12-2023

On Dyadic Parity Check Codes and Their Generalizations

Meraiah Martinez
University of Nebraska-Lincoln, mmartinez26@huskers.unl.edu

ON DYADIC PARITY CHECK CODES AND THEIR GENERALIZATIONS

by

Meraiah Martinez

A DISSERTATION

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Doctor of Philosophy

Major: Mathematics

Under the Supervision of Professor Christine A. Kelley

Lincoln, Nebraska

December, 2023

ON DYADIC PARITY CHECK CODES AND THEIR GENERALIZATIONS

Meraiah Martinez, Ph.D.

University of Nebraska, 2023

Adviser: Christine A. Kelley

In order to communicate information over a noisy channel, error-correcting codes can be used to ensure that small errors don't prevent the transmission of a message. One family of codes that has been found to have good properties is low-density parity check (LDPC) codes. These are represented by sparse bipartite graphs and have low-complexity graph-based decoding algorithms. Various graphical properties, such as the girth and stopping sets, influence when these algorithms might fail. Additionally, codes based on algebraically structured parity check matrices are desirable in applications due to their compact representations, practical implementation advantages, and tractable decoder performance analysis.

This dissertation focuses on codes based on parity check matrices that are dyadic, n-adic, or quasi-dyadic (QD), meaning the parity check matrix representation is block structured with dyadic matrices as blocks. Depending on the number of nonzero positions in the leading row of each block, these codes may be either low density or moderate density. Since each block is reproducible, the resulting QD codes have similar advantages to quasi-cyclic (QC) codes. We examine basic code properties of dyadic, n-adic, and QD parity check codes, including bounds on the dimension and minimum distance, cycle structure of the corresponding Tanner graph, and their possible use in quantum code constructions. We also consider the relationship between cycle codes of graphs and cycle codes of their lifts.

ACKNOWLEDGMENTS

All thanks to Jesus for the amazing gifts He has given to me.

I also want to say a huge thank you to my advisor, Christine Kelley, who has guided me through this process with much patience and encouragement. I am so grateful for your mentorship.

Thank you also to my readers, Jamie Radcliffe and Xavier Perez-Gimenez, as well as the other members of my committee, Mark Walker and Vinod Variyam, and the rest of the math department at UNL. All of your support over the past several years has kept me persevering through graduate school, especially with the extra trials during the pandemic season. Thanks to my classmates for all of your support, as well, with many study sessions over the years.

To the math department and others at Benedictine College, who supported me through my undergraduate studies, as well as in this first year as a professor, thank you! Thank you, as well, to my professors at Front Range Community College, especially Ken Monks, who explained math in a way that finally clicked for me and helped me learn to love the subject.

I am so grateful for the dear friends I have made in Lincoln, both in and out of the math department. Thank you all for the company in ice cream runs, board games, long walks, Shakespeare nights, and good conversations.

Thank you also to my parents, siblings, other family, and friends who have supported me day in and day out. I am so grateful for each of you! (There is not enough space here to thank everyone.)

Table of Contents

List of Figures

Chapter 1

Introduction

Consider a situation in which we have some information we would like to transmit. Some examples include saving a file on a computer, calling a friend on the phone, and making a QR code to send people to a particular website. In each situation, there is a chance that something may go wrong in the transmission process. In the QR code example, the entire code may not be visible, or perhaps a section is smudged. In each situation, though, error-correcting codes can be used in order to ensure that small errors do not prevent recovery of the information.

There are a wide variety of methods for constructing codes, including graph-based codes with iterative decoders, which give a method for correcting errors when using these codes. These have been shown to achieve near-capacity performance on several communication channels, efficiently correcting errors nearly up to the theoretical bound, and have replaced classical codes in many applications [14]. Low density parity check (LDPC) codes, characterized by having sparse parity check matrix representations, are one such family of graph-based codes. For practical implementation, the design of short to moderate length codes with algebraic structure is desired. Thus, array-based LDPC codes can be constructed, which have parity check matrices with block decomposition form. When the blocks are circulants, one obtains quasi-cyclic low density parity check (QC-LDPC) codes. Such codes have long been regarded as

good candidates in practice due to their efficient practical implementation, compact representation, and good decoder performance, as in [8, 20, 37].

Codes with compact representation are also desirable. For example, in code-based cryptography, the McEliece cryptosystem, proposed in 1978 by Robert McEliece [19], uses Goppa codes and relies on the difficulty of decoding a vector in a random linear code. The McEliece cryptosystem is a public key cryptosystem in which a generator matrix with an efficient decoding algorithm is the private key, and another generator matrix for the code without such a decoding algorithm is the public key. To encrypt a message, the message is multiplied by the generator matrix, and a random error is added to the message. To decrypt the message, this codeword with error is decoded by the indented receiver with the efficient decoding algorithm [19]. The primary limitation of the cryptosystem is that the public key is too large [1], which has motivated searches for codes with compact representation.

In an effort to find such codes, reproducible and quasi-reproducible codes were introduced in [28]. These codes have parity check matrices specifiable using a small subset of their rows and a set of transformations. While cyclic and quasi-cyclic codes belong to this class, so do many others. Quasi-dyadic parity check codes have a block decomposition using dyadic matrices as blocks. The first row of each dyadic matrix is enough to specify the block, and the number of entries in the first row determines the density of the matrix. Quasi-dyadic low density parity check (QD-LDPC) codes have similar advantages to QC-LDPC codes, such as efficient representation and comparable parameters, and therefore have potential to be useful in many applications. Indeed, codes with QD structure were considered in [21], and later in [23] and [28]. Moreover, [2] analyzed QD arithmetic and showed that the structure may be exploited to yeild efficient encoding and decoding algorithms.

In this dissertation, we investigate the properties of QD-LDPC codes and gener-

alizations to quasi-n-adic LDPC codes to understand their potential for other coding theory applications. In particular, we ask basic questions on the dimension, girth, and cycle structure of these code families, the latter two of which are relevant for iterative decoding performance. We also derive an upper bound on the minimum distance of these codes. Moreover, we give an explicit example of a quasi-triadic LDPC code that has similar parameters to a well-known QC-LDPC code, further demonstrating that QD codes and their variations have potential to be useful in practice. We also take an initial look at the use of QD codes in quantum coding theory.

This dissertation is organized as follows. Chapter 2 contains background information on codes and n-adic matrices. We then consider various properties of quasi-n-adic matrices in Chapter 3, including an isomorphism to a polynomial quotient ring for the special case of $n = 2$. In Chapter 4, we continue looking at the special case of $n = 2$ to analyze some code parameters and consider applications in quantum coding theory. The analysis of code parameters for n-adic and quasi-n-adic codes continues in Chapters 5 and 6. In Chapter 7, we consider taking lifts of graphs and the cycle codes resulting from lifts. Finally, Chapter 8 concludes the dissertation.

Parts of Chapters 3, 5, and 6 appear in joint work with Kelley [17]. The material in Chapters 3, 4, 5, and 6 also appears in joint work with Pllaha and Kelley [18].

Chapter 2

Preliminaries

Recall our situation at the beginning of Chapter 1, where we wanted to transmit information and correct some number of errors in the process. We address this problem using a *code*. A *code* $\mathcal{C}$ of length n over an alphabet A is a subset $\mathcal{C} \subseteq A^n$. If $\mathcal{C}$ is a k-dimensional subspace of $\mathbb{F}_q^n$, then $\mathcal{C}$ is an $[n,k]_q$ *linear code*. Here, k is the *dimension* of the code, i.e. $\dim(\mathcal{C}) = k$, and we are able to send k pieces of information using n transmitted elements. These transmitted tuples are called *codewords*. When a codeword is transmitted over a *channel*, some errors may be introduced, as noted earlier.

Various properties of codes are used to determine how well a particular code can transmit information. One can measure the efficiency of a code $\mathcal{C}$ using its *rate*, k/n. Additionally, we can consider properties which can be helpful when decoding messages which may contain errors. The *Hamming distance* between two codewords $d(u,v)$ is the number of entries in which they differ. The *minimum distance* $d_{\min}$ of a code is the smallest distance between any two codewords, i.e.

$$d_{\min}(\mathcal{C}) := \min_{u,v \in \mathcal{C}} d(u,v).$$

Computing the minimum distance for a general code can be difficult, but it is

straightforward to show that in a linear code, the minimum distance can be found by considering the nonzero entries in each codeword. The *weight* $\mathrm{wt}(u)$ of a codeword is the number of nonzero entries in u, and if $\mathcal{C}$ is a linear code, the minimum distance of $\mathcal{C}$ is equal to the smallest weight of a nonzero codeword.

A larger minimum distance for a code guarantees a greater ability to correct errors in messages. If the Hamming distance between a received message and a codeword is at most $\lfloor \frac{d_{\min}-1}{2} \rfloor$, then that codeword is the unique codeword closest to the received message. Because the probability of errors in transmission is assumed to be relatively low, the closest codeword is the most likely to have been transmitted. Thus, if we have at most $\lfloor \frac{d_{\min}-1}{2} \rfloor$ errors, each received message can be decoded to its unique closest codeword. This implies that codes with larger minimum distances can correct more errors.

Linear codes are often defined by a matrix which generates the subspace or for which the subspace is the kernel. A *generator matrix* G of a linear code $\mathcal{C}$ is a matrix such that the rows of G form a generating set for $\mathcal{C}$. Similarly, a *parity check matrix* H of a linear code $\mathcal{C}$ is a matrix such that $Hc^T = 0$ if and only if c is a codeword in $\mathcal{C}$.

The *dual code* of a linear code $\mathcal{C}$ in $\mathbb{F}_q^n$ is defined as

$$\mathcal{C}^\perp = \{u \in \mathbb{F}_q^n \mid u \cdot c \equiv uc^T = 0 \text{ for all } c \in \mathcal{C}\}. \tag{2.1}$$

Thus, a parity check matrix of a code is a generator matrix for its dual, and a generator matrix for a code is a parity check matrix for its dual. This also gives $\dim(\mathcal{C}^\perp) = n - \dim(\mathcal{C})$.

Example 2.0.1. *Consider the* $[3,2]_2$ *linear code* $\mathcal{C}$ *given by:*

$$\mathcal{C} = \{000, 011, 110, 101\}.$$

We see that $d_{min}(\mathcal{C}) = 2$*, since that is the smallest weight of a nonzero codeword, and we have* $\dim(\mathcal{C}) = 2$*. A generator matrix for* $\mathcal{C}$ *is given by*

$$G = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix},$$

since every element in $\mathcal{C}$ *is given by a linear combination of the rows of* G*. Additionally, a parity check matrix for* $\mathcal{C}$ *is given by*

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

We see that for $101 \in \mathcal{C}$*, we have*

$$H(1,0,1)^T = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = 1 + 0 + 1 = 0.$$

□

2.1 LDPC Codes

Gallager introduced low density parity check (LDPC) codes in [10]. An LDPC code is a code that has a sparse parity check matrix representation. Typically, a family of LDPC codes will have a small constant row weight. These codes work well with graph-based iterative decoding algorithms, which allow for the correction of trans-

mitted errors in a relatively short amount of time. Moderate density parity check (MDPC) codes are similarly characterized by a moderately dense parity check matrix and have been explored particularly for cryptographic applications [1]. In this case, families have row weights that scale in $O(\sqrt{n \log n})$ but do not work quite as well with iterative decoding algorithms. LDPC codes are well suited for graph-based decoding due to their sparse representations as *Tanner graphs* [33]. The complexity for these algorithms is linear in the number of edges [14].

Definition 2.1.1. *The* Tanner graph *of a linear code $\mathcal{C}$ from a parity check matrix H is the bipartite graph $G = (V, W; E)$ for which H is the adjacency matrix. The vertices W corresponding to the rows of H are called* check nodes*, and the vertices V corresponding to the columns of H are called* variable nodes*. There is an edge (v, w) if and only if the (w, v) entry of H, denoted $h_{w,v}$, is nonzero. For non-binary codes, the edge (v, w) has weight $h_{w,v}$.*

Example 2.1.2. *Consider the code $\mathcal{C}$ over $\mathbb{F}_2$ given by the parity check matrix*

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

We can construct the corresponding Tanner graph in Figure 2.1, where the variable nodes are represented by circles and the check nodes are represented by squares.

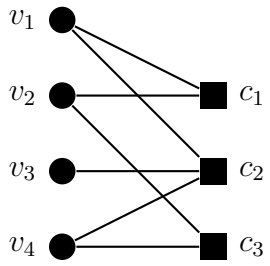


Figure 2.1: A Tanner graph for a linear code of length 4 with 3 check nodes

We see that $1101 \in \mathcal{C}$*, since*

$$H(1,1,0,1)^T = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

We can also see that this is a codeword by considering the Tanner graph. Here, we view the codeword coordinates as values for the variable nodes and add the values adjacent to each check node. In this case, as shown in Figure 2.2, each check node has a sum of 0 and so is satisfied. Thus, $1101 \in \mathcal{C}$*.*

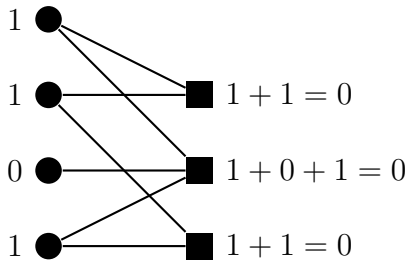


Figure 2.2: Using a Tanner graph to verify a codeword

Similarly, we see that $0011 \notin \mathcal{C}$*, since*

$$H(0,0,1,1)^T \neq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

In the Tanner graph, we find values as in Figure 2.3. Since there is at least one check node that does not add to 0, it is not satisfied, and so $0011 \notin \mathcal{C}$*.* □

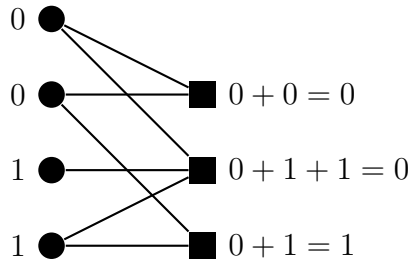


Figure 2.3: Using a Tanner graph to verify a non-codeword

A parity check matrix is (j, ℓ)-*regular* if each row has weight j and each column has weight ℓ. This implies that in the corresponding Tanner graph, each variable node has degree j and each check node has degree ℓ.

Various decoding algorithms make use of Tanner graph representations, and different properties of these graphs can improve the performance of these algorithms. For example, the degree distribution of a graph is related to the decoding threshold, which characterizes the worst channel on which a decoder can operate [5].

Additionally, we can consider the *girth* of a graph, which is the length of the smallest cycle in the graph. For example, for the graph in Figure 2.1, the girth is 6, since the smallest cycle is given by the edges connecting vertices v_1, c_1, v_2, c_3, v_4, and c_2. Graph-based iterative decoding algorithms, such as maximum-likelihood soft-decision decoding, work best on codes with a cycle-free Tanner graph, because different parts

of the message remain independent for more iterations. However, Etzion showed in [7] that codes with cycle-free Tanner graphs have poor distance in relation to the rate. Thus, in order to have good decoding properties while maintaining other good code parameters, it is desirable that codes have Tanner graph representations with large girth, e.g. [6, 16, 29].

Another graph structure that affects decoding is the presence of *stopping sets*. A *stopping set* of a Tanner graph is a set of variable nodes $S \subseteq V$ for which each check node neighbor of S in W has at least two neighbors in S. For example, the set $S = \{v_1, v_2, v_4\}$ is a stopping set of the graph in Figure 2.1, because each of the check node neighbors of some element of this set, c_1, c_2, and c_3, has at least two neighbors in S. However, the set $S = \{v_3, v_4\}$ is not a stopping set, because c_3 only has one neighbor in S. The *stopping distance* of a code with a particular Tanner graph representation is the size of a minimum stopping set in the Tanner graph. In Example 2.1.2, the stopping distance of $\mathcal{C}$ with the Tanner graph representation in Figure 2.1 is 3, because for any set of one or two variable nodes, there is some check node neighbor that only has one neighbor in the set of variable nodes. However, we found a stopping set of size 3, so the stopping distance is 3. When decoding from the erasure channel, where errors are given by the erasure of data in the message, stopping sets can prevent successful decoding. If each variable node in a stopping set is erased in a message, the check node neighbors cannot be used to determine what the missing variable node values should be. Thus, small stopping sets can be problematic, and so a large stopping distance is desirable for a code.

2.1.1 Graph lifts

To construct LDPC codes, we need sparse bipartite graphs, and long codes require large vertex sets. Constructing a *lift* of a graph is one way to obtain graphs that can

be used for LDPC codes. In the literature, such graphs are called protograph codes [36] or codes based on voltage graphs [15].

Let $G = (V, E)$ be a directed graph. To obtain a degree ℓ lift of the base graph G, label each edge e of G with a permutation $\sigma_e \in S_\ell$, the symmetric group on ℓ elements. Then the corresponding degree ℓ lift of G, denoted $\hat{G} = (\hat{V}, \hat{E})$ is constructed by replacing each $v \in V$ with ℓ vertices $\{v_1, \ldots, v_\ell\}$ in $\hat{V}$, and the edges in $\hat{E}$ are given by $(v_i, w_{\sigma_e(i)})$ where $e = (v, w) \in E$. Such a lift of a graph is also known as a permutation voltage graph in [13]. Lifts are particularly helpful for constructing graph-based codes, because they are locally like the base graph and so inherit a lot of its properties, such as the degree distribution.

Example 2.1.3. *Consider the directed graph G with edges labeled with a permutation in S_2, where $\iota = (1)(2)$, the identity permutation, and $\sigma = (1\ 2)$, given in Figure 2.4.*

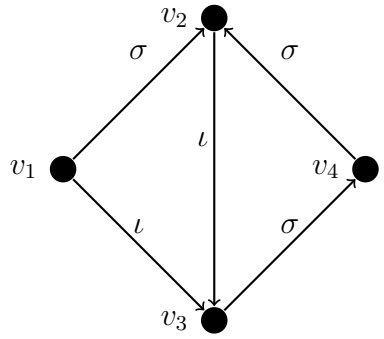


Figure 2.4: A directed graph G

Then the corresponding degree 2 lift of G is $\hat{G}$, as shown in Figure 2.5. □

When a Tanner graph is used as the base graph of a lift, all edges can be considered as directed from a check node to a variable node. Recall that a permutation $\sigma \in S_n$ can be written as an $n \times n$ permutation matrix, where the (i, j) entry is equal to 1 if $\sigma(j) = i$ and is 0 otherwise. In this case, the lifted graph is itself a Tanner

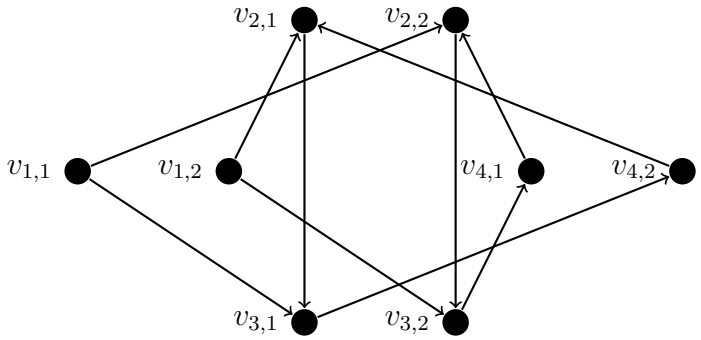


Figure 2.5: The degree 2 lift $\hat{G}$ of graph G in Figure 2.4

graph, and the corresponding parity check matrix can be constructed from the original parity check matrix by replacing each nonzero entry corresponding to edge e with the permutation matrix corresponding to σ_e and replacing each zero entry with a zero matrix.

