REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
09-05-2011

2. REPORT TYPE

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE
Subspace Arrangement Codes and Cryptosystems

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Berg, James Alfred

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Naval Academy
Annapolis, MD 21402

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
Trident Scholar Report no. 395 (2011)

12. DISTRIBUTION/AVAILABILITY STATEMENT
This document has been approved for public release; its distribution is UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Errors often occur in transferring electronic data, ranging from sensitive government information to everyday bar codes. Encoding information with an error-correcting code can alleviate the problem of corrupt or lost data. In order to not overburden computing systems, an efficient code must be used that will quickly encode and decode data while detecting and correcting a large number of errors. The goal of this project is to construct and develop efficient codes using recent advances in algebraic geometry, combinatorics, and commutative algebra.

15. SUBJECT TERMS
coding theory, cryptology, subspace arrangements, simplicial complexes, McEliece Cryptosystem

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)*
				49	

U.S.N.A. — Trident Scholar project report; no. 395 (2011)

SUBSPACE ARRANGEMENT CODES AND CRYPTOSYSTEMS

by

Midshipman 1/C James A. Berg
United States Naval Academy
Annapolis, Maryland

Signature

Certification of Advisor Approval

Assistant Professor Max D. Wakefield
Mathematics Department

Signature

Date

Acceptance for the Trident Scholar Committee

Professor Carl E. Wick
Associate Director of Midshipmen Research

Signature

Date

Abstract. Errors often occur in transferring electronic data, ranging from sensitive government information to everyday bar codes. Encoding information with an error-correcting code can alleviate the problem of corrupt or lost data. In order to not overburden computing systems, an efficient code must be used that will quickly encode and decode data while detecting and correcting a large number of errors. The goal of this project is to construct and develop efficient codes using recent advances in algebraic geometry, combinatorics, and commutative algebra.

The mathematics of subspace arrangements and simplicial complexes lend themselves well for applications to coding theory. A subspace arrangement is a finite collection of subspaces in a vector space. A simplicial complex is an abstract generalization of a polygon or Euclidean solid. Fortunately, both simplicial complexes and subspaces arrangements can be described algebraically by a collection of polynomials, which can be used to construct a code. Then the combinatorial and geometric properties of subspace arrangements and simplicial complexes can be used to enumerate these efficient codes.

Scripts and algorithms were written in the computer algebra systems Sage and Macaulay2 to compute properties of the codes. The data led to the main results of the project: formulas for the the length, dimension, and minimum distance of polygon and skeletal simplicial complex evaluation codes. Scripts were written that aided in the construction of proofs for these formulas. The formulas give favorable lengths (short to minimize computation), dimensions (large to allow for more codewords), and minimum distances (large to allow more errors to be corrected and identified) of these polygon and skeletal simplicial complex evaluation codes.

The last part of the project involved extensions to a cryptosystem based on these codes. A cryptosystem deals with enciphering a message, which is an algorithmic process designed to make a sent message unreadable to an interloper, but, after another algorithmic deciphering process, is readable to the intended receiver (who, unlike the interloper, knows the deciphering key). Work was done on extensions to the code-based McEliece Cryptosystem, which has been demonstrated to withstand theoretical quantum computing attacks that would render common modern ciphers useless.

Keywords
Coding Theory
Cryptology
Subspace Arrangements
Simplicial Complexes
McEliece Cryptosystem

Acknowledgements

I would like to thank everyone who has made this project possible. I owe much to Assistant Professor Max Wakefield for having guided me through three years of research. Through our time together, I have learned much about the process of mathematical research and paper-writing. More importantly, my passion for mathematics has continued to grow under his guidance. I also want to thank the Mathematics Department Chair, Professor Geoffrey Price. I consider myself honored to have had him as my teacher for three semesters. I am indebted to the entire Mathematics Department, especially my previous instructors: Professor Robert Lockhart (who also has been my academic advisor), Professor Craig Bailey (who first fueled my desire for independent research), Dean Frederic Davis, Assistant Professor Sommer Gentry, Professor Charles Hanna, Professor Michael Hoffman, Professor W. David Joyner, Associate Professor Courtney Moen, Professor George Nakos, Assistant Professor Vrej Zarikian, LCDR Kyle Caudle, USN (ret.), and Professor William Traves. I also thank Professor Carl Wick and the Trident Scholar Committee for providing me with the opportunity to conduct this research. In particular, I thank Professor B. Mitchell Baker and Associate Professor Christopher Brown for being my second readers. There are countless other people who have mentored and helped me throughout all of my undergraduate studies: I thank you all. I would be remiss, of course, without thanking those who have engendered my mathematical passions and have never stopped supporting me: my parents, Jim and Kathy; my sister, Amanda; and my love, Sarah.

Contents

1. Introduction

As error-correcting codes have developed over the past century, so have the mathematical disciplines of algebra, geometry, and combinatorics, such as the development of the properties of a characteristic polynomial describing a poset representing a subspace arrangement by Athanasiadis in [1]. The goal of this project is to construct and develop efficient codes using recent advances in algebraic geometry, combinatorics, and commutative algebra. In particular, the mathematics of subspace arrangements and simplicial complexes lend themselves well for applications to coding theory. A subspace arrangement, $\mathcal{A}$, is a finite collection of subspaces in a vector space, V. A simplicial complex, Δ, is an abstract generalization of a polygon or Euclidean solid. Fortunately, both simplicial complexes and subspaces arrangements can be described algebraically by a collection of polynomials (namely, the ideal $I_{\mathcal{A}}$), which can be used to construct a code. Let $C(\mathcal{A}, j)$ be the evaluation code for the arrangement $\mathcal{A}$ with polynomials up to degree j.

A major problem in linear coding theory is finding codes that have a small number of digits (length) with a high number codewords (dimension), as well as good error-correction properties (minimum distance). In addition, for certain classes of codes, finding the length, dimension, and minimum distance becomes very difficult. Then aim of this project is find the length, dimension, and minimum distance of codes of the form $C(\mathcal{A}, j)$. Fortunately, the combinatorial and geometric properties of subspace arrangements and simplicial complexes can be used to find the length, minimum distance, and dimension of these codes.

The formula for the length of $C(\mathcal{A}, j)$ for any subspace arrangement $\mathcal{A}$ can be extracted from the characteristic polynomial of the arrangement's intersection lattice. However, in general, finding the dimension and minimum distance seems to be a much more difficult task. In Section 5, we find the dimension of $C(\mathcal{A}, j)$ for certain cases where $\mathcal{A}$ is the arrangement of coordinate hyperplanes. We then focus on the binary simplicial complex codes $C(\mathcal{A}_\Delta, j)$. The dimension for $C(\mathcal{A}_\Delta, j)$ is given by the face vector of Δ. Nonetheless, the minimum distance is still very difficult to determine. In Section 6.2, we give an explicit formula for the minimum distance of $C(\mathcal{A}_\Delta, j)$ for all values of j when Δ is a polygon.

The most significant class of codes we study is the skeletal codes $C(\mathcal{A}_\Delta, j)$ where Δ is a skeleton (a simplicial complex with all possible faces of a certain dimension and all lower dimensions). These codes turn out to be punctured Reed-Muller codes, as well as being related to Hamming codes. The main result is Theorem 6.22, which gives a formula for the minimum distance of $C(\mathcal{A}_\Delta, 1)$, a skeletal code with $j = 1$. Though the minimum distance for $j > 1$ appears to be very difficult, we prove a few cases of a presented conjecture.

One component of the project involved the production of computer scripts and algorithms in Sage and Macaulay2. For example, a Macaulay2 algorithm allowed the construction of a code (from which the properties of the code could be calculated), whereas another algorithm in Sage created a code but only returned the properties of the code. Other scripts helped speed up lengthy calculations, making proofs easier to see and construct.

The remainder of the project was focused on the McEliece cryptosystem. A cryptosystem deals with enciphering (rather than encoding) a message, which is an algorithmic process designed to make a sent message unreadable to an interloper, but, after another algorithmic deciphering process, is readable to the intended receipt (who, unlike the interloper, knows the key to decipher the message). The McEliece cryptosystem is built on error-correcting codes and is suitable to codes with nice decoding algorithms (see [6], [5], and [18]). Since our subspace arrangement codes are closely related to Hamming codes and Reed-Muller codes, it is possible that they also have nice decoding properties. An important note here is that the McEliece cryptosystem has been demonstrated to withstand some theoretical quantum computing attacks that would render common modern ciphers (such as RSA) useless.

2. Basic Definitions

In this section we establish notation and the basic objects of study.

2.1. Coding Theory.

We begin by systematically defining all objects of study for this project with a focus on linear codes as the main character of study. To do this we define all relevant terms.

A *commutative ring* is a set of elements closed under two operations (usually called addition and multiplication) that, under addition, forms an Abelian group (a set of elements with an operation that has a unit and that is associative and commutative, and inverses exist) and, under multiplication, forms a commutative monoid (a set of elements with an operation that has a unit as well as being associate and commutative; inverses do not necessarily exist). The distributive property also holds in a commutative ring. A *field* is a commutative ring where all multiplicative inverses exist. A *finite field* is a field with finitely many elements. We call $\mathbb{F}_q$ the finite field with q elements and note that q must be a power of a prime. One example is $\mathbb{F}_2$, which is defined as the set $\{0, 1\}$ with addition and multiplication modulo 2. This example is of prime importance because of its applicability to computers.

A *vector space*, V, is a set of vectors, $\{v_1, \dots\}$ closed under addition and scalar multiplication, with the distributive property holding. The central example we study is the vector space $\mathbb{F}_q^n$. A subset of a vector space which is closed under vector addition and scalar multiplication is a *subspace* of the vector space. The minimum number of vectors required to generate all vectors of a subspace [or vector space] (through scalar multiplication and vector addition) is the *dimension* of the subspace [or vector space]. For example, the dimension of $\mathbb{F}_q^n$ is n.

The aim of coding theory is to develop efficient data transmission processes that will detect and correct errors. A code consists of a collection of codewords, each of which are assigned to a symbol (or group of symbols). A simple example is the ISBN 10-digit number used for cataloging books. The first nine digits are used for identification, and the last digit is a linear combination of the identification digits that checks for errors when the bar code is scanned (here, the codewords are the 10-digit numbers assigned to each book).

In order to introduce coding theory terminology, consider the following example. Let $M = \{a, b, c, d\}$ be a set of information. A code for M could consist of the subspace of codewords $C = \{(0, 0, 0, 0), (0, 1, 0, 1), (1, 0, 1, 0), (1, 1, 1, 1)\}$, with each codeword representing one element in M. Note that the codewords in this example all have digits in $\mathbb{F}_2$. Formally, a code C is a subspace of a vector space $V = \mathbb{F}_q^n$ over the finite field $\mathbb{F}_q$; in this example, C is a subspace of $V = \mathbb{F}_2^4$. The *length*, n, of a code refers to how many digits are in a codeword, which is also the dimension of the ambient vector space V; in the example, $n = 4$. The *dimension*, k, of a code is the dimension of the subspace C; in the example, $k = 2$. A basis of vectors for the subspace C provides for the construction of the *generating matrix* of C, which is the matrix with the basis vectors forming the rows. In the example, the generating matrix for C is

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

Finally, the *minimum distance*, d, is the minimum number of digits that must be changed to transform one codeword into another; in this example, $d = 2$. Minimum distance can also be defined as the minimum Hamming weight among the non-zero codewords (the Hamming weight of a codeword is the number of non-zero entries in it). Minimum distance dictates the number of errors a code can detect. In particular, a code can detect up to $d-1$ errors and correct up to $\lfloor \frac{d}{2} \rfloor$ errors (note that $\lfloor z \rfloor$ is the largest integer less than or equal to z). A code with length n, dimension k, and minimum distance d is referred to as an $[n,k,d]_q$ code, where the digits are elements of $\mathbb{F}_q$. The example is a $[4,2,2]_2$ code.

2.2. **Hamming Codes.** An important class of codes which will relate to the codes developed in this project is the class of Hamming codes. All binary Hamming codes can be constructed in a fairly straightforward process, which is described in [9]. For the Hamming code denoted $\mathcal{H}_r$, first form a matrix with the numbers 1 through 2^{r-1} in binary as the columns (so there are $2^r - 1$ columns). Permute the matrix to form a matrix of the form $[A|I_r]$, where I_r is the $r \times r$ identity matrix (this matrix, termed H_r, is called the parity-check matrix of the code). Then, $\mathcal{H}_r$ can be described by the matrix $[I_{2^r-1-r}|A^T]$, where A^T is the transpose of A (the columns of A^T are the rows of A, with the ordering of the rows becoming the ordering of the columns). Each binary Hamming code $\mathcal{H}_r$ has the property of being a $[2^r-1, 2^r-1-r, 3]_2$ code.

Example 2.1. The construction of $\mathcal{H}_4$ is provided. First, the matrix of binary numbers is

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

The columns are then permuted to form

$$H_4 = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

From this matrix, the generating matrix for $\mathcal{H}_4$ is formed:

$$\mathcal{H}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

By inspection, it can be confirmed that $\mathcal{H}_4$ is a $[15,11,3]_2$ code.

2.3. **Reed-Muller Codes.** Another class of closely-related codes is the class of Reed-Muller codes, which are described in detail in [14]. A binary Reed-Muller code, $\mathcal{R}(j,m)$, is formed by examining all polynomials in m variables up to degree j. Each polynomial corresponds to a codeword by evaluating that polynomial on all possible points in $\mathbb{F}_2^m$. For each $\mathcal{R}(j,m)$, a $[2^m, \sum\limits_{i=0}^{r} \binom{m}{i}, 2^{m-r}]_2$ code is produced.

Example 2.2. The construction of the $\mathcal{R}(1,3)$ Reed-Muller code will be demonstrated. For consistency, the following ordering of the points in $\mathbb{F}_2^3$ will be used:

x_1	x_2	x_3
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Each codeword is obtained by creating a binary string that represents the evaluation of each polynomial on the certain states of x_1, x_2, and x_3, with the order given by the above table. Thus, the following table can be made to display all the codewords:

Polynomial	*Codeword*
0	$(0,0,0,0,0,0,0,0)$
x_1	$(0,0,0,0,1,1,1,1)$
x_2	$(0,0,1,1,0,0,1,1)$
x_3	$(0,1,0,1,0,1,0,1)$
x_1+x_2	$(0,0,1,1,1,1,0,0)$
x_1+x_3	$(0,1,0,1,1,0,1,0)$
x_2+x_3	$(0,1,1,0,0,1,1,0)$
$x_1+x_2+x_3$	$(0,1,1,0,1,0,0,1)$
1	$(1,1,1,1,1,1,1,1)$
$1+x_1$	$(1,1,1,1,0,0,0,0)$
$1+x_2$	$(1,1,0,0,1,1,0,0)$
$1+x_3$	$(1,0,1,0,1,0,1,0)$
$1+x_1+x_2$	$(1,1,0,0,0,0,1,1)$
$1+x_1+x_3$	$(1,0,1,0,0,1,0,1)$
$1+x_2+x_3$	$(1,0,0,1,1,0,0,1)$
$1+x_1+x_2+x_3$	$(1,0,0,1,0,1,1,0)$

Note that, since most of the polynomials are linear combinations of one another, $\mathcal{R}(1,3)$ can be represented by a matrix of a basis for the code (the polynomial is in the first column):

$$\begin{array}{c|cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x_1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ x_2 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ x_3 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}.$$

Note that the code is a $[2^3, \sum\limits_{i=0}^{1} \binom{3}{i}, 2^{3-1}]_2 = [8,4,4]_2$ code.

2.4. **Subspace Arrangements.** Let V be a vector space of dimension ℓ over a finite field $\mathbb{F}$ of q elements. A subspace arrangement $\mathcal{A} = \{X_1, \ldots, X_t\}$ in V is a finite collection of linear subspaces $X_i \subseteq V$. Some mathematical objects related to subspace arrangements are now introduced (definitions for some of them are contained in the following sections). These objects are further discussed in [2]. Let $S = \mathbb{F}_q[x_1, \ldots, x_\ell]$ be the symmetric algebra of the dual vector space V^* (the dual vector space is the set of all linear maps $\phi : V \longrightarrow \mathbb{F}_q$). The points

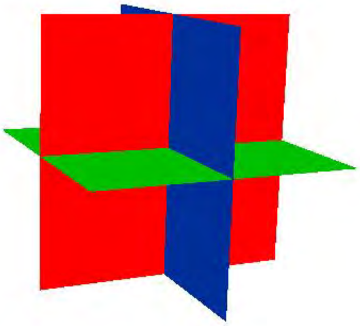

FIGURE 1. The coordinate planes

of V in $\mathcal{A}$ are $P(\mathcal{A}) = \bigcup\limits_{i=1}^{t} X_i = \{p_1, \dots, p_n\}$. Additionally, let $I(\mathcal{A}) = \{f \in S | f(P(\mathcal{A})) = 0\}$ be the defining ideal of $\mathcal{A}$. With these properties of a subspace arrangement, it is possible to develop an error-correcting code from a subspace arrangement.

3. Subspace Arrangement Codes

3.1. **Definition of $C(\mathcal{A}, j)$.** In order to define the codes we want to study we need to understand polynomials and their roots over a finite field. A *polynomial ring*, $\mathbb{F}_q[x_1, \dots, x_\ell]$, is the set of all polynomials over the variables $x_1, \dots, x_\ell$ with usual addition and multiplication and with coefficients in $\mathbb{F}_q$. A *subring* of a ring is a subset of a ring that is closed under the ring operations. An *ideal*, I, is a subring that is closed under multiplication by any element in the ring.

Recall the notation of subspace arrangements. We can now construct a code from $\mathcal{A}$ by evaluating all polynomials of certain degrees. Define the evaluation map $ev_{\mathcal{A}} : S_{\leq j} \to \mathbb{F}^n$ by

$$ev_{\mathcal{A}}(f) = (f(p_1), \dots, f(p_n))$$

where $S_{\leq j}$ is the vector space of all polynomials of less degree than or equal to j in $S = \mathbb{F}_q[x_1, \dots, x_\ell]$. Now we can define our main object of study.

Definition 3.1. The image $C(\mathcal{A}, j) = im(ev_{\mathcal{A}})$ is a linear subspace in $\mathbb{F}_q^n$ that we call a *subspace arrangement code.*

Now we examine a few examples.

Example 3.2. Let $q = 2$, $\ell = 3$, $j = 1$, and $\mathcal{A}$ be the xy-, xz-, and yz-planes, as seen in Figure 1. Then, $I(\mathcal{A}) = < x_1x_2x_3 >$. Thus, $V = \mathbb{F}_2^3$ and $S = \mathbb{F}_q[x_1x_2x_3]$. It follows that $P(\mathcal{A}) = \{(0,0,0)$, (1,0,0), (1,1,0), (1,0,1), (0,1,0), (0,1,1), $(0,0,1)\}$, so $|P(\mathcal{A})| = 7$. Now, to find im$(ev_{\mathcal{A}} : S_{\leq 1} \to \mathbb{F}_2^7)$, we write a basis for the subspace spanned by the image of the evaluation map as a matrix (the top row designates a point in $P(\mathcal{A})$, and the first column delineates the polynomials in $S_{\leq 1}$ at which each point is evaluated):

$$\begin{array}{c|ccccccc}
 & (0,0,0) & (1,0,0) & (1,1,0) & (1,0,1) & (0,1,0) & (0,1,1) & (0,0,1) \\
\hline
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
x_1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
x_2 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
x_3 & 0 & 0 & 0 & 1 & 0 & 1 & 1
\end{array} .$$

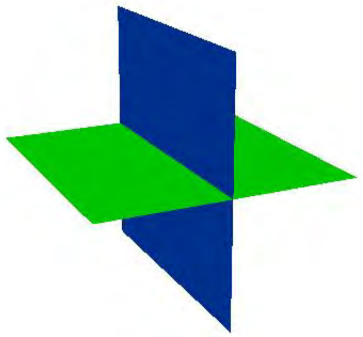

FIGURE 2. The xz- and yz-planes

Close observation yields that the length of the code is 7, the dimension is 4, and the minimum distance is 3. Therefore, this code is a $[7, 4, 3]_2$ code.

