H^*)^* = H$. □

Proposition 4.10. *The combinatorial hypermap corresponding to the topological dual of a hypermap is equal to the combinatorial dual of the hypermap.*

Proof. It is important to remember that the dual has the opposite orientation so all the conventions in Table 4.1 are reversed. Say we begin with a combinatorial hypermap $H = (\sigma, \alpha)$ which of course has combinatorial dual $H^* = (\alpha^{-1}\sigma, \alpha^{-1})$. Consider a cycle (i_1, \dots, i_n) of $\alpha^{-1}\sigma$. This is a face of H and so gives us a vertex of the topological dual with the darts i_1, \dots, i_n clockwise around it. Our usual convention is to take vertices counterclockwise but on the dual graph we go clockwise so the vertices of H^* give us exactly the permutation $\alpha^{-1}\sigma$. Similarly, cycles of α are edges of H counterclockwise and if we take them in H^* clockwise then we get the permutation α^{-1} as desired. □

Example 4.11. If we start with a hypergraph embedded on a torus as pictured in

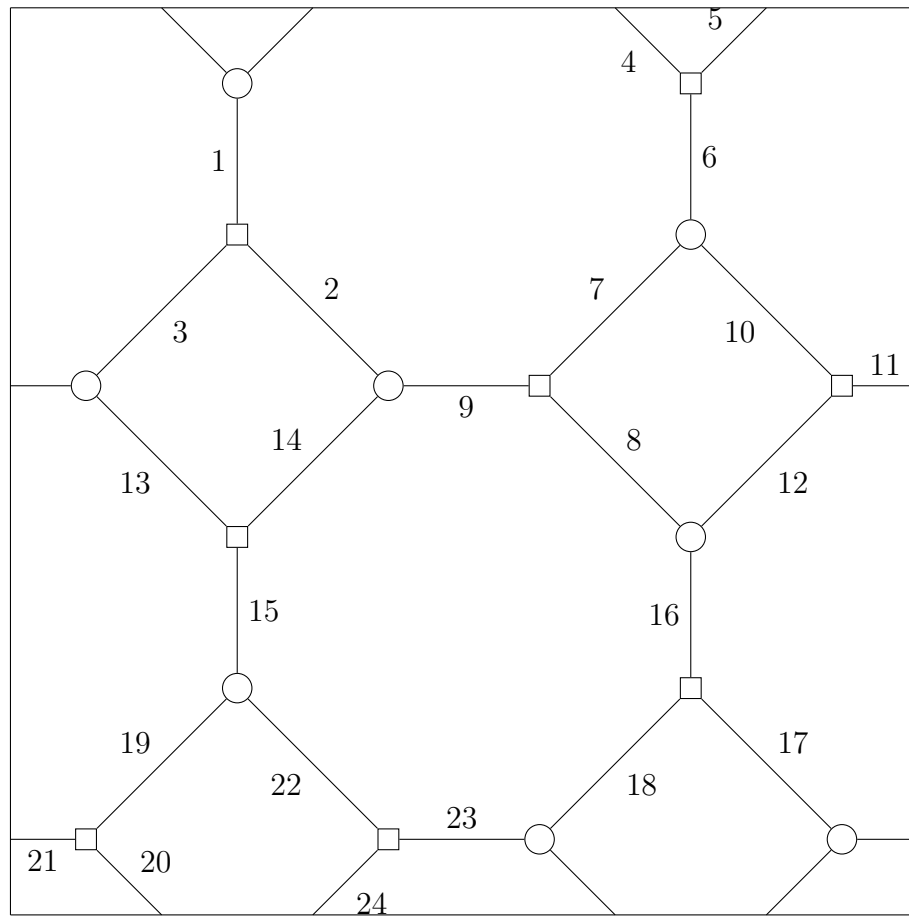


FIGURE 4.2. A hypermap with four square and four octagon faces.

Figure 4.2 then we have a hypermap with

$$\begin{aligned}\sigma &= (1\ 24\ 20)(2\ 14\ 9)(3\ 11\ 13)(4\ 18\ 23)(5\ 21\ 17)(6\ 7\ 10)(8\ 16\ 12)(15\ 19\ 22), \\ \alpha &= (1\ 2\ 3)(4\ 5\ 6)(7\ 8\ 9)(10\ 11\ 12)(13\ 14\ 15)(16\ 17\ 18)(19\ 20\ 21)(22\ 23\ 24), \text{ and} \\ \alpha^{-1}\sigma &= (1\ 11\ 6\ 21)(2\ 24\ 4\ 7)(3\ 14)(5\ 18)(8\ 10)(9\ 16\ 23\ 15)(12\ 13\ 19\ 17)(20\ 22).\end{aligned}$$

Then, using $(\sigma', \alpha') = (\alpha^{-1}\sigma, \alpha^{-1})$, we see the combinatorial dual has

$$\begin{aligned}\sigma' &= (1\ 11\ 6\ 21)(2\ 24\ 4\ 7)(3\ 14)(5\ 18)(8\ 10)(9\ 16\ 23\ 15)(12\ 13\ 19\ 17)(20\ 22) \\ \alpha' &= (1\ 3\ 2)(4\ 6\ 5)(7\ 9\ 8)(10\ 12\ 11)(13\ 15\ 14)(16\ 18\ 17)(19\ 21\ 20)(22\ 24\ 23), \text{ and} \\ (\alpha')^{-1}\sigma' &= (1\ 24\ 20)(2\ 14\ 9)(3\ 11\ 13)(4\ 18\ 23)(5\ 21\ 17)(6\ 7\ 10)(8\ 16\ 12)(15\ 19\ 22).\end{aligned}$$

In particular $(\alpha')^{-1}\sigma' = \sigma$. Notice that this is the same as the topological dual in Figure 4.3 as long as we orient the surface in the opposite way i.e. cycle around vertices clockwise, edges counterclockwise and faces counterclockwise.

4.3 Hypermap homology

Our theory of homology of hypermaps comes from [CM92]. Our main difference is that we are working over \mathbb{F}_2 -vector spaces instead of over \mathbb{Z} -modules.

Let $\mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{W}$ be \mathbb{F}_2 -vector spaces with bases V, E, F, W respectively where $W = \{w_1, \dots, w_n\}$. Define $d_2: \mathcal{F} \rightarrow \mathcal{W}$ by $d_2(f) = \sum_{i \in f} w_i$ extended linearly and $d_1: \mathcal{W} \rightarrow \mathcal{V}$ by $d_1(w_i) = v_{\ni i} + v_{\ni \alpha^{-1}(i)}$. Here we use the notation $v_{\ni i}$ for the element of V containing i (there exists a unique such vertex because V is a partition of B).