Example 2.1.4. *Consider the code $\mathcal{C}$ from Example 2.1.2. We assign permutations to edges as follows using elements of S_3, where $\iota = (1)(2)(3)$, the identity permutation:*

$$c_1v_1 \mapsto \iota, c_2v_1 \mapsto (1\ 2), c_1v_2 \mapsto (1\ 2\ 3), c_3v_2 \mapsto (1\ 3\ 2),$$
$$c_2v_3 \mapsto (1\ 3), c_2v_4 \mapsto (1\ 3), c_3v_4 \mapsto (1\ 2).$$

The result of the lift, $\hat{G}$, is the Tanner graph corresponding to the parity check

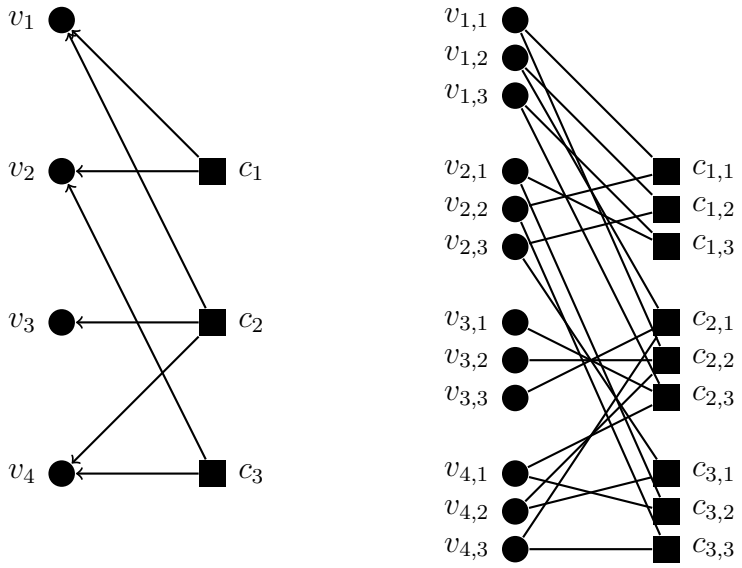


Figure 2.6: A Tanner graph G and a degree 3 lift $\hat{G}$

matrix

$$H' = \left[\begin{array}{ccc|ccc|ccc|ccc} 1&0&0&0&1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&1&0&0&0&0&0&0 \\ 0&0&1&1&0&0&0&0&0&0&0&0 \\ \hline 0&1&0&0&0&0&0&0&1&0&0&1 \\ 1&0&0&0&0&0&0&1&0&0&1&0 \\ 0&0&1&0&0&0&1&0&0&1&0&0 \\ \hline 0&0&0&0&0&1&0&0&0&0&1&0 \\ 0&0&0&1&0&0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&1 \end{array}\right].$$

□

This motivates a definition for *matrix lifts*. A degree ℓ lift of a matrix $M \in \mathbb{F}_q^{k\times N}$ is a matrix $\hat{M} \in \mathbb{F}_q^{k\ell\times N\ell}$ such that the $\ell \times \ell$ submatrix of $\hat{M}$ given by rows $i\ell, i\ell+1, \ldots, i\ell+(\ell-1)$ and columns $j\ell, j\ell+1, \ldots, j\ell+(\ell-1)$ has row and column

weight equal to $m_{i,j}$. In particular, if $m_{i,j} = 1$, this submatrix is a permutation matrix. For ease of notation, we call this submatrix $\hat{M}_{i,j}$.

2.2 Reproducible codes

In addition to lifting graphs to obtain long LDPC codes, large parity check matrices can be obtained through other methods. In the McEliece cryptosystem, as mentioned in Chapter 1, the primary difficulty with using the generator or parity check matrices of codes as keys is that the matrices take too much data to store [1]. This gives some motivation for compact representation of codes, and in response, Santini, et al. introduced the idea of *reproducible codes* in [27].

A matrix $A \in \mathbb{F}_q^{k \times n}$ is said to be *reproducible* [27] if A can be entirely described by a strict subset of its rows (called the signature set) and a family of linear transformations on that subset. Similarly, a linear code $\mathcal{C}$ over $\mathbb{F}_q$ is *reproducible* [27] if it can be represented by a reproducible generator or parity check matrix. One example of a well-studied family of reproducible codes is *cyclic codes*.

Definition 2.2.1 (Cyclic Code)**.** *A linear code $\mathcal{C}$ is a* cyclic code *if every cyclic shift of a codeword in $\mathcal{C}$ is also a codeword in $\mathcal{C}$.*

It is known that $[n, k]$ cyclic codes have generator matrices given by taking a particular codeword and $k - 1$ shifts of the codeword as rows. They also have parity check matrices constructed similarly. Since we can describe the generator or parity check matrix of a cyclic code using a single row and the family of shift permutations, these are reproducible codes.

Example 2.2.2. *Consider the binary cyclic code $\mathcal{C}$ generated by*

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

This matrix is reproducible, since it can be described as the signature row $a = (1, 1, 0, 1, 0, 0, 0)$ with linear transformations given by M, M^2, and M^3, where

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

Alternatively, the rows of G are given by the vectors a, Ma^T, M^2a^T, and M^3a^T. Thus, the code is also reproducible. □

Generally, more structured codes, such as reproducible codes, have worse performance than codes that are closer to random. The condition of reproducibility quickly leads to a less structured generalization of *quasi-reproducible* matrices and codes. A matrix $A \in \mathbb{F}_q^{k \times n}$ is said to be *quasi-reproducible* [27] if A is an array of reproducible matrices $A_{i,j}$. A linear code $\mathcal{C}$ over $\mathbb{F}_q$ is *quasi-reproducible* [27] if it can be represented by a quasi-reproducible generator or parity check matrix. Again, we find a well-studied family of quasi-reproducible codes in *quasi-cyclic codes*.

Definition 2.2.3 (Quasi-Cyclic Code)**.** *A code $\mathcal{C}$ is ℓ-quasi-cyclic if shifting any codeword by ℓ symbol positions yields another codeword.*

In this case, we can consider the generator matrix of a quasi-cyclic matrix as an array of generator matrices for cyclic codes, and thus are quasi-reproducible, as shown in Example 2.2.4.

Example 2.2.4. *Consider the 3-quasi-cyclic code $\mathcal{C}$ generated by*

$$G = \left[\begin{array}{ccc|ccc|ccc|ccc|ccc} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right].$$

By permuting the columns, we obtain a generator matrix G' for a permutation equivalent code (i.e. the rows and columns of a generator matrix of one code can be permuted to obtain a generator matrix for the other):

$$G' = \left[\begin{array}{ccccc|ccccc|ccccc} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right].$$

Note that G' is an array of cyclic matrices, which are each reproducible. Thus, G' is a quasi-reproducible matrix. Since $\mathcal{C}$ is equivalent to a quasi-reproducible code, we consider $\mathcal{C}$ to be a quasi-reproducible code. □

2.2.1 n-adic codes

Cyclic and quasi-cyclic codes have been studied extensively in [25, 26] and elsewhere. In this dissertation, we will consider another family of reproducible and quasi-reproducible codes, that formed by *n-adic matrices*. *Dyadic* (2-adic) matrices have been considered in constructing codes for alternative implementations of the McEliece cryptosystem in [21] and [24], the former of which was cryptanalyzed [9].

To establish some notation before giving the definition of n-adic matrices and codes, note that we will be working with $n^\ell \times n^\ell$ matrices with rows and column indexed by $\mathbb{Z}_n^\ell$. We equip $\mathbb{Z}_n^\ell$ with a total order. Let $[m] = \{0, 1, \ldots, m-1\}$, as indices will begin with 0. The order on $\mathbb{Z}_n^\ell$ is determined by the bijection

$$\varphi : \mathbb{Z}_n^\ell \to [n^\ell],\ a = (a_{\ell-1}, \cdots, a_0) \mapsto \tilde{a} \equiv a_{\ell-1}n^{\ell-1} + a_{\ell-2}n^{\ell-2} + \cdots + a_0.$$

The order of an element a in the group $\mathbb{Z}_n^\ell$ will be denoted by $\#\langle a \rangle$.

Definition 2.2.5 (n-adic Matrix)**.** *A matrix $M \in \mathbb{F}_q^{n^\ell \times n^\ell}$ is called* n-adic *if $m_{a,b} = m_{0,a+b}$ for all $a, b \in \mathbb{Z}_n^\ell$ and $0 \in \mathbb{Z}_n^\ell$. For the special case of $n = 2$, the matrix is called* dyadic*. The* signature row *of an n-adic matrix is its first row. We will write $M_{\mathbf{s}}$ for the n-adic matrix with signature $\mathbf{s}$. The* weight *of an n-adic matrix is the number of nonzero elements in its signature row. When a matrix has weight 1, we will use the notation M_a for $M_{\mathbf{s}}$ where $s_i = \begin{cases} k & i = a \\ 0 & i \neq a. \end{cases}$ for some nonzero k. (Any nonzero value k will give the same results for those considered in this dissertation.)*

Note that a similar definition is given in [3].

Example 2.2.6 (Dyadic Matrix)**.** *Let elements a, b, c, d form the signature row of a*

4×4 matrix, and label each row and column with its binary string representation.

$$\begin{array}{c} \\ (0,0) \\ (0,1) \\ (1,0) \\ (1,1) \end{array} \begin{array}{c} \begin{array}{cccc} (0,0) & (0,1) & (1,0) & (1,1) \end{array} \\ \begin{bmatrix} a & b & c & d \\ m_{01,00} & m_{01,01} & m_{01,10} & m_{01,11} \\ m_{10,00} & m_{10,01} & m_{10,10} & m_{10,11} \\ m_{11,00} & m_{11,01} & m_{11,10} & m_{11,11} \end{bmatrix} \end{array}$$

We form a dyadic matrix as follows. By definition, we know $m_{01,00} = m_{0,(0,1)+(0,0)} = m_{00,01} = b$. Similarly, the values for $m_{i,j}$ for the remaining i, j can be determined by the elements of the signature row. This yields the dyadic matrix

$$\begin{array}{c} \\ (0,0) \\ (0,1) \\ (1,0) \\ (1,1) \end{array} \begin{array}{c} \begin{array}{cccc} (0,0) & (0,1) & (1,0) & (1,1) \end{array} \\ \begin{bmatrix} a & b & c & d \\ b & a & d & c \\ c & d & a & b \\ d & c & b & a \end{bmatrix} \end{array}.$$

Thus, a 4×4 dyadic matrix has the above form, where determining values for a, b, c, d gives the entire matrix. □

We see that n-adic matrices are reproducible, since we determine the matrix by the signature row and a series of permutations of those elements. As in Example 2.2.6, we can consider forming a triadic (3-adic) matrix with a signature row of weight 2 in Example 2.2.7.

Example 2.2.7 (Triadic Matrix)**.** *The following is an example of a weight 2 triadic matrix with rows and columns labeled with their ternary string representations:*

$$
\begin{array}{c}
 \\
(0,0) \\
(0,1) \\
(0,2) \\
(1,0) \\
(1,1) \\
(1,2) \\
(2,0) \\
(2,1) \\
(2,2)
\end{array}
\begin{array}{c}
\begin{array}{ccccccccc}
(0,0) & (0,1) & (0,2) & (1,0) & (1,1) & (1,2) & (2,0) & (2,1) & (2,2)
\end{array} \\
\left[\begin{array}{ccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}\right]
\end{array}
$$

□

We can obtain the quasi-reproducible generalization of n-adic matrices by considering *quasi-n-adic matrices*. A matrix is *quasi-n-adic* if it given by an array of n-adic blocks. As with reproducible and quasi-reproducible codes, a code $\mathcal{C}$ is *n-adic* if it has an n-adic generator or parity check matrix and *quasi-n-adic* if it has a quasi-n-adic generator or parity check matrix. An *n-adic array of matrices* is an $n^\ell \times n^\ell$ array of matrices $[M_{a,b}]_{a,b\in\mathbb{Z}_n^\ell}$ such that $M_{a,b} = M_{0,a+b}$ for all $a, b \in \mathbb{Z}_n^\ell$.

For ease of notation, when referring to a row or column in a quasi-n-adic matrix, we will give an ordered pair (a, b), where a represents the n-adic block in which the row or column is found, indexed from 0, and $b \in \mathbb{Z}_n^\ell$ represents the row or column considered within the n-adic block.

Example 2.2.8. *The following is a quasi-dyadic matrix, since it is a* 2×3 *array of* 4×4 *blocks, where each* 4×4 *block is a dyadic matrix:*

$$\left[\begin{array}{c|c|c} D_{0,0} & D_{0,1} & D_{0,2} \\ \hline D_{1,0} & D_{1,1} & D_{1,2} \end{array}\right] = \left[\begin{array}{cccc|cccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1^{*} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{array}\right]$$

In this example, $D_{1,0}$ is a weight 0 dyadic matrix, $D_{0,0}$, $D_{0,1}$, and $D_{1,1}$ are weight 1 dyadic matrices, and $D_{0,2}$ and $D_{1,2}$ are weight 2 dyadic matrices.

The second nonzero element of the signature row of $D_{1,2}$, denoted with an asterisk in the matrix above, can be described as being in row $(1,(0,0))$ and column $(2,(0,1))$.

□

In this dissertation, we will consider n-adic and quasi-n-adic codes that have n-adic or quasi-n-adic parity check matrix representations.

Chapter 3

Properties of n-adic matrices

We now consider some structural properties of n-adic and quasi-n-adic matrices to better understand their potential for coding theory and other applications. We start our analysis with an illustrative example that establishes a recursive structure of n-adic matrices.

Example 3.0.1. *Consider a general* 8×8 *dyadic matrix:*

$$\begin{bmatrix} a & b & c & d & e & f & g & h \\ b & a & d & c & f & e & h & g \\ c & d & a & b & g & h & e & f \\ d & c & b & a & h & g & f & e \\ e & f & g & h & a & b & c & d \\ f & e & h & g & b & a & d & c \\ g & h & e & f & c & d & a & b \\ h & g & f & e & d & c & b & a \end{bmatrix}$$

This matrix can be written as a dyadic array of general 4×4 *dyadic matrices:*

$$\left[\begin{array}{cccc|cccc} a & b & c & d & e & f & g & h \\ b & a & d & c & f & e & h & g \\ c & d & a & b & g & h & e & f \\ d & c & b & a & h & g & f & e \\ \hline e & f & g & h & a & b & c & d \\ f & e & h & g & b & a & d & c \\ g & h & e & f & c & d & a & b \\ h & g & f & e & d & c & b & a \end{array}\right]$$

Each of the 4×4 *dyadic matrices can be written as a dyadic array of* 2×2 *dyadic matrices:*

$$\left[\begin{array}{cc|cc|cc|cc} a & b & c & d & e & f & g & h \\ b & a & d & c & f & e & h & g \\ \hline c & d & a & b & g & h & e & f \\ d & c & b & a & h & g & f & e \\ \hline e & f & g & h & a & b & c & d \\ f & e & h & g & b & a & d & c \\ \hline g & h & e & f & c & d & a & b \\ h & g & f & e & d & c & b & a \end{array}\right]$$

□

As Example 3.0.1 shows, a dyadic matrix can be written as a 2×2 dyadic array of dyadic matrices with smaller dimension. This property generalizes for all n-adic matrices.

Theorem 3.0.2. *A matrix* $M \in \mathbb{F}_q^{n^\ell \times n^\ell}$ *is* n*-adic with* $\ell \geq 2$ *if and only if* M *can be written as a* $n \times n$ n*-adic array of* n*-adic matrices in* $\mathbb{F}_q^{n^{\ell-1} \times n^{\ell-1}}$.

Proof. If a matrix $M \in \mathbb{F}_q^{n^\ell \times n^\ell}$ is n-adic, then by definition, $m_{a,b} = m_{0,a+b}$ for $a = (a_{\ell-1}, \ldots, a_0), b = (b_{\ell-1}, \ldots, b_0) \in \mathbb{Z}_n^\ell$. Note that the first $n^{\ell-1}$ rows of M have $a_{\ell-1} = 0$, the second $n^{\ell-1}$ rows have $a_{\ell-1} = 1$, and likewise through the last $n^{\ell-1}$ rows with $a_{\ell-1} = n-1$. Similarly, we have values for $b_{\ell-1}$ consistent over blocks of $n^{\ell-1}$ columns.

Consider some such block of $n^{\ell-1}$ rows and columns, M'. We have constant values $a_{\ell-1} = r$, $b_{\ell-1} = c$ for each of the elements in M', and so for each $m_{a,b}$ in this block, we know $(a+b)_{\ell-1}$ is the constant $r+c$. The remaining elements of $(a+b)$ iterate through values of $\mathbb{Z}_n^{\ell-1}$. Thus, we see for any $m_{a,b}$ in M', we have

$$m_{a,b} = m_{0,a+b} = m_{0,(r+c,a_{\ell-2}+b_{\ell-2},\ldots,a_0+b_0)},$$

and there is a corresponding element in the first row of M' given by

$$\begin{aligned} m'_{0,(a_{\ell-2}+b_{\ell-2},\ldots,a_0+b_0)} &= m_{(r,0,\ldots,0),(c,a_{\ell-2}+b_{\ell-2},\ldots,a_0+b_0)} \\ &= m_{0,(r,0,\ldots,0)+(c,a_{\ell-2}+b_{\ell-2},\ldots,a_0+b_0)} \\ &= m_{0,(r+c,a_{\ell-2}+b_{\ell-2},\ldots,a_0+b_0)}. \end{aligned}$$

Because these have the same value, M' is n-adic.

We also see that these n-adic blocks form an n-adic array. Values in the r, c block are determined by elements $m_{0,(r+c,a_{\ell-2}+b_{\ell-2},\ldots,a_0+b_0)}$, which are the values in the $r+c$ block of the first row. Thus, the blocks form an n-adic array.