Example 3.3. Let $q = 7$, $\ell = 2$, $j = 2$, and $\mathcal{A}$ be the xz- and yz-planes as in Figure 2. Then, $I(\mathcal{A}) = < x_1x_2 >$. Thus, $V = \mathbb{F}_7^2$ and $S = \mathbb{F}_7[x_1x_2]$. It follows that $P(\mathcal{A}) = \{(0, 0)$, (1,0), (2,0), (3,0), (4,0), (5,0), (6,0), (0,1), (0,2), (0,3), (0,4), (0,5), $(0, 6)\}$, implying $|P(\mathcal{A})| = 13$. Now, to find $\text{im}(ev_{\mathcal{A}} : S_{\leq 2} \to \mathbb{F}_7^{13})$, determine its corresponding matrix:

	$(0,0)$	$(1,0)$	$(2,0)$	$(3,0)$	$(4,0)$	$(5,0)$	$(6,0)$	$(0,1)$	$(0,2)$	$(0,3)$	$(0,4)$	$(0,5)$	$(0,6)$
1	1	1	1	1	1	1	1	1	1	1	1	1	1
x_1	0	1	2	3	4	5	6	0	0	0	0	0	0
x_2	0	0	0	0	0	0	0	1	2	3	4	5	6
x_1^2	0	1	4	2	2	4	1	0	0	0	0	0	0
x_2^2	0	0	0	0	0	0	0	1	4	2	2	4	1

Close observation yields that the length of the code is 13, the dimension is 5, and the minimum distance is 5. Therefore, this code is a $[13, 5, 5]_7$ code.

Example 3.4. Note that if $\mathcal{A}$ is the entirety of $V = \mathbb{F}_q^\ell$ (so $I(\mathcal{A}) = < 0 >$) and $q = 2$, then $P(\mathcal{A}) = V$. Thus, the subspace arrangement code is equivalent to a Reed-Muller code. Specifically, the code is $C(\mathbb{F}_q^\ell, j) = \mathcal{R}(j, \ell)$. Since skeletal codes (as discussed in Section 6.3) are a punctured $C(\mathbb{F}_q^\ell, j)$ code, they are also a punctured Reed-Muller code.

3.2. **Code Length and Characteristic Polynomial.** Let $\mathcal{A}$ be the subspace arrangement $\{X_1, \ldots, X_k\}$, where for all i, X_i is a subspace of the coordinate hyperplanes. Let $I_{\mathcal{A}}$ be the corresponding ideal in $\mathbb{F}_q^\ell$. Let $L(\mathcal{A})$ consist of all intersections of the subspaces of $\mathcal{A}$ (note that the empty intersection is defined as the entire vector space, V). Then $L(\mathcal{A})$ is a lattice and a poset by reverse inclusion (a poset is a partially-ordered set, meaning that an order relation, such as reverse inclusion, exists on some of the elements of the set; a lattice is a poset where any two elements have a unique least upper bound and greatest lower bound). Next, the Möbius function, μ, on $L(\mathcal{A})$ is $\mu : L(\mathcal{A}) \longrightarrow \mathbb{Z}$ defined recursively by

$$\begin{cases} \mu(V) = 1 \\ \mu(X) = -\sum\limits_{Y \lneq X} \mu(Y) \end{cases} .$$

From this function, the characteristic polynomial for $\mathcal{A}$, $\chi(\mathcal{A}, t)$, is defined as

$$\chi(\mathcal{A}, t) = \sum_{X \in L(\mathcal{A})} (\mu(X) t^{\dim(X)}).$$

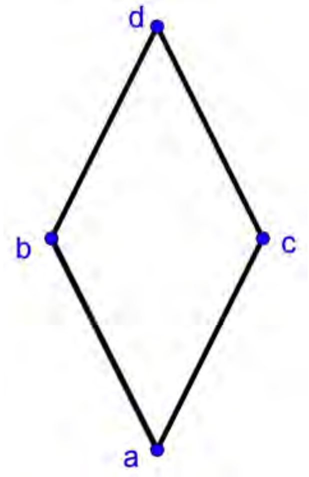

FIGURE 3. Poset of 2-Boolean

In [1], Christos Athanasiadis proved a theorem for determining the number of the set of points of $\mathcal{A}$ (in $\mathbb{F}_q^\ell$):

$$|P(\mathcal{A})| = q^\ell - \chi(\mathcal{A}, q).$$

Since $|P(\mathcal{A})|$ corresponds to the length of a code, we have the following interpretation of Athanasiadis's result.

Corollary 3.5. *If n is the length of a code associated with the subspace arrangement $\mathcal{A}$, with characteristic polynomial $\chi(\mathcal{A}, q)$, then*

$$n = |P(\mathcal{A})| = q^\ell - \chi(\mathcal{A}, q).$$

The following example illustrates this process.

Example 3.6. Let $V = \mathbb{F}_3^2$ and $\mathcal{A}$ be the x-axis and y-axis. Then, $I(\mathcal{A}) = < xy >$, with a corresponding poset $\{a, b, c, d\}$, where $a \leq b, a \leq c, a \leq d, b \leq d$, and $c \leq d$. For an easier understanding of this process, the poset can be interpreted to have a correspondence to $\mathbb{R}^2$ (and the 2-Boolean, which is the Boolean arrangement [discussed in Section 5] in $\mathbb{R}^2$) as follows: $a = V = \mathbb{R}^2$, $b =$y-axis, $c =$x-axis, and $d =$origin. Figure 3 is a picture of the Hasse diagram of this poset. It follows that $L(\mathcal{A}) = \{a, b, c, d\} = \{V, b, c, d\}$. Using the Möbius function, $\mu(V) = 1$, $\mu(b) = -\mu(V) = -1$ $(a \leq b)$, $\mu(c) = -\mu(V) = -1$ $(a \leq c)$, and $\mu(d) = -(\mu(c) + \mu(b) + \mu(V)) = -(-1 - 1 + 1) = 1$ $(c \leq d, b \leq d, a \leq d)$. Thus, the characteristic polynomial is

$$\chi(\mathcal{A}, t) = [1 * t^2] + [-1 * t^1] + [-1 * t^1] + [1 * t^0] = t^2 - 2t + 1$$

(each bracketed addend corresponds to the evaluation of a, b, c, and d, respectively). Finally, evaluating using Athanasiadis's theorem gives

$$n = |\mathcal{A}| = q^\ell - \mathcal{X}(\mathcal{A}, q) = 3^2 - \chi(\mathcal{A}, 3) = 3^2 - [3^2 - 2 * 3 + 1] = 5.$$

Working out $P(\mathcal{A})$ by hand confirms the answer:

$$P(\mathcal{A}) = \{(0,0), (1,0), (2,0), (0,1), (0,2)\} \implies |P(\mathcal{A})| = |\mathcal{A}| = 5.$$

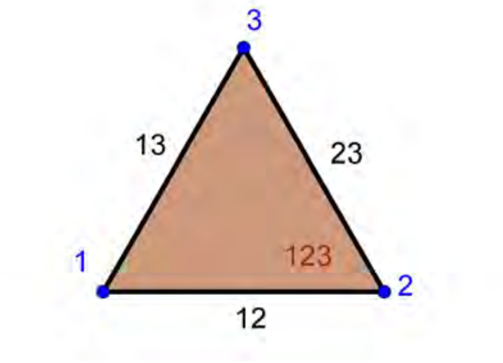

FIGURE 4. 3-simplex

4. Coordinate Arrangement Codes

We want to study the dimension and minimum distance of $C(\mathcal{A}, j)$ for any subspace arrangement $\mathcal{A}$, but doing so is difficult. For the rest of the paper, focus will be placed on subspace arrangements that are coordinate arrangements, which are described in this section.

4.1. Simplicial Complexes.

In order to study subspace arrangements that are intersections of coordinate hyperplanes, we first define and build a theory for what are called simplicial complexes. We follow the standard formulation of the correspondence between coordinate arrangements and simplicial complexes written in [2] by Björner. For convenience, let $[k] = \{1, \ldots, k\}$ be the set of numbers 1 through k. A *simplicial complex*, Δ, is a set of subsets of $[k]$ such that

$$(1)\,\text{if}\ \sigma \in \Delta\ \text{and}\ \tau\ \text{is a subset of}\ \sigma,\ \text{then}\ \tau \in \Delta,\ \text{and}$$
$$(2)\,\text{if}\ x \in [k],\ \text{then}\ \{x\} \in \Delta$$

where $[k]$ are the vertices of Δ and σ is a face of Δ. A k-simplex is a simplicial complex that contains all subsets of $[k]$. Geometrically, we can view the 1-simplex as a point, the 2-simplex as a line segment, the 3-simplex as a triangle, the 4-simplex as a tetrahedron, and such forth.

Example 4.1. Let $k = 3$ and $\Delta = \{\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$. We can think of Δ geometrically as a filled-in triangle by associating a face of size 3 to the 3-simplex. In fact, this simplicial complex is exactly the 3-simplex as pictured in Figure 4.

Example 4.2. Let $k = 3$ and $\Delta = \{\{1\}, \{2\}, \{3\}, \{1, 2\}, \{2, 3\}, \{1, 3\}\}$. Here the simplicial complex is geometrically represented by a triangle with the inside missing, which is pictured in Figure 5.

The combinatorial setting allows one to easily study higher dimensional geometric objects. In order to use simplicial complexes to produce codes, we associate an ideal to a simplicial complex. For a simplicial complex Δ with vertices $[\ell]$, the vertices correspond to the variables of the polynomial ring $\mathbb{F}_q[x_1, \ldots, x_\ell]$, and the ideal is $I_\Delta = \{x_\sigma \mid \sigma \notin \Delta\}$.

4.2. Definition of $C(\mathcal{A}_\Delta, j)$.

The process of correlating a subspace arrangement with a simplicial complex, as described in [2], is as follows. For $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ (a basis of $\mathbb{F}_q^n$) and each subset $\sigma = \{i_1, \ldots, i_s\} \subseteq [n]$, let $K_\sigma = \text{span}\{\mathbf{b}_{i_1}, \ldots, \mathbf{b}_{i_s}\}$. The subspace arrangement

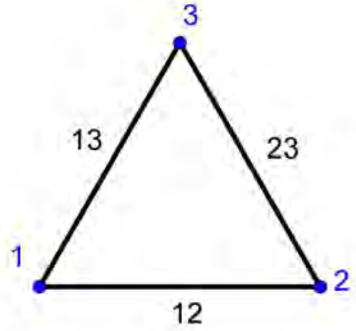

FIGURE 5. Empty triangle

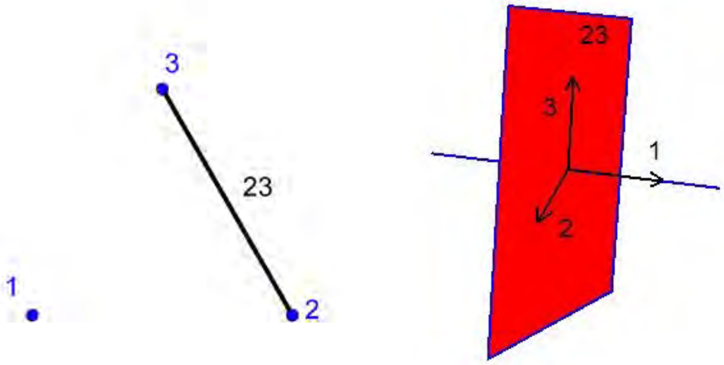

FIGURE 6. Simplicial complex corresponding to the x-axis and yz-plane

$\mathcal{A}_\Delta = \{K_\sigma | \sigma \in \Delta\}$ is thus determined by the simplicial complex Δ. With this correlation in mind, it is now possible to define a coordinate arrangement code.

Definition 4.3. Let Δ be a simplicial complex with corresponding subspace arrangement $\mathcal{A}_\Delta$. Then the *coordinate arrangement code* corresponding to Δ is the subspace arrangement code $C(\mathcal{A}_\Delta, j)$.

The construction of error-correcting codes from simplicial complexes, as well as the correlation between subspace arrangements and simplicial complexes, can be demonstrated through examples.

Example 4.4. Let $\Delta = \{\{1\}, \{2, 3\}\} \subseteq 2^{[3]}$. The subspace arrangement then consists of the span of the first variable joined with the span of the second and third variables (that is, $< x > \cup < y, z >$). This subspace arrangement thus consists of the x-axis together with the yz-plane. The simplicial complex and subspace arrangement are depicted in Figure 6. From the construction described above and recalling the conventions discussed in Section 3.1, it is clear that $\ell = 3$. Let $q = 2$ and $j = 1$ (note that q and j are not contingent on the construction of Δ), so $V = \mathbb{F}_2^3$ and $S = \mathbb{F}_2[x_1 x_2 x_3]$. It follows that $I_\Delta = < x_1 x_2, x_1 x_3 >$.

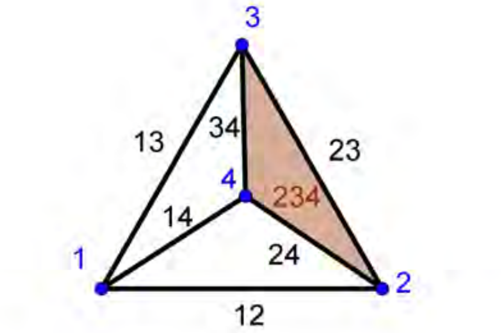

FIGURE 7. Tetrahedron

Hence, $P(\mathcal{A}_\Delta) = \{(0,0,0)$, (1,0,0), (0,1,0), (0,0,1), $(0,1,1)\}$, implying $|P(\mathcal{A}_\Delta)| = 5$. The matrix corresponding to $\text{im}(ev : S_{\leq 1} \to \mathbb{F}_2^5)$ is given below.

	$(0,0,0)$	$(1,0,0)$	$(0,1,0)$	$(0,0,1)$	$(0,1,1)$
1	1	1	1	1	1
x_1	0	1	0	0	0
x_2	0	0	1	0	1
x_3	0	0	0	1	1
x_2x_3	0	0	0	0	1

.

Thus, the length is 5, the dimension is 5, and the minimum distance is 1, so the code is a $[5,5,1]_2$ code.

Example 4.5. Let Δ represent the simplicial complex interpreted geometrically as a tetrahedron with vertices 1,2,3, and 4 and all 1-dimensional faces (the lines connecting each vertex), as well as the 2-dimensional face (a plane) {2,3,4}. Thus, Δ lacks three "planes" and is not "filled in" (since $\{1,2,3,4\} \notin \Delta$). The simplicial complex can be interpreted geometrically as in Figure 7.

From the construction described above and recalling the conventions discussed in Section 3.1, it is clear that $\ell = 4$. Let $q = 2$ and $j = 1$, so $V = \mathbb{F}_2^4$ and $S = \mathbb{F}_2[x_1x_2x_3x_4]$. It follows that $I_{\mathcal{A}_\Delta} = < x_1x_2x_3, x_1x_2x_4, x_1x_3x_4 >$. Hence, $P(\mathcal{A}_\Delta) = \{(0,0,0,0)$, (0,0,0,1), (0,0,1,0), (0,1,0,0), (1,0,0,0), (0,0,1,1), (0,1,0,1), (0,1,1,0), (1,0,0,1), (1,0,1,0), (0,1,0,0), $(0,1,1,1)\}$, implying $|P(\mathcal{A}_\Delta)| = 12$. The matrix corresponding to $\text{im}(ev : S_{\leq 1} \to \mathbb{F}_2^{12})$ is given below.

	(0000)	(0001)	(0010)	(0100)	(1000)	(0011)	(0101)	(0110)	(1001)	(1010)	(0100)	(0111)
1	1	1	1	1	1	1	1	1	1	1	1	1
x_1	0	0	0	0	1	0	0	0	1	1	0	0
x_2	0	0	0	1	0	0	1	1	0	0	1	1
x_3	0	0	1	0	0	1	0	1	0	1	0	1
x_4	0	1	0	0	0	1	1	0	1	0	0	1

.

Thus, the length is 12, the dimension is 5, and the minimum distance is 3, so the code is a $[12,5,3]_2$ code.

4.3. **Upper Bound on Dimension.** In order to find an upper bound on the dimension of a code, the Hilbert function, $H(M,i)$ will be introduced. The Hilbert function deals with a graded S-module, M. $H(M,i)$ is defined as the dimension of M_i, the i^{th} graded component of M, so $H(M,i)$ =dim(M_i). Furthermore, if $M = \mathbb{C}[x_1,\ldots,x_\ell]$, then

$$H(M,i) = \binom{i+\ell-1}{\ell-1} = \binom{i+\ell-1}{i}.$$

This formula results from the realization of (given how many variables are in the polynomial ring) how many different ways there are of choosing a term of a certain degree. The formula follows from a theorem in discrete mathematics [20], where the number of selections with repetition of r objects chosen from s types of objects is $\binom{r+s-1}{r}$ (the r objects are each degree of a term in a polynomial, and the s objects are each variable). For example, if there are three variables, x_1, x_2, and x_3, and it is sought how many monomials are of degree 2, then there are three slots to put a total of 2 degrees, so there are $\binom{3+2-1}{2} = \binom{4}{2} = 6$ different monomials of degree 2 formed from three variables $(\{x_1^2, x_2^2, x_3^2, x_1x_2, x_1x_3, x_2x_3\})$. This formula can be applied in a more general sense to a simplicial complex Δ.

The Hilbert series is obtained from the Hilbert function. The Hilbert series of the graded module M is defined as $HS(M,t) = \sum\limits_{i=0}^{\infty} H(M,i)t^i$, meaning that the Hilbert series is a power series of t such that the terms of degree i refer to the dimension of the i^{th} graded component of M. For a simplicial complex Δ, vector space $V = \mathbb{F}_q^\ell$, and homogeneous ideal I_Δ, let $\mathbb{F}[\Delta] = S/I_\Delta$ be the Stanley-Reisner ring. Since I_Δ is homogeneous, $\mathbb{F}[\Delta]$ is graded. The face vector of a simplicial complex is defined by $f_i = |\{\sigma \in \Delta : |\sigma| = i\}|$ for $0 \leq i$. The face vector refers to the number of faces of dimension $i-1$ in Δ (f_0 is defined as $f_0 = 1$). The Hilbert series of the Stanley-Reisner ring (as discussed in [17]) is

$$HS(\mathbb{F}[\Delta],t) = f_0 + \sum_{i=1}^{\infty}\sum_{m=1}^{D+1}\binom{i-1}{m-1} f_m t^i,$$

where D refers to the dimension of Δ. As a corollary to Stanley's work in [17], an upper bound for dimension exists.