Also define $\iota: \mathcal{E} \rightarrow \mathcal{W}$ by $\iota(e) = \sum_{i \in e} w_i$ extended linearly. Notice ι is an injection (d_2 is also), and then define a projection map $p: \mathcal{W} \rightarrow \mathcal{W}/\iota(\mathcal{E})$.

Proposition 4.12. *We have $d_1 \circ d_2 = 0$ and $d_1 \circ \iota = 0$.*

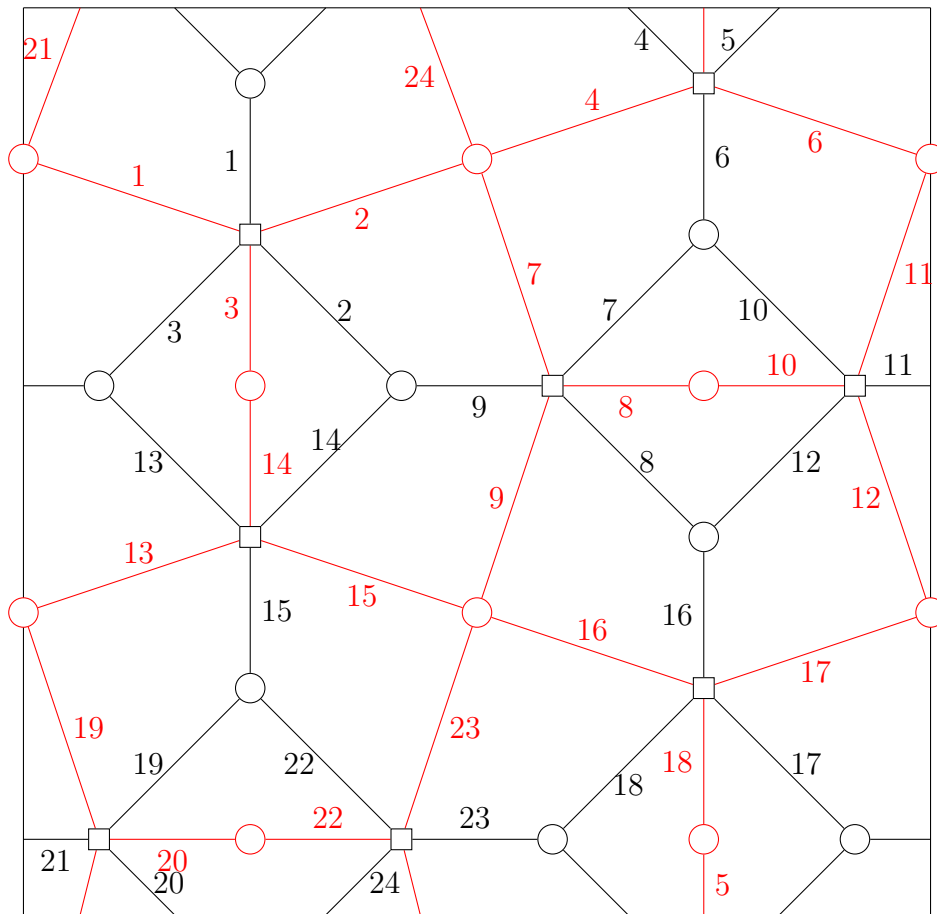


FIGURE 4.3. The octagon-square hypermap with its dual drawn in red.

Proof. We have $(d_1 \circ d_2)(f) = d_1(\sum_{i \in f} w_i) = \sum_{i \in f} v_{\ni i} + \sum_{i \in f} v_{\ni \alpha^{-1}(i)}$. Now both of these sums are the vertices around the face f so extending linearly $d_1 \circ d_2 = 0$. Similarly $(d_1 \circ \iota)(e) = d_1(\sum_{i \in e} w_i) = \sum_{i \in e} v_{\ni i} + \sum_{i \in e} v_{\ni \alpha^{-1}(i)}$ and both of these sums are the vertices adjacent to e . \square

From this we can define $\partial_1: \mathcal{W}/\iota(\mathcal{E}) \rightarrow \mathcal{V}$ by $\partial_1(w + \iota(\mathcal{E})) = d_1(w)$ and $\partial_2 = p \circ d_2$. Then we have the commutative diagram below.

$$\begin{array}{ccccc} \mathcal{F} & \xrightarrow{d_2} & \mathcal{W} & \xrightarrow{d_1} & \mathcal{V} \\ & \searrow \partial_2 & \downarrow p & \nearrow \partial_1 & \\ & & \mathcal{W}/\iota(\mathcal{E}) & & \end{array}$$

From this we see $\partial_1 \circ \partial_2 = d_1 \circ d_2 = 0$. So we have a chain complex

$$\mathcal{F} \xrightarrow{\partial_2} \mathcal{W}/\iota(\mathcal{E}) \xrightarrow{\partial_1} \mathcal{V}$$

$$V_S = \bigcap_{s \in S} V_{\{s\}}.$$

First we see that if \mathcal{V} has basis $v_1, \dots, v_{|V|}$ then $\text{im } \partial_1$ has basis $v_1 - v_j$ for $j = 2, \dots, |V|$. This set is linearly independent because the v_j are a basis of \mathcal{V} and it spans $\text{im } \partial_1$ because $\partial_1(w_i) = v_{\ni i} + v_{\ni \alpha^{-1}(i)} = (v_1 + v_{\ni \alpha^{-1}(i)}) + (v_1 + v_{\ni i})$. Thus $\dim(\text{im } \partial_1) = |V| - 1$ and $\dim H_0 = 1$.

Next $\ker \partial_2 = \{f \in \mathcal{F} : d_2(f) \in \iota(\mathcal{E})\}$ which has the same dimension as $d_2(\mathcal{F}) \cap \iota(\mathcal{E})$ because d_2 is injective. But an element in this intersection corresponds to both a union of cycles of $\alpha^{-1}\sigma$ and a union of cycles of α . But such a union must be empty or the whole of B and so $\dim H_2 = \dim(\ker \partial_2) = 1$.

From this we have $\dim(\text{im } \partial_2) = \dim \mathcal{F} - \dim(\ker \partial_2) = |F| - 1$ and $\dim(\ker \partial_1) = \dim(\mathcal{W}/\iota(\mathcal{E})) - \dim(\text{im } \partial_1) = |W| - |E| - (|V| - 1) = n - |V| - |E| + 1$. So then $\dim H_1 = \dim(\ker \partial_1) - \dim(\text{im } \partial_2) = (n - |V| - |E| + 1) - (|F| - 1) = n + 2 - |V| - |E| - |F| = 2g$.