Similarly, consider an $n \times n$ n-adic array of n-adic matrices in $\mathbb{F}_q^{n^{\ell-1} \times n^{\ell-1}}$, $M = [M'_{a,b}]_{a,b \in \mathbb{Z}_n}$. Here, the $m_{a,b}$ element is in $M'_{a_{\ell-1},b_{\ell-1}}$ in row $(a_{\ell-2}, \ldots, a_0)$ and column

$(b_{\ell-2}, \ldots, b_0)$. Thus, since $M'_{a_{\ell-1},b_{\ell-1}}$ is n-adic, we know

$$\begin{aligned} m_{a,b} &= m'_{(a_{\ell-2},\ldots,a_0),(b_{\ell-2},\ldots,b_0)} \\ &= m'_{0,(a_{\ell-2}+b_{\ell-2},\ldots,a_0+b_0)} \\ &= m_{(a_{\ell-1},0,\ldots,0),(b_{\ell-1},a_{\ell-2}+b_{\ell-2},\ldots,a_0+b_0)}, \end{aligned}$$

and since the array of matrices is n-adic, we know $M'_{a_{\ell-1},b_{\ell-1}}$ is the same as $M'_{0,(a_{\ell-1}+b_{\ell-1})}$. Thus,

$$m_{a,b} = m_{(a_{\ell-1},0,\ldots,0),(b_{\ell-1},a_{\ell-2}+b_{\ell-2},\ldots,a_0+b_0)} = m_{0,(a_{\ell-1}+b_{\ell-1},a_{\ell-2}+b_{\ell-2},\ldots,a_0+b_0)} = m_{0,a+b},$$

and so M is n-adic. □

This recursive definition yields a simple characterization for when the lift of an n-adic matrix is itself n-adic.

Corollary 3.0.3. *A degree d lift $\hat{M}$ of an n-adic matrix $M \in \mathbb{F}_q^{n^\ell \times n^\ell}$ is also n-adic if and only if each $\hat{M}_{a,b}$ is n-adic and $\hat{M}_{a,b} = \hat{M}_{0,a+b}$ for all $a, b \in \mathbb{Z}_n^\ell$.*

Proof. Note that if $\hat{M}$ and M are both n-adic, then they have dimensions $n^{\ell+d} \times n^{\ell+d}$ and $n^\ell \times n^\ell$, respectively, for some $\ell, d \in \mathbb{N}$. Also, if each $\hat{M}_{a,b}$ is n-adic, then each has dimension $n^d \times n^d$ for some d. Then, since M is n-adic with dimension $n^\ell \times n^\ell$ for some ℓ, $\hat{M}$ will have dimension $n^{\ell+d} \times n^{\ell+d}$. Thus, each matrix has the appropriate dimensions to be n-adic.

From Theorem 3.0.2, we know that $\hat{M}$ is n-adic if and only if $\hat{M}$ is an $n \times n$ n-adic array of n-adic matrices. Applying this theorem ℓ times gives us that $\hat{M}$ is n-adic if and only if $\hat{M}$ is an $n^\ell \times n^\ell$ n-adic array of $n^d \times n^d$ n-adic matrices, which gives us the conclusion. □

From the definition of n-adic matrices, it immediately follows that n-adic matrices over a field are symmetric, since for an n-adic matrix M, $m_{a,b} = m_{0,a+b} = m_{0,b+a} = m_{b,a}$. Additionally, Santini et al. demonstrated that dyadic matrices over a field form a commutative ring [28]. We show that for other values of n, while the matrices form an abelian group under addition, the set of n-adic matrices is neither closed under multiplication nor commutative.

Theorem 3.0.4. *For all $n \geq 3$ and all $\ell \geq 1$, the set of n-adic matrices of dimension $n^\ell \times n^\ell$ over a field $\mathbb{F}$ is neither closed under multiplication nor commutative. Additionally, I_{n^ℓ} is not n-adic for $n \geq 3$ and $\ell \geq 1$.*

Proof. First, we claim that the anti-diagonal matrix with all 1's on the anti-diagonal, D, is an n-adic matrix. Since we know from Theorem 3.0.2 that n-adic matrices of dimension $n^\ell \times n^\ell$ are given by an n-adic matrix of dimension $n \times n$ with entries n-adic matrices of dimension $n^{\ell-1} \times n^{\ell-1}$, it suffices to show that this holds for $\ell = 1$. For larger values of ℓ, we can make an n-adic array of n-adic matrices by replacing each nonzero entry of D with another copy of D, yielding the anti-diagonal matrix for the next size of n-adic matrices. For elements $d_{a,b}$ along the anti-diagonal, we have $a + b = n - 1$, so each of these entries is equal to $d_{0,n-1}$, and thus, this matrix is n-adic.

We know that given any matrix M, MD is given by reversing the order of the columns of M and DM is given by reversing the order of the rows of M. In particular, we know that $D^2 = I_{n^\ell}$. Note that I_{n^ℓ} is not n-adic for $n \geq 3$, since the first row only has one nonzero entry, which occurs in the first column, but the second row has a nonzero entry in the second column. This gives $i_{(0,\ldots,0,1),(0,\ldots,0,1)} = 1 \neq 0 = i_{0,(0,\ldots,0,2)} = i_{0,(0,\ldots,0,1)+(0,\ldots,0,1)}$. Thus, the set of n-adic matrices for $n \geq 3$ is not closed under multiplication, and I_{n^ℓ} is not n-adic.

Since D is an n-adic matrix, it remains to show that there is an n-adic matrix for which reversing the rows and reversing the columns results in two different matrices. Consider M to be the n-adic permutation matrix with signature row $(1, 0, \ldots, 0)$. We see that the only corner of M which has a nonzero element is $M_{0,0}$, since $n \geq 3$, and so $(n-1)+(n-1) \neq 0$ as elements of $\mathbb{Z}_n$. Thus, the only nonzero corner of MD is in the first row and final column, but the only nonzero corner of DM is in the final row and first column, and so $MD \neq DM$. Thus, the set of n-adic matrices of dimension $n^\ell \times n^\ell$ over $\mathbb{F}$ is not commutative for all $n \geq 3$. □

3.1 Dyadic matrices

As shown in Theorem 3.0.4, n-adic matrices are generally neither commutative nor closed under multiplication. However, since dyadic matrices fulfill these conditions and form a commutative ring, we consider properties of the ring of dyadic matrices. Smarandache and Vontobel give an upper bound for the minimum distance of quasi-cyclic codes in [30] by considering an isomorphism between the ring of cyclic matrices over $\mathbb{F}$ and a polynomial ring. We consider a similar isomorphism between the ring of dyadic matrices over $\mathbb{F}$ and another polynomial ring.

Theorem 3.1.1. *The ring of $2^\ell \times 2^\ell$ dyadic matrices over $\mathbb{F}$ is isomorphic to*

$$\mathbb{F}[x_0, \ldots, x_{\ell-1}]/\langle x_0^2 - 1, x_1^2 - 1, \ldots, x_{\ell-1}^2 - 1\rangle.$$

Proof. Any dyadic matrix $M \in \mathbb{F}^{2^\ell \times 2^\ell}$ can be represented by its signature row $(m_a)_{a \in \mathbb{Z}_2^\ell}$. Consider the function ϕ from the ring of dyadic matrices (represented

by their signature rows) in $\mathbb{F}^{2^\ell \times 2^\ell}$ to $\mathbb{F}[x_0, \ldots, x_{\ell-1}]/\langle x_0^2 - 1, \ldots, x_{\ell-1}^2 - 1\rangle$ given by

$$\phi\left((m_a)_{a\in\mathbb{Z}_2^\ell}\right) = \sum_{a\in\mathbb{Z}_2^\ell} m_a x_0^{a_0} \cdots x_{\ell-1}^{a_{\ell-1}}.$$

We will show that this function is an isomorphism of rings.

First, since the identity matrix is dyadic with signature row $(1, 0, \ldots, 0)$, we have $\phi(I_{2^\ell}) = \phi(1, 0, \ldots, 0) = 1$. Additionally, for dyadic matrices M and W, the matrix $M + W$ is also dyadic with signature row $(m_a + w_a)_{a\in\mathbb{Z}_2^\ell}$. Thus,

$$\begin{aligned}
\phi(M+W) &= \sum_{a\in\mathbb{Z}_2^\ell} (m_a + w_a) x_0^{a_0} \cdots x_{\ell-1}^{a_{\ell-1}} \\
&= \sum_{a\in\mathbb{Z}_2^\ell} m_a x_0^{a_0} \cdots x_{\ell-1}^{a_{\ell-1}} + \sum_{a\in\mathbb{Z}_2^\ell} w_a x_0^{a_0} \cdots x_{\ell-1}^{a_{\ell-1}} \\
&= \phi(M) + \phi(W).
\end{aligned}$$

Moreover, the a, b entry of $Z := MW$ is given by $\sum_{c\in\mathbb{Z}_2^\ell} m_{a,c} w_{c,b}$. Since both M and W are dyadic matrices, we have

$$\begin{aligned}
z_{a,b} &= \sum_{c\in\mathbb{Z}_2^\ell} m_{a,c} w_{c,b} \\
&= \sum_{c\in\mathbb{Z}_2^\ell} m_{0,a+c} w_{0,c+b} \\
&= \sum_{c\in\mathbb{Z}_2^\ell} m_{0,c} w_{0,c+a+b} && \text{by reindexing} \\
&= \sum_{c\in\mathbb{Z}_2^\ell} m_{0,c} w_{c,a+b} \\
&= z_{0,a+b}.
\end{aligned}$$

Thus, MW is a dyadic matrix with signature row $\left(\sum_{b\in\mathbb{Z}_2^\ell} m_b w_{a+b}\right)_{a\in\mathbb{Z}_2^\ell}$. This gives

$$\begin{aligned}\phi(MW) &= \sum_{a\in\mathbb{Z}_2^\ell}\left(\sum_{b\in\mathbb{Z}_2^\ell} m_b w_{a+b}\right) x_0^{a_0}\cdots x_{\ell-1}^{a_{\ell-1}} \\ &= \left(\sum_{a\in\mathbb{Z}_2^\ell} m_a x_0^{a_0}\cdots x_{\ell-1}^{a_{\ell-1}}\right)\left(\sum_{a\in\mathbb{Z}_2^\ell} w_a x_0^{a_0}\cdots x_{\ell-1}^{a_{\ell-1}}\right) \\ &\quad \text{by factoring and reindexing} \\ &= \phi(M)\phi(W).\end{aligned}$$

So ϕ is a ring homomorphism.

Next, we want to show ϕ is an isomorphism. Given $(m_a)_{a\in\mathbb{Z}_2^\ell} \in \ker\phi$, we know

$$\sum_{a\in\mathbb{Z}_2^\ell} m_a x_0^{a_0}\cdots x_{\ell-1}^{a_{\ell-1}} = 0,$$

and so $m_a = 0$ for all a. Thus, ϕ is injective. We also see that any element of

$$\mathbb{F}[x_0,\ldots,x_{\ell-1}]/\langle x_0^2-1,\ldots,x_{\ell-1}^2-1\rangle$$

can be written as $\sum_{a\in\mathbb{Z}_2^\ell} m_a x_0^{a_0}\cdots x_{\ell-1}^{a_{\ell-1}}$ for some $(m_a)_{a\in\mathbb{Z}_2^\ell}$ and so equals $\phi\left((m_a)_{a\in\mathbb{Z}_2^\ell}\right)$. Thus, ϕ is surjective. □

Note that as n-adic matrices are not closed under multiplication for $n \geq 3$, this isomorphism does not generalize to n-adic matrices. The proof of Theorem 3.1.1 fails to generalize particularly at the point where $Z := MW$ is shown to be dyadic. We conclude this chapter with an observation of the square of a dyadic matrix.

Theorem 3.1.2. *For a dyadic matrix M over $\mathbb{F}_2$, either $M^2 = 0$ if M has even weight or $M^2 = I$ if M has odd weight.*

Proof. First, note that $M = M^T$, since M is a dyadic matrix. Thus, we will consider the dot product of rows i and j of M in both cases to give the i, j entry of M^2. Let m_i represent row i of M. If M has weight k and $a_1, \ldots, a_k$ are the positions of the nonzero signature row entries of M, we know m_i has nonzero entries in columns $a_1 + i, \ldots, a_k + i$ and m_j has nonzero entries in columns $a_1 + j, \ldots, a_k + j$.

If $i = j$, we see $m_i \cdot m_j = m_i \cdot m_i = k$. Thus, if k is even, $m_i \cdot m_j = 0$, and if k is odd, $m_i \cdot m_j = 1$.

If $i \neq j$ and there are some i', j' such that $a_{i'} + i = a_{j'} + j$, note that we also have $a_{j'} + i = a_{i'} + j$. Thus, we have pairs of entries that are nonzero in both rows m_i and m_j, and so $m_i \cdot m_j = 0$. If there is no i', j' such that $a_{i'} + i = a_{j'} + j$, then the dot product is again 0.

Thus, the claim holds. □

Chapter 4

Dual and dimension of dyadic codes

The dimension of a code is a fundamental parameter and is used to calculate the rate and other code parameters, as well. Thus, it is natural to consider the dimension of dyadic codes. In this chapter, we will consider codes with dyadic parity check matrices, along with their duals. As a result, the codes will have length $n = 2^\ell$. By Corollary 3.1.2, dyadic matrices with signatures of odd weight are invertible, and so we will consider only matrices with signatures of even weight. For a dyadic matrix $M_{\mathbf{v}}$, we will denote $\mathcal{C}_{\mathbf{v}}$ the linear code with parity check matrix $M_{\mathbf{v}}$.

We begin by showing that a code with a dyadic parity check matrix of even weight contains its dual, which then has implications for the dimension of the code.

Theorem 4.0.1. *Let $M_{\mathbf{v}}$ be a dyadic matrix where $\mathbf{v}$ has even weight, and consider the linear code $\mathcal{C}_{\mathbf{v}}$ with parity check matrix $M_{\mathbf{v}}$. Then $\mathcal{C}_{\mathbf{v}}$ is* dual-containing*, that is, $\mathcal{C}_{\mathbf{v}}^\perp \subseteq \mathcal{C}_{\mathbf{v}}$. As a consequence,* $\dim(\mathcal{C}_{\mathbf{v}}) \geq n/2 = 2^{\ell-1}$.

Proof. Since $M_{\mathbf{v}}$ is a parity check matrix for $\mathcal{C}_{\mathbf{v}}$, it is a generator matrix for $\mathcal{C}_{\mathbf{v}}^\perp$. Thus, $\mathcal{C}_{\mathbf{v}}$ is dual-containing if and only if $M_{\mathbf{v}} M_{\mathbf{v}}^T = 0$, since this would imply that every row in the generator matrix of $\mathcal{C}_{\mathbf{v}}^\perp$ is also in $\mathcal{C}_{\mathbf{v}}$. Since $M_{\mathbf{v}}$ is a dyadic matrix with even weight, by Corollary 3.1.2, we have $M_{\mathbf{v}} M_{\mathbf{v}}^T = M_{\mathbf{v}}^2 = 0$, and thus $\mathcal{C}_{\mathbf{v}}^\perp \subseteq \mathcal{C}_{\mathbf{v}}$. Because $\dim(\mathcal{C}^\perp) = n - \dim(\mathcal{C})$ for any length n linear code $\mathcal{C}$, we also see $n =$

$\dim(\mathcal{C}_\mathbf{v}) + \dim(\mathcal{C}_\mathbf{v}^\perp) \leq 2\dim(\mathcal{C}_\mathbf{v})$, and so $\dim(\mathcal{C}_\mathbf{v}) \geq n/2 = 2^{\ell-1}$. □

Along with the lower bound of the dimension of the code, the fact that these codes are dual-containing give results on the distance of the code, as well as a condition for when the dual also has a dyadic parity check matrix.

Corollary 4.0.2. *Let $M_\mathbf{v}$ be a dyadic matrix where $\mathbf{v}$ has even weight, and consider the linear code $\mathcal{C}_\mathbf{v}$ with parity check matrix $M_\mathbf{v}$. Then $d_{min}(\mathcal{C}_\mathbf{v}) \leq \mathrm{wt}(\mathbf{v})$.*

Proof. Note that $\mathbf{v} \in \mathcal{C}_\mathbf{v}^\perp$, since $\mathbf{v}$ is the first row of $M_\mathbf{v}$. Since $\mathcal{C}_\mathbf{v}^\perp \subseteq \mathcal{C}_\mathbf{v}$ by Theorem 4.0.1, we know $\mathbf{v} \in \mathcal{C}_\mathbf{v}$, and so $d_{\min}(\mathcal{C}_\mathbf{v}) \leq \mathrm{wt}(\mathbf{v})$. □

Corollary 4.0.3. *Let $M_\mathbf{v}$ be a dyadic matrix where $\mathbf{v}$ has even weight, and consider the linear code $\mathcal{C}_\mathbf{v}$ with parity check matrix $M_\mathbf{v}$. Then $\mathcal{C}_\mathbf{v}^\perp$ has a dyadic parity check matrix if and only if $\mathcal{C}_\mathbf{v}^\perp = \mathcal{C}_\mathbf{v}$.*

Proof. By Theorem 4.0.1, we know $\mathcal{C}_\mathbf{v}^\perp \subseteq \mathcal{C}_\mathbf{v}$, so $\dim(\mathcal{C}_\mathbf{v}^\perp) \leq \dim(\mathcal{C}_\mathbf{v})$. Additionally, if $\mathcal{C}_\mathbf{v}^\perp$ has a dyadic parity check matrix, it is also dual-containing, and so $\mathcal{C}_\mathbf{v} \subseteq \mathcal{C}_\mathbf{v}^\perp$. Thus, $\dim(\mathcal{C}_\mathbf{v}) \leq \dim(\mathcal{C}_\mathbf{v}^\perp)$, and so $\dim(\mathcal{C}_\mathbf{v}) = \dim(\mathcal{C}_\mathbf{v}^\perp)$. Therefore, $\mathcal{C}_\mathbf{v} = \mathcal{C}_\mathbf{v}^\perp$. The converse is trivial. □

As we will consider later in this chapter, dual-containing classical codes can be used to construct quantum codes. Before that, we consider the case of self-dual codes, that is, $\mathcal{C}^\perp = \mathcal{C}$.