Corollary 4.6. *For a simplicial complex Δ with corresponding subspace arrangement $\mathcal{A}_\Delta$, an upper bound for $k = \dim(C(\mathcal{A}_\Delta, j))$ exists:*

$$k \leq f_0 + \sum_{i=1}^{j}\sum_{m=1}^{D+1}\binom{i-1}{m-1} f_m.$$

Using the combinatorial formula discussed at the beginning of this section, the upper bound can also be described as

$$k \leq \sum_{m=0}^{j}\binom{m+\ell-1}{m}.$$

An example illustrates this upper bound.

Example 4.7. For $V = \mathbb{F}_2^2$, let Δ correspond to two connected points, so $\mathcal{A}_\Delta$ is a plane (and the entirety of V), so $I_\Delta = <0>$. Thus, $f_0 = 1$, $f_1 = 2$, (for each of the two 0-dimensional points), and $f_2 = 1$ (for the single edge). An upper bound for the dimension of $C(\mathcal{A}_\Delta, 1)$ can be found. Using the above formula for the Hilbert series,

$$HS(\mathbb{F}[\Delta],t) = f_0 + \sum_{i=1}^{j}\sum_{m=1}^{D+1}\binom{i-1}{m-1} f_m$$

$$\begin{aligned}
&= f_0 + \sum_{i=1}^{1} \sum_{m=1}^{1+1} \binom{i-1}{m-1} f_m \\
&= f_0 + \sum_{i=1}^{1} \left(\binom{i-1}{0} f_1 + \binom{i-1}{1} f_2 \right) \\
&= f_0 + \binom{0}{0} f_1 + \binom{0}{1} f_2 \\
&= (1)(1) + (1)(2) + (0)(1) = 3.
\end{aligned}$$

As a check, note that there are exactly 3 polynomials in $\mathbb{F}_2[x_1 x_2]$ of degree less than or equal to 1: $\{1, x_1, x_2\}$. Thus, the upper bound is 3.

5. Boolean Arrangement Codes

In this section, the Boolean arrangement (consisting of all coordinate hyperplanes for a vector space) is used to produce ideals and codes. Notation concerning Boolean arrangement codes follows.

Definition 5.1. A **Boolean arrangement code**, BC(ℓ, j), is a code generated by the coordinate hyperplane arrangement and the ideal $I = < x_1 \cdots x_\ell >$.

This section is concerned with results dealing with the dimension of Boolean arrangement codes in certain circumstances.

Proposition 5.2. The dimension of BC(ℓ,1) is $k = \ell + 1$.

Proof. Previous work allowed the extension of a theorem by Richard Stanley to be applied to this class of error-correcting codes, establishing an upper bound for dimension:

$$k \leq \sum_{m=0}^{j} \binom{m+\ell-1}{m} = \sum_{m=0}^{1} \binom{m+\ell-1}{m} = \binom{\ell-1}{0} + \binom{\ell}{1} = \ell + 1.$$

The dimension of $\ell + 1$ refers to the number of linearly independent row vectors produced by the matrix construction of the evaluation map $ev_{\mathcal{A}}$ for generating the code. Each row corresponds to a polynomial. The polynomials are $1, x_1, \ldots, x_\ell$. Thus, it suffices to show that the row vectors $\mathbf{v_0}, \mathbf{v_1}, \ldots, \mathbf{v}_\ell$ produced by the respective polynomials $1, x_1, \ldots, x_\ell$ are linearly independent. Define e_i as the point $(0, 0, \ldots, 0, 1, 0, \ldots, 0)$ in $\mathbb{F}_2^\ell$ such that all components in e_i are 0 except for the ith position. Furthermore, define e_0 as the origin, $(0, 0, \ldots, 0, \ldots, 0)$. Note that regardless of the selection of q, e_i for $i = 0, \ldots, \ell$ evaluates to 0 on I, so $e_i \in P(\mathcal{A})$ for $i = 0, \ldots, \ell$. For simplicity (but without loss of generality), let $e_0, \ldots, e_\ell$ correspond to the first $\ell + 1$ columns in the generating matrix G:

$$\begin{array}{c|ccccccc}
 & e_0 & e_1 & e_2 & \cdots & e_{\ell-1} & e_\ell & \cdots \\
\hline
1 & 1 & 1 & 1 & \cdots & 1 & 1 & \cdots \\
x_1 & 0 & 1 & 0 & \cdots & \cdots & 0 & \cdots \\
x_2 & 0 & 0 & 1 & 0 & \cdots & 0 & \cdots \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \cdots \\
x_{\ell-1} & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots \\
x_\ell & 0 & 0 & \cdots & \cdots & 0 & 1 & \cdots
\end{array}.$$

G is clearly upper triangular and hence (by the properties of an upper triangular matrix) has linearly independent rows. $\square$

Remark 5.3. For $j > 1$, the process becomes increasingly difficult. Some results for specific selections of ℓ, q, and j follow. For results concerning the case $q = 2$, see Section 6.3.

Proposition 5.4. The dimension of BC(2,2) whenever $q > 2$ is $k = 5$.

Proof. Since $q > 2$, $\mathbb{F}_q^2$ contains at least 3 elements, $0, 1$, and ω. In the matrix construction, let the first 5 columns correspond to the respective points $(0,0), (1,0), (0,1), (\omega,0)$, and $(0,\omega)$, and let the rows correspond to the respective polynomials $1, x_1, x_2, x_1^2$, and x_2^2. Thus, the resulting generating matrix, G, is as follows:

$$\begin{array}{c|ccccc}
 & (0,0) & (1,0) & (0,1) & (\omega,0) & (0,\omega)\ldots \\
\hline
1 & 1 & 1 & 1 & 1 & 1\ldots \\
x_1 & 0 & 1 & 0 & \omega & 0\ldots \\
x_2 & 0 & 0 & 1 & 0 & \omega\ldots \\
x_1^2 & 0 & 1 & 0 & \omega^2 & 0\ldots \\
x_1^2 & 0 & 0 & 1 & 0 & \omega^2\ldots
\end{array}.$$

We show the rows are linearly independent by computing a maximal minor and showing it is not equal to zero. The minor we examine is the minor with columns 1 through 5. Computing the cofactor expansion down the first column gives that the determinant is equal to the determinant of the following matrix:

$$G' = \begin{bmatrix} 1 & 0 & \omega & 0 \\ 0 & 1 & 0 & \omega \\ 1 & 0 & \omega^2 & 0 \\ 0 & 1 & 0 & \omega^2 \end{bmatrix}.$$

The determinant of G' (and the maximal minor) is $\omega^2(\omega - 1)^2 \neq 0$ since $\mathbb{F}_q$ is a field and thus an integral domain, meaning it has no zero divisors (note that $\omega \neq 0$, $\omega \neq 1$). □

Proposition 5.5. The dimension of BC(3,2) whenever $q > 2$ is $k = 10$.

Proof. First note that 10 corresponds to the upper bound:

$$\sum_{m=0}^{j} \binom{m+\ell-1}{m} = \sum_{m=0}^{2} \binom{m+2}{m} = \binom{2}{0} + \binom{3}{1} + \binom{4}{2} = 10.$$

The upper bound corresponds to the polynomials $1, x_1, x_2, x_3, x_1^2, x_2^2, x_3^2, x_1x_2, x_1x_3, x_2x_3$.

Thus, carefully selecting the points in $P(\mathcal{A})$ for the columns (which can be ordered in any way), the left part of the generating matrix G is

$$\begin{array}{c|cccccccccc}
 & \mathbf{0} & (1,0,0) & (0,1,0) & (0,0,1) & (\omega,0,0) & (0,\omega,0) & (0,0,\omega) & (1,\omega,0) & (1,0,\omega) & (0,1,\omega)\ldots \\
\hline
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\ldots \\
x_1 & 0 & 1 & 0 & 0 & \omega & 0 & 0 & 1 & 1 & 0\ldots \\
x_2 & 0 & 0 & 1 & 0 & 0 & \omega & 0 & \omega & 0 & 1\ldots \\
x_3 & 0 & 0 & 0 & 1 & 0 & 0 & \omega & 0 & \omega & \omega\ldots \\
x_1^2 & 0 & 1 & 0 & 0 & \omega^2 & 0 & 0 & 1 & 1 & 0\ldots \\
x_1^2 & 0 & 0 & 1 & 0 & 0 & \omega^2 & 0 & \omega^2 & 0 & 1\ldots \\
x_3^2 & 0 & 0 & 0 & 1 & 0 & 0 & \omega^2 & 0 & \omega^2 & \omega^2\ldots \\
x_1x_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega & 0 & 0\ldots \\
x_1x_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega & 0\ldots \\
x_2x_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega\ldots
\end{array}.$$

Then subtract rows 1, 2, and 3 from rows 4, 5, and 6, respectively. The resulting 10×10 matrix to the left is upper triangular with $\omega^2 - \omega$ on the diagonal of columns 4, 5, and 6:

$$\begin{array}{c|cccccccccc}
 & (0,0,0) & (1,0,0) & (0,1,0) & (0,0,1) & (\omega,0,0) & (0,\omega,0) & (0,0,\omega) & (1,\omega,0) & (1,0,\omega) & (0,1,\omega)\ldots \\
\hline
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\ldots \\
x_1 & 0 & 1 & 0 & 0 & \omega & 0 & 0 & 1 & 1 & 0\ldots \\
x_2 & 0 & 0 & 1 & 0 & 0 & \omega & 0 & \omega & 0 & 1\ldots \\
x_3 & 0 & 0 & 0 & 1 & 0 & 0 & \omega & 0 & \omega & \omega\ldots \\
x_1^2 & 0 & 0 & 0 & 0 & \omega^2-\omega & 0 & 0 & 1 & 1 & 0\ldots \\
x_1^2 & 0 & 0 & 0 & 0 & 0 & \omega^2-\omega & 0 & \omega^2 & 0 & 1\ldots \\
x_3^2 & 0 & 0 & 0 & 0 & 0 & 0 & \omega^2-\omega & 0 & \omega^2 & \omega^2\ldots \\
x_1x_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega & 0 & 0\ldots \\
x_1x_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega & 0\ldots \\
x_2x_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega\ldots
\end{array}.$$

Since $q > 2$, these entries (as well as all other diagonal entries) are not zero, so the determinant of the matrix (and the maximal minor) is non-zero, directly implying that the rows are linearly independent. □

Proposition 5.6. The dimension of BC(ℓ,2) whenever $q > 2$ and $\ell \geq 3$ is $k = \frac{\ell^2+\ell}{2} + \ell + 1$.

Proof. First note that $k = \frac{\ell^2+\ell}{2} + \ell + 1$ corresponds to the upper bound:

$$\sum_{m=0}^{j} \binom{m+\ell-1}{m} = \sum_{m=0}^{2} \binom{m+\ell-1}{m}$$

$$= \binom{\ell-1}{0} + \binom{\ell}{1} + \binom{\ell+1}{2} = 1 + \ell + \frac{(\ell+1)!}{(\ell-1)!(2)} = 1 + \ell + \frac{(\ell+1)(\ell)}{2} = \frac{\ell^2+\ell}{2} + \ell + 1.$$

Notice that since this number is the upper bound, the proposed dimension corresponds to all polynomials in $\mathbb{F}_q[x_1, \ldots, x_\ell]$ of degree 0, 1, and 2. Thus, it is sufficient to show that all $\frac{\ell^2+\ell}{2} + \ell + 1$ rows in the generating matrix, G, are linearly independent. Since $q > 2$ there exists $\omega \in \mathbb{F}_q$ such that $\omega \neq 0, 1$. We choose a specific order on the points of the arrangement so that the matrix G can be viewed as

$$\begin{array}{c|ccccccccccc}
 & \mathbf{0} & (10\ldots0) & \ldots & (\omega0\ldots0) & \ldots & (0\ldots0\omega) & (1\omega0\ldots0) & \ldots & (10\ldots0\omega) & \ldots & (0\ldots01\omega)\ldots \\
\hline
1 & 1 & 1 & \ldots & 1 & \ldots & 1 & 1 & \ldots & 1 & \ldots & 1\ldots \\
x_1 & 0 & 1 & \ldots & \omega & \ldots & 0 & 1 & \ldots & 1 & \ldots & 0\ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\ldots \\
x_\ell & 0 & 0 & \ldots & 0 & \ldots & \omega & 0 & \ldots & \omega & \ldots & \omega\ldots \\
x_1^2 & 0 & 1 & \ldots & \omega^2 & \ldots & 0 & 1 & \ldots & 1 & \ldots & 0\ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\ldots \\
x_\ell^2 & 0 & 0 & \ldots & 0 & \ldots & \omega^2 & 0 & \ldots & \omega^2 & \ldots & \omega^2\ldots \\
x_1x_2 & 0 & 0 & \ldots & 0 & \ldots & 0 & \omega & \ldots & 0 & \ldots & 0\ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\ldots \\
x_1x_\ell & 0 & 0 & \ldots & 0 & \ldots & 0 & 0 & \ldots & \omega & \ldots & 0\ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\ldots \\
x_{\ell-1}x_\ell & 0 & 0 & \ldots & 0 & \ldots & 0 & \ldots & 0 & \ldots & 0 & \omega\ldots
\end{array}.$$

Subtract rows $1, \ldots, \ell$ from rows $\ell+1, \ldots, 2\ell$, respectively. The resulting leftmost square matrix is upper triangular with non-zero entries on the diagonal, so it has a non-zero determinant. Thus, the maximal minor is non-zero, implying that the rows are linearly independent. □

Proposition 5.7. The dimension of BC(4,3) whenever $q > 3$ is $k = 35$.

Proof. Of first note is the upper bound for these parameters:

$$k \leq \sum_{m=0}^{j} \binom{m+\ell-1}{m} = \sum_{m=0}^{3} \binom{m+3}{m} = \binom{3}{0} + \binom{4}{1} + \binom{5}{2} + \binom{6}{3} = 1 + 4 + 10 + 20 = 35.$$

Thus, in order to show that the dimension is 35, it suffices to show that each row vector in the matrix construction of the code (of which there are 35 for each polynomial of degree less than or equal to 3) is linearly independent. Furthermore, careful ordering of the columns of the matrix construction will lend themselves to a clear determination of linear independence. Let $\omega \in \mathbb{F}_q \backslash \{0,1\}$ such that $\omega \neq \omega^2$ and $\omega \neq -1$. Thus, the matrix construction can be as follows:

$$G = \begin{bmatrix} M_1 & * & * \\ 0 & M_2 & * \\ 0 & 0 & M_3 \end{bmatrix}.$$

M_1, M_2, and M_3 are all square sub-matrices (of different dimensions) along the diagonal of G (thus, 0 and $*$ represent square sub-matrices elsewhere in the matrix, where 0 denotes all entries in the sub-matrix are 0). It will be demonstrated that M_1, M_2, and M_3 are all able to be made to be upper triangular, meaning that G can be made to be upper triangular and implying that all of the rows are linearly independent.

First, M_1 contains the rows corresponding to the polynomials { 1, x_1, x_2, x_3, x_4, x_1^2, x_2^2, x_3^2, x_4^2, x_1^3, x_2^3, x_3^3, x_4^3 } and the columns corresponding to the points { (0,0,0,0), (1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1) (ω,0,0,0), (0,ω,0,0) (0,0,ω,0) (0,0,0,ω), (ω^2,0,0,0), (0,ω^2,0,0), (0,0,ω^2,0), (0,0,0,ω^2)}. Observe that M_1 is as follows:

$$M_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & \omega & 0 & 0 & 0 & \omega^2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \omega & 0 & 0 & 0 & \omega^2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \omega & 0 & 0 & 0 & \omega^2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \omega & 0 & 0 & 0 & \omega^2 \\ 0 & 1 & 0 & 0 & 0 & \omega^2 & 0 & 0 & 0 & \omega^4 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \omega^2 & 0 & 0 & 0 & \omega^4 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \omega^2 & 0 & 0 & 0 & \omega^4 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \omega^2 & 0 & 0 & 0 & \omega^4 \\ 0 & 1 & 0 & 0 & 0 & \omega^3 & 0 & 0 & 0 & \omega^6 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \omega^3 & 0 & 0 & 0 & \omega^6 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \omega^3 & 0 & 0 & 0 & \omega^6 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \omega^3 & 0 & 0 & 0 & \omega^6 \end{bmatrix}.$$

In order to make M_1 upper triangular, it suffices to use row-reduction techniques on M_1. To describe the row-reduction techniques being used, some new notation will be introduced. Let the rows 1-4 of the above matrix be collective called **A**, the next four **B**, and the last four **C** (the 0th row does not need to change during the operations and thus is not included). Let operations $c\mathbf{A} + d\mathbf{B}$ be defined as the multiplication of each row in **A** and **B** by c and d, respectively, and then the component-wise addition of **A** and **B** (so the first row in **A** is added to the first row in **B**; this process is analogous to creating a new vector space). Thus, (through row-reduction) replace **B** by **B** − **A**, and replace **C** by **C** − **B** − ω(**B** − **C**):

$$\begin{bmatrix}
1&1&1&1&1&1&1&1&1&1&1&1&1\\
0&1&0&0&0&\omega&0&0&0&\omega^2&0&0&0\\
0&0&1&0&0&0&\omega&0&0&0&\omega^2&0&0\\
0&0&0&1&0&0&0&\omega&0&0&0&\omega^2&0\\
0&0&0&0&1&0&0&0&\omega&0&0&0&\omega^2\\
0&0&0&0&0&\omega^2-\omega&0&0&0&\omega^4-\omega^2&0&0&0\\
0&0&0&0&0&0&\omega^2-\omega&0&0&0&\omega^4-\omega^2&0&0\\
0&0&0&0&0&0&0&\omega^2-\omega&0&0&0&\omega^4-\omega^2&0\\
0&0&0&0&0&0&0&0&\omega^2-\omega&0&0&0&\omega^4-\omega^2\\
0&0&0&0&0&0&0&0&0&\omega^6-\omega^5-\omega^4+\omega^3&0&0&0\\
0&0&0&0&0&0&0&0&0&0&\omega^6-\omega^5-\omega^4+\omega^3&0&0\\
0&0&0&0&0&0&0&0&0&0&0&\omega^6-\omega^5-\omega^4+\omega^3&0\\
0&0&0&0&0&0&0&0&0&0&0&0&\omega^6-\omega^5-\omega^4+\omega^3
\end{bmatrix}.$$

Note that this is an upper triangular matrix, so the determinant of it is the product of the diagonal entries:

$$1^4(\omega^2-\omega)^4(\omega^6-\omega^5-\omega^4+\omega^3)^4$$

$$=\omega^4(\omega-1)^4\Big(\omega^3\big((\omega-1)(\omega+1)(\omega-1)\big)\Big)^4$$

$$=\omega^{16}(\omega-1)^{12}(\omega+1)^4.$$

Thus, the determinant is 0 if and only if $\omega=0,1,-1$, which is not the case. Therefore, M_1 can be made to be upper triangular with a non-zero determinant.

