MSRI-UP 2009

Technical Reports

Preface

This publication contains the technical reports written by the students who participated in the 2009 Mathematical Sciences Research Institute - Undergraduate Program (MSRI-UP) in Berkeley, CA. MSRI-UP is a six-week Research Experience for Undergraduates (REU) funded by National Science Foundation (NSF, grant No. DMS-0754872) and the National Security Agency (NSA, grant No. H98230-09-0103).

The seventeen students who participated in the MSRI-UP 2009 came from universities in Arizona, California, Georgia, Hawaii, Louisiana, Minnesota, New Jersey, New York, Pennsylvania and Puerto Rico. They worked in groups on undergraduate research projects in the field of coding theory designed by and under the direction of Professor John B. Little, College of the Holy Cross. Professor Little and the students were supported by an academic staff consisting of Dr. Emille Davie, Postdoctoral Fellow, University of California, Santa Barbara; Candice Price, Graduate Student, University of Iowa; and Ashley Wheeler, Graduate Student, University of Michigan.

The reports contained herein are the culmination of hundreds of hours of work by the MSRI-UP 2009 students and staff. We are confident that the interested reader will find the work done by these undergraduates mathematically rich, interesting and impressive. (We mention that Prof. Little and the rest of the MSRI-UP staff has done some editing of the reports, but because the quantity of work produced by the students during the short six-week program is so great and because of other time constraints, these reports should be characterized as "not-fully edited".)

MSRI-UP's primary goal is *to increase the number of graduate degrees in the mathematical sciences, especially doctorates, earned by U.S. citizens and permanent residents by cultivating heretofore untapped mathematical talent.* The summer research experience along with subsequent professional development opportunities and mentoring are designed to cultivate the mathematical talent of the MSRI-UP undergraduates.

Much support for the program was provided by many individuals at MSRI; in particular we thank Robert Bryant, Hélène Barcelo, Kathy M. O'Hara, Enrico Hernandez, Jonathan Rubinsky, Anna Foster and Arne Jensen. In addition, MSRI-UP co-directors Duane Cooper, Morehouse College; Ricardo Cortez, Tulane University; Ivelisse Rubio, University of Puerto Rico at Río Piedras; and Suzanne Weeks, Worcester Polytechnic Institute contributed significantly towards the organization and design of the program.

Best of luck to the MSRI-UP 2009 students!

Herbert A. Medina
Director, MSRI-UP 2009
Berkeley, CA

Table of Contents

Algebraic structure of (generalized) toric codes

Warren Chancellor
Morehouse College

Jonathon Henry
Cal Poly Pomona

Ellen Lê
Pomona College

July 2009

Abstract

Building on the recent work surrounding toric codes, introduced in 2000 by J. Hansen, we further investigate the properties of this interesting class of error correcting cyclic codes. A toric code C is generated by creating monomials from a set of lattice points P in dimension m, and evaluating each of those monomials over all m-tuples of non-zero elements in a finite field of size q. Just as "ordinary" cyclic codes can be studied via properties of polynomials in one variable, we show that toric codes, which are m-dimensional cyclic codes, can be studied via m-variable polynomials. We aim in our work to generalize explicitly what the algebraic structure is for toric codes. In particular, we give formulas for finding the roots of (generalized) toric codes and their dual codes, and from these roots we derive a formula for an idempotent polynomial that generates the toric code.

1 Introduction

Researchers such as J. Hansen and D. Ruano have considered the properties of generalized toric codes which have given background for future research to be done. In particular, in [DGV] the authors found that all (generalized) toric codes are m-dimensional cyclic. For the case $m = 2$, if a codeword is written as a $(q-1) \times (q-1)$ array then it is closed under *row-wise and column-wise cyclic shifts*. In classical coding theory, a cyclic code is identified by both a one-dimensional generating polynomial and a unique generating idempotent. We find a generating idempotent polynomial in m variables for toric codes which is analogous to the generating polynomial for cyclic codes. For the purpose of this paper, unless otherwise stated, we take $m = 2$ but our results can be generalized for all m.

2 Background

Definition 1. Given a polytope $P \subset \mathbb{R}^2$ each integer lattice point (a, b) in P determines a monomial $x^a y^b$. Each monomial is evaluated at all pairs of non-zero elements in a finite field to produce codewords in a basis for a *toric code C*.

Recall that for a binary cyclic code C of length n, each codeword of length n can be represented as a polynomial $c(x) \bmod x^n + 1$ of degree n. A polynomial I in one variable x is said to be *idempotent* if it satisfies the relation

$$(I(x))^2 \equiv I(x) \mod x^n + 1.$$

There exists a unique idempotent polynomial $I(x)$ in C that generates C.