Theorem 4.0.4. *Consider the dyadic matrix*

$$M_\mathbf{v} = \begin{bmatrix} M_\mathbf{u} & M_\mathbf{w} \\ M_\mathbf{w} & M_\mathbf{u} \end{bmatrix}.$$

If the signatures $\mathbf{u}$ *and* $\mathbf{w}$ *both have odd weight, then the corresponding dyadic code* $\mathcal{C}_{\mathbf{v}}$ *is self-dual.*

Proof. By Corollary 3.1.2, we have that $M_{\mathbf{u}}^2 = M_{\mathbf{w}}^2 = I$ and both $M_{\mathbf{u}}$ and $M_{\mathbf{w}}$ are invertible. Thus, multiplying $M_{\mathbf{v}}$ with the invertible matrix below does not change its rank:

$$\begin{bmatrix} M_{\mathbf{u}} & 0 \\ M_{\mathbf{u}} & M_{\mathbf{w}} \end{bmatrix} \cdot \begin{bmatrix} M_{\mathbf{u}} & M_{\mathbf{w}} \\ M_{\mathbf{w}} & M_{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} M_{\mathbf{u}}^2 & M_{\mathbf{u}}M_{\mathbf{w}} \\ M_{\mathbf{u}}^2 + M_{\mathbf{w}}^2 & M_{\mathbf{u}}M_{\mathbf{w}} + M_{\mathbf{w}}M_{\mathbf{u}} \end{bmatrix}$$
$$= \begin{bmatrix} I & M_{\mathbf{u}}M_{\mathbf{w}} \\ 0 & 2M_{\mathbf{u}}M_{\mathbf{w}} \end{bmatrix}$$
$$= \begin{bmatrix} I & M_{\mathbf{u}}M_{\mathbf{w}} \\ 0 & 0 \end{bmatrix}.$$

Considering the last matrix, we see this product has rank $n/2$ and thus the code $\mathcal{C}_{\mathbf{v}}$ is self-dual by Theorem 4.0.1. □

With the structure of dyadic matrices, there is a relation between splitting the signature row into the first and second halves and splitting the signature row into the first and third quarters, as well as the second and fourth quarters. In particular, row and column permutations can be applied to a dyadic matrix to swap the second and third quarters of the signature and yielding another dyadic matrix. This is described in Thoerem 4.0.5.

Theorem 4.0.5. *If* M *is a* $2^\ell \times 2^\ell$ *dyadic matrix with even weight and nonzero signature row entries* $a_1 = (a_{1,\ell-1}, \ldots, a_{1,0}), \ldots, a_k = (a_{k,\ell-1}, \ldots, a_{k,0})$ *such that there is some* i *such that* $\#\{a_j | a_{j,i} = 1\}$ *is odd, then* M *can be transformed by permuting*

rows and columns to form another dyadic matrix $M' = \begin{bmatrix} M_{\mathbf{u}} & M_{\mathbf{w}} \\ M_{\mathbf{w}} & M_{\mathbf{u}} \end{bmatrix}$ *with dyadic submatrices* $M_{\mathbf{u}}$ *and* $M_{\mathbf{w}}$ *with the weights of signature rows* $\mathbf{u}$ *and* $\mathbf{w}$ *both odd.*

Proof. Let M be a $2^\ell \times 2^\ell$ dyadic matrix as above, and let b be some index such that $\#\{a_j | a_{j,b} = 1\}$ is odd. Note that for some $a \in \mathbb{Z}_2^\ell$, for the corresponding integer $\tilde{a} = a_{\ell-1}n^{\ell-1} + a_{\ell-2}n^{\ell-2} + \cdots + a_0$, we have that $\tilde{a} \in [2^{\ell-1}]$, i.e. a is in the first half of the signature row, if and only if $a_{\ell-1} = 0$. Thus, if $b = \ell - 1$, M is already of the form $M = \begin{bmatrix} M_{\mathbf{u}} & M_{\mathbf{w}} \\ M_{\mathbf{w}} & M_{\mathbf{u}} \end{bmatrix}$ with dyadic submatrices $M_{\mathbf{u}}$ and $M_{\mathbf{w}}$ with the weights of signature rows $\mathbf{u}$ and $\mathbf{w}$ both odd.

If $b \neq \ell - 1$, we can construct M' by swapping pairs of rows and columns $j, \phi(j)$ of M, where $j \in \mathbb{Z}_2^\ell$ and $\phi : \mathbb{Z}_2^\ell \to \mathbb{Z}_2^\ell$ is given by

$$\phi((j_i)_{i\in[\ell]}) = (j'_i)_{i\in[\ell]} \text{ such that } j'_i = \begin{cases} j_b & \text{if } i = \ell - 1 \\ j_{\ell-1} & \text{if } i = b \\ j_i & \text{otherwise} \end{cases}.$$

Note that $\phi^2 = \mathrm{id}$, and so ϕ gives a permutation of the rows and columns of M to obtain M', as well as for M' to obtain M. We see that M' is a dyadic matrix, since $M'_{i,j} = M_{\phi(i),\phi(j)} = M_{0,\phi(i)+\phi(j)} = M'_{\phi(0),\phi(\phi(i)+\phi(j))}$, by applying ϕ and recalling that M is a dyadic matrix. Note that $\phi(0) = 0$ and $\phi(i+j) = \phi(i)+\phi(j)$. Thus, $M'_{i,j} = M'_{0,i+j}$, and so M' is a dyadic matrix. Note that M' now has signature row entries a'_j such that $\#\{a'_j | a_{j,\ell-1} = 1\}$ is odd. Thus, M' can be written as $M' = \begin{bmatrix} M_{\mathbf{u}} & M_{\mathbf{w}} \\ M_{\mathbf{w}} & M_{\mathbf{u}} \end{bmatrix}$ with dyadic submatrices $M_{\mathbf{u}}$ and $M_{\mathbf{w}}$ with the weights of signature rows $\mathbf{u}$ and $\mathbf{w}$ both odd. □

This result gives the following reformulation of Theorem 4.0.4 and a generalization.

Corollary 4.0.6. *If M is a $2^\ell \times 2^\ell$ dyadic matrix with even weight and nonzero signature row entries $a_1 = (a_{1,\ell-1}, \ldots, a_{1,0}), \ldots, a_k = (a_{k,\ell-1}, \ldots, a_{k,0})$ such that $\#\{a_j | a_{j,\ell-1} = 1\}$ is odd, then the dyadic code $\mathcal{C}$ with parity check matrix M is self-dual.*

Theorem 4.0.7. *If M is a $2^\ell \times 2^\ell$ dyadic matrix with even weight and nonzero signature row entries $a_1 = (a_{1,\ell-1}, \ldots, a_{1,0}), \ldots, a_k = (a_{k,\ell-1}, \ldots, a_{k,0})$ such that there is some i such that $\#\{a_j | a_{j,i} = 1\}$ is odd, then the dyadic code $\mathcal{C}$ with parity check matrix M is self-dual.*

Proof. By Theorem 4.0.5, M' can be obtained from M from permuting rows and columns so that $M' = \begin{bmatrix} M_{\mathbf{u}} & M_{\mathbf{w}} \\ M_{\mathbf{w}} & M_{\mathbf{u}} \end{bmatrix}$ with dyadic submatrices $M_{\mathbf{u}}$ and $M_{\mathbf{w}}$ with the weights of signature rows $\mathbf{u}$ and $\mathbf{w}$ both odd. By Corollary 4.0.6, the dyadic code corresponding to M' is self-dual. Since M' is obtained from permuting rows and columns of M, they have the same rank, and the result follows. □

4.1 Dyadic quantum codes

Quantum coding theory uses *qubits* as fundamental units of information. These are elements of $\mathbb{C}^2$ written as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$, and

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

are orthonormal basis vectors. Additional qubits are given by taking tensor products of qubits. For example, two-qubit states are given by $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle +$

$\delta|11\rangle$, where $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$ and

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad |01\rangle = |0\rangle \otimes |1\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix},$$

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \qquad |11\rangle = |1\rangle \otimes |1\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

The fundamental linear transformations on a single qubit are given by the four Pauli matrices:

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

The Pauli group consists of these matrices and their multiplicative factors: $G = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}$. This is then extended to the Pauli group G_n on n qubits by taking n-fold tensor products of matrices in this group.

In [22], Nielsen and Chuang note that rather than describing a quantum state as a vector in $\mathbb{C}^{2n}$, one can describe it by the operators that stabilize it. It turns out that this is often easier than describing the state itself. For example, the quantum

state

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} = \begin{bmatrix} \sqrt{2}/2 \\ 0 \\ 0 \\ \sqrt{2}/2 \end{bmatrix}$$

is stabilized by the operators

$$XX = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \text{ and } ZZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

since $XX|\psi\rangle = |\psi\rangle$ and $ZZ|\psi\rangle = |\psi\rangle$. It is harder to see but still true that $|\psi\rangle$ is the unique quantum state (up to phase shift, which is given by multiplication by a constant) stabilized by these operators [22].

Gottesman introduced *stabilizer codes* in [11]. A *stabilizer group* $\mathcal{S}$ is an abelian subgroup of G_n that does not contain $-I$, and the corresponding *stabilizer code* $\mathcal{C}(\mathcal{S})$ is given by

$$\mathcal{C}(\mathcal{S}) := \{|\psi\rangle : S|\psi\rangle = |\psi\rangle \quad \forall S \in \mathcal{S}\}[11].$$

A stabilizer group $\mathcal{S} \subseteq G_n$ with k independent generators can be represented by the $k \times 2n$ *check matrix* $H_{\mathcal{S}} = (H_X|H_Z)$ over $\mathbb{F}_2$, where for each row corresponding to a generator, the i^{th} entry of H_X is 1 if the generator has an X in the i^{th} position and the i^{th} entry of H_Z is 1 if the generator has a Z in the i^{th} position. In this case, a Y in the generator is indicaded by 1's in both H_X and H_Z. A $[\![n, k, d]\!]$ stabilizer code has length n, dimension k, and minimum distance d. For more information on quantum codes, we refer the reader to [22].

Example 4.1.1. *The five qubit ⟦5,1,3⟧ stabilizer code, which is the smallest code capable of correcting any error in a single qubit, is given by the following stabilizer group [22]:*

$$\mathcal{S} = \langle XZZXI, IXZZX, XIXZZ, ZXIXZ \rangle$$

We can write the corresponding check matrix

$$H_{\mathcal{S}} = [H_X | H_Z] = \left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right].$$

Note that the first generator, $XZZXI$, has an X in the first and fourth positions and a Z in the second and third positions. This corresponds to the first row in the check matrix, where the first and fourth positions of H_X are 1 and the second and third positions of H_Z are 1. The remaining rows are found similarly from the remaining generators. □

The commutivity of the stabilizers of $\mathcal{S}$ is equivalent to self-orthogonality of $H_{\mathcal{S}}$ with respect to the *symplectic inner product*, which is given by $h \odot g := h_X g_Z^T + h_Z g_X^T$ for vectors $h = (h_X, h_Z), g = (g_X, g_Z)$. Thus, we must have some matrix $H_{\mathcal{S}}$ such that $H_{\mathcal{S}} \odot H_{\mathcal{S}} = H_X H_Z^T + H_Z H_X^T = 0$. Because dyadic matrices are commutative and symmetric, choosing dyadic H_X and H_Z over $\mathbb{F}_2$ satisfies this condition immediately, which makes dyadic matrices candidates to consider in the construction of check matrices for stabilizer codes. Additionally, since a dyadic matrix can be written as a dyadic array of dyadic matrices by Theorem 3.0.2, if we have a dyadic matrix H of

even weight over $\mathbb{F}_2$, we have

$$H = \begin{bmatrix} M_1 & M_2 \\ M_2 & M_1 \end{bmatrix}$$

for dyadic matrices M_1 and M_2 of the same parity weight. This gives

$$\begin{aligned} H \odot H &= \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \cdot \begin{bmatrix} M_2 & M_1 \end{bmatrix} + \begin{bmatrix} M_2 \\ M_1 \end{bmatrix} \cdot \begin{bmatrix} M_1 & M_2 \end{bmatrix} \\ &= \begin{bmatrix} M_1 M_2 & M_1^2 \\ M_2^2 & M_2 M_1 \end{bmatrix} + \begin{bmatrix} M_2 M_1 & M_2^2 \\ M_1^2 & M_1 M_2 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

The last equation holds by applying Theorem 3.1.2 and recalling that dyadic matrices commute. Thus, we can also use dyadic matrices of even weight as check matrices for stabilizer codes.

When H_X and H_Z are sparse, they define a quantum LDPC (QLDPC) code. The minimum distance of a stabilizer code $\mathcal{C}(\mathcal{S})$ is found by taking the minimum weight of an operator which commutes with all operators in $\mathcal{S}$ but is not in $\mathcal{S}$. Equivalently, it is the minimum *quantum weight* of a codeword in $\mathcal{C}(\mathcal{S}) \setminus \mathcal{C}(\mathcal{S})^{\perp}$. We can find codewords of $\mathcal{C}(\mathcal{S})$ using a generator or check matrix as in a classical code. To calculate the *quantum weight* (or *symplectic weight*) of a codeword in a stabilizer code, we first note that the code always has even length. Thus, we can consider the first half and the second half of a codeword: $c = (x, z)$. We then find the number of indices such that $x_i = 1$ or $z_i = 1$. This gives us the weight of a codeword, and the minimum distance of a code is found by taking the minimum of this weight over codewords in $\mathcal{C}(\mathcal{S}) \setminus \mathcal{C}(\mathcal{S})^{\perp}$. If $\mathcal{C}(\mathcal{S})$ is self-dual, the minimum is found over nonzero codewords of

$\mathcal{C}(\mathcal{S})$.

Example 4.1.2. *Consider M the following 16×16 dyadic matrix with signature row weight 4 and support $\{0, 1, 11, 15\}$:*

$$M = \left[\begin{array}{c|c} 11000000 & 00010001 \\ 11000000 & 00100010 \\ 00110000 & 01000100 \\ 00110000 & 10001000 \\ 00001100 & 00010001 \\ 00001100 & 00100010 \\ 00000011 & 01000100 \\ 00000011 & 10001000 \\ \hline 00010001 & 11000000 \\ 00100010 & 11000000 \\ 01000100 & 00110000 \\ 10001000 & 00110000 \\ 00010001 & 00001100 \\ 00100010 & 00001100 \\ 01000100 & 00000011 \\ 10001000 & 00000011 \end{array}\right]$$

This matrix has rank 8 and is self-dual by Theorem 4.0.7. Thus, it yields an $[\![8, 0]\!]$

stabilizer code. After row-reduction, the generator matrix of the stabilizer is

$$G = \left[\begin{array}{c|c} 10000100 & 00010010 \\ 01000100 & 00000011 \\ 00100001 & 01001000 \\ 00010001 & 00001100 \\ 00001100 & 00010001 \\ 00000011 & 01000100 \\ 00000000 & 11001100 \\ 00000000 & 00110011 \end{array}\right]$$

Consider the codeword $c = (1,1,1,0,0,0,0,1,0,1,0,1,1,0,0,1)$ *found by adding the first three rows of* G*. We can calculate the quantum weight of* c *by splitting* c *into* $x = (1,1,1,0,0,0,0,1)$ *and* $z = (0,1,0,1,1,0,0,1)$*. Indexing from 0, we have* $x_i = 1$ *or* $z_i = 1$ *for* $i \in \{0,1,2,3,4,7\}$*. Thus, the quantum weight of* c *is 6.*

In Magma, we calculated the minimum quantum weight of nonzero codewords to find that this stabilizer code has minimum distance 4. This is the largest possible minimum distance for a stabilizer code with length 8 and dimension 0, and it is not equivalent with the self-dual cyclic $[\![8,0,4]\!]$ *code given in [12].* □

The self-orthogonality condition for the construction of a stabilizer code can be difficult to satisfy, so we also consider the Calderbank-Shor-Steane (CSS) construction [4, 32]. This gives a method for constructing a self-orthogonal code from two other codes, which can then be used as a stabilizer code.

In the CSS construction of a stabilizer code, we consider two matrices H_X and

H_Z such that $H_Z H_X^T = 0$. This ensures that the matrix

$$H = \left[\begin{array}{c|c} H_X & 0 \\ 0 & H_Z \end{array}\right]$$

satisfies the conditions for a stabilizer code. In this case, to compute the minimum distance of the stabilizer code as above, we can take the minimum Hamming weight of a codeword in $(\mathcal{C}_X \setminus \mathcal{C}_Z^{\perp}) \cup (\mathcal{C}_Z \setminus \mathcal{C}_X^{\perp})$, where $\mathcal{C}_X$ and $\mathcal{C}_Z$ are the classical codes with parity check matrices H_X and H_Z, respectively. Because dyadic parity check codes contain their duals, if one uses $\mathcal{C}_X = \mathcal{C}_Z = \mathcal{C}$, where $\mathcal{C}$ is a dyadic parity check code, this reduces to finding the minimum Hamming weight over $\mathcal{C} \setminus \mathcal{C}^{\perp}$.

If we begin with some dyadic parity check matrix H_X, we can find a dyadic parity check matrix H_Z such that $H_Z H_X^T = 0$ by taking a codeword of $\mathcal{C}_X$ as the signature row for H_Z. Thus, we can use dyadic matrices to give stabilizer codes in the CSS construction, as well, as is shown in Example 4.1.3.

Example 4.1.3. *Consider M a 32×32 dyadic matrix with signature row weight 4, where the nonzero elements of the signature row are in positions 9, 14, 17, and 18 (entries indexed from 0 to 31). As noted above, we can construct a second dyadic matrix using M as the parity-check matrix for a code. From Magma, we found a random codeword, which had weight 12 with nonzero entries in positions 8, 9, 11, 14, 17, 18, 19, 20, 25, 26, 29, and 30. If M' is the dyadic matrix which uses this codeword as its signature row, we see that we can use*

$$H = \left[\begin{array}{c|c} M & 0 \\ 0 & M' \end{array}\right]$$

as a parity check matrix for a QLDPC code. From calculations in Magma, the quan-

tum distance for this code is 4. □

Taking into account various bounds for quantum codes, the stabilizer code in Example 4.1.2 has the largest possible minimum distance given its length and dimension [12]. It remains to be seen whether dyadic constructons of quantum codes, whether directly as check matrices for stabilizer codes or used in the CSS construction, continue to maximize parameters for greater lengths and dimensions.

Chapter 5

Minimum distance of n-adic and quasi-n-adic codes

As shown in Corollary 4.0.2, the minimum distance of a code with dyadic parity check matrix with even signature row weight has minimum distance at most that weight. In this chapter, we derive upper bounds on the minimum distance for quasi-dyadic and quasi-n-adic codes. We will continue to consider n-adic and quasi-n-adic matrices as parity check matrices for codes.

5.1 Minimum distance bound for quasi-dyadic codes

Smarandache and Vontobel give an upper bound for the minimum distance of quasi-cyclic codes in [30] by considering an isomorphism between the ring of cyclic matrices over $\mathbb{F}$ and a polynomial ring. We will use the analogous isomorphism in Theorem 3.1.1 to establish an analogous upper bound. Since n-adic matrices for $n \geq 3$ are neither closed under multiplication nor commutative, this method does not generalize to n-adic matrices.

For ease of notation, we define

$$\mathbb{F}\{\ell\} := \mathbb{F}[x_0, \ldots, x_{\ell-1}]/\langle x_0^2 - 1, x_1^2 - 1, \ldots, x_{\ell-1}^2 - 1\rangle.$$

In this ring, we define the weight of a polynomial $\mathrm{wt}(c(\mathbf{x}))$ to be the number of

nonzero terms of the polynomial. From this isomorphism, we can consider a relation between quasi-dyadic codes and codes over $\mathbb{F}\{\ell\}$.