Now, examine M_2, which contains the rows corresponding to the polynomials { x_1x_2, x_1x_3, x_1x_4, x_2x_3, x_2x_4, x_3x_4, $x_1^2x_2$, $x_1^2x_3$, $x_1^2x_4$, $x_2^2x_3$, $x_2^2x_4$, $x_3^2x_4$, $x_1x_2^2$, $x_1x_3^2$, $x_1x_4^2$, $x_2x_3^2$, $x_2x_4^2$, $x_3x_4^2$ } and the columns corresponding to the points { (1,1,0,0), (1,0,1,0), (1,0,0,1), (0,1,1,0), (0,1,0,1), (0,0,1,1), (1,ω,0,0), (1,0,ω,0), (1,0,0,ω), (0,1,ω,0), (0,1,0,ω), (0,0,1,ω), (ω,ω^2,0,0), (ω,0,ω^2,0), (ω,0,0,ω^2), (0,ω,ω^2,0), (0,ω,0,ω^2), (0,0,ω,ω^2)}:

$$M_2=\begin{bmatrix}
1&0&0&0&0&0&\omega&0&0&0&0&0&\omega^3&0&0&0&0&0\\
0&1&0&0&0&0&0&\omega&0&0&0&0&0&\omega^3&0&0&0&0\\
0&0&1&0&0&0&0&0&\omega&0&0&0&0&0&\omega^3&0&0&0\\
0&0&0&1&0&0&0&0&0&\omega&0&0&0&0&0&\omega^3&0&0\\
0&0&0&0&1&0&0&0&0&0&\omega&0&0&0&0&0&\omega^3&0\\
0&0&0&0&0&1&0&0&0&0&0&\omega&0&0&0&0&0&\omega^3\\
1&0&0&0&0&0&\omega&0&0&0&0&0&\omega^4&0&0&0&0&0\\
0&1&0&0&0&0&0&\omega&0&0&0&0&0&\omega^4&0&0&0&0\\
0&0&1&0&0&0&0&0&\omega&0&0&0&0&0&\omega^4&0&0&0\\
0&0&0&1&0&0&0&0&0&\omega&0&0&0&0&0&\omega^4&0&0\\
0&0&0&0&1&0&0&0&0&0&\omega&0&0&0&0&0&\omega^4&0\\
0&0&0&0&0&1&0&0&0&0&0&\omega&0&0&0&0&0&\omega^4\\
1&0&0&0&0&0&\omega^2&0&0&0&0&0&\omega^5&0&0&0&0&0\\
0&1&0&0&0&0&0&\omega^2&0&0&0&0&0&\omega^5&0&0&0&0\\
0&0&1&0&0&0&0&0&\omega^2&0&0&0&0&0&\omega^5&0&0&0\\
0&0&0&1&0&0&0&0&0&\omega^2&0&0&0&0&0&\omega^5&0&0\\
0&0&0&0&1&0&0&0&0&0&\omega^2&0&0&0&0&0&\omega^5&0\\
0&0&0&0&0&1&0&0&0&0&0&\omega^2&0&0&0&0&0&\omega^5
\end{bmatrix}.$$

In order to make M_2 upper triangular, it suffices to use row-reduction techniques on M_2. Let **D**, **E**, **F** represent (respectively) the first 6, next 6, and last 6 rows. To make M_2 upper-triangular, switch **E** and **F**. Then, replace **E** by **E** − **D** and **F** by **F** − **D**:

$$\begin{bmatrix}
1\,0\,0\,0\,0\,0 & \omega & 0 & 0 & 0 & 0 & 0 & \omega^3 & 0 & 0 & 0 & 0 & 0 \\
0\,1\,0\,0\,0\,0 & 0 & \omega & 0 & 0 & 0 & 0 & 0 & \omega^3 & 0 & 0 & 0 & 0 \\
0\,0\,1\,0\,0\,0 & 0 & 0 & \omega & 0 & 0 & 0 & 0 & 0 & \omega^3 & 0 & 0 & 0 \\
0\,0\,0\,1\,0\,0 & 0 & 0 & 0 & \omega & 0 & 0 & 0 & 0 & 0 & \omega^3 & 0 & 0 \\
0\,0\,0\,0\,1\,0 & 0 & 0 & 0 & 0 & \omega & 0 & 0 & 0 & 0 & 0 & \omega^3 & 0 \\
0\,0\,0\,0\,0\,1 & 0 & 0 & 0 & 0 & 0 & \omega & 0 & 0 & 0 & 0 & 0 & \omega^3 \\
0\,0\,0\,0\,0\,0 & \omega^2-\omega & 0 & 0 & 0 & 0 & 0 & \omega^5-\omega^3 & 0 & 0 & 0 & 0 & 0 \\
0\,0\,0\,0\,0\,0 & 0 & \omega^2-\omega & 0 & 0 & 0 & 0 & 0 & \omega^5-\omega^3 & 0 & 0 & 0 & 0 \\
0\,0\,0\,0\,0\,0 & 0 & 0 & \omega^2-\omega & 0 & 0 & 0 & 0 & 0 & \omega^5-\omega^3 & 0 & 0 & 0 \\
0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & \omega^2-\omega & 0 & 0 & 0 & 0 & 0 & \omega^5-\omega^3 & 0 & 0 \\
0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 0 & \omega^2-\omega & 0 & 0 & 0 & 0 & 0 & \omega^5-\omega^3 & 0 \\
0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 0 & 0 & \omega^2-\omega & 0 & 0 & 0 & 0 & 0 & \omega^5-\omega^3 \\
0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega^4-\omega^3 & 0 & 0 & 0 & 0 & 0 \\
0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega^4-\omega^3 & 0 & 0 & 0 & 0 \\
0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega^4-\omega^3 & 0 & 0 & 0 \\
0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega^4-\omega^3 & 0 & 0 \\
0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega^4-\omega^3 & 0 \\
0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega^4-\omega^3
\end{bmatrix}.$$

The determinant is $\omega^{24}(\omega-1)^{12}$, which is nonzero because $\omega \neq 0, 1$. Therefore, M_1 can be made to be upper triangular with a non-zero determinant.

Lastly, M_3 can be examined, which is as follows:

	$(1,1,1,0)$	$(1,1,0,1)$	$(1,0,1,1)$	$(0,1,1,1)$
$x_1x_2x_3$	1	0	0	0
$x_1x_2x_4$	0	1	0	0
$x_1x_3x_4$	0	0	1	0
$x_2x_3x_4$	0	0	0	1

.

Clearly, M_3 is upper triangular with a non-zero determinant. Thus, G can be expressed as an upper triangular matrix with the determinant being non-zero. Therefore, all 35 row vectors are linearly independent, so $k = 35$. □

Proposition 5.8. For BC(ℓ,j) with $q \gg 0$ ($l \geq j$), $k \geq \ell j + 1$.

Proof. First, some definitions are necessary. Note that in all of the above matrices in the rows which correspond to the monomials, for each degree of each monomial, a diagonal submatrix was formed. For example, examine the following:

	$(1,0,0)$	$(0,1,0)$	$(0,0,1)$	$(\omega,0,0)$	$(0,\omega,0)$	$(0,0,\omega)$
x_1	1	0	0	ω	0	0
x_2	0	1	0	0	ω	0
x_3	0	0	1	0	0	ω
x_1^2	1	0	0	ω^2	0	0
x_1^2	0	1	0	0	ω^2	0
x_3^2	0	0	1	0	0	ω^2

.

In order to allow for arbitrary ℓ to be included in the argument, consider the equivalent abbreviated matrix for this matrix (which is not contingent upon the choice of ℓ):

	(1)	(ω)
x_i	1	ω
x_i^2	1	ω^2

.

Each entry in this abbreviated matrix corresponds to an $\ell \times \ell$ block of the larger matrix of the form (entry element)(I_ℓ), where I_ℓ is the $\ell \times \ell$ identity matrix. If the determinant of this abbreviated matrix is shown to be non-zero for certain q, then the determinant of the larger matrix is non-zero.

Now we recall how a determinant is calculated. An entry in the first row of the matrix is multiplied by 1 or -1 and then multiplied by the determinant of its cofactor. Repeating this process until all cofactor determinants are calculated reveals that this process is analogous to a permutation. Thus, for a matrix A, $\det(A) = \sum(-1)^a A_\sigma$, where A_σ is a permutation and a is dependent upon whether A_σ is even or odd. Examine the matrix G that consists of the following $\ell j + 1$ rows of the matrix that generates the code:

$$G = \left[\begin{array}{c|ccccccc} & (1) & (\omega) & (\omega^2) & (\omega^3) & (\omega^4) & (\omega^5) & \dots \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & \dots \\ x_i & 1 & \omega & \omega^2 & \omega^3 & \omega^4 & (\omega^5) & \dots \\ x_i^2 & 1 & \omega^2 & \omega^4 & \omega^6 & \omega^8 & (\omega^{10}) & \dots \\ x_i^3 & 1 & \omega^3 & \omega^6 & \omega^9 & \omega^{12} & (\omega^{15}) & \dots \\ x_i^4 & 1 & \omega^4 & \omega^8 & \omega^{12} & \omega^{16} & (\omega^{20}) & \dots \\ x_i^5 & 1 & \omega^5 & \omega^{10} & \omega^{15} & \omega^{20} & (\omega^{25}) & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{array}\right].$$

Observe that for a permutation σ, for the mth row (the initial row, corresponding to the constant polynomial, is termed the 0th row), a column is uniquely identified (which is identified with a term of degree i_m). It follows then that the total degree of the term in the maximal minor of G for the permutation σ is $\sum_{m=0}^{j}(m)(i_m)$. The term of highest degree (treating ω as a variable) is sought. The main diagonal has the property that $(i_m) = m$, so $\sum_{m=0}^{j}(m)(i_m) = \sum_{m=0}^{j} m^2 = \frac{j(j+1)(2j+1)}{6}$. Thus, if the main diagonal provides the highest degree term in the maximal minor, it must be demonstrated that $\sum_{m=0}^{j}(m)(i_m) < \frac{j(j+1)(2j+1)}{6}$, where it is assumed that $s \neq i_s$ for at least one such s.

Proof by induction will be used. The base case is $j = 1$ (the $j = 0$ case is a 1×1 matrix, so the conditions of the inequality cannot be fulfilled). The matrix for the $j = 1$ case is

$$\begin{array}{c|cc} & (1) & (\omega) \\ \hline 1 & 1 & 1 \\ x_i & 1 & \omega \end{array}.$$

The main diagonal term has degree 1, which is collaborated by the summation formula: $\frac{1(1+1)(2+1)}{6} = \frac{6}{6} = 1$. The other diagonal (which is the only option for $\sum_{m=0}^{j}(m)(i_m)$ such that $s \neq i_s$ for at least one such s) has degree $\sum_{m=0}^{j}(m)(i_m) = (0)(1) + (1)(0) = 0$. Thus, the inequality holds for the base case.

Now, assume $\sum_{m=0}^{j}(m)(i_m) < \frac{j(j+1)(2j+1)}{6}$. We must show

$$\sum_{m=0}^{j+1}(m)(i_m) < \frac{(j+1)((j+1)+1)(2(j+1)+1)}{6},$$

with $s \neq i_s$ for at least one such s. Without the loss of generality, assume that s is the maximal index such that $i_s \neq s$. Thus, $i_t = s$ for some $t < s$, and $i_r = r$ for all $r > s$.

Then $\sum_{m=0}^{j+1}(m)(i_m)$ is as follows:

$$\sum_{m=0}^{j+1}(m)(i_m) = \sum_{\substack{m=0\\ m\neq t}}^{s}(m)(i_m) + (t)(i_t) + \sum_{m=s+1}^{j+1}(m)(i_m)$$

$$= \sum_{m=0}^{t-1}(m)(i_m) + \sum_{m=t+1}^{s}(m)(i_m) + (t)(s) + \sum_{m=s+1}^{j+1} m^2$$

$$= \sum_{m=0}^{t-1}(m)(i_m) + \sum_{m=t}^{s-1}(m+1)(i_{m+1}) + (t)(s) + \sum_{m=s+1}^{j+1} m^2$$

$$= \sum_{m=0}^{t-1}(m)(i_m) + \sum_{m=t}^{s-1}(m)(i_{m+1}) + \sum_{m=t}^{s-1} i_{m+1} + (t)(s) + \sum_{m=s+1}^{j+1} m^2.$$

Let $\{i_0, \ldots, i_{t-1}, i_{t+1}, \ldots, i_s\} = \{a_0, \ldots, a_{s-1}\} \subseteq [s-1]$ to combine the first two summations:

$$\sum_{m=0}^{j+1}(m)(i_m) = \sum_{m=0}^{s-1}(m)(a_m) + \sum_{m=t}^{s-1} i_{m+1} + (t)(s) + \sum_{m=s+1}^{j+1} m^2.$$

Use the induction hypothesis on the first term:

$$\sum_{m=0}^{j+1}(m)(i_m) < \frac{(s-1)(s)(2(s-1)+1)}{6} + \sum_{m=t}^{s-1} i_{m+1} + (t)(s) + \sum_{m=s+1}^{j+1} m^2.$$

Summation formulas can be used to simplify the expression:

$$\sum_{m=0}^{j+1}(m)(i_m) < \frac{(s-1)(s)(2(s-1)+1)}{6} + \frac{(s-1)s}{2} - \frac{(t-1)t}{2} + ts$$
$$+ \frac{(j+1)((j+1)+1)(2(j+1)+1)}{6} - \frac{(s(s+1)(2s+1)}{6}.$$

Thus, it suffices to show

$$\frac{(s-1)(s)(2(s-1)+1)}{6} + \frac{(s-1)s}{2} - \frac{(t-1)t}{2} + ts$$
$$+ \frac{(j+1)((j+1)+1)(2(j+1)+1)}{6} - \frac{(s(s+1)(2s+1)}{6}$$
$$< \frac{(j+1)((j+1)+1)(2(j+1)+1)}{6},$$

which is analogous to

$$\frac{(s-1)(s)(2(s-1)+1)}{6} + \frac{(s-1)s}{2} - \frac{(t-1)t}{2} + ts - \frac{(s(s+1)(2s+1)}{6} < 0$$

and to

$$\frac{(s(s+1)(2s+1)}{6} - \left(\frac{(s-1)(s)(2(s-1)+1)}{6} + \frac{(s-1)s}{2} - \frac{(t-1)t}{2} + ts\right) > 0.$$

Algebraic manipulation of the left-hand side of the last inequality follows:

$$\frac{(s(s+1)(2s+1)}{6} - \left(\frac{(s-1)(s)(2(s-1)+1)}{6} + \frac{(s-1)s}{2} - \frac{(t-1)t}{2} + ts\right)$$

$$= \frac{1}{6}\left((s(s+1)(2s+1) - ((s-1)s(2s-1) + 3(s-1)s - 3(t-1)t + 6ts))\right)$$
$$= \frac{1}{6}(2s^3 + 3s^2 + s - (2s^3 - 2s - 3t^2 + 3t + 6st)) = \frac{1}{6}(3s^2 + 3s + 3t^2 - 3t - 6st)$$
$$= \frac{1}{2}(s^2 + s + t^2 - t - 2st) = \frac{1}{2}((s-t)(s-t+1))$$

Since $t < s$, $s - t > 0$, and $s - t + 1 > 0$. Thus,

$$\frac{(s(s+1)(2s+1)}{6} - \left(\frac{(s-1)(s)(2(s-1)+1)}{6} + \frac{(s-1)s}{2} - \frac{(t-1)t}{2} + ts\right)$$
$$= \frac{1}{2}((s-t)(s-t+1)) > 0,$$

directly implying $\sum\limits_{m=0}^{j+1}(m)(i_m) < \frac{(j+1)((j+1)+1)(2(j+1)+1)}{6}$, completing the proof that the term of maximal degree in the determinant of the abbreviated form of G is obtained by multiplication along the main diagonal, which evaluates to $\frac{(j)(j+1)(2j+1)}{6}$. However, this expression is not the term of highest degree in G, since each term in the abbreviated form of G describes an $\ell \times \ell$ submatrix. Since this $\ell \times \ell$ submatrix is the identity matrix multiplied by a constant, the only way to keep a term in the determinant of G non-zero is by multiplying along the diagonal of the submatrix. Thus, for the submatrix corresponding to ω^i, instead of contributing a term of degree i to a term in the determinant using the submatrix, the submatrix contributes a term of degree $(\ell)(i)$ to the term. Thus, the term of highest degree in the determinant is $\ell\frac{(j)(j+1)(2j+1)}{6}$. It follows from the existence of this term that the determinant is also not zero. By an extension of the Fundamental Theorem of Algebra to finite fields, there can be as many as $\ell\frac{(j)(j+1)(2j+1)}{6}$ zeros in the expression describing the determinant of G. Let $q > \ell\frac{(j)(j+1)(2j+1)}{6}$. Thus, there are more elements in the finite field than zeros in the determinant. Let ω be an element of the finite field which is not a zero of the determinant, directly implying that the determinant of G is not zero. Thus, all of the row vectors of G are linearly independent. Therefore, as long as q is sufficiently large, $k \geq \ell(j) + 1$. □

6. Binary Simplicial Complex Codes

Error-correcting codes are typically binary codes because of the prominence of the binary system in computer operations. Thus, in this section, only binary codes ($q = 2$) are examined. These codes derive from different classes of simplicial complexes. Starting with a proposition concerning the dimension for any binary simplicial complex code, the section then moves onto corollaries concerning specific cases of these codes. An examination of polygons and then skeletons (defined later) follows. The section ends in a significant result regarding minimum distance for skeletal codes.

6.1. **Dimension.**

An important part of the examination of the dimension of binary simplicial complex codes is the concept of a face vector, f_i, in the simplicial complex.

Definition 6.1. A *face vector*, f_i, represents the number of faces of dimension $i-1$ for a given simplicial complex. That is, $f_i = |\{\sigma \in \Delta : |\sigma| = i\}|$ for $0 \leq i$. The trivial case, f_0, is defined as $f_0 = 1$.

Proposition 6.2. For binary simplicial complex codes, $k = \sum\limits_{i=0}^{j} f_i$.