3 Roots of Toric Codes

We need the following definition.

Definition 2. Consider the evaluation mapping

$$ev : \mathbb{F}_q[x, y] \to (\mathbb{F}_q^*)^{(q-1)^2},$$

defined by $ev(f) = (f(\alpha^i, \alpha^j))_{i=0\ldots q-2, j=0\ldots q-2}$. Each vector $ev(f)$ corresponds to a polynomial

$$F(s,t) = \sum_{i=0}^{q-2} \sum_{j=0}^{q-2} f(\alpha^i, \alpha^j) s^i t^j.$$

For $(a, b) \in \mathbb{Z}^2$, we define $D(a, b)$ as the $F(s, t)$ corresponding to $ev(x^a y^b)$. We say $D(a, b)$ is the *representative polynomial for* (a, b).

Example 1. Let P be the polytope $P = \text{conv}(0, e_1, e_2, e_1+e_2)$, We construct the generator matrix, G, for the corresponding code C_P over $\mathbb{F}_4$ by evaluating the monomials $1, x, y, xy$ over each 2-tuple in $(\mathbb{F}_4^*)^2$.

$$G = \begin{array}{c} \\ 1 \\ x \\ y \\ xy \end{array} \begin{array}{c} \begin{pmatrix}1\\1\end{pmatrix} \begin{pmatrix}\alpha\\1\end{pmatrix} \begin{pmatrix}\alpha^2\\1\end{pmatrix} \begin{pmatrix}1\\\alpha\end{pmatrix} \begin{pmatrix}\alpha\\\alpha\end{pmatrix} \begin{pmatrix}\alpha^2\\\alpha\end{pmatrix} \begin{pmatrix}1\\\alpha^2\end{pmatrix} \begin{pmatrix}\alpha\\\alpha^2\end{pmatrix} \begin{pmatrix}\alpha^2\\\alpha^2\end{pmatrix} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 \\ 1 & 1 & 1 & \alpha & \alpha & \alpha & \alpha^2 & \alpha^2 & \alpha^2 \\ 1 & \alpha & \alpha^2 & \alpha & \alpha^2 & 1 & \alpha^2 & 1 & \alpha \end{pmatrix} \end{array}.$$

The fourth row in G corresponds to

$$D(1,1) = 1 + \alpha s + \alpha^2 s^2 + \alpha t + \alpha^2 st + s^2 t^2 + \alpha^2 t + st^2 + \alpha s^2 t^2.$$

3.1 Relation of $D(a, b)$ and the toric code C_P

Let $\mathfrak{D} = \{D(a, b) \mid (a, b) \in P\}$. This generates the toric code C_P. Note that the roots of C_P are the common zeroes of $\mathfrak{D}$. We can find these roots and a basis for $C_P^\perp$ using the following theorems. We write α for a primitive element of $\mathbb{F}_q$.

Theorem 1. *The roots of $D(a, b)$ are all elements of $(\mathbb{F}_q^*)^2$ except (α^i, α^j) where $(i, j) \equiv (-a, -b) \mod q - 1$.*

Proof. Given that

$$D(a,b) = (1+\alpha^a s+\ldots+\alpha^{a(q-2)}s^{q-2}+\alpha^b t+\ldots+\alpha^{al+bm}s^l t^m+\ldots+\alpha^{(a+b)(q-2)}s^{(q-2)}t^{(q-2)}),$$

we can factor $D(a,b)$ and solve for zero to find the roots.

$$D(a,b) = (1+\alpha^a s+\ldots+\alpha^{al}s^l\alpha^{a(q-2)}s^{q-2})\cdot(1+\alpha^b t+\ldots+\alpha^{b^m}t^m+\ldots+\alpha^{b(q-2)}t^{(q-2))} = 0.$$

Let $u = \alpha^a s$, and $w = \alpha^b t$. Substituting into the equation above gives us

$$(1+u+u^2+\ldots+u^{q-2})(1+w+w^2+\ldots+w^{q-2}) = 0.$$

We know that each $\beta \in \mathbb{F}_q^*$ has the property $\beta^{q-1} = 1$ since $\mathbb{F}_q^*$ is a multiplicative group. Hence, every element of $\mathbb{F}_q^*$ is a solution of the equation $u^{q-1}-1=0$. We know that $u^{q-1}-1$ factors as $(u-1)(u^{q-2}+u^{q-3}+\ldots+u^2+u+1)$. The only root of $u-1$ in $\mathbb{F}_q^*$ is $u=1$. Moreover, $u=1$ is not a root of the second factor, but every other element of $\mathbb{F}_q^*$ is a root of the second factor. Since $u=\alpha^a s$, we see that $u=\beta$ is a root of the second factor except for when $s=(\alpha^a)^{-1}=\alpha^i$ where $i \equiv -a \bmod q-1$.