The following well known lemma allows us to create a homomorphism of homology groups induced from a homomorphism of chains.

Lemma 4.13. *If we have a commutative diagram of vector spaces and vector space homomorphisms (i.e. linear functions)*

$$\begin{array}{ccccc} A_2 & \xrightarrow{f_2} & A_1 & \xrightarrow{f_1} & A_0 \\ \downarrow \psi_2 & & \downarrow \psi_1 & & \downarrow \psi_0 \\ B_2 & \xrightarrow{g_2} & B_1 & \xrightarrow{g_1} & B_0 \end{array}$$

with $f_1 \circ f_2 = 0$ and $g_1 \circ g_2 = 0$ then $\psi_1(\text{im } f_2) \subseteq \text{im } g_2$, $\psi_1(\ker f_1) \subseteq \ker g_1$ which gives us a homomorphism

$$(\psi_1)_* : \frac{\ker f_1}{\text{im } f_2} \rightarrow \frac{\ker g_1}{\text{im } g_2}.$$

Proof. An element of $\psi_1(\text{im } f_2)$ is of the form $\psi_1(f_2(a_2)) = g_2(\psi_2(a_2)) \in \text{im } g_2$. Next an element of $\psi_1(\ker f_1)$ is of the form $\psi_1(a_1)$ with $f_1(a_1) = 0$. But then $g_1(\psi_1(a_1)) = \psi_0(f_1(a_1)) = \psi_0(0) = 0$ so $\psi_1(a_1) \in \ker g_1$.

Then we can define $(\psi_1)_*(a_1 + \text{im } f_2) = \psi_1(a_1) + \text{im } g_2$ and the facts above give us that this function is well defined and has an allowable codomain. Linearity follows from the linearity of ψ_1 . \square

The next lemma gives us a condition on the above diagram that lead to an isomorphism of homology.

Lemma 4.14. *If we have the setup in Lemma 4.13 and furthermore we have*

$$\psi_1(\text{im } f_2) = \text{im } g_2,$$

$$\psi_1(\ker f_1) = \ker g_1, \text{ and}$$

$$\ker \psi_1 \cap \ker f_1 \subseteq \text{im } f_2$$

then $(\psi_1)_*$ is an isomorphism.

Proof. We have $(\psi_1)_*$ a surjection because ψ_1 is a surjection from $\ker f_1$ to $\ker g_1$. To see $(\psi_1)_*$ is injective, if $(\psi_1)_*(a_1 + \text{im } f_2) = 0$ for $a_1 \in \ker f_1$ then $\psi_1(a_1) \in \text{im } g_2 = \psi_1(\text{im } f_2)$ so $\psi_1(a_1) = \psi_1(f_2(a_2))$ and thus $\psi_1(a_1 - f_2(a_2)) = 0$ and also

$f_1(a_1 - f_2(a_2)) = 0$ so $a_1 - f_2(a_2) \in \ker \psi_1 \cap \ker f_1 \subseteq \text{im } f_2$ so $a_1 - f_2(a_2) = f_2(a'_2)$ so $a_1 = f_2(a_2 + a'_2) \in \text{im } f_2$ as required. \square

The following proposition gives us another way of looking at hypermap-homology which may be informative.

Proposition 4.15.

$$H_1 = \frac{\ker \partial_1}{\text{im } \partial_2} \cong \frac{\ker d_1}{\text{im } d_2 + \iota(\mathcal{E})}.$$

Proof. Consider the following diagram.

$$\begin{array}{ccccc} \mathcal{F} \oplus \mathcal{E} & \xrightarrow{d_2 \oplus \iota} & \mathcal{W} & \xrightarrow{d_1} & \mathcal{V} \\ \downarrow \pi_{\mathcal{F}} & & \downarrow p & & \downarrow id \\ \mathcal{F} & \xrightarrow{\partial_2} & \mathcal{W}/\iota(\mathcal{E}) & \xrightarrow{\partial_1} & \mathcal{V} \end{array}$$

This diagram is commutative because $p((d_2 \oplus \iota)(f + e) = p(d_2(f)) = \partial_2(f) = \partial_2(\pi_{\mathcal{F}}(f + e))$ and $\partial_1(p(w)) = d_1(w)$. Next note that

$$\text{im } \partial_2 = \text{im}(p \circ d_2) = p(\text{im } d_2) = p(\text{im } d_2 \oplus \iota)$$

and using the fact that p is surjective,

$$\ker \partial_1 = \{p(w) : \partial_1(p(w)) = 0\} = \{p(w) : d_1(w) = 0\} = p(\ker d_1).$$

Finally, $\ker p = \iota(\mathcal{E}) \subseteq \text{im}(d_2 \oplus \iota)$ so by Lemma 4.14 we have p_* an isomorphism. \square

Next we consider how to relate hypermap-homology to the homology we can get by considering the embedded bipartite graph representation of the hypermap. The chain complex of this ‘classical homology’ is

$$\mathcal{F} \xrightarrow{\bar{d}_2} \mathcal{W} \xrightarrow{\bar{d}_1} \mathcal{V} \oplus \mathcal{E}$$

where \bar{d}_2 and \bar{d}_1 are defined by $\bar{d}_2(f) = \sum_{i \in f} w_i + w_{\alpha^{-1}(i)}$ and $\bar{d}_1(w_i) = v_{\ni i} + e_{\ni i}$ extended linearly.

Proposition 4.16.

$$H_1 = \frac{\ker \partial_1}{\text{im } \partial_2} \cong \frac{\ker \bar{d}_1}{\text{im } \bar{d}_2}.$$

Proof. Consider the following diagram

$$\begin{array}{ccccc} \mathcal{F} & \xrightarrow{\partial_2} & \mathcal{W}/\iota(\mathcal{E}) & \xrightarrow{\partial_1} & \mathcal{V} \\ \downarrow id & & \downarrow \mu & & \downarrow i_{\mathcal{V}} \\ \mathcal{F} & \xrightarrow{\bar{d}_2} & \mathcal{W} & \xrightarrow{\bar{d}_1} & \mathcal{V} \oplus \mathcal{E} \end{array}$$

where $\mu(w_i + \iota(\mathcal{E})) = w_i + w_{\alpha^{-1}(i)}$. This is well defined because

$$\mu(\iota(e)) = \sum_{i \in e} w_i + \sum_{i \in e} w_{\alpha^{-1}(i)} = 0.$$

The diagram is commutative because

$$\mu(\partial_2(f)) = \mu\left(\sum_{i \in f} w_i\right) = \sum_{i \in f} w_i + w_{\alpha^{-1}(i)} = \bar{d}_2(f)$$

and

$$\begin{aligned} \bar{d}_1(\mu(w_i + \iota(\mathcal{E}))) &= \bar{d}_1(w_i + w_{\alpha^{-1}(i)}) \\ &= v_{\ni i} + e_{\ni i} + v_{\ni \alpha^{-1}(i)} + e_{\ni \alpha^{-1}(i)} \\ &= v_{\ni i} + v_{\ni \alpha^{-1}(i)} \\ &= \partial_1(w_i + \iota(\mathcal{E})). \end{aligned}$$

We have $\text{im } \bar{d}_2 = \text{im}(\mu \circ \partial_2) = \mu(\text{im } \partial_2)$ and the earlier lemma gives us $\mu(\ker \partial_1) \subseteq \ker \bar{d}_1$. We now aim to prove the other inclusion.