Proof. This proposition is analogous to the claim that the evaluation of the Hilbert function upon a Stanley-Reisner ring based upon a simplicial complex ($\mathbb{F}(\Delta)$; the ring is assumed to be over $\mathbb{F}_2^\ell$ for some ℓ) is $H(SR(\Delta), m) = f_m$ (the dimension is merely the sum for these Hilbert function evaluations from 0 to j, which constitutes the upper bound previously determined). The main reason that these formulas hold is that in a polynomial ring over $\mathbb{F}_2$, x^n and x (for n a positive integer) evaluate the same over $\mathbb{F}_2$. Examine the formation of an ideal based upon a simplicial complex: the ideal is based upon faces that are not a part of the simplicial complex. Thus, if a point i and a line segment jk are not a part of the simplicial complex, then x_i and x_jx_k (respectively) are contained in the ideal, meaning that the polynomials x_i and x_jx_k do not contribute unique row vectors in the matrix describing the code (these row vectors would be the zero vector since the evaluation of the polynomial at each point that represents a column would be zero since each such point is in $P(\mathcal{A}_\Delta)$). The argument holds true for cases for higher dimensional faces. Thus, it readily becomes clear that the maximum number of linearly independent row vectors in this case directly corresponds to how many faces of each dimension there are in the simplicial complex. Furthermore, this maximum number is the exact number because analogously to the previous proofs dealing with the Boolean arrangement, points in the ideal can be carefully chosen to create an upper triangular matrix. Therefore, the total number of face vectors of dimension m that are in the simplicial complex is equal to the evaluation of the Hilbert function at m. Thus, the dimension of the code is the summation of the Hilbert function from $m = 0$ to $m = j$, which is the summation of the face vectors from f_0 through f_j: $k = \sum\limits_{i=0}^{j} f_i$. □

Corollary 6.3. *Let M denote the dimension of the minimum dimensional non-face in a simplicial complex Δ. Then, if $j \leq M - 1$, the dimension of $C(\mathcal{A}_\Delta, j)$ is $k = \sum\limits_{m=0}^{j} \binom{\ell}{m}$.*

Proof. First note that in $\mathbb{F}_2[x_1, \dots, x_\ell]$, for n a positive integer, the polynomials x^n and x evaluate the same on 0 and 1. Thus, in the matrix construction of the code, squarefree monomials need only to be considered (any further ones will result in a row vector being identical to the row vector corresponding to the squarefree monomial). Thus, there are $\binom{\ell}{i}$ monomials with unique evaluation over $\mathbb{F}_2$ of degree i. Since $j \leq M - 1$, all of these monomials with unique evaluation are in the Stanley-Reisner Ring (the faces corresponding to these monomials are in the simplicial complex). For $I = \{i_1, \dots, i_t\}$, let $e_I = e_{i_1,\dots,i_t}$ correspond to the point in $\mathbb{F}_2^\ell$ with 1s in the $i_1, \dots, i_t$ positions and 0s elsewhere. For example, if $I = \{0, 2, 3\}$, then $e_I = e_{0,2,3} = (1, 0, 1, 1, 0, \dots)$. Let X_I be defined as $x_1 \cdots x_t$ (note that the origin, $(0, \dots, 0)$ and the constant polynomial are defined as $e_{\{\}}$ and $X_{\{\}}$, respectively). Thus, for $I \neq \{\}$, $X_I(e_J) = \left\{ \begin{array}{ll} 1 & I \subseteq J \\ 0 & I \nsubseteq J \end{array} \right.$. Now, construct G such that the polynomials are normally ordered (from lowest to highest degree, in lexicographic order within the same degree). Order the columns such that if X_I is in the jth row, then e_I is in the jth column (note that $A = \{\}$):

$$G = \begin{array}{c|ccccc}
 & e_A & e_B & e_C & e_D \dots & \\
\hline
X_A & 1 & 1 & 1 & 1 & \dots \\
X_B & 0 & 1 & * & \dots & \dots \\
X_C & 0 & 0 & 1 & * & \dots \\
X_D & 0 & 0 & 0 & 1 & \dots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots
\end{array}.$$

Thus, with this careful ordering, an upper triangular matrix is formed, thereby implying that each of the row vectors are linearly independent. Since this method of ordering covers all monomials in the Stanley-Reisner Ring, all row vectors in G are linearly independent, so the dimension of the code is equal to the number of polynomials up to degree j: $k = \sum\limits_{m=0}^{j} \binom{\ell}{m}$. □

Corollary 6.4. *If the subspace arrangement is the Boolean arrangement ($I = < x_1 \cdots x_\ell >$), then the length is $n = 2^\ell - 1$ and the dimension is $k = \sum\limits_{m=0}^{j} \binom{\ell}{m}$.*

Proof. The formula for the length of the code is trivial: there are a total of 2^ℓ points in $\mathbb{F}_2^\ell$, and the only point not in the ideal is the point with each component being 1, $(1, 1, 1, \ldots, 1)$. As for dimension, the previous result can be used. Since the Boolean arrangement is used, $M = \ell$. Clearly, $j < \ell - 1$, so $k = \sum\limits_{m=0}^{j} \binom{\ell}{m}$. □

Remark 6.5. While a convenient formula for dimension has been fairly easily determined, the case for minimum distance is much more difficult. In order to obtain some results on minimum distance, specific cases (such as letting $j = 1$) are examined in later portions of this paper.

6.2. **Polygons.** A specific class of simplicial complexes are now examined; namely, the simplicial complex is a polygon.

Definition 6.6. Let P_m be the simplicial complex that is an m-gon. Thus, $P_m = \{\{1\}, \{2\}, \ldots, \{m\}, \{1, 2\}, \{2, 3\}, \ldots, \{m-1, m\}, \{1, m\}\}$.

Proposition 6.7. The characteristic polynomial for a code based upon P_m is $\chi(\mathcal{A}_{P_m}, t) = t^m - mt^2 + mt - 1$.

Proof. Recall that $L(\mathcal{A}_{P_m})$ is a lattice that refers to intersections of all subspaces of the coordinate hyperplanes corresponding to the simplicial complex Δ. The Hasse Diagram for $L(\mathcal{A}_{P_m})$ is depicted in Figure 8. Clearly, at the bottom of the hierarchy is P_m itself, which corresponds to the vector space $\mathbb{F}_q^m$ of dimension m. The level of atoms consists of the edges in P_m (which has dimension 2), whose intersections from the collection of vertices at the third level (which has dimension 1). At the top of the diagram is the empty set (with dimension 0), which is the only element in the intersection of two vertices. The next step in determining the characteristic polynomial is evaluating the Möbius function upon the components of the diagram (in Figure 8, the evaluation of the Möbius function on each component is in red). By definition, the Möbius function evaluates to 1 on P_m. The atoms of $L(\mathcal{A}_{P_m})$ are then each -1 (which is the negative of the evaluation of the Möbius function on P_m). Each edge has two vertices ordered below it, along with P_m, so the Möbius function evaluates to $-(2(-1)+1) = 1$. For the empty set, there are m vertices (with value 1) and m edges (with value -1), along with P_m (with value 1), so the Möbius function evaluates to $-((m)(1) + (m)(-1) + 1) = -1$. Thus, the characteristic polynomial can be formed by combing the dimension of each level of the diagram along with evaluation of the Möbius function: $\chi(\mathcal{A}_{P_m}, t) = t^m - mt^2 + mt - 1$. □

Corollary 6.8. *The length of $C(\mathcal{A}_{P_m}, j)$ is $n = 2m + 1$.*

Proof. The formula for length, when $q = 2$ is

$$\begin{aligned} n = 2^m - \chi(\mathcal{A}_{P_m}, 2) &= 2^m - (2^m - 2^2 m + 2m - 1) \\ &= 2^m - 2^m + 4m - 2m + 1 = 2m + 1. \end{aligned}$$

□

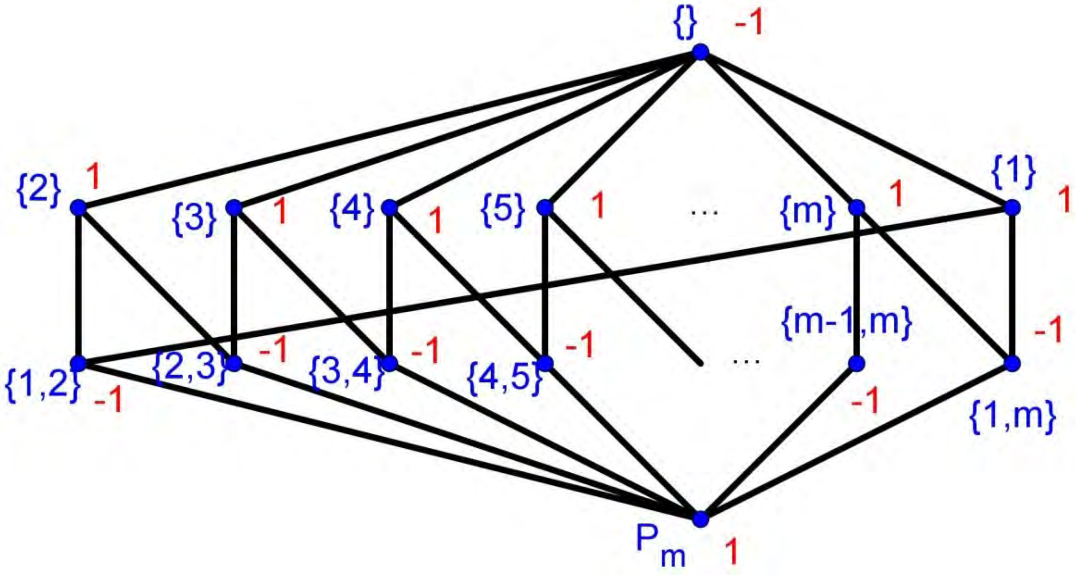

Figure 8. Hasse diagram of $L(\mathcal{A}_{P_m})$

Proposition 6.9. The dimension of $C(\mathcal{A}_{P_m}, 1)$ is $m+1$, and the minimum distance is 3.

Proof. Recall that $k = \sum_{i=0}^{j} f_i$. Thus, $k = \sum_{i=0}^{1} f_i = f_0 + f_1 = 1 + m = m + 1$.

In $\mathbb{F}_2^m$, let $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$, where the 1 is in the ith position, where i ranges from 0 to $m-1$ (so the initial position is the 0th position). For consistency, the origin is defined as $e_{\{\}}$. Let $e_{(i)(j)}$ then be defined as $e_i + e_j$. For P_m, $\mathcal{A}_{P_m}$ corresponds to a collection of m coordinate planes in $\mathbb{F}_2^m$ which are spanned by adjacent coordinate axis (that is, the planes spanned by x_0 and x_1; x_1 and x_2; $\ldots$; x_{m-2} and x_{m-1}; and x_1 and x_{m-1}). The points in the subspace arrangement are thus

$$P(\mathcal{A}_{P_m}) = \bigcup_{X \in \mathcal{A}_{P_m}} X_i = \{e_{\{\}}\} \cup \{e_i\} \cup \{e_{(i)(i+1)}\}$$

for $0 \leq i \leq m-1$. Note that $i+1$ is addition modulo m. For simplicity, let the polynomials in the Stanley-Reisner ring up to degree $j = 1$ be labeled as 1, x_0, x_1, $\ldots$, x_{m-2}, and x_{m-1}. For $p \in \mathcal{A}_{P_m}$,

$$x_i(p) = \begin{cases} 1 & \text{if } p = e_i, e_{(i-1)(i)}, e_{(i)(i+1)} \\ 0 & \text{else} \end{cases}.$$

Thus, since each non-constant polynomial evaluates to 1 for only 3 points in $P(\mathcal{A}_{P_m})$, the minimum Hamming weight among the codewords that form the basis of the code is 3. Observe that when two codewords (corresponding to polynomials of degree 1) of the basis are added, two possibilities exist: $x_i + x_j$ ($j > i+1 \bmod m$) and $x_i + x_{i+1}$. For $x_i + x_j$, $\{e_i, e_{(i-1)(i)}, e_{(i)(i+1)}\} \cap \{e_j, e_{(j-1)(j)}, e_{(j)(j+1)}\} = \{\}$ gives the location of overlapping ones, so the Hamming weight is $3+3=6$. For $x_i + x_{i+1}$, $\{e_i, e_{(i-1)(i)}, e_{(i)(i+1)}\} \cap \{e_{i-1}, e_{(i-2)(i-1)}, e_{(i-1)(i)}\} = \{e_{(i-1)(i)}\}$ gives the location of overlapping ones, so the Hamming weight is $3+3-(2)(1) = 4$. Both cases yield a Hamming weight greater than 3. Additionally, if 3 or more codewords (corresponding to polynomials of degree 1) are added $(x_i + x_j + x_k + \cdots)$, then $e_i, e_j, e_k, \dots$ are not in the intersection, so there are at least 3 non-overlapping ones.

Now, care must be exercised in adding the codeword $(1, 1, \dots, 1)$. When this codeword is added to any other codeword with Hamming weight w, the Hamming weight becomes $2m + 1 - w$. Note that when $m = 3$, when two codewords corresponding to polynomials of degree 1 are added, the case is never $x_i + x_j$, making the Hamming weight 4. Adding any two codewords (corresponding to polynomials of degree 1) with the constant polynomial results in $2m + 1 - 4 = 7 - 4 = 3$. For $m \geq 4$, $2m + 1 - 6 \geq (2)(4) - 5 = 3$ (the x_i+x_j case) and $2m + 1 - 4 \geq (2)(4) - 4 = 5$ (the $x_i + x_{i+1}$ case), so the minimum Hamming weight is still 3 after any two codewords (corresponding to polynomials of degree 1) are added with the codeword $(1, 1, \dots, 1)$. When 3 or more codewords (corresponding to polynomials of degree 1) are added, two cases arise: either 3 or more points evaluate to 1 on multiple polynomials, or fewer than 3 points have this overlap. If the case is the former, then adding the sum with the codeword $(1, 1, \dots, 1)$ yields a codeword with Hamming weight of at least 3 (since the overlapped points had ones that added to 0 since only two codewords $[x_i, x_{i+1}]$ can overlap at $e_{(i)(i+1)}$). If only 2 points overlap, then it is the case $x_{i-1} + x_i + x_{i+1} + \dots$, with $x_{i+2} \neq x_{i-1}$ (or else 3 points would overlap). Note then that x_{i+2} cannot be in the sum of codewords (since it would evaluate to one on the $e_{(i+1)(i+2)}$ with x_{i+1}, creating an additional overlapping point). Thus, the component of the sum corresponding to e_{i+1} is 0. When the codewords are added, the components corresponding to the overlapping points become zero (since only two codewords overlap at those points), in addition to the component corresponding to e_{i+1}. Thus, when $(1, 1, \dots, 1)$ is added, at least 3 components of the vector are 1, so the Hamming weight is at least 3. In the case with 1 point overlapping, the sum must be $x_i + x_{i+1} + x_j + \dots$ with x_{i-1} and x_{i+2} not in the sum. Thus, the components corresponding to e_{i-1} and e_{i+1} are 0, along with $e_{(i)(i+1)}$ (because of the overlap). Hence when the sum is added to $(1, 1, \dots, 1)$, the Hamming weight is at least 3. If no points are overlapping, then the sum of codewords is $x_i + x_j + x_k + \dots$ with x_{i+1}, x_{j+1}, and x_{k+1} all not in the sum (clearly, $x_{i+1} \neq x_{j+1} \neq x_{k+1}$). Thus, in the sum of these codewords, the components corresponding to e_{i+1}, e_{j+1}, and e_{k+1} are 0. Therefore, when the sum is added to $(1, 1, \dots, 1)$, the Hamming weight is at least 3. Thus, no matter how many codewords are added, the Hamming weight remains at least 3. Therefore, $d = 3$. □

Proposition 6.10. The dimension of $C(\mathcal{A}_{P_m}, 2)$ is $2m + 1$, and the minimum distance is 1.

Proof. Recall that $k = \sum_{i=0}^{j} f_i$. Thus, $k = \sum_{i=0}^{2} f_i = f_0 + f_1 + f_2 = 1 + m + m = 2m + 1$. By Proposition 6.2, $n = 2m + 1$. Thus, the generating matrix for the code can be made into an upper triangular $(2m+1) \times (2m+1)$ matrix with 1s along the diagonal (since the row vectors are linearly independent). Thus the last row of the vector must have have all nonzero entries with the exception of the last entry (that is, $(0, 0, \dots, 0, 1)$), giving it a Hamming weight of 1. Therefore, the minimum Hamming weight (and the minimum distance) of the codes is 1. □

Remark 6.11. The general characterization of binary codes based upon a simplicial complex which is an m-gon is as follows. The case with $j = 0$ is trivial and not included. Work above indicates that if $j = 1$, a $[2m+1, m+1, 3]_2$ code results, whereas if $j \geq 2$, a $[2m+1, 2m+1, 1]_2$ code results. Note that if $j > 3$, no more linearly independent row vectors are created (since there are no points in the ideal with three components that are 1 since only edges are vertices are present in the simplicial complex).

6.3. **Skeletal Codes.** Computing the minimum distance for a general simplicial complex evaluation code at this time seems difficult. In this section we focus on one case where the minimum distance can be calculated. The proof is elementary but slightly involved. To begin we define the codes and examine a few examples.

Definition 6.12. For $0 \leq h \leq \ell$ an *h-skeleton* is a simplicial complex, denoted by $\Delta(\ell, h)$, on ℓ vertices consisting of all possible $h-1$ to 0-dimensional faces.

Definition 6.13. A binary *h-skeleton code* is the binary evaluation code $C(\mathcal{A}_{\Delta(\ell,h)}, j)_2$ of the associated coordinate arrangement to the h-skeleton and is denoted $K(\ell, h, j)$.

The length of these codes is obtained by counting the number of points that have at most h non-zero entries. Hence, the length of $K(\ell, h, j)$ is

$$n = \sum_{i=0}^{h} \binom{\ell}{i}. \tag{1}$$

The dimension can be found as a specialization of Proposition 6.2.

Corollary 6.14. *For* $0 \leq j \leq h \leq \ell$

$$\dim(K(\ell, h, j)) = \sum_{i=0}^{j} \binom{\ell}{i}.$$

For the remainder of this section we study the minimum distance of the codes $K(\ell, h, j)$. In order to obtain more information about $K(\ell, h, j)$ we need to examine and carefully construct a convenient generating matrix. To do this we need a little notation. Let $\sigma = \{i_1, \ldots, i_r\} \subseteq [\ell]$ and let $x_\sigma = x_{i_1} x_{i_2} \cdots x_{i_r}$. With this notation, the Stanley-Reisner ideal of $\Delta(\ell, h)$ is

$$I_{\Delta(\ell,h)} = (x_\sigma : |\sigma| = h+1).$$

To denote points in the arrangement $\mathcal{A}_{\Delta(\ell,h)}$ we let $\{e_1 \ldots, e_\ell\}$ be the standard basis for $V = \mathbb{F}_2^\ell$ (that is, e_i has all components 0 except a 1 in the i-th coordinate). Now for $\tau = \{i_1, \ldots, i_s\} \subseteq [\ell]$ let $e_\tau = \sum\limits_{k=1}^{s} e_{i_k}$. Then the set of all points in the arrangement $\mathcal{A}_{\Delta(\ell,h)}$ is

$$\left[\bigcup_{X \in \mathcal{A}_{\Delta(\ell,h)}} X \right] = \{e_\tau : 1 \leq |\tau| \leq h\}.$$

Now we will construct the blocks of the generating matrix for $K(\ell, h, j)$. Let B_{rs} be the matrix defined as

$$B_{rs} = (x_\sigma(e_\tau)) \tag{2}$$

where $|\sigma| = r$, $|\tau| = s$, and the rows and columns are ordered lexicographically. Then a generating matrix $G(\ell, h, j)$ of $K(\ell, h, j)$ constructed block-wise is

$$G(\ell, h, j) = (B_{rs})_{\substack{0 \leq r \leq j \\ 0 \leq s \leq h}}.$$

We can now denote column and row blocks of the generating matrix.

Definition 6.15. Let $CB_t = \{B_{rt} : 0 \leq r \leq h\}$ be the union of the blocks of columns in the matrix of the code with t ones in each point. Let $RB_t = \{B_{tr} : 0 \leq r \leq j\}$ be the union of the blocks of rows in the matrix of the code with t ones in each point.

This notation for the generating matrix streamlines the computation of minimum distance. We begin by presenting an upper bound for the minimum distance.

Lemma 6.16. *For $0 \leq j \leq h \leq \ell$ the minimum distance of $K(\ell, h, j)$ satisfies*

$$d \leq \sum_{i=0}^{h-j} \binom{\ell - j}{i}.$$

Proof. In the generating matrix $G(\ell, h, j)$ the rows in the last row block RB_j have the smallest weight. The smallest t such that B_{jt} has no zero entries is when $t = j$. The Hamming weight of any row of B_{jt} for $j \leq t \leq h$ is $\binom{\ell-j}{t-j}$. Hence, the Hamming weight of an entire row in RB_j is

$$\sum_{i=j}^{h} \binom{\ell - j}{i - j}.$$

□

Remark 6.17. If $j = h$ then $K(\ell, h, j)$ is a maximum distance separable (MDS) code but the minimum distance is 1 because the upper bound here is 1. An MDS code is one that satisfies the Singleton bound of $k \leq n - d + 1$.