Theorem 5.1.1. *Given a quasi-dyadic code $\mathcal{C}$ over $\mathbb{F}$ with parity check matrix*

$$H = \begin{bmatrix} H_{0,0} & H_{0,1} & \cdots & H_{0,J-1} \\ H_{1,0} & H_{1,1} & \cdots & H_{1,J-1} \\ \vdots & \vdots & \ddots & \vdots \\ H_{I-1,0} & H_{I-1,1} & \cdots & H_{I-1,J-1} \end{bmatrix}$$

(where each $H_{i,j}$ is an $2^\ell \times 2^\ell$ dyadic matrix over $\mathbb{F}$), we can consider an associated code $\mathcal{C}'$ over $\mathbb{F}\{\ell\}$ with polynomial parity check matrix

$$H(\mathbf{x}) = \begin{bmatrix} h_{0,0}(\mathbf{x}) & h_{0,1}(\mathbf{x}) & \cdots & h_{0,J-1}(\mathbf{x}) \\ h_{1,0}(\mathbf{x}) & h_{1,1}(\mathbf{x}) & \cdots & h_{1,J-1}(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ h_{I-1,0}(\mathbf{x}) & h_{I-1,1}(\mathbf{x}) & \cdots & h_{I-1,J-1}(\mathbf{x}) \end{bmatrix}$$

(where $h_{i,j}(\mathbf{x}) = \phi(H_{i,j})$ as defined in Theorem 3.1.1).

Given a vector $\mathbf{c} = \left((c_{j,a})_{a\in\mathbb{Z}_2^\ell}\right)_{j=0}^{J-1} \in \mathbb{F}^{(2^\ell)J}$, we have an associated polynomial vector $\mathbf{c}(\mathbf{x}) = (c_0(\mathbf{x}), \ldots, c_{J-1}(\mathbf{x})) \in \mathbb{F}\{\ell\}^J$, where $c_j(\mathbf{x}) = \sum_{a\in\mathbb{Z}_2^\ell} c_{j,a} x_0^{a_0} \cdots x_{\ell-1}^{a_{\ell-1}}$. Then

$$H\mathbf{c}^T = \mathbf{0}^T \text{ if and only if } H(\mathbf{x})\mathbf{c}(\mathbf{x})^T = \mathbf{0}^T.$$

Proof. Let $c_j := (c_{j,a})_{a\in\mathbb{Z}_2^\ell}$. We see that $H\mathbf{c}^T = \mathbf{0}^T$ if and only if $\sum_{j=0}^{J-1} H_{i,j}c_j^T = \mathbf{0}^T$ for all $i \in [I]$. Similarly, $H(\mathbf{x})\mathbf{c}(\mathbf{x})^T = \mathbf{0}^T$ if and only if $\sum_{j=0}^{J-1} h_{i,j}(\mathbf{x})c_j(\mathbf{x}) = 0$ for all $i \in [I]$.

Let $\mathbf{s}$ be the signature row of $H_{i,j}$ for some i, j. Note that row k of $H_{i,j}c_j^T$ is

$\sum_{a\in\mathbb{Z}_2^\ell} s_{k+a}c_{j,a}$ and the coefficient of $x_0^{k_0}\dots x_{\ell-1}^{k_{\ell-1}}$ in $h_{i,j}(\mathbf{x})c_j(\mathbf{x})$ is also $\sum_{a\in\mathbb{Z}_2^\ell} s_{k+a}c_{j,a}$. Thus, $\sum_{j=0}^{J-1} H_{i,j}c_j^T = \mathbf{0}^T$ if and only if $\sum_{j=0}^{J-1} h_{i,j}(\mathbf{x})c_j(\mathbf{x}) = 0$, and so the claim holds. □

Lemma 5.1.2. *Let $\mathcal{C}$ be the quasi-dyadic code defined by a polynomial parity check matrix $H(\mathbf{x}) \in \mathbb{F}_2\{\ell\}^{I\times J}$. Let S be a subset of $[J]$ with size $I+1$, and let $\mathbf{c}(\mathbf{x}) = (c_0(\mathbf{x}),\dots,c_{J-1}(\mathbf{x})) \in \mathbb{F}_2^{(2^\ell)J}\{\ell\}$ be defined by*

$$c_j(\mathbf{x}) = \begin{cases} \operatorname{perm}(H_{S\setminus j}(\mathbf{x})) & \text{if } j \in S \\ 0 & \text{otherwise} \end{cases},$$

where $H_{S\setminus j}(\mathbf{x}) \in \mathbb{F}_2\{\ell\}^{I\times I}$ consists of the columns of $H(\mathbf{x})$ with indices in $S\setminus j$. Then $\mathbf{c}(\mathbf{x})$ is a codeword in $\mathcal{C}$.

Proof. Let $S = \{j_0,\dots,j_I\} \subseteq [J]$. To show $\mathbf{c}(\mathbf{x})$ is a codeword in $\mathcal{C}$, we will show that $s(\mathbf{x})^T = H(\mathbf{x})\mathbf{c}(\mathbf{x})^T = \mathbf{0}$. We see that for each $i \in [I]$, the i^{th} component of $s(\mathbf{x})^T$ is given by

$$\begin{aligned} s_i(\mathbf{x}) &= \sum_{j\in[J]} h_{i,j}(\mathbf{x})c_j(\mathbf{x}) \\ &= \sum_{j\in S} h_{i,j}(\mathbf{x})c_j(\mathbf{x}) \\ &= \sum_{j\in S} h_{i,j}(\mathbf{x}) \operatorname{perm}(H_{S\setminus j}(\mathbf{x})) \\ &= \sum_{j\in S} h_{i,j}(\mathbf{x}) \det(H_{S\setminus j}(\mathbf{x})), \end{aligned}$$

with the last equality holding because we are working in $\mathbb{F}_2$. Note that

$$s_i(\mathbf{x}) = \det\left(\begin{bmatrix} h_{i,j_0}(\mathbf{x}) & h_{i,j_1}(\mathbf{x}) & \cdots & h_{i,j_I}(\mathbf{x}) \\ h_{0,j_0}(\mathbf{x}) & h_{0,j_1}(\mathbf{x}) & \cdots & h_{0,j_I}(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ h_{I-1,j_0}(\mathbf{x}) & h_{I-1,j_1}(\mathbf{x}) & \cdots & h_{I-1,j_I}(\mathbf{x}) \end{bmatrix}\right),$$

and so $s_i(\mathbf{x}) = 0$, since the first row of the matrix is repeated at another point in the matrix. Therefore, $\mathbf{c}(\mathbf{x})$ is a codeword of $\mathcal{C}$. □

Theorem 5.1.3. *Let $\mathcal{C}$ be the quasi-dyadic code defined by a polynomial parity-check matrix $H(\mathbf{x}) \in \mathbb{F}_2\{\ell\}^{I\times J}$. Then*

$$d_{\min}(\mathcal{C}) \leq \min_{\substack{S\subseteq[J] \\ |S|=I+1}} \sum_{j\in S} \mathrm{wt}(\mathrm{perm}(H_{S\setminus j}(\mathbf{x}))),$$

where the minimum is only over non-zero values and $\min(\emptyset) := \infty$.

Proof. Let $S \subseteq [J]$ with $|S| = I + 1$ and $\mathbf{c}(\mathbf{x})$ the corresponding codeword of $\mathcal{C}$ from Lemma 5.1.2. Then $\mathbf{c}(\mathbf{x})$ has weight

$$\begin{aligned} \mathrm{wt}(\mathbf{c}(\mathbf{x})) &= \sum_{j\in[J]} \mathrm{wt}(c_j(\mathbf{x})) \\ &= \sum_{j\in S} \mathrm{wt}(c_j(\mathbf{x})) \\ &= \sum_{j\in S} \mathrm{wt}(\mathrm{perm}(H_{S\setminus j}(\mathbf{x}))) \end{aligned}$$

which gives the result. □

We can generalize the result in 5.1.3, as well as the analogous bound in [30], to codes over a general field $\mathbb{F}$ by altering the codewords we find in the process.

Corollary 5.1.4. *Let $\mathcal{C}$ be the quasi-dyadic code defined by a polynomial parity-check matrix $H(\mathbf{x}) \in \mathbb{F}\{\ell\}^{I\times J}$. Then*

$$d_{\min}(\mathcal{C}) \leq \min_{\substack{S=(j_0,\ldots,j_I)\\ j_k\in[J]}} \sum_{k\in[I+1]} \mathrm{wt}(\det(H_{S\setminus j_k}(\mathbf{x}))),$$

where the minimum is only over non-zero values and $\min(\emptyset) := \infty$.

Proof. As in the proof of Lemma 5.1.2, we see $\mathbf{c}(\mathbf{x}) = (c_0(\mathbf{x}), \ldots, c_{J-1}(\mathbf{x})) \in \mathbb{F}^{(2^\ell)J}\{\ell\}$ as given by

$$c_j(\mathbf{x}) = \begin{cases} (-1)^k \det(H_{S\setminus j_k}(\mathbf{x})) & \text{if } j = j_k \in S \\ 0 & \text{otherwise} \end{cases}$$

is a codeword in $\mathcal{C}$, since the i^{th} component of $s(\mathbf{x})^T$ is given by

$$\begin{aligned} s_i(\mathbf{x}) &= \sum_{j\in[J]} h_{i,j}(\mathbf{x})c_j(\mathbf{x}) \\ &= \sum_{k\in[I+1]} h_{i,j_k}(\mathbf{x})c_{j_k}(\mathbf{x}) \\ &= \sum_{k\in[I+1]} h_{i,j_k}(\mathbf{x})(-1)^k \det(H_{S\setminus j_k}(\mathbf{x})). \end{aligned}$$

This gives

$$s_i(\mathbf{x}) = \det\left(\begin{bmatrix} h_{i,j_0}(\mathbf{x}) & h_{i,j_1}(\mathbf{x}) & \cdots & h_{i,j_I}(\mathbf{x}) \\ h_{0,j_0}(\mathbf{x}) & h_{0,j_1}(\mathbf{x}) & \cdots & h_{0,j_I}(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ h_{I-1,j_0}(\mathbf{x}) & h_{I-1,j_1}(\mathbf{x}) & \cdots & h_{I-1,j_I}(\mathbf{x}) \end{bmatrix}\right),$$

and so $s_i(\mathbf{x}) = 0$.

The result follows as in Theorem 5.1.3 by noting that the weight of the codeword

given above is given by $\sum_{k\in[I+1]} \mathrm{wt}(\det(H_{S\setminus j_k}(\mathbf{x})))$. □

5.2 Quasi-n-adic distance bound

The minimum distance of a quasi-n-adic code, where each block is a weight one n-adic matrix, is bounded by the number of rows of n-adic blocks in the parity check matrix. We use the case with only two columns of n-adic blocks to obtain the bound.

Lemma 5.2.1. *Given a length N code $\mathcal{C}$ with parity-check matrix H such that H has row weight 2, and some vector $\mathbf{c}'$ of length N, a codeword $\mathbf{c} \in \mathcal{C}$ can be obtained through the following process:*

1. *Consider the nonzero elements of $\mathbf{c}'$. These correspond to columns of H. Now consider the rows containing nonzero elements in these columns.*
2. *Note that there are only two nonzero elements in each of these rows. Thus, to ensure the checks are satisfied, both must correspond to zero or both to nonzero entries in a codeword. Thus, if only one of the elements corresponds to a nonzero entry in $\mathbf{c}'$, add the other as an appropriate nonzero entry to fulfill the check. Call this new vector $\mathbf{c}_1$.*
3. *If we have $H\mathbf{c}_1^T = \mathbf{0}^T$, we are done. Otherwise, repeat steps 1 and 2 with $\mathbf{c}_1$, continuing until $H\mathbf{c}_i^T = \mathbf{0}^T$. Call this $\mathbf{c}_i$ where the process terminates $\mathbf{c}$.*

Additionally, $\mathrm{wt}(\mathbf{c})$ *will be the minimum over codewords in $\mathcal{C}$ containing the support of $\mathbf{c}'$.*

Proof. First, note that the process given above will terminate, since N is finite and H has row weight 2, so the all 1's vector is a codeword in $\mathcal{C}$. Thus, we also know that $H\mathbf{c}^T = \mathbf{0}^T$, and so $\mathbf{c} \in \mathcal{C}$.

We see that $\mathbf{c}$ will have the fewest possible nonzero elements in addition to those in the support of $\mathbf{c}'$, because each added element was necessary to satisfy some check. Thus, the value of $\mathbf{c}$ does not depend on the order in which we add nonzero elements to $\mathbf{c}'$. □

Example 5.2.2. *Consider*

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

and $\mathbf{c}' = (1,0,0,0,0,0)$. *Following the above process to obtain* $\mathbf{c}$ *with* $H\mathbf{c}^T = \mathbf{0}^T$, *we first consider the first column, which corresponds to the only nonzero entry of* $\mathbf{c}'$. *Thus, we consider the first and third rows, since they have nonzero entries in the first column, and we add nonzero entries to* $\mathbf{c}'$ *corresponding to the other nonzero entries of those rows to obtain* $\mathbf{c}_1 = (1,0,0,1,1,0)$. *We see* $H\mathbf{c}_1^T \neq \mathbf{0}^T$, *so we repeat the process with* $\mathbf{c}_1$.

With $\mathbf{c}_1$, *the nonzero entries correspond to the first, fourth, and fifth columns, which have nonzero entries in the first, third, and fourth rows. Adding the other nonzero entries from those rows gives us* $\mathbf{c}_2 = (1,0,1,1,1,0)$. *We see* $H\mathbf{c}_2^T = \mathbf{0}^T$, *and so we set* $\mathbf{c} = (1,0,1,1,1,0)$. *This gives us a codeword for* $\mathcal{C}$ *with parity-check matrix* H *with minimum weight containing the support of* $\mathbf{c}'$. □

Lemma 5.2.3. *Let* H *be a parity-check matrix of a code* $\mathcal{C}$ *where* H *is a* $j \times 2$ *array*

of weight 1 n-adic matrices:

$$H = \begin{bmatrix} H_{1,1} & H_{1,2} \\ H_{2,1} & H_{2,2} \\ \vdots & \vdots \\ H_{j,1} & H_{j,2} \end{bmatrix}.$$

Given a codeword $\mathbf{c}'$ *of the code* $\mathcal{C}'$ *with parity check matrix*

$$H' = \begin{bmatrix} H_{1,1} & H_{1,2} \\ H_{2,1} & H_{2,2} \\ \vdots & \vdots \\ H_{j-1,1} & H_{j-1,2} \end{bmatrix},$$

let $\mathbf{c}$ *be the vector obtained by adding the fewest possible nonzero elements to* $\mathbf{c}'$ *so that* $\mathbf{c}$ *satisfies the parity check matrix*

$$H'' = \begin{bmatrix} H_{1,1} & H_{1,2} \\ H_{j,1} & H_{j,2} \end{bmatrix}.$$

Then $\mathbf{c}$ *is a codeword in* $\mathcal{C}$*.*

Proof. First, consider some $\mathbf{c}' \in \mathcal{C}'$. Since H'' is made up of four permutation matrices, each row and column has exactly two nonzero entries. Thus, we can make a new vector $\mathbf{c}$ with the fewest possible additional nonzero elements such that $H''\mathbf{c}^T = \mathbf{0}^T$ by following the process given in Lemma 5.2.1. Also, note that this process is equivalent to adding, for each position in the support of $\mathbf{c}'$, all of the variable nodes in the unique cycle of the Tanner graph of H'' containing the variable node corresponding to that position.

Next, we want to show that $H\mathbf{c}^T = \mathbf{0}^T$. We already know that $\mathbf{c}$ satisfies the checks for the submatrices $[H_{1,1}\ H_{1,2}]$ and $[H_{j,1}\ H_{j,2}]$, so consider some check in row m of a submatrix $[H_{i,1}\ H_{i,2}]$ for some $1 < i < j$. If both nonzero entries in the row corresponded to nonzero entries in $\mathbf{c}'$, then this check is still satisfied, because we did not remove any nonzero entries in the process of making $\mathbf{c}$. Otherwise, since $\mathbf{c}'$ satisfied the checks, both nonzero entries in row m must correspond to zero entries in $\mathbf{c}'$. Thus, $\mathbf{c}$ will satisfy the check if and only if both entries became nonzero or both entries remained zero.

Let $a_1, a_2, b_1, b_2, c_1, c_2 \in \mathbb{Z}_n^\ell$ be the positions of the nonzero signature row elements of $H_{1,1}, H_{1,2}, H_{i,1}, H_{i,2}, H_{j,1}$, and $H_{j,2}$, respectively. Then for any $m \in \mathbb{Z}_n^\ell$, row m of $[H_{i,1}\ H_{i,2}]$ has nonzero elements in columns $(0, b_1 - m)$ and $(1, b_2 - m)$ (recall notation for quasi-n-adic matrices as in Example 2.2.8). We are considering the case where both of these entries are zero in $\mathbf{c}'$. Assume without loss of generality that $(0, b_1 - m)$ corresponds to a nonzero entry in $\mathbf{c}$. It suffices to show that $(1, b_2 - m)$ also corresponds to a nonzero entry in $\mathbf{c}$.

Since $(0, b_1 - m)$ corresponds to a nonzero entry in $\mathbf{c}$ but a zero entry in $\mathbf{c}'$, there must be some nonzero entry in $\mathbf{c}'$ which is in the same cycle of the Tanner graph of H'' as the vertex corresponding to $(0, b_1 - m)$. Let α be the index of this entry.

Consider the Tanner graph of H''. Since this graph is 2-regular, each node is in a unique cycle. From the properties of n-adic matrices, we see that the cycle containing the variable node corresponding to $(0, b_1 - m)$ contains exactly the variable nodes corresponding to rows of the form $(0, k(c_1 - c_2 + a_2 - a_1) + b_1 - m)$ and $(1, a_2 - a_1 + k(c_1 - c_2 + a_2 - a_1) + b_1 - m)$ for $k \in \mathbb{Z}_n$. These values can be found by considering the structure of the cycles in the Tanner graph as in the proof of Theorem 6.1.3. Thus, we know α has one of these forms.

Let H_i be the submatrix of H' given by

$$H_i = \begin{bmatrix} H_{1,1} & H_{1,2} \\ H_{i,1} & H_{i,2} \end{bmatrix}.$$

Since $\mathbf{c}'$ satisfies $H'\mathbf{c}'^T = \mathbf{0}^T$, we also have $H_i\mathbf{c}'^T = \mathbf{0}^T$. Also, since H_i is 2-regular, the variable node corresponding to α is in a unique cycle in the Tanner graph. Since the α entry was nonzero in $\mathbf{c}'$, all of the entries of $\mathbf{c}'$ corresponding to elements of this cycle must also have been nonzero. It suffices to show that there is some column which has a corresponding variable node in this cycle, as well as a corresponding variable node in the unique cycle of the Tanner graph of H'' containing the variable node corresponding to $(1, b_2 - m)$. This would ensure that the $(1, b_2 - m)$ entry becomes nonzero in the process of creating $\mathbf{c}$.