If $x \in \ker \bar{d}_1$ then at each edge (i.e. square vertex of the bipartite graph representation) of the hypermap there are an even number of adjacent darts of x . Choose an ordering $w_{i_0}, w_{i_1}, \dots, w_{i_{2m-1}}$ of the darts at each edge so that there are no other darts in x as we go counterclockwise from $w_{i_{2k}}$ to $w_{i_{2k+1}}$. Then if the darts in between $w_{i_{2k}}$ and $w_{i_{2k+1}}$ are w_{l_1}, \dots, w_{l_a} we have $\mu(w_i + w_{l_1} + \dots + w_{l_a} + \iota(\mathcal{E})) = w_i + w_j$. Now by

linearity, since all darts in x are adjacent to one edge of the hypermap, this gives us a recipe to write $x = \mu(w + \iota(\mathcal{E}))$ for some $w \in \mathcal{W}$.

Finally, note that if $w = \sum_{i \in I} w_i$ for some index set I then $\mu(w + \iota(\mathcal{E})) = \mu(\sum_{i \in I} w_i + \iota(\mathcal{E})) = \sum_{i \in I} (w_i + w_{\alpha^{-1}(i)})$. Thus if $\mu(w + \iota(\mathcal{E})) = 0$ then w is fixed by α^{-1} and must be in $\iota(\mathcal{E})$ so in fact μ is injective. Thus Lemma 4.14 shows that μ_* is an isomorphism. \square

This gives us three ways to think about H_1 :

1. ∂_1 -cycles in $\mathcal{W}/\iota(\mathcal{E})$ modulo ∂_2 -boundaries,
2. d_1 -cycles in \mathcal{W} modulo d_2 -boundaries and ι -boundaries, or
3. \bar{d}_1 -cycles in \mathcal{W} modulo \bar{d}_2 -boundaries.

4.4 Hypermap-homology codes

Choose a basis for $\mathcal{W}/\iota(\mathcal{E})$ (the bases for \mathcal{F} and \mathcal{V} have already been fixed).

Let $H_X = [\partial_1]$ and $H_Z^T = [\partial_2]$. Then we have $H_X H_Z^T = [\partial_1 \circ \partial_2] = [0] = 0$ so we can create a CSS code from H_X and H_Z . Now H_X is a $|V| \times (|W| - |E|)$ matrix and H_Z is a $|F| \times (|W| - |E|)$ matrix. By the theory of CSS codes this code has parameters $[|W| - |E|, 2g, D]$ with

$$D = \min\{\text{wt}(c) : c \in (C_X \setminus C_Z^\perp) \cup (C_Z \setminus C_X^\perp)\}$$

where $C_X = \ker(H_X)$, $C_X^\perp = \text{im}(H_X^T)$, $C_Z = \ker(H_Z)$ and $C_Z^\perp = \text{im}(H_Z^T)$.

Example 4.17. In this example we create a hypermap-homology code from example 4.11. Since $\mathcal{W}/\iota(\mathcal{E}) = \langle w_1, w_2, w_3, \dots, w_{24} \mid w_1 + w_2 + w_3, \dots, w_{22} + w_{23} + w_{24} \rangle$ we

may take as a basis $w_1, w_2, w_4, w_5, \dots, w_{23}$. Choosing this basis we can calculate

$$H_X = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

and

$$H_Z = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

In both matrices the columns are labelled by $w_1, w_2, w_4, w_5, w_7, w_8, \dots, w_{23}$. In H_X the rows are labelled by vertices v_1, \dots, v_8 while in H_Z the rows are labelled by faces f_1, \dots, f_8 . We give a few examples of how to compute these matrices:

$$\partial_1(w_{10}) = v_{\ni 10} + v_{\ni 12} = v_6 + v_7$$

$$\partial_2(f_1) = w_1 + w_{11} + w_6 + w_{21} = w_1 + w_4 + w_5 + w_{11} + w_{19} + w_{20}.$$

By computer search (see Section A.1) we calculate $D = 2$ so this is a $[16, 2, 2]$ code.

Example 4.18. The hypermap in Figure 4.4 has 48 darts, 16 vertices, 16 edges and 16 faces. We choose a basis for $\mathcal{W}/\iota(\mathcal{E})$ corresponding to all the darts except the ones to the bottom right of each edge. Computer search shows that the minimum distance is 3 and thus this is a $[32, 2, 3]$ code.

Example 4.19. The hypermap in Figure 4.5 has 32 darts, 8 vertices, 8 edges and 16 faces. The dual hypermap can be seen in Figure 4.6. We choose a basis for $\mathcal{W}/\iota(\mathcal{E})$

corresponding to all the darts except the ones below each edge. Computer search shows that the minimum distance is 4 and thus this is a $[24, 2, 4]$ code. Compare this to the toric code of minimum distance 4 which is a $[32, 2, 4]$ code.

Example 4.20. The hypermap in Figure 4.7 is based on a graph embedding from [GKN03] which is described as the dual of the line graph of a cube. We choose a basis for $\mathcal{W}/\iota(\mathcal{E})$ corresponding to all the darts whose labels are not divisible by 6. Computer calculations show that this gives rise to a $[20, 2, 3]$ code.

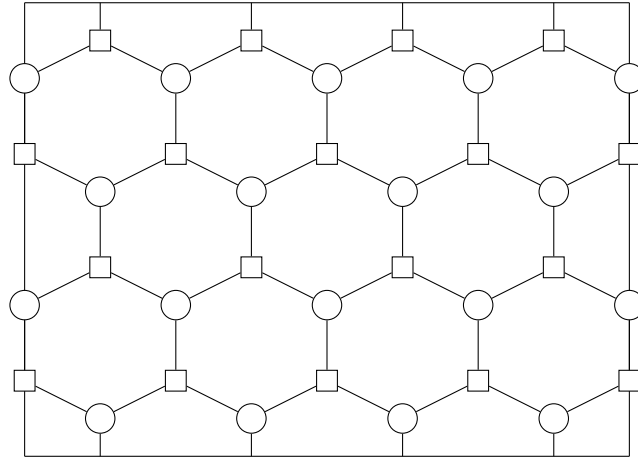


FIGURE 4.4. A hypermap with hexagonal faces, embedded on a torus.