If $j = 1$ then the minimum distance is bounded by $d \leq \sum\limits_{i=0}^{h-1} \binom{\ell-1}{i}$. It seems difficult to determine the actual minimum distance. The main result of this paper (Theorem 6.22) is that this upper bound is exactly the minimum distance for the case $j = 1$. First, we obtain a formula for the Hamming weight of adding s rows of the generating matrix. In order to do develop this formula, we need a little more notation. Suppose $x_{i_1}, \ldots, x_{i_s}$ are the degree one monomials that correspond to the s rows we are to sum in B_{1a}. Let P_a be the set of all points in $\mathbb{F}_2^\ell$ that have exactly a nonzero entries. Note that P_a corresponds to the columns of B_{1a}.

Definition 6.18. For $1 \leq t \leq s$, let $X_r := \{p \in P_a : x_{i_r}(p) = 1\}$. Let $\mathcal{L}_t^{a,s}$ be the set of all the sets of points that evaluate to 1 on at least t degree one monomials, so

$$\mathcal{L}_t^{a,s} = \{X_{k_1} \cap \cdots \cap X_{k_t} : \{k_1, \ldots, k_t\} \subseteq \{i_1, \ldots, i_s\}\}.$$

If we wanted to calculate the size of the union of the sets $X_{i_1} \cup \cdots \cup X_{i_s}$, then we could use a standard inclusion-exclusion formula

$$|X_{i_1} \cup \cdots \cup X_{i_s}| = \sum_{t=1}^{s} (-1)^{t+1} \sum_{Y \in \mathcal{L}_t^{a,s}} |Y|.$$

However, we want to calculate the Hamming weight of the sum of these row vectors of which not all points will sum to 1. To do this we will create a generalized inclusion-exclusion formula.

Lemma 6.19. *The Hamming weight of adding s row vectors of B_{1a} of the code $C(\mathcal{A}_{\Delta(l,h)}, 1)$ is*

$$\sum_{t=1}^{s} (-2)^{t-1} \sum_{Y \in \mathcal{L}_t^{a,s}} |Y|.$$

Proof. We prove this by induction. The critical idea here is that if a point p is contained in exactly t sets $X_{k_1}, \ldots, X_{k_t}$ and not in any others, then the entry corresponding to this point in the sum will be 0 if t is even and 1 if t is odd. Let c_t be the coefficient that will be multiplied to the point p that is contained in exactly t sets $X_{k_1}, \ldots, X_{k_t}$ in the sum (note that points are the objects being summed here because the Ys consist of points). In the case when $t = 1$, we want to count all the points that are in exactly 1 set. We therefore sum the entirety of the sets of just one intersection: $\sum\limits_{r=1}^{s} |X_{i_r}|$. Thus, the coefficient is $c_1 = 1$ for the $t = 1$ term. However, if $t > 1$, the point p has already been counted in lower terms because it is also a subset of all possible intersections of these t sets:

$$X_{k_1}, \dots, X_{k_t}, X_{k_1} \cap X_{k_2}, \dots, X_{k_{t-1}} \cap X_{k_t}, \dots, X_{k_1} \cap \cdots \cap X_{k_{t-1}}, \dots, X_{k_2} \cap \cdots X_{k_t}.$$

Because we want the coefficient for t odd to be 1 and for t even to be zero, we now have that

$$\sum_{r=1}^{t} \binom{t}{r} c_r = \begin{cases} 1 & t \text{ odd} \\ 0 & t \text{ even} \end{cases}.$$

One method to do this is to set

$$\sum_{r=1}^{t} \binom{t}{r} c_r = \frac{(-1)^t - 1}{-2}.$$

Now we prove by induction on t that $c_t = (-2)^{t-1}$. The base is already provided above.

By construction, $c_{t+1} = \frac{(-1)^{t+1}-1}{-2} - \sum\limits_{r=1}^{t} \binom{t+1}{r} c_r$. Then by the induction hypothesis,

$$c_{t+1} = \frac{(-1)^{t+1} - 1}{-2} - \sum_{r=1}^{t} \binom{t+1}{r} (-2)^{r-1}.$$

Using the binomial expansion formula, we see that that

$$(-1)^{t+1} = (-2+1)^{t+1} = \sum_{i=0}^{t+1} \binom{t+1}{i} (-2)^i 1^{t+1-i} = \sum_{i=0}^{t+1} \binom{t+1}{i} (-2)^i$$

$$= 1 + \sum_{i=1}^{t+1} \binom{t+1}{i} (-2)^i = 1 + (-2) \sum_{i=1}^{t+1} \binom{t+1}{i} (-2)^{i-1}.$$

Hence,

$$\frac{(-1)^{t+1} - 1}{-2} = \sum_{i=1}^{t+1} \binom{t+1}{i} (-2)^{i-1} = (-2)^t + \sum_{i=1}^{t} \binom{t+1}{i} (-2)^{i-1}.$$

Now add the sum to both sides of this equation to obtain

$$(-2)^t = \frac{(-1)^{t+1} - 1}{-2} - \sum_{i=1}^{t} \binom{t+1}{i} (-2)^{i-1} = c_t.$$

□

Lemma 6.19 gives a nice method to compute the Hamming weight of the sum of s row vectors.

Lemma 6.20. *The Hamming weight of adding s vectors in RB_1 is*

$$\sum_{a=1}^{h} \sum_{t=1}^{s} (-2)^{t-1} \binom{s}{t} \binom{\ell - t}{a - t}.$$

Proof. Note that for any $Y \in \mathcal{L}_t^{a,s}$, the size is $|Y| = \binom{\ell-t}{a-t}$ since t of the nonzero entries must match up with the t monomials (so there are $a-t$ ones left to chose from the remaining $\ell - t$ components of the point). Since we can choose any t subsets of the monomials, we have

$$|\mathcal{L}_t^{a,s}| = \binom{s}{t}.$$

Then the formula for the Hamming weight is given by applying Lemma 6.19 and summing over all possible column blocks CB_a where $1 \leq a \leq h$. □

Next we prove a technical lemma that will be used in the proof of the main theorem.

Lemma 6.21. *If* $g_i^s = \sum_{t=1}^{i} \binom{s-t}{i-t}\binom{s}{t}(-2)^{t-1}$, *then*

$$g_i^s = \begin{cases} \binom{s}{i} & 2|i \\ 0 & 2\nmid i \end{cases}.$$

Proof. We prove this in two cases. Case 1 is when $i = 2m$. Then

$$g_{2m}^s = \sum_{t=1}^{2m} \binom{s-t}{2m-t}\binom{s}{t}(-2)^{t-1} = \sum_{t=1}^{2m} \frac{(s-t)!s!}{(2m-t)!(s-2m)!t!(s-t)!}(-2)^{t-1}$$

$$= \sum_{t=1}^{2m} \frac{s!}{(2m-t)!(s-2m)!t!}\frac{(2m)!}{(2m)!}(-2)^{t-1} = \frac{s!}{(s-2m)!(2m)!}\sum_{t=1}^{2m}\frac{(2m)!}{(2m-t)!t!}(-2)^{t-1}$$

$$= \binom{s}{2m}\left(-\frac{1}{2}\right)\left[\sum_{t=1}^{2m}\binom{2m}{t}(-2)^t\right] = \binom{s}{2m}\left(-\frac{1}{2}\right)\left[\sum_{t=0}^{2m}\binom{2m}{t}(-2)^t - 1\right]$$

$$= \binom{s}{2m}\left(-\frac{1}{2}\right)\left[\sum_{t=0}^{2m}\binom{2m}{t}(-2)^t(1)^{2m-t} - 1\right] = \binom{s}{2m}\left(-\frac{1}{2}\right)\left[(1-2)^{2m} - 1\right]$$

$$= \binom{s}{2m}\left(-\frac{1}{2}\right)(1-1) = 0.$$

Case 2 is when $i = 2m+1$. Then

$$g_{2m+1}^s = \sum_{t=1}^{2m+1}\binom{s-t}{2m+1-t}\binom{s}{t}(-2)^{t-1}$$

$$= \sum_{t=1}^{2m+1}\frac{(s-t)!s!}{(2m+1-t)!(s-2m-1)!t!(s-t)!}(-2)^{t-1}$$

$$= \sum_{t=1}^{2m+1}\frac{s!}{(2m+1-t)!(s-2m-1)!t!}\frac{(2m+1)!}{(2m+1)!}(-2)^{t-1}$$

$$= \frac{s!}{(s-2m-1)!(2m+1)!}\sum_{t=1}^{2m+1}\frac{(2m+1)!}{(2m+1-t)!t!}(-2)^{t-1}$$

$$= \binom{s}{2m+1}\left(-\frac{1}{2}\right)\left[\sum_{t=1}^{2m+1}\binom{2m+1}{t}(-2)^t\right]$$

$$= \binom{s}{2m+1}\left(-\frac{1}{2}\right)\left[\sum_{t=0}^{2m+1}\binom{2m+1}{t}(-2)^t - 1\right]$$

$$
\begin{aligned}
&= \binom{s}{2m+1}\left(-\frac{1}{2}\right)\left[\sum_{t=0}^{2m+1}\binom{2m+1}{t}(-2)^t(1)^{2m+1-t} - 1\right] \\
&= \binom{s}{2m+1}\left(-\frac{1}{2}\right)\left[(1-2)^{2m+1} - 1\right] = \binom{s}{2m}\left(-\frac{1}{2}\right)(-1-1) \\
&= \binom{s}{2m}\left(-\frac{1}{2}\right)(-2) = \binom{s}{2m}.
\end{aligned}
$$

□

Notice that the formula in Lemma 6.20 for the case $s = 1$ is exactly the computation made in Lemma 6.16. In order to show that this value for $s = 1$ is exactly the minimum distance, it is enough to show that the sum for $s > 1$ is greater than that for $s = 1$, since $q = 2$. Now we can state and prove the main theorem of this paper.

Theorem 6.22. *The minimum distance of the codes $C(\mathcal{A}_{\Delta(\ell,h)}, 1) = K(\ell, h, 1)$ is*

$$\sum_{a=1}^{h}\binom{\ell-1}{a-1}.$$

Proof. We need to show that the Hamming weight of adding s rows given in Lemma 6.20 is always larger than the Hamming weight of one row given in Lemma 6.16:

$$\sum_{a=1}^{h}\sum_{t=1}^{s}(-2)^{t-1}\binom{s}{t}\binom{\ell-t}{a-t} \geq \sum_{a=1}^{h}\binom{\ell-1}{a-1}$$

This is equivalent to showing

$$\sum_{a=1}^{h}\left[\left(\sum_{t=1}^{s}(-2)^{t-1}\binom{s}{t}\binom{\ell-t}{a-t}\right) - \binom{\ell-1}{a-1}\right] \geq 0. \tag{3}$$

Now we use Pascal's formula to allow for the exchange of terms of 3. We examine the term

$$\binom{\ell-t}{a-t} = \binom{\ell-t-1}{a-t} + \binom{\ell-t-1}{a-t-1}$$

$$= \left(\binom{\ell-t-2}{a-t} + \binom{\ell-t-2}{a-t-1}\right) + \left(\binom{\ell-t-2}{a-t-1} + \binom{\ell-t-3}{a-t-2}\right) = \dots$$

Since, whenever Pascal's formula is used, each binomial coefficient is broken down into two binomial coefficients, the process is analogous to Pascal's triangle: the top number, $l - t - x$, corresponds to the xth row, and the bottom number, $a-t-x$, corresponds to the xth column. Thus, there are $\binom{s-t}{i-t}$ occurrences of each $\binom{\ell-s}{a-i}$ for each t. Therefore, $\binom{\ell-t}{a-t} = \sum\limits_{i=1}^{s}\binom{\ell-s}{a-i}\binom{s-t}{i-t}$, so the inequality we are trying to prove is now

$$\sum_{a=1}^{h}\left[\left(\sum_{t=1}^{s}(-2)^{t-1}\binom{s}{t}\sum_{i=1}^{s}\binom{\ell-s}{a-i}\binom{s-t}{i-t}\right) - \sum_{i=1}^{s}\binom{\ell-s}{a-i}\binom{s-1}{i-1}\right] \geq 0. \tag{4}$$

Now focusing on the $\binom{\ell-s}{a-s}$ terms, 4 becomes

$$\sum_{a=1}^{h}\left[\sum_{i=1}^{s}\left(\left(\sum_{t=1}^{s}(-2)^{t-1}\binom{s}{t}\binom{s-t}{i-t}\right) - \binom{s-1}{i-1}\right)\binom{\ell-s}{a-i}\right] \geq 0 \tag{5}$$

Notice that the third sum is only nonzero when $t \leq i$ and that the $\binom{s-1}{i-1}$ term only affects the $t = 1$ term of the third sum. Hence, we can rewrite 5 as

$$\sum_{a=1}^{h}\left[\sum_{i=1}^{s}\left((s-1)\binom{s-1}{i-1}+\sum_{t=2}^{i}(-2)^{t-1}\binom{s}{t}\binom{s-t}{i-t}\right)\binom{\ell-s}{a-i}\right] \geq 0 \tag{6}$$

Let d_i^s be the coefficient of $\binom{\ell-s}{a-i}$ in 6:

$$d_i^s = (s-1)\binom{s-1}{i-1}+\sum_{t=2}^{i}(-2)^{t-1}\binom{s}{t}\binom{s-t}{i-t}.$$

Recall the numbers g_i^s from Lemma 6.21:

$$g_i^s = \sum_{t=1}^{i}\binom{s-t}{i-t}\binom{s}{t}(-2)^{t-1}.$$

Then

$$\begin{aligned} g_i^s - d_i^s &= \sum_{t=1}^{i}\binom{s-t}{i-t}\binom{s}{t}(-2)^{t-1} - (s-1)\binom{s-1}{i-1} - \sum_{t=2}^{i}(-2)^{t-1}\binom{s}{t}\binom{s-t}{i-t} \\ &= \binom{s-1}{i-1}. \end{aligned}$$

Thus,

$$d_{2m+1}^s = g_{2m+1}^s - \binom{s-1}{2m}, \tag{7}$$

which, by Lemma 6.21, gives that 7 becomes

$$d_{2m+1}^s = -\binom{s-1}{2m}.$$

Using Lemma 7 again, we get

$$d_{2m}^s = g_{2m}^s - \binom{s-1}{2m-1} = \binom{s}{2m} - \binom{s-1}{2m-1} = \binom{s-1}{2m}.$$

Hence,

$$d_{2m+1}^s = -d_{2m}^s. \tag{8}$$

The main inequality we are trying to prove, 6, can now be written as

$$\sum_{a=1}^{h}\left[\sum_{i=1}^{s} d_i^s\binom{\ell-s}{a-i}\right] \geq 0. \tag{9}$$

Assume $s = 2m$ is even and expand the left hand side of 9 via odds and evens:

$$\sum_{a=1}^{h}\left[\left(\sum_{r=1}^{m} d_{2r}^{2m}\binom{\ell-2m}{a-2r}\right)+\left(\sum_{r=0}^{m-1} d_{2r+1}^{2m}\binom{\ell-2m}{a-2r-1}\right)\right].$$

Then using 9 on the odd terms, we get

$$\sum_{a=1}^{h}\left[\left(\sum_{r=1}^{m} d_{2r}^{2m}\binom{\ell-2m}{a-2r}\right)-\left(\sum_{r=0}^{m-1} d_{2r}^{2m}\binom{\ell-2m}{a-2r-1}\right)\right] \tag{10}$$

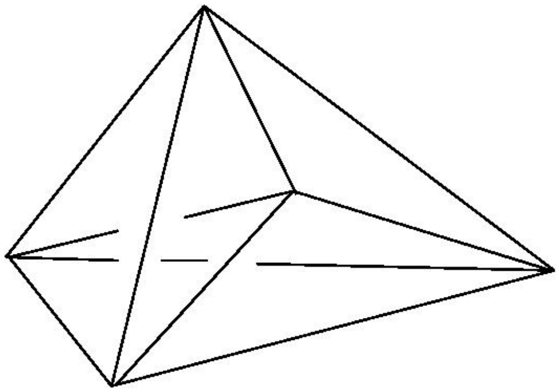

FIGURE 9. $\Delta(5,2)$

Then using Pascal's formula on 10, we have

$$\sum_{a=1}^{h}\left[d_{2m}^{2m}\binom{\ell-2m}{a-2m}-d_{0}^{2m}\binom{\ell-2m}{a-1}+\sum_{r=1}^{m-1}d_{2r}^{2m}\left(\binom{\ell-2m}{a-2r}-\binom{\ell-2m}{a-2r-1}\right)\right]. \tag{11}$$

Then switch sums on 11 to get

$$d_{2m}^{2m}\sum_{a=1}^{h}\binom{\ell-2m}{a-2m}-d_{0}^{2m}\sum_{a=1}^{h}\binom{\ell-2m}{a-1}+\left[\sum_{r=1}^{m-1}d_{2r}^{2m}\sum_{a=1}^{h}\left(\binom{\ell-2m}{a-2r}-\binom{\ell-2m}{a-2r-1}\right)\right]. \tag{12}$$

Then the sum in the left portion of 12 telescopes:

$$d_{2m}^{2m}\sum_{a=1}^{h}\binom{\ell-2m}{a-2m}-d_{0}^{2m}\sum_{a=1}^{h}\binom{\ell-2m}{a-1}+\left[\sum_{r=1}^{m-1}d_{2r}^{2m}\left(-\binom{\ell-2m}{-2r}+\binom{\ell-2m}{h-2r}\right)\right]. \tag{13}$$

Then notice that $d_{0}^{2m}=0$ and that 13 becomes

$$d_{2m}^{2m}\sum_{a=1}^{h}\binom{\ell-2m}{a-2m}+\left[\sum_{r=1}^{m-1}d_{2r}^{2m}\binom{\ell-2m}{h-2r}\right]. \tag{14}$$

Since 14 is the left hand side of 6 and each term is positive, we have proved the theorem. □

Example 6.23. $\Delta(5,2)$consists of 5 vertices and all 1-dimensional and 0-dimensional faces (which are the edges and vertices, respectively). This object is a 4-dimensional object, but it can be approximated in Figure 9. The matrix generating $K(5,2,1)$ is as follows:

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

Thus, by the formulas given in Equation 1, Corollary 6.14, and Theorem 6.22, $K(5,2,1)$ is a $[\sum_{i=0}^{h}\binom{\ell}{i}, \sum_{i=0}^{j}\binom{\ell}{i}, \sum_{a=1}^{h}\binom{\ell-1}{a-1}]_2 = [\sum_{i=0}^{2}\binom{5}{i}, \sum_{i=0}^{1}\binom{5}{i}, \sum_{a=1}^{2}\binom{5-1}{a-1}]_2 = [16,6,5]_2$ code.

Now we focus on the case when $j > 1$. It is more complicated, and we are not able to calculate the minimum distance. However, we are able to find formulas for summing row vectors of the generating matrix and and are able to compare these formulas to the conjectured upper bound.

Definition 6.24. Let $B(\sigma_1, \dots, \sigma_s)$ be the Hamming weight of adding s rows of the generating matrix $G(\ell, h, j)$ where each row corresponds to a set $\sigma_i \subseteq [\ell] = \{1, \dots, \ell\}$.

The $j > 1$ analogue to Definition 6.18 is the following.