As before, we see that the cycle of the Tanner graph of H'' containing the variable node corresponding to $(1, b_2 - m)$ contains exactly the variable nodes of the forms $(0, c_1 - c_2 + k'(c_1 - c_2 + a_2 - a_1) + b_2 - m)$ and $(1, k'(c_1 - c_2 + a_2 - a_1) + b_2 - m)$ for $k' \in \mathbb{Z}_n$.

If α is in the first half of $\mathbf{c}$, then $\alpha = (0, k(c_1 - c_2 + a_2 - a_1) + b_1 - m)$ for some $k \in \mathbb{Z}_n$. Let $\alpha' = k(c_1 - c_2 + a_2 - a_1) + b_1 - m$. Then, in the Tanner graph of H_i, the variable nodes in the unique cycle containing the variable node corresponding to α include those of the form $(0, k''(b_1 - b_2 + a_2 - a_1) + \alpha')$ for $k'' \in \mathbb{Z}_n$. Taking

$k'' = -1(= n-1)$ and $k' = k-1$, we see:

$$\begin{aligned}
&(0, k''(b_1 - b_2 + a_2 - a_1) + \alpha') \\
=&(0, -(b_1 - b_2 + a_2 - a_1) + k(c_1 - c_2 + a_2 - a_1) + b_1 - m) \\
=&(0, c_1 - c_2 + (k-1)(c_1 - c_2 + a_2 - a_1) + b_2 - m) \\
=&(0, c_1 - c_2 + k'(c_1 - c_2 + a_2 - a_1) + b_2 - m).
\end{aligned}$$

Thus, there is a column such that the corresponding vertex in the Tanner graph of H'' is in the same cycle as the variable node corresponding to $(1, b_2 - m)$ and the corresponding vertex in the Tanner graph of H_i is in the same cycle as the variable node corresponding to α. Thus, $(1, b_2 - m)$ is nonzero in $\mathbf{c}$, and the claim holds in this case.

In the other case, $\alpha = (1, a_2 - a_1 + k(c_1 - c_2 + a_2 - a_1) + b_1 - m)$ for some $k \in \mathbb{Z}_n$, and we set $\alpha' = a_2 - a_1 + k(c_1 - c_2 + a_2 - a_1) + b_1 - m$. In the Tanner graph of H_i, the columns with corresponding variable nodes in the cycle containing the variable node corresponding to α include those of the form $(1, k''(b_1 - b_2 + a_2 - a_1) + \alpha')$ for $k'' \in \mathbb{Z}_n$. Taking $k'' = -1(= n-1)$ and $k' = k$, we see:

$$\begin{aligned}
&(1, k''(b_1 - b_2 + a_2 - a_1) + \alpha') \\
=&(1, -(b_1 - b_2 + a_2 - a_1) + a_2 - a_1 + k(c_1 - c_2 + a_2 - a_1) + b_1 - m) \\
=&(1, k(c_1 - c_2 + a_2 - a_1) + b_2 - m) \\
=&(1, k'(c_1 - c_2 + a_2 - a_1) + b_2 - m).
\end{aligned}$$

Again, there is a column such that the corresponding vertex in the Tanner graph of H'' is in the same cycle as the variable node corresponding to $(1, b_2 - m)$ and the corresponding vertex in the Tanner graph of H_i is in the same cycle as the variable

node corresponding to α. Thus, $(1, b_2 - m)$ is nonzero in $\mathbf{c}$, and so $\mathbf{c} \in \mathcal{C}$ in all cases. $\square$

Theorem 5.2.4. *The minimum distance for a quasi-n-adic code with a parity check matrix given by a $j \times k$ array of weight 1 n-adic blocks with $k \geq 2$ is at most $2n^{j-1}$.*

Proof. Let H be a $j \times k$ array of weight 1 n-adic blocks, and let H' be a $j \times 2$ array consisting of the first two columns of blocks of H. Note that for any $\mathbf{c}'$ with $H'\mathbf{c}'^T = \mathbf{0}^T$, we can extend $\mathbf{c}'$ by adding zeros to make it the appropriate length to make a vector $\mathbf{c}$ with $H\mathbf{c}^T = \mathbf{0}^T$. Thus, we will consider the code with parity check matrix H', since the minimum distance for the code with parity check matrix H is less than or equal to the minimum distance for the code with parity check matrix H'.

We proceed by induction on j. If $j = 1$, since each column of the parity check matrix has weight $j > 0$, a nonzero codeword must have weight at least $2 = 2n^{1-1}$. Thus, the bound holds.

For $j > 1$, let H' be the given parity check matrix and H'' be the submatrix of H' given by removing the last row of weight 1 n-adic blocks. So H'' is a $(j-1) \times 2$ array of weight 1 n-adic blocks. By our inductive hypothesis, a minimum weight vector $\mathbf{c}$ such that $H''\mathbf{c}^T = \mathbf{0}^T$ has weight at most $2n^{(j-1)-1}$. By the process in Lemma 5.2.3, we can add nonzero elements to $\mathbf{c}$ to obtain a vector $\mathbf{c}'$ such that $H'\mathbf{c}'^T = \mathbf{0}^T$.

We know that each of the added nonzero elements in $\mathbf{c}'$ correspond to variable nodes in the Tanner graph of a 2×2 array of weight 1 n-adic matrices. By Theorem 6.1.3, which we will prove in the next chapter, it follows that each of these nodes is in a unique cycle and that each of these cycles contains at most $2n$ variable nodes. Since $\mathbf{c}$ satisfied each of the checks in H'' and each cycle contains some check node corresponding to a row in H'', any cycle with a variable node corresponding to a nonzero element in $\mathbf{c}$ must have at least one additional variable node corresponding

to a nonzero element in $\mathbf{c}$ to satisfy that check node. Thus, we add at most an additional $2n-2$ nonzero elements to get $\mathbf{c}'$ for each pair of nonzero elements in $\mathbf{c}$. Thus, the weight of $\mathbf{c}$ is at most $n^{(j-1)-1} \cdot (2n-2) + 2n^{(j-1)-1} = 2n^{j-1}$. □

This bound is tight for the case where $k = 2$ and for values of j such that $j \leq \ell+1$ with $n^\ell \times n^\ell$ n-adic blocks. We give a characterization for certain cases of when this bound is met.

Theorem 5.2.5. *Let $\mathcal{C}$ be the code with parity check matrix H, where H is a $j \times 2$ array of n-adic blocks of weight 1 and dimension $n^\ell \times n^\ell$, and the nonzero signature row entries for the blocks in the first row and column are all in position $0 \in \mathbb{Z}_n^\ell$. Then if $a_1, \ldots, a_{j-1} \in \mathbb{Z}_n^\ell$ are the positions of the signature row entries for the remaining blocks, we have $d_{min}(\mathcal{C}) = 2n^{j-1}$ if and only if the order of a_i is n and $a_i \notin \langle a_1, \ldots, a_{i-1} \rangle$ for each i.*

Proof. Let M_a be the n-adic block of weight 1 and dimension $n^\ell \times n^\ell$ with nonzero signature row entry in position a. Then H has the form:

$$H = \begin{bmatrix} M_{\mathbf{0}} & M_{\mathbf{0}} \\ M_{\mathbf{0}} & M_{a_1} \\ M_{\mathbf{0}} & M_{a_2} \\ \vdots & \vdots \\ M_{\mathbf{0}} & M_{a_{j-1}} \end{bmatrix}.$$

Consider some nonzero codeword $\mathbf{c} \in \mathcal{C}$. Let $\mathbf{c} = \left((c_{i,a})_{a \in \mathbb{Z}_n^\ell}\right)_{i \in \{1,2\}}$. Since each M_a is a permutation matrix, there must be some α such that $c_{1,\alpha}$ is nonzero. In order to satisfy the checks in the block row $[M_{\mathbf{0}} \;\; M_{a_i}]$, we see $c_{2,a_i+\alpha}$ must also be nonzero for each i, and $c_{2,\alpha}$ must be nonzero to satisfy $[M_{\mathbf{0}} \;\; M_{\mathbf{0}}]$. Then, in order to satisfy the checks in the block row $[M_{\mathbf{0}} \;\; M_{\mathbf{0}}]$, we see $c_{1,a_i+\alpha}$ must be nonzero for each i.

Repeating this process of satisfying the checks in $[M_{\mathbf{0}}\ M_{a_i}]$ and in $[M_{\mathbf{0}}\ M_{\mathbf{0}}]$ with each new nonzero entry, we see that each $c_{1,\beta+\alpha}$ and $c_{2,\beta+\alpha}$ must be nonzero for $\beta = \sum_{i=1}^{j-1} b_i a_i$ with nonnegative integers b_i. However, since each a_i is in $\mathbb{Z}_n^\ell$, we only need to consider $b_i \in \{0, \ldots, n-1\}$ to give unique values. Note that this gives $2n^{j-1}$ possible nonzero entries from this process, depending on choices of b_i.

In order for the distance of $\mathcal{C}$ to be maximized at $2n^{j-1}$, each choice of $(b_1, \ldots, b_{j-1})$ must yield a different sum β. This happens if and only if a_i has order at least n and $a_i \notin \langle a_1, \ldots, a_{i-1} \rangle$ for all i. □

Corollary 5.2.6. *A code $\mathcal{C}$ with parity check matrix H, where H is a $j \times k$ array of n-adic blocks of weight 1 and dimension $n^\ell \times n^\ell$, can have minimum distance $2n^{j-1}$ only if $j \leq \ell + 1$.*

Proof. We know $2n^{j-1}$ is the upper bound of the distance for $\mathcal{C}$ by Theorem 5.2.4. Additionally, $d_{\min}(\mathcal{C}) \leq d_{\min}(\mathcal{C}')$, where $\mathcal{C}'$ is the code with parity check matrix H' given by the first two columns of blocks of n-adic matrices of H. We also know $\mathcal{C}'$ is equivalent to a code of the form given in Theorem 5.2.5, so we can reduce to this case.

In $\mathbb{Z}_n^\ell$, there can be at most ℓ elements satisfying the conditions of Theorem 5.2.5. Note that if $a_1, \ldots, a_m$ satisfy that the order of a_i is n and $a_i \notin \langle a_1, \ldots, a_{i-1} \rangle$ for each i, $\langle a_1, \ldots, a_m \rangle$ has order n^m, and $\mathbb{Z}_n^\ell$ has order n^ℓ. Thus, $j - 1 \leq \ell$. □

Thus, quasi-dyadic and quasi-n-adic codes have poor minimum distances as the block lengths increase.

5.3 Examples

The Tanner-Sridhara-Fuja (TSF) codes introduced in [35] are one family of array-based codes with algebraically chosen circulant matrices. In this section, we construct a quasi-triadic code with comparable parameters to such a code. We first review the construction in [35].

Example 5.3.1. *Let a and b be two elements from the multiplicative group of $\mathbb{F}_\ell$, with orders j and r, respectively, and ℓ a prime. Then, the parity check matrix H is defined as*

$$H = \begin{bmatrix} I_1 & I_b & I_{b^2} & \dots & I_{b^{r-1}} \\ I_a & I_{ab} & I_{ab^2} & \dots & I_{ab^{r-1}} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ I_{a^{j-1}} & I_{a^{j-1}b} & I_{a^{j-1}b^2} & \dots & I_{a^{j-1}b^{r-1}} \end{bmatrix},$$

where $I_{a^ib^k}$ is the $p \times p$ identity matrix cyclically shifted to the left by a^ib^k positions, where a^ib^k is interpreted as an element in the integers $\mathbb{Z}_\ell = \{0, 1, \dots, \ell - 1\}$.

For example, consider the $[155, 64, 20]$ code that is obtained by choosing $\ell = 31, a = 5, b = 2$ and $j = 3, r = 5$ in the above construction. The matrix is given by

$$H = \begin{bmatrix} I_1 & I_2 & I_4 & I_8 & I_{16} \\ I_5 & I_{10} & I_{20} & I_9 & I_{18} \\ I_{25} & I_{19} & I_7 & I_{14} & I_{28} \end{bmatrix},$$

where I_x is the 31×31 identity matrix cyclically shifted to the left by x positions. This code has block length 155, dimension 64, minimum distance 20 and the corresponding Tanner graph has a girth of 8. This code is notable for its good distance properties which are competitive with a BCH code of comparable length. □

Example 5.3.2 (Quasi-triadic code)**.** *As a result of Theorem 5.2.4, a construction of a quasi-dyadic code as a* 3×5 *array of dyadic matrices, which would give a similar construction to the TSF code given above, has minimum distance at most 8. However, we can do better by considering a quasi-triadic structure, as shown next.*

Consider the quasi-triadic code $\mathcal{C}$ *of blocklength 135 defined by parity check matrix*

$$H = \begin{bmatrix} T_0 & T_0 & T_0 & T_0 & T_0 \\ T_0 & T_{25} & T_{21} & T_{26} & T_9 \\ T_0 & T_6 & T_{17} & T_{24} & T_{16} \end{bmatrix}$$

where T_i *is the* 27×27 *triadic matrix of weight 1 with the nonzero entry in position* i *of the signature row (entries indexed from 0 to 26). From calculations in Magma, we found that this code has minimum distance 16 and girth 8. The entries for this matrix were found by fixing the first row and column to be 0 and randomly testing other entries to find the best girth and distance. The relative minimum distance of this example is* $\frac{16}{135} \approx 0.1185$ *and is comparable to that of the* $[155, 64, 20]$ *TSF code, which is* $\frac{20}{155} \approx 0.1290$*.* □

Furthermore, several graph automorphism properties were shown to exist within the Tanner graph for TSF codes [34] that provide an algebraic framework for analyzing and understanding the structure and properties of these codes. It will be interesting to apply similar analysis to quasi-n-adic codes.

Chapter 6

Girth and stopping sets

In this chapter, we will comment on the possible girth and stopping set sizes of Tanner graphs for codes with n-adic and quasi-n-adic parity check matrices.

6.1 Girth and cycle structure

While efficient, iterative decoding algorithms are optimal only on cycle-free graphs [38], such graphs do not give rise to good codes [7]. Thus, large girth is desired so that the code graph is locally cycle-free. Moreover, decoder failure is often attributed to the presence of combinatorial structures such as absorbing sets and stopping sets which contain cycles. First, we consider the girth of a Tanner graph corresponding to an n-adic matrix.

Theorem 6.1.1. *The Tanner graph corresponding to an n-adic matrix M of weight 2 with nonzero elements in positions $a, b \in \mathbb{Z}_n^\ell$ of the signature row has girth $2 \cdot \#\langle b - a \rangle$.*

Proof. Let M be a n-adic matrix with nonzero elements in the signature row in positions a and b, and let $p = \#\langle b - a \rangle$.

First, we know that all cycles in the Tanner graph must be even, since it is bipartite, so we can describe a cycle of length $2k$ by indices $c_1, \ldots, c_{2k}$, where $m_{c_i, c_{i+1}}$ is nonzero for odd i and m_{c_{i+1}, c_i} is nonzero for even i (we will consider $2k + 1 \equiv 1$ in

this context). Here, even values of i correspond to variable nodes and odd values of i to check nodes or vice versa. Note that for this to be a cycle of length $2k$, we must have $c_i \neq c_j$ for i, j the same parity and $i \neq j$. Additionally, since $m_{c_i,c_{i+1}}$ is nonzero for odd i and m_{c_{i+1},c_i} is nonzero for even i, we know $c_i + c_{i+1} \in \{a, b\}$ for all i.

Consider $\{c_i\}_{i=1}^{2p}$ given by

$$c_i = \begin{cases} \frac{i-1}{2}(b-a) & i \text{ odd} \\ a + \frac{i-2}{2}(a-b) & i \text{ even} \end{cases}.$$

We want to show that $\{m_{c_i,c_{i+1}} | i \text{ odd}\} \cup \{m_{c_{i+1},c_i} | i \text{ even}\}$ gives a cycle in the Tanner graph. First, we will show that $c_i \neq c_j$ for i, j the same parity and $i \neq j$. For odd i, j, if $c_i = c_j$, we see

$$\begin{aligned} \frac{i-1}{2}(b-a) &= \frac{j-1}{2}(b-a) \\ \left(\frac{j-1}{2} - \frac{i-1}{2}\right)(b-a) &= 0. \end{aligned}$$

Thus, $\left(\frac{j-1}{2} - \frac{i-1}{2}\right) = mp$ for some $m \in \mathbb{Z}$. Additionally, we know $i, j \in \{1, 3, \ldots, 2p-1\}$, and so $\frac{i-1}{2}, \frac{j-1}{2} \in \{0, 1, \ldots, p-1\}$. Thus, $-(p-1) \leq \left(\frac{j-1}{2} - \frac{i-1}{2}\right) \leq p-1$, and so $m = 0$, and so $i = j$.

For even i, j, if $c_i = c_j$, we see

$$\begin{aligned} a + \frac{i-2}{2}(a-b) &= a + \frac{j-2}{2}(a-b) \\ \left(\frac{j-2}{2} - \frac{i-2}{2}\right)(b-a) &= 0. \end{aligned}$$

Thus, $\left(\frac{j-2}{2} - \frac{i-2}{2}\right) = mp$ for some $m \in \mathbb{Z}$. Additionally, we know $i, j \in \{2, 4, \ldots, 2p\}$, and so $\frac{i-2}{2}, \frac{j-2}{2} \in \{0, 1, \ldots, p-1\}$. Thus, $-(p-1) \leq \left(\frac{j-2}{2} - \frac{i-2}{2}\right) \leq p-1$, and so

$m = 0$, and so $i = j$.

To show this is a cycle, it remains to show that $c_i + c_{i+1} \in \{a, b\}$ for all i. For odd i, we see

$$\begin{aligned} c_i + c_{i+1} &= \frac{i-1}{2}(b-a) + a + \frac{(i+1)-2}{2}(a-b) \\ &= a, \end{aligned}$$

and for even i, we see

$$\begin{aligned} c_i + c_{i+1} &= a + \frac{i-2}{2}(a-b) + \frac{(i+1)-1}{2}(b-a) \\ &= a + b - a \\ &= b. \end{aligned}$$

Since we also know $c_{2p} + c_1 = a + \frac{2p-2}{2}(a-b) + \frac{1-1}{2}(b-a) = a + (p-1)(a-b) = a - (a-b) = b$, this gives us a cycle of length $2p$ in the Tanner graph.

Next, we will show that there is no cycle of length less than $2p$ in the Tanner graph. Since we know each cycle is even, we consider some cycle of length $2q$ given by indices $\{c_i\}_{i=1}^{2q}$, where $\{m_{c_i,c_{i+1}} | i \text{ odd}\} \cup \{m_{c_{i+1},c_i} | i \text{ even}\}$ are nonzero elements of the matrix. We can assume that $c_i \neq c_j$ for i, j the same parity and $i \neq j$, and we know $c_i + c_{i+1}, c_1 + c_{2q} \in \{a, b\}$ for all i, as before. We also see that for any i, $c_i + c_{i+1} \neq c_{i+1} + c_{i+2}$, since i and $i+2$ have the same parity with $i \neq i+2$ (since we cannot have 2-cycles), and so $c_i \neq c_{i+2}$. Thus, without loss of generality, we have $c_i + c_{i+1} = a$ for odd i and $c_i + c_{i+1} = b$ for even i (so $c_{2q} + c_1 = b$).