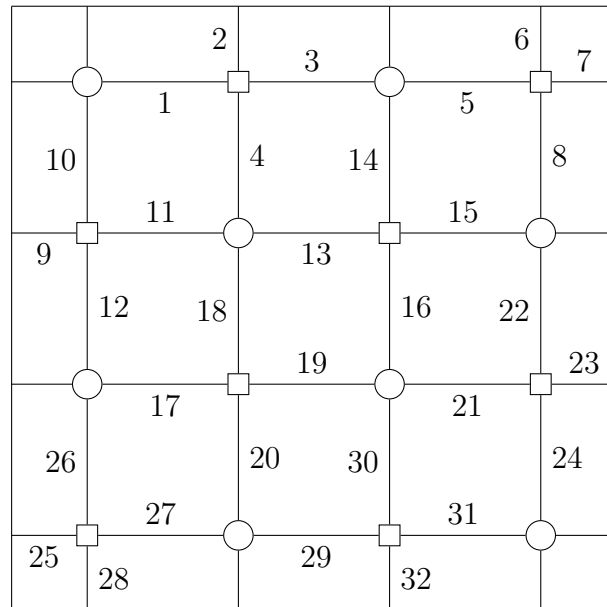


FIGURE 4.5. A hypermap with square faces, embedded on a torus.

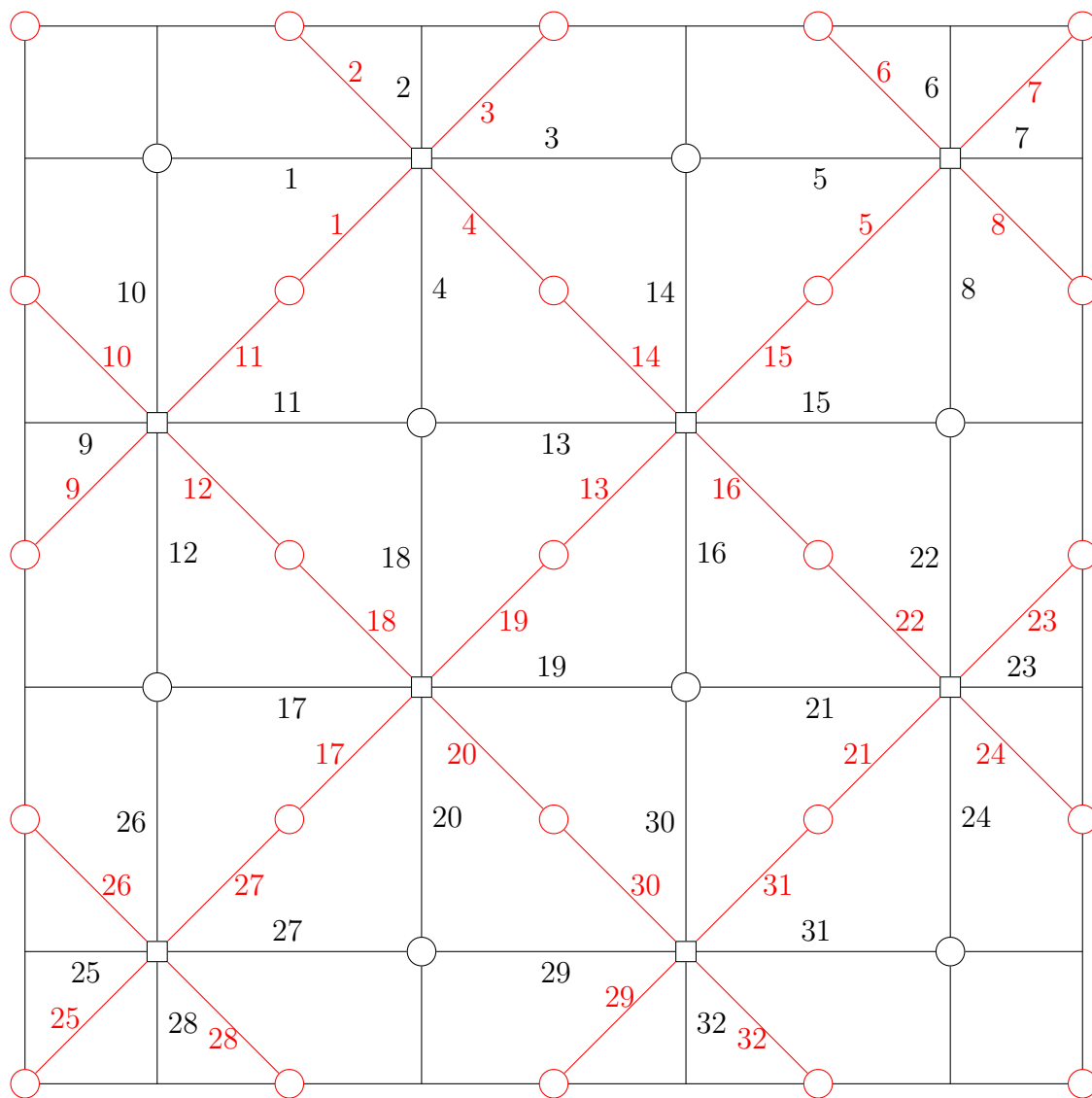


FIGURE 4.6. A hypermap with square faces, embedded on a torus. The dual hypermap is shown in red.