Definition 6.25. For $1 \le t \le s$ let $X_{\sigma_r} := \{p \in P_a : x_{\sigma_r}(p) = 1\}$. Let $\mathcal{L}_t^{a,s}$ be the set of all the sets of points that evaluate to 1 on at least t of the s monomials $x_{\sigma_1}, \dots, x_{\sigma_s}$. Thus,

$$\mathcal{L}_t^{a,s} = \{X_{k_1} \cap \cdots \cap X_{k_t} : \{k_1, \dots, k_t\} \subseteq \{\sigma_1, \dots, \sigma_s\}\}.$$

Proposition 6.26. For any skeletal code $K(l, h, j)$, the Hamming weight of adding n rows of the generating matrix is

$$B(\sigma_1, \dots, \sigma_n) = \sum_{e=1}^{n} \left[(-2)^{e-1} \sum_{\substack{I \subseteq [n] \\ |I| = e}} B\left(\bigcup_{i \in I} \sigma_i \right) \right] = \sum_{e=1}^{n} \left[(-2)^{e-1} \sum_{\substack{I \subseteq [n] \\ |I| = e}} \sum_{i=0}^{h - |\cup \sigma_i|} \binom{\ell - |\cup \sigma_i|}{i} \right].$$

Proof. The $(-2)^{t-1}$ coefficient follows with the same argument as in Lemma 6.19. Then we realize that the number of points in a t-fold intersection $Y = X_{k_1} \cap \cdots \cap X_{k_t} \in \mathcal{L}_t^{a,s}$ is equal to the Hamming weight of the row corresponding to the union of the sets

$$\bigcup_{i=1}^{t} X_{k_i}.$$

Note that this row might not actually exist in the generating matrix. However, we can consider it as a row in the full matrix where $j = \ell$. Finally, the remainder of the formula is realized by applying Lemma 6.16. □

In order to show that the minimum distance is equal to the upper bound, it must be demonstrated that

$$\sum_{i=1}^{h-j} \binom{\ell - j}{i} \le \sum_{e=1}^{n} \left[(-2)^{e-1} \sum_{\substack{I \subseteq [n] \\ |I| = e}} \sum_{i=0}^{h - |\cup \sigma_i|} \binom{\ell - |\cup \sigma_i|}{i} \right]. \tag{15}$$

However, this proof turns out to be difficult. For the remainder, we examine a few cases.

Proposition 6.27. For $j > 1$, $B(\sigma_1, \sigma_2) \ge \sum_{i=1}^{h-j} \binom{\ell-j}{i}$.

Proof. Let $|\sigma_1| = i_1$, $|\sigma_2| = i_2$. Without the loss of generality, assume (relabeling as necessary) $i_1 \le i_2$ and $\sigma_1 \ne \sigma_2$ (else, $B(\sigma_1, \sigma_2) = 0$). Note that $|\sigma_1 \cup \sigma_2| \ge i_1 + 1$. By Proposition 6.26,

$$B(\sigma_1, \sigma_2) = B(\sigma_1) + B(\sigma_2) - 2B(\sigma_1 \cup \sigma_2)$$

$$= \sum_{i=1}^{h-i_1} \binom{\ell - i_1}{i} + \sum_{i=1}^{h-i_2} \binom{\ell - i_2}{i} - 2 \sum_{i=1}^{h - |\sigma_1 \cup \sigma_2|} \binom{\ell - |\sigma_1 \cup \sigma_2|}{i}.$$

Since $|\sigma_1 \cup \sigma_2| \geq i_1 + 1$,

$$\sum_{i=1}^{h-i_1} \binom{\ell - i_1}{i} + \sum_{i=1}^{h-i_2} \binom{\ell - i_2}{i} - 2 \sum_{i=1}^{h-|\sigma_1 \cup \sigma_2|} \binom{\ell - |\sigma_1 \cup \sigma_2|}{i}$$

$$\geq \sum_{i=1}^{h-i_1} \binom{\ell - i_1}{i} + \sum_{i=1}^{h-i_2} \binom{\ell - i_2}{i} - 2 \sum_{i=1}^{h-(i_1+1)} \binom{\ell - (i_1+1)}{i}.$$

Hence, we prove the inequality for $|\sigma_1 \cup \sigma_2| = i_1 + 1$. There are now two cases: (1) $i_1 = i_2$ or (2) $i_1 + 1 = i_2$ and $\sigma_1 \subseteq \sigma_2$. For Case 1, assume $i_1 = i_2 = a$, so the left-hand side of the proposed inequality becomes

$$2\left(\sum_{i=1}^{h-a} \binom{\ell - a}{i}\right) - 2\left(\sum_{i=1}^{h-(a+1)} \binom{\ell - (a+1)}{i}\right).$$

Since $j \geq a$, $\sum\limits_{i=1}^{h-j} \binom{\ell-j}{i} \leq \sum\limits_{i=1}^{h-a} \binom{\ell-a}{i}$, so it suffices to show

$$2\left(\sum_{i=1}^{h-a} \binom{\ell - a}{i}\right) - 2\left(\sum_{i=1}^{h-(a+1)} \binom{\ell - (a+1)}{i}\right) \geq \sum_{i=1}^{h-a} \binom{\ell - a}{i}.$$

This last expression can be rewritten as

$$\sum_{i=1}^{h-a} \binom{\ell - a}{i} - 2\left(\sum_{i=1}^{h-(a+1)} \binom{\ell - (a+1)}{i}\right) \geq 0.$$

We can phrase this last inequality by saying, “The Hamming Weight decreases by more than half in each group of rows (each group of rows corresponds to polynomials of the same degree).” Since $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$,

$$\sum_{i=1}^{h-a} \binom{\ell - a}{i} - 2\left(\sum_{i=1}^{h-(a+1)} \binom{\ell - (a+1)}{i}\right)$$

$$= \sum_{i=1}^{h-a} \left(\binom{\ell - a - 1}{i} + \binom{\ell - a - 1}{i-1}\right) - 2\left(\sum_{i=1}^{h-a-1} \binom{\ell - a - 1}{i}\right)$$

$$= 2\left(\sum_{i=1}^{h-a-1} \binom{\ell - a - 1}{i}\right) - 2\left(\sum_{i=1}^{h-a-1} \binom{\ell - a - 1}{i}\right) + \binom{\ell - a - 1}{0} + \binom{\ell - a - 1}{h-a}$$

$$= 1 + \binom{\ell - a - 1}{h-a} \geq 0.$$

Thus, for Case 1, $B(\sigma_1, \sigma_2) \geq \sum\limits_{i=1}^{h-j} \binom{\ell-j}{i}$.

For Case 2, since the Hamming Weight decreases by more than half in each group of rows,

$$B(\sigma_1, \sigma_2) \geq B(\sigma_1) - B(\sigma_2) \geq \frac{1}{2} B(\sigma_1) \geq B(\sigma_2) \geq B(\sigma_j),$$

where $B(\sigma_j) = \sum\limits_{i=1}^{h-j} \binom{\ell-j}{i}$, since $|\sigma_j| \geq |\sigma_2|$. Thus, $B(\sigma_1, \sigma_2) \geq \sum\limits_{i=1}^{h-j} \binom{\ell-j}{i}$. □

Proposition 6.28. $B(\sigma_1,\dots,\sigma_n) = B(\sigma_1,\dots,\sigma_{n-1}) + B(\sigma_n) - 2B(\sigma_1 \cup \sigma_n,\dots,\sigma_{n-1}\cup\sigma_n)$.

Proof. From Proposition 6.26,

$$B(\sigma_1,\dots,\sigma_n) = \sum_{i=1}^{n} B(\sigma_i) - 2\sum_{i_1,i_2} B(\sigma_{i_1}\cup\sigma_{i_2}) + \cdots + (-2)^{n-1}B(\sigma_1\cup\cdots\cup\sigma_n). \tag{16}$$

Rearranging so that all terms involving σ_n on the right-hand side of 16 are isolated yields

$$\begin{aligned}\left[\sum_{i=1}^{n-1} B(\sigma_i) - 2\sum_{i_1,i_2\neq n} B(\sigma_{i_1}\cup\sigma_{i_2}) + \cdots + (-2)^{n-2}B(\sigma_1\cup\cdots\cup\sigma_{n-1})\right]& \\ + \left[B(\sigma_n) - 2\sum_{i=1}^{n-1} B(\sigma_i\cup\sigma_n) + \cdots + (-2)^{n-1}B(\sigma_1\cup\cdots\cup\sigma_n)\right]&.\end{aligned} \tag{17}$$

Again, by Proposition 6.26,

$$B(\sigma_1,\dots,\sigma_{n-1}) = \sum_{i=1}^{n-1} B(\sigma_i) - 2\sum_{i_1,i_2\neq n} B(\sigma_{i_1}\cup\sigma_{i_2}) + \cdots + (-2)^{n-2}B(\sigma_1\cup\cdots\cup\sigma_{n-1}), \tag{18}$$

and, since $\sigma_i \cup \sigma_j \cup \sigma_n = (\sigma_i\cup\sigma_n)\cup(\sigma_j\cup\sigma_n)$,

$$\begin{aligned} & B(\sigma_1\cup\sigma_n,\dots,\sigma_{n-1}\cup\sigma_n) \\ &= \sum_{i=1}^{n-1} B(\sigma_i\cup\sigma_n) - 2\sum_{i_1,i_2} B(\sigma_{i_1}\cup\sigma_{i_2}\cup\sigma_n) + \cdots + (-2)^{n-2}B(\sigma_1\cup\cdots\cup\sigma_n).\end{aligned} \tag{19}$$

Substituting 18 and 19 back into the right-hand side of 17 yields the claimed formula. □

Proposition 6.29. Let $|\sigma_1| \le |\sigma_2| \le \cdots \le |\sigma_k| \le |\sigma_{k+1}| = \cdots = |\sigma_{n-1}| = |\sigma_n|$ for $k \le n$. If $|\sigma_k| < (n-k) + |\sigma_n|$, then $B(\sigma_1,\dots,\sigma_n) \ge B(\sigma_n)$.

Proof. The proof will be by induction on n. The base case $n = 2$ is covered by the proof of Proposition 6.27. Thus, assume $B(\sigma_1,\dots,\sigma_x) \ge B(\sigma_x)$ for $x < n$. It must be demonstrated that $B(\sigma_1,\dots,\sigma_n) \ge B(\sigma_n)$. By the formula in Proposition 6.28,

$$B(\sigma_1,\dots,\sigma_n) = B(\sigma_1,\dots,\sigma_{n-1}) + B(\sigma_n) - 2B(\sigma_1\cup\sigma_n,\dots,\sigma_{n-1}\cup\sigma_n).$$

Note that $B(\sigma_1\cup\sigma_n,\dots,\sigma_{n-1}\cup\sigma_n)$ is the Hamming weight of $n-1$ row vectors being added together, fulfilling the property that all nonzero entries of the rows $\sigma_i\cup\sigma_n$ are also nonzero entries in the row σ_n. Hence,

$$B(\sigma_n) \ge B(\sigma_1\cup\sigma_n,\dots,\sigma_{n-1}\cup\sigma_n).$$

Thus,

$$\begin{aligned} B(\sigma_1,\dots,\sigma_n) &\ge B(\sigma_1,\dots,\sigma_{n-1}) + B(\sigma_n) - 2B(\sigma_n) \\ &= B(\sigma_1,\dots,\sigma_{n-1}) - B(\sigma_n).\end{aligned}$$

Then using Proposition 6.28 repeatedly, we have

$$\begin{aligned} B(\sigma_1,\dots,\sigma_n) &\ge B(\sigma_1,\dots,\sigma_{n-2}) + B(\sigma_{n-1}) - 2B(\sigma_1\cup\sigma_{n-1},\dots,\sigma_{n-2}\cup\sigma_{n-1}) - B(\sigma_n) \\ &\ge B(\sigma_1,\dots,\sigma_{n-2}) + B(\sigma_{n-1}) - 2B(\sigma_{n-1}) - B(\sigma_n) \\ &= B(\sigma_1,\dots,\sigma_{n-2}) - B(\sigma_{n-1}) - B(\sigma_n) \\ &\ge \ldots \ge B(\sigma_1,\dots,\sigma_k) - (B(\sigma_{k+1}) + \cdots + B(\sigma_n)).\end{aligned}$$

By the induction hypothesis,

$$\begin{aligned} B(\sigma_1, \ldots, \sigma_n) &\geq B(\sigma_1, \ldots, \sigma_k) - (B(\sigma_{k+1}) + \cdots + B(\sigma_n)) \\ &\geq B(\sigma_k) - (B(\sigma_{k+1}) + \cdots + B(\sigma_n)) \\ &= B(\sigma_k) - (n-k)B(\sigma_n), \end{aligned}$$

since $|\sigma_{k+1}| = \cdots = |\sigma_n|$. Let $T_{k+m} \subseteq [\ell]$ such that $|T_{k+m}| = |\sigma_k| + m$ (so $B(\sigma_n) \leq B(T_{k+m})$ for $m \leq (n-k)$) and recall from Proposition 6.27 that the Hamming weight between groups of rows decreases by more than half (also, note that T_{k+i} is a row with higher Hamming weight than the row T_{k+i+1}). Thus,

$$\begin{aligned} B(\sigma_1, \ldots, \sigma_n) &\geq B(\sigma_k) - (n-k)B(\sigma_n) \\ &\geq [B(\sigma_k) - B(T_{k+1})] - B(T_{k+2}) - \cdots - B(\sigma_n) \\ &\geq [B(T_{k+1}) - B(T_{k+2})] - B(T_{k+3}) - \cdots - B(\sigma_n) \\ &\geq \ldots \geq B(T_{n-1}) - B(\sigma_n) \geq B(\sigma_n). \end{aligned}$$

Note that there are exactly enough T_{k+m} terms for the above inequalities since $|\sigma_k| < (n-k) + |\sigma_n|$. Thus, for $|\sigma_k| < (n-k) + |\sigma_n|$, $B(\sigma_1, \ldots, \sigma_n) \geq B(\sigma_n)$. □

With respect to Equation 15 Propositions 6.26, 6.27, 6.28, and 6.29 give evidence for the following conjecture.

Conjecture 6.30. The minimum distance of $K(\ell, h, j)$ is

$$d = \sum_{i=1}^{h-j} \binom{\ell - j}{i}.$$

Proposition 6.31. The codes $K(\ell, \ell-1, \ell-2)$ and $\mathcal{H}_\ell$ are permutation equivalent.

Proof. By Theorem 1.8.1 in Huffman and Pless [9], any $[2^r - 1, 2^r - 1 - r, 3]$ code is equivalent to $\mathcal{H}_r$. The length of $K(\ell, \ell-1, \ell-2)$ by Equation 1 is

$$\sum_{i=0}^{h} \binom{\ell}{i} = \sum_{i=0}^{\ell-1} \binom{\ell}{i} = \sum_{i=0}^{\ell} \binom{\ell}{i} - \binom{\ell}{\ell} = 2^\ell - 1.$$

The dimension of $K(\ell, \ell-1, \ell-2)$ by Corollary 6.14 is

$$\sum_{i=0}^{j} f_i = \sum_{i=0}^{\ell-2} f_i = \sum_{i=0}^{\ell-2} \binom{\ell}{i} = \sum_{i=0}^{\ell} \binom{\ell}{i} - \binom{\ell}{\ell-1} - \binom{\ell}{\ell} = 2^\ell - \ell - 1 = 2^\ell - 1 - \ell.$$

The upper bound on the minimum distance Lemma 6.16 is by

$$d \leq \sum_{i=0}^{h-j} \binom{\ell - j}{i} = \sum_{i=0}^{\ell-(\ell-1)} \binom{\ell - (\ell-2)}{i} = \sum_{i=0}^{1} \binom{2}{i} = \binom{2}{0} + \binom{2}{1} = 3.$$

By Corollary 1.4.14 in [9], since the rows in the matrix generating $K(\ell, \ell-1, \ell-2)$ are linearly independent, $d \geq 3$, implying $d = 3$. Thus, $K(\ell, \ell-1, \ell-2) \cong \mathcal{H}_\ell$. □

Remark 6.32. Since the two codes are permutation-equivalent, there must be some means by which to transform $K(\ell, \ell-1, \ell-2)$ into $\mathcal{H}_\ell$. This process is achieved by adding rows, permuting columns, and permuting rows in the matrix generating $K(\ell, \ell-1, \ell-2)$. Since the matrix (if the points in $V(I)$ are ordered appropriately) can be made into an upper-triangular matrix adjoined to a $(2^\ell - \ell - 1) \times \ell$ matrix, adding all rows to rows above them will result in a $[I_{2^\ell - \ell - 1}|A]$ matrix. In the A block (corresponding to points in $V(I)$ that have only one 0 [and thus $\ell{-}1$ ones] as a component), there are (since there are $\ell{-}2$ rows) $\sum\limits_{0}^{\ell-2-j} \binom{\ell-1-j}{i} = 2^{\ell-1-j}-1$ ones, which is odd. Thus, when all the rows are added, the A block remains unchanged. This A block consists of binary strings of length ℓ that corresponding to polynomials up to degree $\ell-2$ being evaluated on points with $\ell-1$ ones. These strings, once transposed and combined with a transposed identity matrix, will consist of all numbers in binary from 1 to 2^{ℓ}, indicating its equivalence to $\mathcal{H}_{\ell}$.

7. McEliece Cryptosystem

An important application of the working on coding theory and subspace arrangement codes is the creation of a cryptosystem. Work has already been done on making ciphers based upon error-correcting codes. The most prominent example is the McEliece Cryptosystem, which is described in detail in [6] and [18].

7.1. **History and Description.** Robert McEliece first proposed the idea of a public key cryptosystem built from error-correcting codes in 1978. Since McEliece's development of the scheme, various proposals have been made with different error-correcting codes, but most of them have been proven to not be as efficient as McEliece's proposal to use Goppa Codes [18]. Goppa codes have a closed formula for length, dimension, and minimum distance based upon two parameters, m and t: $n = 2^m$, $k = n - mt$, and $d = 2t + 1$. The strength of the McEliece cryptosystem is that it can be very hard to determine what codeword is closest to a random string of digits. The process for encryption and decryption follows a few basic guidelines. The receiver will make a public key, and the sender will encrypt their message with the key so that the receiver can decrypt it (note that in the standard cryptologic terminology, the receiver and sender are Bob and Alice, respectively). The receiver selects the generating matrix G for C, a $[n, k, d]_2$ code, so G is a k x n matrix. The receiver also selects S, an invertible k x k matrix over $\mathbb{F}_2$, and P, an n x n permutation matrix. The receiver then calculates $G_1 = SGP$. While keeping S, G, and P private, the receiver publishes G_1 as a public key. The sender who wants to send a message, x, randomly selects a vector, e, of length n with weight t. The sender then forms $y = xG_1 + e$ and sends y to the receiver. The receiver then needs to decrypt y to find x. First, the receiver finds y_1 by calculating yP^{-1}. Using the error decoder for C (such as a parity check matrix) the receiver finds x_1, the codeword in C that the is the closest to y_1. Next, the receiver finds x_0, where $x_0G = x_1$. Finally, the receiver recovers x by calculating x_0S^{-1}. This process can best by explained by an example.

Example 7.1. Let C be the $[15, 5, 7]_2$ skeletal code corresponding to $K(4, 2, 1)$. The matrix for C is

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

However, to make calculating a parity check matrix easier, replace the first row with the sum of all five rows:

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

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

and let $P = \begin{bmatrix} 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$.