For every even i, we see $c_{i+1} - c_{i-1} = (c_i + c_{i+1}) - (c_{i-1} + c_i) = b - a$. We claim that for all i, $c_{2i-1} - c_1 = (i-1)(b-a)$. We know the claim is satisfied for $i = 1$ trivially. For $i > 1$, by induction, we have $c_{2(i-1)-1} - c_1 = (i-2)(b-a)$, and so we

see $c_{2i-1} - c_1 = c_{2i-1} - c_{2i-3} + c_{2i-3} - c_1 = b - a + (i-2)(b-a) = (i-1)(b-a)$. Thus, in particular, $c_{2q-1} - c_1 = (q-1)(b-a)$. Additionally, we see $c_{2q-1} - c_1 = c_{2q-1} + c_{2q} - (c_1 + c_{2q}) = a - b$. Thus, $a - b = (q-1)(b-a)$, and so $0 = q(b-a)$, which implies $q = mp$ for some $m \in \mathbb{Z}$. Since the cycle is nontrivial, we know $q \geq p$, and so the cycle has length at least $2p$. □

Since n-adic matrices are always square, it is likely that quasi-n-adic matrices will give more interesting codes, so we consider possible girths of Tanner graphs from quasi-n-adic matrices. We now look at 2×2 arrays of weight 1 n-adic matrices and the effect on cycle length.

Lemma 6.1.2. *If* $M = \begin{bmatrix} M_u & M_v \\ M_x & M_w \end{bmatrix}$, *where* M_a *is an* n*-adic matrix of weight 1 with the nonzero signature row entry in position* $a \in \mathbb{Z}_n^\ell$, *then each cycle in the Tanner graph corresponding to* M *has the same length.*

Proof. First, note that a cycle of length $2k$ in the Tanner graph corresponding to M can be represented as a pair of sequences $(c_i)_{i=1}^k$, $(r_i)_{i=1}^k$ where m_{r_i,c_i} and $m_{r_i,c_{i+1}}$ are nonzero for each i, and $c_i \neq c_j$, $r_i \neq r_j$ for $i \neq j$.

Next, we see that the nonzero elements corresponding to a cycle in the Tanner graph must go through the submatrices M_a in order: $\ldots, M_u, M_v, M_w, M_x, M_u, \ldots$. Since each M_a is a permutation matrix, we know M is a 2-regular matrix, and so if we have $m_{r_i,c_i} \in M_u$, then the only other nonzero element in the row is $m_{r_i,c_{i+1}} \in M_v$. Similarly, the only other nonzero element in that column is $m_{r_{i+1},c_{i+1}} \in M_w$, and the only other nonzero element in the new row is $m_{r_{i+1},c_{i+2}} \in M_x$ (then back to M_u in the

column). Thus, without loss of generality, we can consider

$$m_{r_i,c_i} \in \begin{cases} M_u & i \text{ odd} \\ M_w & i \text{ even} \end{cases}, \qquad m_{r_i,c_{i+1}} \in \begin{cases} M_v & i \text{ odd} \\ M_x & i \text{ even} \end{cases}.$$

Additionally, since M is 2-regular, we know each nonzero entry is contained in exactly one cycle.

For notation, as in Chapter 2 we will use $(0, a)$ to denote a row or column that corresponds to the a entry of the first block and $(1, a)$ to denote a row or column that corresponds to the a entry of the second block. (So for example, we know there are nonzero entries:

$$M_{(0,0),(0,u)}, M_{(0,0),(1,v)}, M_{(1,0),(0,x)}, M_{(1,0),(1,w)}$$

corresponding to the signature row entries of the four weight 1 dyadic matrices.) Thus, we have

$$m_{r_i,c_i} = \begin{cases} m_{(0,r_i'),(0,c_i')} \in M_u & i \text{ odd} \\ m_{(1,r_i'),(1,c_i')} \in M_w & i \text{ even} \end{cases},$$

$$m_{r_i,c_{i+1}} \in \begin{cases} m_{(0,r_i'),(1,c_{i+1}')} \in M_v & i \text{ odd} \\ m_{(1,r_i'),(0,c_{i+1}')} \in M_x & i \text{ even} \end{cases},$$

where r_i', c_i' are the appropriate values in $\mathbb{Z}_n^\ell$ such that m_{r_i,c_i} is in the r_i', c_i' position of the n-adic matrix. Additionally, since these elements of M are nonzero, $m_{(0,r_i'),(0,c_i')} \in M_u$ implies $r_i' + c_i' = u$ (and similarly for the other submatrices).

Consider the cycle that contains $m_{(0,0),(0,u)} \neq 0$. We can describe the cycle as $(c_i)_{i=1}^k, (r_i)_{i=1}^k$, where $c_1 = (0, u), r_1 = (0, 0)$, and the other elements are determined by the other nonzero elements in the respective rows and columns. We will show that

any cycle has length equal to this cycle.

Let $(p_i)_{i=1}^t, (q_i)_{i=1}^t$ represent a cycle of length $2t$ with $m_{p_i,q_i}, m_{p_i,q_{i+1}} \neq 0$ for each i and $p_i \neq p_j$, $q_i \neq q_j$ for each $i \neq j$. As above, without loss of generality, we have

$$m_{p_i,q_i} = \begin{cases} m_{(0,p_i'),(0,q_i')} \in M_u & i \text{ odd} \\ m_{(1,p_i'),(1,q_i')} \in M_w & i \text{ even} \end{cases},$$

$$m_{p_i,q_{i+1}} \in \begin{cases} m_{(0,p_i'),(1,q_{i+1}')} \in M_v & i \text{ odd} \\ m_{(1,p_i'),(0,q_{i+1}')} \in M_x & i \text{ even} \end{cases}.$$

Consider $(p_i' - p_1')_{i=1}^t$ and $(q_i' + u - q_1')_{i=1}^t$. Note that if $p_i' + q_i' = a$, then $(q_i' + u - q_1') + (p_i' - p_1') = a + u - u = a$ (since $m_{(0,p_1'),(0,q_1')} \in M_u$). Similarly, if $p_i' + q_{i+1}' = a$, then $(q_i' + u - q_1) + (p_{i+1}' - p_1) = a + u - u = a$. Additionally, if $p_i' \neq p_j'$, then $p_i' - p_1' \neq p_j' - p_1'$, and if $q_i' \neq q_j'$, then $q_i' + a - q_1' \neq q_j' + u - q_1'$. Thus, we have another cycle of length t given by adding the corresponding prefixes to $(q_i' + u - q_1')_{i=1}^t$ and $(p_i' - p_1')_{i=1}^t$ (determined by the sequence of n-adic matrices $M_u, M_v, M_w, M_x, M_u, \ldots$). Note that for $i = 1$, this gives us $m_{(0,p_1'-p_1'),(0,q_1'+u-q_1')} = m_{(0,0),(0,u)}$ as an element in the cycle. However, we already considered the unique cycle containing $m_{(0,0),(0,u)}$ above, and so $t = k$, and thus, any cycle in the Tanner graph of M has length $2k$. □

Theorem 6.1.3. *If* $M = \begin{bmatrix} M_u & M_v \\ M_x & M_w \end{bmatrix}$, *where* M_a *is an* n*-adic matrix of weight 1 with the nonzero signature row entry in position* $a \in \mathbb{Z}_n^\ell$, *then the girth of the Tanner graph is* $4 \cdot \#\langle u + w - (v + x)\rangle$.

Proof. By Lemma 6.1.2, it suffices to show that the Tanner graph corresponding to M has a cycle of length $4 \cdot \#\langle u + w - (v + x)\rangle$. Let $\alpha = u + w - (v + x)$. We claim

that columns $((c_i, c'_i))_{i=1}^{2\#\langle\alpha\rangle}$ and rows $((r_i, r'_i))_{i=1}^{2\#\langle\alpha\rangle}$ with:

$$\begin{aligned} r_{2j+1} &= 0 & r'_{2j+1} &= \alpha j \\ r_{2j} &= 1 & r'_{2j} &= w - v + \alpha(j-1) \\ c_1 &= 0 & c'_1 &= a \\ c_{2j+1} &= 0 & c'_{2j+1} &= v + x - w - \alpha(j-1) \\ c_{2j} &= 1 & c'_{2j} &= v - \alpha(j-1) \end{aligned}$$

gives a cycle of length $4\#\langle\alpha\rangle$ in the Tanner graph corresponding to M.

It suffices to show that $m_{(r_i,r'_i),(c_i,c'_i)}, m_{(r_i,r'_i),(c_{i+1},c'_{i+1})} \neq 0$ for each i and $(c_i, c'_i) \neq (c_j, c'_j), (r_i, r'_i) \neq (r_j, r'_j)$ for $i \neq j$.

For odd i, say $i = 2j+1$, we see $r'_{2j+1} + c'_{2j+1} = \alpha j + (v + x - w - \alpha(j-1)) = v + x - w + \alpha = u$ (and $r'_1 + c'_1 = 0 + u = u$). Since for such i, we have $r_i, c_i = 0$, we have $m_{(r_i,r'_i),(c_i,c'_i)} \in M_u$ is nonzero for odd i. Additionally, $r'_{2j+1} + c'_{2(j+1)} = \alpha j + (v - \alpha(j+1-1)) = v$, and for such i, we have $r_i = 0, c_{i+1} = 1$. Thus, we have $m_{(r_i,r'_i),(c_{i+1},c'_{i+1})} \in M_v$ is nonzero for odd i.

For even i, say $i = 2j$, we see $r'_{2j} + c'_{2j} = (w - v + \alpha(j-1)) + (v - \alpha(j-1)) = w$, and so $m_{(r_i,r'_i),(c_i,c'_i)} \in M_w$ is nonzero for even i. Additionally, $r'_{2j} + c'_{2j+1} = (w - v + \alpha(j-1)) + (v + x - w - \alpha(j-1)) = x$ (and $r'_{2\#\langle\alpha\rangle} + c'_1 = (w - v + \alpha(\#\langle\alpha\rangle - 1)) + u = w - v - \alpha + u = x$), and so $m_{(r_i,r'_i),(c_{i+1},c'_{i+1})} \in M_x$ is nonzero for even i.

We see that $(c_i, c'_i) \neq (c_j, c'_j), (r_i, r'_i) \neq (r_j, r'_j)$ for $i \neq j$ since for different parities of i, j, $c_i \neq c_j$ and $r_i \neq r_j$, and for i, j the same parity, the values of c'_i, r'_i are distinct elements in a coset of $\langle\alpha\rangle$ and so have no repeated values in the given range. Thus, this gives us a cycle of length $4\#\langle\alpha\rangle$, and so the girth of the Tanner graph is $4\#\langle u + w - (v + x)\rangle$. □

From this result, we see that the length of any cycle in a Tanner graph with this construction will be divisible by 4. In particular, since the order of any element in $\mathbb{Z}_n^\ell$ is at most n, the girth of the Tanner graph of a 2×2 array of weight 1 n-adic matrices is at most $4n$.

6.2 Stopping sets

In addition to considering the girth of a Tanner graph, we consider the minimum possible size of a stopping set for n-adic and quasi-n-adic codes. Considering various weights of n-adic matrices, note that since weight 1 matrices are permutation matrices and therefore the Tanner graphs are perfect matchings, there can be no stopping sets. Additionally, a weight 2 n-adic matrix or a $2 \times \ell$ array of weight 1 n-adic matrices is 2-left-regular, and so the stopping distance will be exactly half the girth.

Theorem 6.2.1. *If M is a dyadic matrix of weight 3, then the Tanner graph has stopping distance 3.*

Proof. Let M be a dyadic matrix of weight 3 with nonzero entries in signature row positions u, v, w. We see that there are no two identical columns of M, because if there were two columns c_1, c_2 with nonzero entries in the same rows, this would imply that $\{u + c_1, v + c_1, w + c_1\} = \{u + c_2, v + c_2, w + c_2\}$. Since $c_1 \neq c_2$, without loss of generality, we must have $u+c_1 = v+c_2$, $v+c_1 = w+c_2$, and $w+c_1 = u+c_2$. Note that since we are working with dyadic matrices, the first equation implies $u = v + c_1 + c_2$ and the second equation implies $w = v + c_1 + c_2$, and so $u = w$. Thus, we can't have any two identical columns of M, and so there is no stopping set of size 2.

We can find a stopping set of size 3 by taking the columns corresponding to the nonzero elements of the signature row: u, v, w. We know in column u, the nonzero entries are exactly in rows 0, $v + u$, and $w + u$. Similarly, for column v, the nonzero

entries are in rows 0, $v+u$, and $w+v$, and for column w, the nonzero entries are in rows 0, $w+u$, and $w+v$. Thus, these three columns correspond to a stopping set of size 3 in the Tanner graph. □

This bound does not apply for all n-adic matrices of weight 3. For example, the following weight 3 triadic matrix has a Tanner graph with stopping distance 2:

$$\left[\begin{array}{ccccccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right].$$

One stopping set of size 2 would be given by the vertices corresponding to the first and last columns of the matrix.

Chapter 7

Graph lifts and cycle codes

Early graph-theoretic codes focused on constructions such as cut-set codes, which are generated by cut-sets of a graph, and cycle codes which are generated by the cycles of a graph [31]. From their relation in graph theory, these can be shown to be dual codes, and in [7], Etzion et al. showed that LDPC codes with cycle-free Tanner graphs are all cut-set codes. In this last chapter, we focus on the relation between the cycle code of a graph and the cycle code of one of its lifts.

For this chapter, we will use $[n] := \{1, \ldots, n\}$.

For a graph $\mathcal{G}$ with edge set $E = \{e_1, \ldots, e_m\}$, we can construct the *cycle code* of $\mathcal{G}$, denoted $\mathcal{C}(\mathcal{G})$, by using all incidence vectors $c_\psi \in \mathbb{F}_2^m$ of cycles ψ in $\mathcal{G}$ as the rows of a generator matrix.

Note that all codewords of $\mathcal{C}(\mathcal{G})$ correspond to cycles or unions of edge-disjoint cycles. If c_1, c_2 are two incidence vectors generating $\mathcal{C}(\mathcal{G})$ corresponding to edge-disjoint cycles, they will have disjoint support, and so $c_1 + c_2$ corresponds to the union of the two disjoint cycles. If c_1, c_2 have some intersection, though, $c_1 + c_2$ corresponds to taking the symmetric difference of the two cycles, which gives another set of edge-disjoint cycles.

Example 7.0.1. *Consider the graph $\mathcal{G}$ in Figure 7.1 with edges $e_1, \ldots, e_5$.*

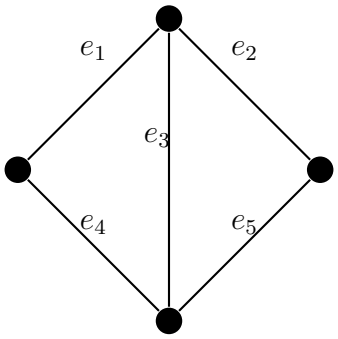


Figure 7.1: A graph $\mathcal{G}$ with edge labels

There are three cycles in this graph: $e_1e_2e_5e_4$, $e_1e_3e_4$, *and* $e_2e_3e_5$. *Thus, the cycle code for this graph has generator matrix*

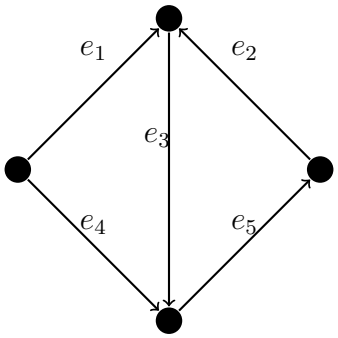
$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

This is a $[5, 2]$ *linear code, as the third row of* G *can be obtained by adding the first two. This addition,* $(1, 1, 0, 1, 1) + (1, 0, 1, 1, 0) = (0, 1, 1, 0, 1)$, *corresponds to the symmetric difference between the cycles* $e_1e_2e_5e_4$ *and* $e_1e_3e_4$, *leaving the cycle* $e_2e_3e_5$.

□

In Chapter 2.1.1, we discussed how to take a lift of a directed graph. Similarly, we can take a lift of an undirected graph by giving arbitrary directions to each edge and then taking a lift. Thus, we can consider both the cycle code of a graph and the cycle code of a lift of a graph. When constructing the lift of a graph, each edge is assigned a permutation, and so we can consider the *net permutation* (also called *net voltage* in [13]) of a walk by composing the permutations of the edges along the walk. If a directed edge is traversed backwards, the inverse permutation is added to the composition, rather than the original permutation.

Example 7.0.2. *Consider the graph $\mathcal{G}$ with edges that are directed and have assignments to permutations in S_3, where $\iota = (1)(2)(3)$, the identity permutation, as in Figure 7.2.*



$$e_1 \mapsto (1\ 2\ 3), e_2 \mapsto (1\ 2),$$
$$e_3 \mapsto (1\ 3\ 2), e_4 \mapsto \iota, e_5 \mapsto \iota$$

Figure 7.2: A directed graph $\mathcal{G}$ with edge labels and permutations in S_3

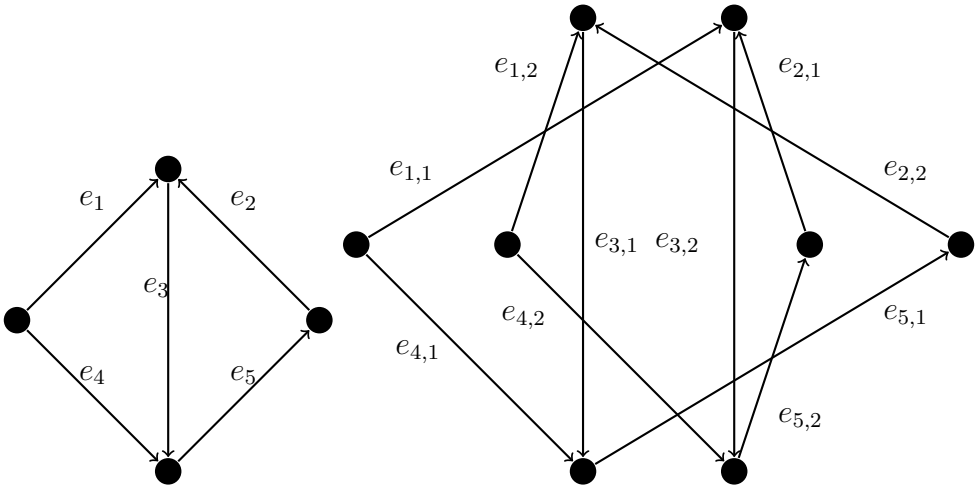
We see that the walk $e_1e_2^{-1}e_5^{-1}e_4^{-1}$ (where an edge e^{-1} denotes traversing in the opposite direction) has net permutation $\iota^{-1} \circ \iota^{-1} \circ (1\ 2)^{-1} \circ (1\ 2\ 3) = (2\ 3)$ and the walk $e_2^{-1}e_5^{-1}e_4^{-1}e_1$ has net permutation $(1\ 2\ 3) \circ \iota^{-1} \circ \iota^{-1} \circ (1\ 2)^{-1} = (1\ 3)$. □

While we can find the net permutation of a walk, the net permutation of a cycle is not necessarily unique, as shown in Example 7.0.2. Recall that the *cycle type* $c_1, \ldots, c_m$ of a permutation σ is the non-decreasing list of the sizes of cycles in σ. For example, if $\sigma = (1\ 4)(2\ 7\ 8)(3\ 6)(5) \in S_8$, then the cycle type of σ is $1, 2, 2, 3$. We can see that the cycle type of a net permutation of a cycle is unique. It is known that the conjugates have the same cycle type. Thus, a net permutation and its inverse, which would be obtained by traversing the cycle in the opposite direction, have the same cycle type. Also, we can begin a net permutation at different points in the cycle, which would give us permutations of the form $\pi\sigma$ and $\sigma\pi$, which are again conjugates

and so have the same cycle type. Thus, we can define a unique *net cycle type* for a cycle C when its edges are assigned permutations for a lift.