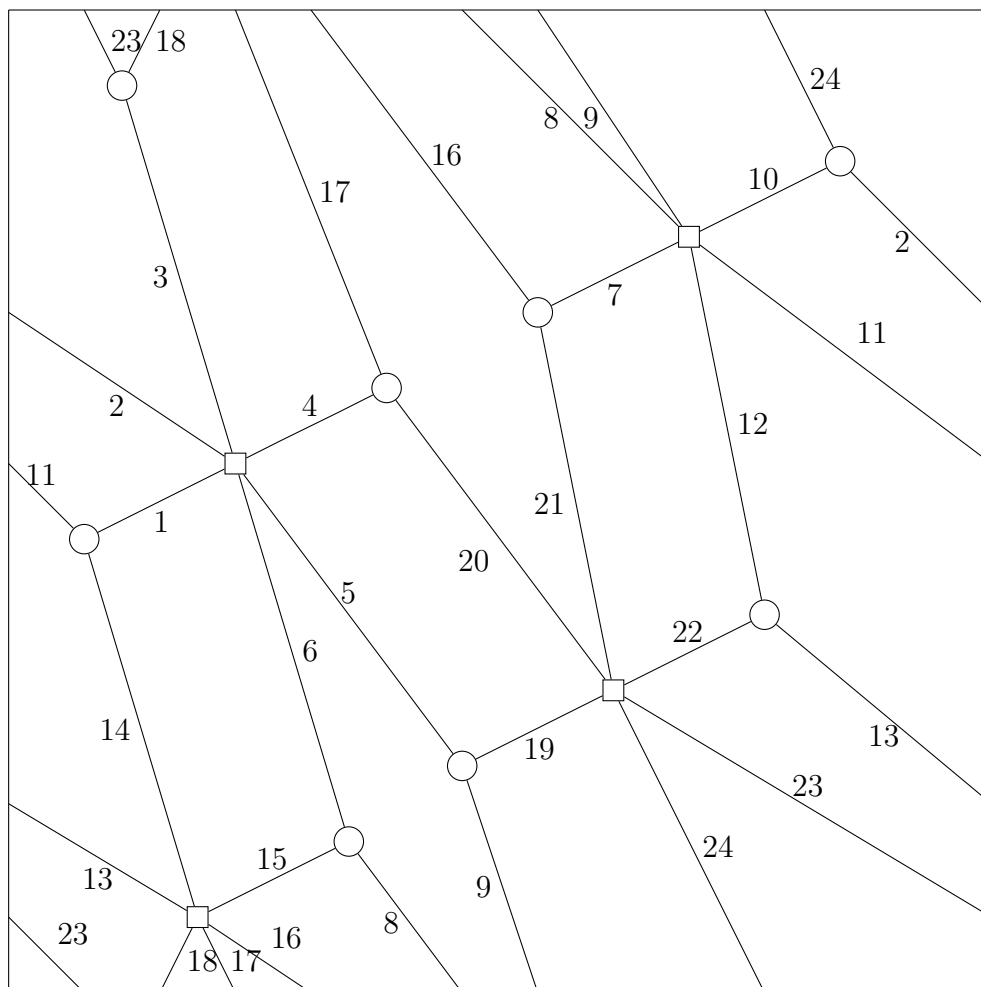


FIGURE 4.7. A hypermap based on the dual of the line graph of a cube.

4.5 Finding weights given a special basis

We choose, for each edge of the hypermap, one dart adjacent to that edge. Call this set of darts S , for ‘special darts’. Then we choose a basis of $\mathcal{W}/\iota(\mathcal{E})$ which is $W \setminus S$. With a basis chosen we have a weight function on $\mathcal{W}/\iota(\mathcal{E})$ with the weight of $x + \iota(\mathcal{E})$ given by the number of nonzero basis vectors in its representation in this ‘special basis’.

If we look at the proof of Proposition 4.16 then we see that $\mu: \ker \partial_1 \setminus \text{im } \partial_2 \rightarrow \ker \bar{d}_1 \setminus \text{im } \bar{d}_2$ is a bijection. Thus we can say that

$$\text{minwt}(C_X \setminus C_Z^\perp) = \text{minwt}\{\mu^{-1}(x): x \in \ker \bar{d}_1 \setminus \text{im } \bar{d}_2\}.$$

Now we seek to understand how we can find the weight of $\mu^{-1}(x)$ given x . We can choose an ordering of the darts around each edge using two cases. If the special dart adjacent to the edge is in x then choose the pairing of darts so that the edge is at the end of a pair not the start. If the special dart is not in x then we can choose an ordering of the darts so that the special dart is not between two paired darts. In both cases $\mu^{-1}(x)$ does not contain any special darts. Now $\mu^{-1}(x)$ is uniquely determined and we have chosen a representation of $\mu^{-1}(x)$ without any special darts so the weight of $\mu^{-1}(x)$ is just the number of darts in its sum. We see also that

$$\text{wt}(\mu^{-1}(x)) = \frac{1}{2} \text{wt}(x) +$$

of non- x darts as we go counterclockwise around edges avoiding special darts.

One way to understand this is to imagine that at each of the darts that we skip as we go around an edge our cycle traverses that dart twice. Then every cycle in $\ker \bar{d}_1$ can be thought of as a cycle that does not skip darts around edges and μ^{-1} of that cycle is of weight half the weight of the cycle.

With the basis $W \setminus S$ chosen then we can identify vector spaces with their duals in the standard way and then the coboundary operators are given by transposes of

boundary operators. If we look at the matrices $\delta_1 = H_X^T$ and $\delta_2 = H_Z$ then we see that δ_1 takes vertices to non-special darts and δ_2 takes non-special darts to faces so we will identify $(\mathcal{W}/\iota(\mathcal{E}))^*$ with $\mathcal{W}/\langle S \rangle$. Thus our hypermap-cohomology will be

$$\mathcal{V} \xrightarrow{\delta_1} \mathcal{W}/\langle S \rangle \xrightarrow{\delta_2} \mathcal{F}$$

where

$$\begin{aligned} \delta_1(v) &= \text{sum of non-special darts whose } \partial_1\text{-boundary contains } v \\ &= \text{map-boundary of face } v \text{ in dual hypermap without special darts} \end{aligned}$$

and

$$\begin{aligned} \delta_2(w_i) &= \text{faces whose } \partial_2\text{-boundaries contain } w_i \\ &= f_{\ni i} + f_{\ni \text{special dart of } e_{\ni i}}. \end{aligned}$$

We can also consider the classical cohomology of the bipartite graph representation of the hypermap. This is

$$\mathcal{V} \xrightarrow{\bar{\delta}_1} \mathcal{W} \xrightarrow{\bar{\delta}_2} \mathcal{F} \oplus \mathcal{E}$$

where

$$\bar{\delta}_1(v) = \text{map-boundary of face } v \text{ in dual-hypermap}$$

and

$$\bar{\delta}_2(w_i) = f_{\ni i} + e_{\ni i}.$$

Proposition 4.21.

$$H^1 = \frac{\ker \delta_2}{\text{im } \delta_1} \cong \frac{\ker \bar{\delta}_2}{\text{im } \bar{\delta}_1}.$$

Proof. Consider the diagram

$$\begin{array}{ccccc} \mathcal{V} & \xrightarrow{\bar{\delta}_1} & \mathcal{W} & \xrightarrow{\bar{\delta}_2} & \mathcal{E} \oplus \mathcal{F} \\ \downarrow \text{id} & & \downarrow \pi & & \downarrow \lambda \\ \mathcal{V} & \xrightarrow{\delta_1} & \mathcal{W}/\langle S \rangle & \xrightarrow{\delta_2} & \mathcal{F} \end{array}$$

where π is the projection $\mathcal{W} \rightarrow \mathcal{W}/\langle S \rangle$ and λ is defined by $\lambda(f_i) = f_i$ and $\lambda(e_i) = f_{\ni \text{special dart of } e_i}$.