Thus, $G_1 = SGP = \begin{bmatrix} 0&1&0&0&0&0&1&1&1&1&1&1&0&0&0 \\ 0&1&1&0&0&0&0&0&0&1&1&1&1&1&1 \\ 1&0&1&1&0&0&0&1&1&1&1&0&0&0&1 \\ 0&0&0&1&1&0&1&1&0&0&1&1&0&1&1 \\ 0&0&0&0&1&1&0&1&1&1&1&0&1&1&0 \end{bmatrix}$.

Now, the sender wants to send the message

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

with e randomly chosen as

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

The sender then computes $y = xG_1 + e$, resulting in

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

The receiver then starts decrypting y, first calculating $y_1 = yP^{-1} = yP^{14}$

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

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

The next step is to use the parity check matrix, H, of C to correct y_1 for errors. Since $G = [I_k|Q]$, by principle, $H = [-Q^T|I_{n-\ell}]$. Thus,

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

In order to figure out how to correct y_1, the syndrome of y_1, termed $S(y_1)$, needs to be found. $S(y_1) = y_1 H^T$:

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

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

Note that the syndrome has dimension 10, so there are $2^{10} = 1024$ potential error-correction actions to be taken (with each action being uniquely correlated to a syndrome). Observe that

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

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

implying that $\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ (which is equal to eP^{-1}) is the coset leader for the syndrome $\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$. Thus, to error-correct y_1 to find x_1, add the coset leader to y_1:

$$\begin{aligned} x_1 &= \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \\ &+ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}. \end{aligned}$$

Next, the vector $x_0 G = x_1$ must be found. Careful observation reveals that x_1 is the sum of the second, third, and fourth rows of G, so $x_0 = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \end{bmatrix}$. The last step in the process is to compute $x = x_0 S^{-1}$:

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

The original message has been correctly deciphered.

7.2. **Security.** An important reason to study the McEliece cryptosystem is for its security. Work done by Hang Dinh, Cristopher Moore, and Alexander Russell in [5] indicate that Shor's algorithm is not effective against the McEliece cryptosystem. Shor's algorithm, discovered in 1994, describes a factoring process possible with quantum computing. This algorithm can be applied to modern ciphers as a quantum Fourier attack. This attack has been demonstrated to be successful on a number of cryptosystems, including the widely-used RSA cipher. However, Hang Dinh, Cristopher Moore, and Alexander Russell show in [5] that Shor's algorithm of factoring integers does not aid in solving the nearest codeword problem, which is the central point of security of the McEliece cryptosystem. Hence, the McEliece cryptosystem is immune to this attack. Thus, applying skeletal codes to the McEliece cryptosystem could result in a viable method for encryption if quantum computing is realized.

8. Sage Code

An integral part of this project has been using Sage to execute calculations that would be too difficult to calculate by hand. What follows are four script sessions that were part of the project.

8.1. **Generating Evaluation Codes and Their Properties.**

Work was initially done in the algebraic geometric software Macaulay2. The following code (in conjunction with an example) details using a Macaulay2 interface in Sage to construct an evaluation code. The ideal, finite field, and vector space are used to create the matrix representing the codes. From this matrix, the dimension of the code can be calculated.

```
sage: loadPackage "RationalPoints";
sage: S=ZZ/7[x,y,z];
sage: I=ideal(x*y,x*z,y*z);
sage: p=rationalPoints I;
o3 : Ideal of
sage: codelength=length p
19
sage: p
{{0, 0, 0}, {1, 0, 0}, {2, 0, 0}, {3, 0, 0}, {4, 0, 0}, {5, 0, 0}, {6, 0,
---------------------------------------------------------------------------
0}, {0, 1, 0}, {0, 2, 0}, {0, 3, 0}, {0, 4, 0}, {0, 5, 0}, {0, 6, 0}, {0,
---------------------------------------------------------------------------
0, 1}, {0, 0, 2}, {0, 0, 3}, {0, 0, 4}, {0, 0, 5}, {0, 0, 6}}

List
sage: k=basis(0,4,S)
| 1 x x2 x3 x4 x3y x3z x2y x2y2 x2yz x2z x2z2 xy xy2 xy3 xy2z xyz xyz2 xz
---------------------------------------------------------------------------
xz2 xz3 y y2 y3 y4 y3z y2z y2z2 yz yz2 yz3 z z2 z3 z4 |

       1       35
Matrix S  <---
sage: e=apply(0 .. (rank source k)-1, i->(g_i=(a,b,c)->sub(k_(0,i),
{x=>a,y=>b,z=>c})));
sage: T=matrix toList apply(0 .. (rank source k)-1,j->
toList apply(0 .. #p-1, i->e_j(toSequence p_i)));
ZZ 35      ZZ 19
o43 : Matrix (--)   <--- (--)
               7           7
sage: C=image transpose T;
sage: C
image |1 0  0  0  0  0000000000000000 0  0  0  0  000000 0  0  0  0 |
      |1 1  1  1  1  0000000000000000 0  0  0  0  000000 0  0  0  0 |
      |1 2  -3 1  2  0000000000000000 0  0  0  0  000000 0  0  0  0 |
      |1 3  2  -1 -3 0000000000000000 0  0  0  0  000000 0  0  0  0 |
      |1 -3 2  1  -3 0000000000000000 0  0  0  0  000000 0  0  0  0 |
      |1 -2 -3 -1 2  0000000000000000 0  0  0  0  000000 0  0  0  0 |
      |1 -1 1  -1 1  0000000000000000 0  0  0  0  000000 0  0  0  0 |
      |1 0  0  0  0  0000000000000000 1  1  1  1  000000 0  0  0  0 |
      |1 0  0  0  0  0000000000000000 2  -3 1  2  000000 0  0  0  0 |
      |1 0  0  0  0  0000000000000000 3  2  -1 -3 000000 0  0  0  0 |
      |1 0  0  0  0  0000000000000000 -3 2  1  -3 000000 0  0  0  0 |
      |1 0  0  0  0  0000000000000000 -2 -3 -1 2  000000 0  0  0  0 |
      |1 0  0  0  0  0000000000000000 -1 1  -1 1  000000 0  0  0  0 |
      |1 0  0  0  0  0000000000000000 0  0  0  0  000000 1  1  1  1 |
      |1 0  0  0  0  0000000000000000 0  0  0  0  000000 2  -3 1  2 |
      |1 0  0  0  0  0000000000000000 0  0  0  0  000000 3  2  -1 -3|
      |1 0  0  0  0  0000000000000000 0  0  0  0  000000 -3 2  1  -3|
      |1 0  0  0  0  0000000000000000 0  0  0  0  000000 -2 -3 -1 2 |
      |1 0  0  0  0  0000000000000000 0  0  0  0  000000 -1 1  -1 1 |

ZZ                      ZZ 19
---module, submodule of (--)
 7                       7
sage: codedimension=rank C
13
```

The above example describes an evaluation code with length 19 and dimension 13 over the finite field $\mathbb{F}_7$. Note that Macaulay2 lacks any function to calculate minimum distance.

The following code uses a file assembled through help from Professor W. David Joyner that has been uploaded to the Sage server. The following production of the code shows the various components of the code simultaneously with an example.

```
sage: attach /home/wakefield/Downloads/evaluation_codes.sage
sage: R.<x,y,z,w>=PolynomialRing(GF(2),4,"x,y,z,w");
sage: I=R.ideal([x*y*z*w]);
sage: P=get_exps(4,2);
sage: P
[[0, 0, 0, 0], [1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]
sage: Pts = get_pts(I)
sage: Pts
[[0, 0, 0, 0], [1, 0, 0, 0], [0, 1, 0, 0], [1, 1, 0, 0], [0, 0, 1, 0],
[1, 0, 1, 0], [0, 1, 1, 0], [1, 1, 1, 0], [0, 0, 0, 1], [1, 0, 0, 1],
[0, 1, 0, 1], [1, 1, 0, 1], [0, 0, 1, 1], [1, 0, 1, 1], [0, 1, 1, 1]]
sage: C = evaluation_code(P, Pts); C
Linear code of length 15, dimension 5 over Finite Field of size 2
sage: C.minimum_distance()
7
```

The first line of code attaches the file Professor Joyner created. The second establishes the finite field, and the third creates the ideal. The next two lines find the points on the ideal based on the selection of j, with the following line formally establishing the code. A built-in Sage function for minimum distance is used in the last line. The example uses the ideal $< x_1x_2x_3x_4 >$ in $\mathbb{F}_2^4$ with $j = 1$ to find a $[15, 5, 7]_2$ code. An important note is that with moderately-high parameters used, this Sage code is not efficient and can take significant amounts of time, especially to calculate minimum distance. Thus, it is very beneficial to develop formulas to calculate the properties.

8.2. **Boolean Arrangement Codes.** A key part of the arguments involved with developing formulas for the dimension of error-correcting codes generated by the ideal $I = < x_1 \cdots x_\ell >$ in $\mathbb{F}_q^\ell$ was creating a matrix that represented the code. While a straightforward task for small values, the process was considerably speed up by the following Sage code that produced the matrix. The program is described and then examples are provided.

```
sage: def matrix_code_generator(j):
...       """
...       A function designed to create the abbreviated sub-matrix of a
matrix for the code corresponding to the coordinate hyperplane arrangement.
The sub-matrix corresponds exclusively to polynomials consisting of one
variable raised to a power.  The input is j, the upper bound on the
polynomials used in generation.
...       EXAMPLES: See below for j=1,...,10
...       INPUT: j
...       OUTPUT: factored determinant of the abbreviated sub-matrix
...       """
...       w=var(’w’)
...       row=[0]*j
...       for k in range(j):
...           row[k]=w^k
...       CodeMatrix=[[0]*j]*j
...       for l in range(j):
...           newrow=[0]*j
...           for k in range(j):
...               newrow[k]=w^k
...           for m in range(j):
...               newrow[m]=row[m]**(l+1)
...           CodeMatrix[l]=newrow
...       CodeMatrixlist=[0]*(j**2)
...       for n in range(j):
...           for p in range(j):
...               CodeMatrixlist[n+j*p]=CodeMatrix[p][n]
...       MS=MatrixSpace(PolynomialRing(ZZ,w),j,j)
...       Code_matrix=MS.matrix(CodeMatrixlist)
...       determinant=det(Code_matrix)
...       factored_det=determinant.factor()
...       return factored_det
sage: def matrix_code_echelon(j):
...       """
...       A function designed to create the abbreviated sub-matrix of a
matrix for the code corresponding to the coordinate hyperplane arrangement.
The sub-matrix corresponds exclusively to polynomials consisting of one
variable raised to a power. The input is j, the upper bound on the
polynomials used in generation.
...       EXAMPLES: See below for j=1,...,5
...       INPUT: j
...       OUTPUT: echelon form of the abbreviated sub-matrix
...       """
...       w=var('w')
...       row=[0]*j
...       for k in range(j):
...           row[k]=w^k
...       CodeMatrix=[[0]*j]*j
...       for l in range(j):
...           newrow=[0]*j
...           for k in range(j):
...               newrow[k]=w^k
...           for m in range(j):
...               newrow[m]=row[m]**(l+1)
...           CodeMatrix[l]=newrow
...       CodeMatrixlist=[0]*(j**2)
...       for n in range(j):
...           for p in range(j):
...               CodeMatrixlist[n+j*p]=CodeMatrix[p][n]
...       MS=MatrixSpace(PolynomialRing(ZZ,w),j,j)
...       Code_matrix=MS.matrix(CodeMatrixlist)
...       echelon=Code_matrix.echelon_form()
...       return echelon
sage: matrix_code_generator(1)
1
sage: matrix_code_echelon(1)
[1]
sage: matrix_code_generator(2)
(w - 1) * w
sage: matrix_code_echelon(2)
[      1          w]
[      0    w^2 - w]
sage: matrix_code_generator(3)
(w + 1) * (w - 1)^3 * w^4
sage: matrix_code_echelon(3)
[1    w                        w^2]
[0   w^2 - w             w^4 - w^2]
[0    0      w^6 - w^5 - w^4 + w^3]
sage: matrix_code_generator(4)
(w + 1)^2 * (w - 1)^6 * w^10 * (w^2 + w + 1)
sage: matrix_code_echelon(4)
[1    w        w^2                                    w^3]
[0  w^2-w    w^4-w^2                            w^6 - w^3]
[0    0    w^6-w^5-w^4+w^3                w^9-w^7-w^6+w^4]
[0    0         0            w^12-w^11-w^10+w^8 + w^7-w^6]
sage: matrix_code_generator(5)
(w + 1)^4 * (w - 1)^10 * w^20 * (w^2 + 1) * (w^2 + w + 1)^2
sage: matrix_code_echelon(5)
```

```
[1 w    w^2              w^3                                              w^4]
[0 w^2-w w^4-w^2         w^6-w^3                                      w^8-w^4]
[0 0  w^6-w^5-w^4+w^3    w^9-w^7-w^6+w^4                     w^12-w^9-w^8+w^5]
[0 0    0  w^12-w^11-w^10+w^8+w^7-w^6  w^16-w^14-w^13-w^12+w^11+w^10+w^9-w^7]
[0 0    0                0              w^20-w^19-w^18+2*w^15-w^12-w^11+w^10]
sage: matrix_code_generator(6)
(w + 1)^6 * (w - 1)^15 * w^35 * (w^2 + 1)^2 * (w^2 + w + 1)^3
* (w^4 + w^3 + w^2 + w + 1)
sage: matrix_code_generator(7)
(w + 1)^9 * (w - 1)^21 * w^56 * (w^2 - w + 1) * (w^2 + 1)^3 * (w^2 + w + 1)^5
* (w^4 + w^3 + w^2 + w + 1)^2
sage: matrix_code_generator(8)
(w + 1)^12 * (w - 1)^28 * w^84 * (w^2 - w + 1)^2 * (w^2 + 1)^4
* (w^2 + w + 1)^7 * (w^4 + w^3 + w^2 + w + 1)^3
* (w^6 + w^5 + w^4 + w^3 + w^2 + w + 1)
sage: matrix_code_generator(9)
(w + 1)^16 * (w - 1)^36 * w^120 * (w^2 - w + 1)^3 * (w^2 + 1)^6
* (w^2 + w + 1)^9 * (w^4 + 1) * (w^4 + w^3 + w^2 + w + 1)^4
* (w^6 + w^5 + w^4 + w^3 + w^2 + w + 1)^2
sage: matrix_code_generator(10)
(w + 1)^20 * (w - 1)^45 * w^165 * (w^2 - w + 1)^4 * (w^2 + 1)^8
* (w^2 + w + 1)^12 * (w^4 + 1)^2 * (w^4 + w^3 + w^2 + w + 1)^5
* (w^6 + w^3 + 1) * (w^6 + w^5 + w^4 + w^3 + w^2 + w + 1)^3
```

8.3. **Binary Skeleton Codes.** When working on the $j = 1$ case for codes developed from the simplicial complex $\Delta(\ell, h)$, to prove the minimum distance formula, observations where made concerning what the Hamming weight of the sum of s row vectors became. Some Sage scripts, listed below, facilitated the easy calculation of the Hamming weight of this sum.

```
def F(l,h,s):
     sum = 0
     for a in range(1,h+2):
          for t in range(1,s+1):
               sum=sum + (-2)**(t-1) * (binomial(s,t)) * (binomial(l-t,a-t))
     return sum
```

9. Conclusion

While much has been achieved in the work of this project, room for further research is still present. There are many other classes of subspace arrangement codes and simplicial complex codes where we know neither the dimension nor the minimum distance. Nonetheless, by varying the parameters ℓ, h, and j of the skeletal codes $K(\ell, h, j)$, we are able to produce many useful codes with various values of length, dimension, and minimum distance. In addition, it is possible that the minimum distance and dimension formulas developed through this project might prove useful in understanding theoretical problems in subspace arrangements and simplicial complexes. Furthermore, we examined a few examples of our skeletal codes in the McEliece cryptosystem. It has not yet been determined (though it seems likely) that there are nice decoding algorithms for our codes, thereby making them well-suited for the McEliece cryptosystem. However, more work is needed to provide a larger class of codes with the same parameters to make the system practical. This project was exciting, especially since it has provided, through open problems such as Conjecture 6.30, many intriguing future research topics.

References

[1] Christos A. Athanasiadis. Characteristic polynomials of subspace arrangements and finite fields. *Adv. Math.*, 122(2):193–233, 1996.

[2] Anders Björner. Subspace arrangements. In *First European Congress of Mathematics, Vol. I (Paris, 1992)*, volume 119 of *Progr. Math.*, pages 321–370. Birkhäuser, Basel, 1994.

[3] Maria Bras-Amorós. Algebraic-geometry codes, one-point codes, and evaluation codes. *Designs, Codes and Cryptography*, 43(2-3):137–145, June 2007.

[4] David A. Cox, John Little, and Donal O'Shea. *Using Algebraic Geometry*, volume 185 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2005.

[5] Hang Dinh, Cristopher Moore, and Alexander Russell. The mceliece cryptosystem resists quantum fourier sampling attacks. *CoRR*, abs/1008.2390, 2010.

[6] D. Engelbert, R. Overbeck, and A. Schmidt. A summary of mceliece- type cryptosystems and their security. Preprint, May 2006.

[7] Leah Gold, John Little, and Hal Schenck. Cayley-bacharach and evaluation codes on complete intersections. *Journal of Pure and Applied Algebra*, 196:91–99, 2005.

[8] Joe Harris. *Algebraic Geometry: A First Course*, volume 133 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1992.

[9] W. Cary Huffman and Vera Pless. *Fundamentals of Error-Correcting Codes*. Cambridge University Press, New York, 2003.

[10] Amanda A. Johnson. Constructing codes from curves. Unpublished, May 1997.

[11] Relinde Jurrius and Ruud Pellikaan. Extended and generalized weight enumerators. *Proceedings of the International Workshop on Coding and Cryptography, WCC 2009, Ullensvang*, pages 76–91, May 2009.

[12] Relinde Jurrius and Ruud Pellikaan. The extended coset leader weight enumerator. *Proceedings 30th Symposium on Information Theory on the Benelux*, pages 217–224, May 2009.

[13] Peter Orlik and Hiroaki Terao. *Arrangements of Hyperplanes*, volume 300 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1992.

[14] Steven Roman. *Coding and Information Theory*, volume 134 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1996.

[15] Bruce E. Sagan. Why the characteristic polynomial factors. *Bulletin of the American Mathematical Society*, 36(2):113–133, February 1999.

[16] Karen E. Smith, Lauri Kahanpää, Pekka Kekäläinen, and William Traves. *An Invitation to Algebraic Geometry*. Universitext. Springer-Verlag, New York, 2000.

[17] Richard P. Stanley. *Combinatorics and Commutative Algebra*, volume 41 of *Progress in Mathematics*. Birkhäuser Boston Inc., Boston, MA, second edition, 1996.

[18] Wade Trappe and Lawrence C. Washington. *Introduction to Cryptography with Coding Theory*. Pearson Prentice Hall, Upper Saddle River, NJ, second edition, 2006.

[19] Michael Tsfasman, Serge Vlăduţ, and Dmitry Nogin. *Algebraic Geometric Codes: Basic Notions*, volume 139 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2007.

[20] Alan Tucker. *Applied Combinatorics*. John Wiley & Sons Inc., New York, fifth edition, 2007.

[21] J.H. van Lint. *Introduction to Coding Theory*, volume 86 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, third edition, 1999.