In this chapter, we will consider how to obtain a generator matrix for the cycle code of a lifted graph when given the generator matrix for the cycle code of the original graph. Example 7.0.3 shows the relation for a particular graph and lift.

Example 7.0.3. *Consider the graph $\mathcal{G}$ and lift $\hat{\mathcal{G}}$ given in Figure 7.3, where permutations $\iota = (1)(2)$ and $\sigma = (1\ 2)$ are elements of S_2.*



$$e_1 \mapsto \sigma, e_2 \mapsto \sigma, e_3 \mapsto \iota, e_4 \mapsto \iota, e_5 \mapsto \sigma$$

Figure 7.3: A graph $\mathcal{G}$ with degree 2 lift $\hat{\mathcal{G}}$

We can find that the $\mathcal{C}(\mathcal{G})$ is generated by

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

and $\mathcal{C}(\hat{\mathcal{G}})$ is generated by

$$G' = \begin{array}{c} \begin{matrix} e_{1,1} & e_{1,2} & e_{2,1} & e_{2,2} & e_{3,1} & e_{3,2} & e_{4,1} & e_{4,2} & e_{5,1} & e_{5,2} \end{matrix} \\ \left[\begin{array}{cccccccccc} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}\right] \end{array}.$$

Note that some characteristics of G' can be seen from G by considering the net cycle type along each cycle. The net cycle type of $C_1 = e_1 e_2^{-1} e_5^{-1} e_4^{-1}$ is 2, and there is one cycle in $\hat{\mathcal{G}}$ corresponding to C_1, which has twice the length of C_1. Similarly, the net cycle type of $C_2 = e_1 e_3 e_4^{-1}$ is 2, and there is again one cycle in $\hat{\mathcal{G}}$ corresponding to C_2, which has twice the length of C_2. Each of these corresponds to a row in G' with twice the weight of a row in G. In contrast, the net cycle type of $C_3 = e_2 e_3 e_5$ is $1, 1$. In $\hat{\mathcal{G}}$, there are two cycles corresponding to C_3, each with length equal to C_3. These correspond to two rows in G', each of which has the same weight as the third row of G. This relation between the cycles in a graph and its lift is given in Theorem 7.0.4.

□

Gross and Tucker look at permutation voltage graphs in [13]. Here, we reconsider their result on the number of cycles in the permutation voltage graphs and give the particular structure of the cycle in order to find the generator matrix for the cycle code of a lifted graph.

Theorem 7.0.4. *Let $C = (e_1, \ldots, e_k)$ be a k-cycle in a graph $\mathcal{G}$ with a degree ℓ lift $\hat{\mathcal{G}}$ such that the edge e_i is assigned permutation $\sigma_i \in S_\ell$. Let $\sigma_{i,j} = \sigma_i \circ \sigma_j$ and $\sigma = \sigma_{k,\ldots,1} = \sigma_k \circ \cdots \circ \sigma_1$. If C has net cycle type $c_1, c_2, \ldots, c_m$, then in the degree ℓ*

lift $\hat{\mathcal{G}}$, C corresponds to $\#\{i|c_i = j\}$ edge-disjoint cycles of length jk with the form

$$C_a = \left(e_{1,a}, e_{2,\sigma_1(a)}, e_{3,\sigma_{2,1}(a)}, \ldots, e_{k,\sigma_{k-1,\ldots,1}(a)}, e_{1,\sigma(a)}, \ldots, e_{k,(\sigma_k^{-1}\circ\sigma^j)(a)}\right)$$

for $a \in [\ell]$.

Proof. Consider $C = (e_1, \ldots, e_k)$ a k-cycle in a graph $\mathcal{G}$ with a degree ℓ lift $\hat{\mathcal{G}}$ such that the net cycle type of C is $c_1, c_2, \ldots, c_m$. We can find a particular net permutation corresponding to this cycle by taking $\sigma = \sigma_k \circ \cdots \circ \sigma_1$.

Without loss of generality, consider the cycle in $\hat{\mathcal{G}}$ corresponding to C and containing $e_{1,a}$ (call this C_a). Let c_i be the length of the cycle in σ containing a. Note that in C_a, we have the path $e_{1,a}, e_{2,\sigma_1(a)}, e_{3,\sigma_{2,1}(a)}, \ldots, e_{k,\sigma_{k-1,\ldots,1}(a)}, e_{1,\sigma(a)}$. If $\sigma(a) = a$, then C_a also has length k. Otherwise, we can continue the path through the cycle of σ containing a to get $C_a = \left(e_{1,a}, e_{2,\sigma_1(a)}, e_{3,\sigma_{2,1}(a)}, \ldots, e_{k,\sigma_{k-1,\ldots,1}(a)}, e_{1,\sigma(a)}, \ldots, e_{k,(\sigma_k^{-1}\circ\sigma^{c_i})(a)}\right)$. Thus, the length of C_a is kc_i. Repeating this process with the other cycles in σ gives the result. □

Considering how lifts affect cycles in a graph, we can consider the relation between the cycle code of the lift of a graph $\mathcal{G}$ and the cycle code of $\mathcal{G}$. A cycle in the lift $\hat{\mathcal{G}}$ can correspond to a cycle in $\mathcal{G}$ or a closed walk in $\mathcal{G}$. We plan to make this relationship more precise in a future work, along with examining the duals of these codes, which are cut-set codes. Here, we can begin to consider the relationship between the generator matrices of these two cycle codes.

Corollary 7.0.5. *Let $\mathcal{G}$ be a graph with degree ℓ lift $\hat{\mathcal{G}}$. A generator matrix G' of $\mathcal{C}(\hat{\mathcal{G}})$ can be obtained from the generator matrix G of $\mathcal{C}(\mathcal{G})$ with rows given by all incidence vectors of cycles by the following process:*

1. *Consider each row of G corresponding to a cycle C. Without loss of generality, $C = e_1 \dots e_k$.*

2. *Let m be the number of cycles in a net permutation σ of C, i.e. $\sigma = \sigma_k \circ \cdots \circ \sigma_1$, and order the cycles: $\pi_1, \dots, \pi_m$.*

3. *Replace each entry $g_{C,e}$ with an $m \times \ell$ matrix such that:*

   - *If $g_{C,e} = 0$, the matrix is all zeros.*
   - *If $g_{C,e} = 1$, the (p, q) entry of the matrix is 1 if there is some $a \in \pi_p$ such that $\sigma_{i-1,\dots,1}(a) = q$, where $e = e_i$, and it is 0 otherwise.*

Proof. From Theorem 7.0.4, we know that a k-cycle $C = (e_1, \dots, e_k)$ corresponds to $\#\{i | c_i = j\}$ edge-disjoint cycles of length jk with the form

$$C_a = \left(e_{1,a}, e_{2,\sigma_1(a)}, e_{3,\sigma_{2,1}(a)}, \dots, e_{k,\sigma_{k-1,\dots,1}(a)}, e_{1,\sigma(a)}, \dots, e_{k,(\sigma_k^{-1} \circ \sigma^j)(a)}\right)$$

for $a \in [\ell]$. It remains to show that the construction of G' given above yields the incidence matrix for these cycles.

For a cycle of this form, note that all the entries corresponding to e_1 are exactly the ones corresponding to the particular cycle of σ containing a, and so G' is accurate for these entries. Call this cycle π. All the entries in this cycle corresponding to e_i have the form $e_{i,(\sigma_{i-1,\dots,1} \circ \sigma^n)(a)}$ for some n. However, note that $\sigma^n(a) \in \pi$, so there is some $a' \in \pi$ such that $e_{i,(\sigma_{i-1,\dots,1} \circ \sigma^n)(a)} = e_{i,\sigma_{i-1,\dots,1}(a')}$. Thus G' gives an incidence matrix for cycles of the form above, that is, for cycles in $\hat{\mathcal{G}}$. □

In some special cases, we find that the process of obtaining the generator matrix for some $\mathcal{C}(\hat{\mathcal{G}})$ from the generator matrix of $\mathcal{C}(\mathcal{G})$ can be simpler than that given above. For example, when $\mathcal{G}$ is 2-regular and so is a union of disjoint cycles, we find

that the cycle code of $\hat{\mathcal{G}}$ is generated by a matrix lift of the generator matrix G of $\mathcal{C}(\mathcal{G})$ under certain permutation conditions.

Theorem 7.0.6. *For $\mathcal{G}$ a 2-regular graph, if G is the generator matrix of the cycle code $\mathcal{C}(\mathcal{G})$, then the generator matrix for the cycle code $\mathcal{C}(\hat{\mathcal{G}})$ of a lift of $\mathcal{G}$ is a matrix lift of G if and only if each cycle in $\mathcal{G}$ has net cycle type $1, \ldots, 1$.*

Proof. First, we know that since $\mathcal{G}$ is 2-regular, it is a union of disjoint cycles. Note that since lifting a graph preserves the degree of each vertex, $\hat{\mathcal{G}}$ is also 2-regular, and so is a union of disjoint cycles. We will begin by considering a single component of $\mathcal{G}$, i.e. a single cycle. Let $C = (e_1, \ldots, e_k)$ be this cycle. In this case, we know from Theorem 7.0.4 that a degree ℓ lift $\hat{\mathcal{G}}$ where each cycle in $\mathcal{G}$ has net cycle type $1, \ldots, 1$ will contain ℓ cycles of length k corresponding to C. Thus, the cycle code of $\hat{C}$ is generated by ℓ codewords of weight k with disjoint support. Because the cycles in $\mathcal{G}$ are disjoint, we see that $\mathcal{C}(\hat{\mathcal{G}})$ is generated by codewords with disjoint support, and there are $j\ell$ codewords in this generating set with weight k, where j is the number of k-cycles in $\mathcal{G}$.

Similarly, since $\mathcal{G}$ is a union of disjoint cycles, $\mathcal{C}(\mathcal{G})$ is generated by codewords with disjoint support, and there are j codewords in this generating set with weight k, where j is again the number of k-cycles in $\mathcal{G}$. Taking a degree ℓ lift of the generator matrix replaces each nonzero element with an $\ell \times \ell$ permutation matrix. Since the codewords in the generator matrix have disjoint support, this process results in $j\ell$ codewords of weight k, with all codewords having disjoint support. Thus, this generates a code equivalent to $\mathcal{C}(\hat{\mathcal{G}})$.

Conversely, note that if some cycle of length k in $\mathcal{G}$ has a net cycle type $c_1, \ldots, c_m$ where $c_m \neq 1$, then by Theorem 7.0.4, this corresponds to cycles of length $c_m k$ in $\hat{\mathcal{G}}$. Again, $\hat{\mathcal{G}}$ is a union of disjoint cycles, and so $\mathcal{C}(\hat{\mathcal{G}})$ is generated by codewords with

disjoint support. Again, in the degree ℓ lift of the generator matrix of $\mathcal{C}(\mathcal{G})$, this cycle corresponds to ℓ codewords of weight k. Since all codewords in the generator matrix have disjoint support and we have codewords of different weights in $\hat{G}$ and the generator matrix of $\mathcal{C}(\hat{\mathcal{G}})$, it follows that $\hat{G}$ does not generate $\mathcal{C}(\hat{\mathcal{G}})$. $\square$

Chapter 8

Conclusions

In this dissertation, we considered various properties of n-adic matrices and n-adic parity check codes. In particular, we derived bounds on the minimum distance of n-adic and quasi-n-adic codes and found a quasi-triadic code with similar parameters to another code from the literature. Additionally, we found properties of the dimension and dual of dyadic codes and used them to construct quantum codes, including one which maximizes parameters. We examined other parameters, such as the girth and stopping distance of n-adic codes. Finally, we considered how to obtain cycle codes of lifts of graphs from the cycle code of a base graph.

This work prompts several questions for future research. It would be interesting to generalize the various n-adic and quasi-n-adic results to larger arrays and matrices of different weights. Additionally, one could search for additional classical and quantum codes with good parameters or give proofs that such codes exist.

Bibliography

[1] M. Baldi. LDPC and MDPC codes in cryptography: are (decoding) failures acceptable? Algebraic Coding and Cryptography on the East Coast Seminar Series, 2020.

[2] G. Banegas, P. S. L. M. Barreto, E. Persichetti, and P. Santini. Designing efficient dyadic operations for cryptographic applications. Cryptology ePrint Archive, Report 2018/650, 2018. `https://ia.cr/2018/650`.

[3] P. Barreto, R. Lindner, and R. Misoczki. Monoidic codes in cryptography. volume 2011, pages 179–199, 11 2011.

[4] A. R. Calderbank and P. W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, Aug 1996.

[5] S.-Y. Chung, T. Richardson, and R. Urbanke. Analysis of sum-product decoding of low-density parity-check codes using a gaussian approximation. *IEEE Transactions on Information Theory*, 47(2):657–670, 2001.

[6] I. B. Djordjevic, L. Xu, T. Wang, and M. Cvijetic. Large girth low-density parity-check codes for long-haul high-speed optical communications. In *OFC/NFOEC 2008 - 2008 Conference on Optical Fiber Communication/National Fiber Optic Engineers Conference*, pages 1–3, 2008.

[7] T. Etzion, A. Trachtenberg, and A. Vardy. Which codes have cycle-free tanner graphs? *IEEE Transactions on Information Theory*, 45(6):2173–2181, 1999.

[8] J. L. Fan. Array codes as ldpc codes. 2001.

[9] J.-C. Faugère, A. Otmani, L. Perret, and J.-P. Tillich. Algebraic cryptanalysis of mceliece variants with compact keys. In H. Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 279–298, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[10] R. Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1):21–28, 1962.

[11] D. Gottesman. Stabilizer codes and quantum error correction. *PhD thesis, California Institute of Technology*, 1997.

[12] M. Grassl. Bounds on the minimum distance of linear codes and quantum codes. Online available at `http://www.codetables.de`, 2007. Accessed on 2023-09-19.

[13] J. L. Gross and T. W. Tucker. *Topological Graph Theory*. Wiley-Interscience, USA, 1987.

[14] C. A. Kelley. Codes over graphs. In W. C. Huffman, J.-L. Kim, and P. Solé, editors, *A Concise Encyclopedia of Coding Theory*. CRC Press, March 2021.

[15] C. A. Kelley and J. L. Walker. Ldpc codes from voltage graphs. In *2008 IEEE International Symposium on Information Theory*, pages 792–796, 2008.

[16] Y. Kou, S. Lin, and M. Fossorier. Low-density parity-check codes based on finite geometries: a rediscovery and new results. *IEEE Transactions on Information Theory*, 47(7):2711–2736, 2001.

[17] M. Martinez and C. A. Kelley. Minimum distance and other properties of quasi-dyadic parity check codes. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 2118–2123, 2022.

[18] M. Martinez, T. Pllaha, and C. A. Kelley. On codes based on dyadic matrices and their generalizations. *Submitted*.

[19] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *Coding Thv*, 4244:114–116, 1978.

[20] R. Michael, T. Sridhara, and T. Fuja. A class of group-structured ldpc codes. 07 2001.

[21] R. Misoczki and P. S. L. M. Barreto. Compact mceliece keys from goppa codes. In M. J. Jacobson, V. Rijmen, and R. Safavi-Naini, editors, *Selected Areas in Cryptography*, pages 376–392, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[22] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge Series on Information and the Natural Sciences. Cambridge University Press, 2000.

[23] E. Persichetti. Compact mceliece keys based on quasi-dyadic srivastava codes. *IACR Cryptology ePrint Archive*, 2011:179, 01 2011.

[24] E. Persichetti. Compact mceliece keys based on quasi-dyadic srivastava codes. *Journal of Mathematical Cryptology*, 6(2):149–169, 2012.

[25] W. Peterson and E. Weldon. *Error-correcting Codes*. Cambridge, MA, 1972.

[26] E. Prange. *Cyclic error-correcting codes in two symbols*. Air force Cambridge research center, 1957.

[27] P. Santini, E. Persichetti, and M. Baldi. Reproducible codes and cryptographic applications. *IACR Cryptol. ePrint Arch.*, 2018:666, 2018.

[28] P. Santini, E. Persichetti, and M. Baldi. Reproducible families of codes and cryptographic applications. *Journal of Mathematical Cryptology*, 16(1):20–48, 2022.

[29] M. Sipser and D. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

[30] R. Smarandache and P. O. Vontobel. Quasi-cyclic ldpc codes: Influence of proto- and tanner-graph structure on minimum hamming distance upper bounds. *IEEE Transactions on Information Theory*, 58(2):585–607, 2012.

[31] P. Solé and T. Zaslavsky. The covering radius of the cycle code of a graph. *Discrete applied mathematics*, 45(1):63–70, 1993.

[32] A. Steane. Multiple particle interference and quantum error correction. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 452, 01 1996.

[33] R. Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27(5):533–547, 1981.

[34] R. M. Tanner. A transform theory for a class of group-invariant codes. *IEEE Trans. Inf. Theory*, 34:752–775, 1988.

[35] R. M. Tanner, D. Sridhara, and T. Fuja. A class of group-structured ldpc codes, 2001.

[36] J. Thorpe. Low-density parity-check (ldpc) codes constructed from protographs. *IPN progress report*, 42(154):42–154, 2003.

[37] R. Townsend and E. Weldon. Self-orthogonal quasi-cyclic codes. *IEEE Transactions on Information Theory*, 13(2):183–195, 1967.

[38] N. Wiberg. Codes and decoding on general graphs. 02 2001.