This diagram is commutative because

$$\delta_1(v) = \text{map-boundary of face } v \text{ in dual hypermap without special darts} = \pi(\bar{\delta}_1(v))$$

and

$$\begin{aligned} \lambda(\bar{\delta}_2(w_i)) &= \lambda(f_{\ni i} + e_{\ni i}) \\ &= f_{\ni i} + f_{\ni \text{special dart of } e_{\ni i}} \\ &= \delta_2(\pi(w_i)) \end{aligned}$$

because $f_{\ni i} + f_{\ni \text{special dart of } e_{\ni i}} = 0$ precisely when $w_i \in S$.

Now $\text{im } \delta_1 = \text{im}(\pi \circ \bar{\delta}_1) = \pi(\text{im } \bar{\delta}_1)$ and as usual we know that $\pi(\ker \delta_2) \subseteq \ker \delta_2$. We now show the opposite inclusion. If $x \in \ker \delta_2$, define $w \in \mathcal{W}$ to contain all the darts of x and then for each edge of the hypermap, if there are an odd number of darts in x adjacent to that edge, add the special dart adjacent to that edge to w . Clearly $\pi(w) = x$, we now claim that $w \in \ker \bar{\delta}_2$. To see this, note that

$$\lambda(\bar{\delta}_2(w)) = \delta_2(\pi(w)) = \delta_2(x) = 0$$

and

$$\bar{\delta}_2(w) = \bar{\delta}_2\left(\sum_{i \in I} w_i\right) = \sum_{i \in I} (f_{\ni i} + e_{\ni i}) = \sum_{i \in I} f_{\ni i}$$

because each edge has an even number of darts of w adjacent. Thus $\lambda(\bar{\delta}_2(w)) = \bar{\delta}_2(w)$ and $w \in \ker \bar{\delta}_2$ as desired.

Finally if $w \in \ker \pi \cap \ker \bar{\delta}_2$ then w is a sum of special darts which is a classical cycle in the dual hypermap. In particular each edge is adjacent to an even number of darts in w . But each edge is adjacent to only one special dart in the whole hypermap so we must have $w = 0$. Thus $\ker \pi \cap \ker \bar{\delta}_2 \subseteq \text{im } \bar{\delta}_1$ and Lemma 4.14 tells us that π_* is an isomorphism. \square

Corollary 4.22. *There is a bijection*

$$\pi: \ker \bar{\delta}_2 \setminus \text{im } \bar{\delta}_1 \rightarrow \ker \delta_2 \setminus \text{im } \delta_1$$

which takes non-boundary cycles of hypermap-cohomology to non-boundary classical cycles of the dual hypermap.

Corollary 4.23. *$\text{minwt}(C_Z \setminus C_X^\perp)$ is given by the minimum weight of a non-boundary classical cycle in the dual hypermap where non-special darts have weight 1 and special darts have weight 0.*

Example 4.24. Take an $m \times m$ square grid hypermap embedded on the torus with m even and the special darts chosen to be the darts underneath each edge (see Figure 4.6 for the hypermap and its dual in the $m = 4$ case, where the special darts are those with labels divisible by 4). Then we have an $[N, k, d]$ code with $N = n - |E| = 2m^2 - (1/2)m^2 = (3/2)m^2$ and $k = 2g = 2$.

For $\text{minwt}(C_Z \setminus C_X^\perp)$ notice that there is a path in the dual hypermap of length $2m$ with half of the darts special darts. Also there are no paths in the dual hypermap of classical weight less than $2m$ so $\text{minwt}(C_Z \setminus C_X^\perp) = m$.

For $\text{minwt}(C_X \setminus C_Z^\perp)$ notice that a horizontal or vertical classical cycle $x \in \ker \bar{d}_1 \setminus \text{im } \bar{d}_2$ in the hypermap has $\text{wt}(\mu^{-1}(x)) = m$. To see this is the minimum weight, if $x \in \ker \bar{d}_1 \setminus \text{im } \bar{d}_2$ then x is not a boundary so without loss of generality x has at least m horizontal darts. But $\mu(\mu^{-1}(x)) = x$ and μ takes w_i to w_i plus w_i rotated around an edge. Therefore μ acting on one dart can only ever lead to 1 horizontal dart so we must in fact have $\text{wt}(\mu^{-1}(x)) \geq m$.

Thus we have a $[(3/2)m^2, 2, m]$ code with $k = 2$ and $d = \sqrt{\frac{2}{3}N}$, so $kd^2 = \frac{4}{3}N$.

4.6 Future work

Many questions about hypermap-homology codes remain unanswered. We mention a few of them here:

1. Must hypermap-homology codes with a special basis satisfy $kd^2 < cN$ for some constant c ?
2. Can we analyze hypermap-homology codes with a non-special basis?
3. Can we find families of hypermaps which lead to better performance than the square grid hypermap?
4. Can we construct ‘planar hypermap-homology codes’?
5. Can we analyze the performance of randomly generated hypermap-homology codes?

APPENDIX A
SOFTWARE

A.1 Matlab software for hypermap-homology codes

We implemented an object-oriented package in Matlab to work with hypermap-homology codes. This includes classes for permutations, hypermaps and CSS codes. The use of this package in this dissertation is restricted to determining the parameters $[n, k, d]$ of a code generated by an input hypermap (σ, α) . We now describe the simple algorithms we use to do this.

First we check if $\langle \sigma, \alpha \rangle \leq S_n$ is transitive. This algorithm comes from [HEO05].

Algorithm 1 CheckTransitive(σ, α)

Require: $\sigma, \alpha \in S_n$

Ensure: Output whether $\langle \sigma, \alpha \rangle$ is transitive

```

1: Set Orbit = {1}
2: repeat
3:   for  $i = 1, \dots, n$  do
4:     Set Orbit = Orbit  $\cup$   $\{\sigma(i), \alpha(i)\}$ 
5:   end for
6: until Orbit did not change in last iteration
7: if  $|\text{Orbit}| = n$  then
8:   return 'Yes'
9: else
10:  return 'No'
11: end if

```

After finding the matrices H_X and H_Z using the definitions in Chapter 4 we can then determine the parameters of the associated code.

Algorithm 2 FindParameters(H_X, H_Z)

Require: Binary matrices H_X and H_Z with $H_X H_Z^T = 0$

Ensure: Output parameters $[n, k, d]$

- 1: Set $n =$ width of matrix H_X
 - 2: Find $\text{rank}(H_X)$ and $\text{rank}(H_Z)$ by putting these matrices in RREF
 - 3: Set $k = n - \text{rank}(H_X) - \text{rank}(H_Z)$
 - 4: Find generator matrices G_X, G_Z by finding nullspace of H_X, H_Z
 - 5: Set k_X to the height of G_X and k_Z to the height of G_Z
 - 6: Set $d = \infty$
 - 7: **for all** $u \in \mathbb{F}_2^{k_X}$ **do**
 - 8: Set $c = G_X u$
 - 9: **if** $0 < \text{wt}(c) < d$ and $G_Z^T c \neq 0$ **then**
 - 10: Set $d = \text{wt}(c)$
 - 11: **end if**
 - 12: **end for**
 - 13: **for all** $u \in \mathbb{F}_2^{k_Z}$ **do**
 - 14: Set $c = G_Z u$
 - 15: **if** $0 < \text{wt}(c) < d$ and $G_X^T c \neq 0$ **then**
 - 16: Set $d = \text{wt}(c)$
 - 17: **end if**
 - 18: **end for**
 - 19: **return** n, k, d
-

Index

- binary linear code, 14
- chain complex, 27
- check matrix, 23
- cohomology, 28
- CSS code, 25
- dual
 - chain complex, 27
 - code, 15
 - hypermap, 42
 - Poincaré, 30
- embedded graph, 29
- generator matrix, 14
- homology, 27
 - bipartite graph representation, 49
 - hypermap, 45
- hypergraph, 38
- hypermap
 - combinatorial, 39
 - topological, 39
- induced homomorphism, 48
- map, *see* embedded graph
- minimum distance
 - classical, 16
 - quantum, 22
- observable, 17
- parity check matrix, 14
- Pauli group, 18
 - effective, 22
- Pauli matrices, 18
- planar code, 33
- quantum mechanics, 16
- qubit, 16
- special basis, 57
- stabilizer code, 20
- stabilizer group, 20
- Steane's 7 qubit code, 26
- syndrome, 20
- toric code, 31

REFERENCES

- [BMD07] H. Bombin and MA Martin-Delgado. Homological error correction: classical and quantum codes. *Journal of mathematical physics*, 48(5):052105–052105, 2007.
- [BPT10] S. Bravyi, D. Poulin, and B. Terhal. Tradeoffs for reliable quantum information storage in 2d systems. *Physical review letters*, 104(5):50503, 2010.
- [CDZ11] A. Couvreur, N. Delfosse, and G. Zémor. A construction of quantum ldpc codes from cayley graphs. In *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pages 643–647. IEEE, 2011.
- [CM92] R. Cori and A. Machi. Maps, hypermaps and their automorphisms: a survey i, ii, iii. *Exposition. Math*, 10(5):403–427, 1992.
- [CP80] R. Cori and J.G. Penaud. The complexity of a planar hypermap and that of its dual. *Annals of Discrete Mathematics*, 9:53–62, 1980.
- [CS96] A.R. Calderbank and P.W. Shor. Good quantum error-correcting codes exist. *Physical Review A*, 54(2):1098, 1996.
- [DCP10] G. Duclos-Cianci and D. Poulin. Fast decoders for topological quantum codes. *Physical review letters*, 104(5):50504, 2010.
- [DKLP02] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill. Topological quantum memory. *Journal of Mathematical Physics*, 43:4452, 2002.
- [Fet11] E. Fetaya. Homological error correcting codes and systolic geometry. *Arxiv preprint arXiv:1108.2886*, 2011.
- [FML02] M.H. Freedman, D.A. Meyer, and F. Luo. Z_2 -systolic freedom and quantum codes. *Mathematics of quantum computation, Chapman & Hall/CRC*, pages 287–320, 2002.
- [Gal62] Robert Gallager. Low-density parity-check codes. *Information Theory, IRE Transactions on*, 8(1):21–28, 1962.
- [GKN03] A. Gagarin, W. Kocay, and D. Neilson. Embeddings of small graphs on the torus. *Cubo. v5*, pages 351–371, 2003.
- [Got97] D. Gottesman. Stabilizer codes and quantum error correction. *Arxiv preprint quant-ph/9705052*, 1997.

- [Gur10] V. Guruswami. Course notes for introduction to coding theory. *Carnegie Mellon University*, 2010.
- [Hat02] Allen Hatcher. *Algebraic topology*. Cambridge University Press, 2002.
- [HEO05] Derek F Holt, Bettina Eick, and Eamonn A O'Brien. *Handbook of computational group theory*, volume 24. Chapman and Hall/CRC, 2005.
- [Kit03] A.Y. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, 2003.
- [KP12] A.A. Kovalev and L.P. Pryadko. Improved quantum hypergraph-product ldpc codes. *Arxiv preprint arXiv:1202.0928*, 2012.
- [LZZ04] S.K. Lando, A.K. Zvonkin, and D.B. Zagier. *Graphs on surfaces and their applications*, volume 141. Springer Verlag, 2004.
- [Mac03] D.J.C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge Univ Pr, 2003.
- [Maz11] F. Mazoit. Tree-width of hypergraphs and surface duality. *Journal of Combinatorial Theory, Series B*, 2011.
- [MMM04] D.J.C. MacKay, G. Mitchison, and P.L. McFadden. Sparse-graph codes for quantum error correction. *Information Theory, IEEE Transactions on*, 50(10):2315–2330, 2004.
- [NC10] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.
- [RU08] T.J. Richardson and R.L. Urbanke. *Modern coding theory*. Cambridge Univ Pr, 2008.
- [SB12] P. Sarvepalli and K.R. Brown. Topological subsystem codes from graphs and hypergraphs. *arXiv preprint arXiv:1207.0479*, 2012.
- [TZ09] J.P. Tillich and G. Zémor. Quantum ldpc codes with positive rate and minimum distance proportional to $n^{1/2}$. In *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, pages 799–803. IEEE, 2009.
- [Wal75] TRS Walsh. Hypermaps versus bipartite maps. *Journal of Combinatorial Theory, Series B*, 18(2):155–163, 1975.
- [Zém09] G. Zémor. On cayley graphs, surface codes, and the limits of homological coding for quantum error correction. *Coding and Cryptology*, pages 259–273, 2009.