A similar argument for w yields the result that the roots of $D(a,b)$ are all elements of $(\mathbb{F}_q^*)^2$ except the pair (α^i,α^j), where $(i,j) \equiv (-a,-b) \bmod q-1$. □

Corollary 1. *Let P be a polytope in $\mathbb{R}^2$. A polynomial represents a word in C_P if and only if it has zeroes at all elements of $(\mathbb{F}_q^*)^2$ except (α^i,α^j) where $(i,j) \equiv (-a,-b) \bmod q-1$ for some $(a,b) \in P \cap \mathbb{Z}^2$.*

Proof. This is immediate from Theorem 1. □

3.2 A Basis for the Dual Code

We use Corollary 1 to a identify a basis for the dual code $C_P^\perp$ using the following correspondence. Let (α^i,α^j) be a zero of C_P. Then for some polynomial $c(s,t) = a_0+a_1 s+\ldots+a_{q-2}s^{q-2}+a_{q-1}t+\ldots+a_{(q-2)^2}s^{q-2}t^{q-2}$ in C_P, we have that $c(\alpha^i,\alpha^j)=0$. Thus,

$$\begin{aligned} c(\alpha^i,\alpha^j) &= a_0 + a_1\alpha^i + \ldots + a_{q-2}\alpha^{i(q-2)} + a_{q-1}\alpha^j + \ldots + a_{(q-2)^2}\alpha^{i(q-2)}\alpha^{j(q-2)} \\ &= (a_0, a_1, \ldots, a_{(q-2)^2}) \cdot (1, \alpha^i, \ldots, \alpha^{(i+j)(q-2)}) \\ &= 0. \end{aligned}$$

Hence, the vector $v_{i,j} = (1,\alpha^i,\ldots,\alpha^{(i+j)(q-2)})$ is an element of $C_P^\perp$.

Example 2. Consider the polytope $P = \mathrm{conv}(0,e_1,e_2,e_1+e_2)$ in $\mathbb{R}^2$. The common roots of C_P are $R = \{(\alpha,1),(1,\alpha),(\alpha,\alpha),(\alpha,\alpha^2),(\alpha^2,\alpha)\}$. For each element of R, the correspondence to vectors in $C_P^\perp$ is given as follows.

$$\begin{aligned} (1,\alpha) &\leftrightarrow (1\ 1\ 1\ \alpha\ \alpha\ \alpha\ \alpha^2\ \alpha^2\ \alpha^2) \\ (\alpha,1) &\leftrightarrow (1\ \alpha\ \alpha^2\ 1\ \alpha\ \alpha^2\ 1\ \alpha\ \alpha^2) \\ (\alpha,\alpha) &\leftrightarrow (1\ \alpha\ \alpha^2\ \alpha\ \alpha^2\ 1\ \alpha^2\ 1\ \alpha) \\ (\alpha,\alpha^2) &\leftrightarrow (1\ \alpha\ \alpha^2\ \alpha^2\ 1\ \alpha\ \alpha\ \alpha^2\ 1) \\ (\alpha^2,\alpha) &\leftrightarrow (1\ \alpha^2\ \alpha\ \alpha\ 1\ \alpha^2\ \alpha^2\ \alpha\ 1). \end{aligned}$$

Theorem 2. *Let*

$$R = \{(\alpha^i, \alpha^j) \in (\mathbb{F}_q^*)^2 | (i,j) \not\equiv (-a,-b) \bmod q-1 \text{ for each } (a,b) \in P\}$$

be the common roots of the code C_P. Then the set of vectors $V = \{v_{i,j}\}$ corresponding to the elements of the set R form a basis for the dual code $C_P^\perp$.

Proof. Let S be the matrix shown below, with rows constructed by $ev(x^a y^b)$ for each $(a,b) \in [0, q-2]$ evaluated at all pairs (α^i, α^j) in $(\mathbb{F}_q^*)^2$. We note that S is a generator matrix of the toric code C_P for the polytope $P = [0, q-2]^2$.

$$\begin{array}{c} \\ 1 \\ x \\ x^2 \\ \cdot \\ \cdot \\ y \\ xy \\ \cdot \\ \cdot \\ y^{q-2} \\ \cdot \\ \cdot \\ \cdot \\ x^{q-2}y^{q-2} \end{array} \begin{array}{c} \binom{1}{1} \\ \left(\begin{array}{c} 1 \\ 1 \end{array}\right. \\ \left(\begin{array}{c} \alpha \\ 1 \end{array}\right. \\ \left(\begin{array}{c} \alpha^2 \\ 1 \end{array}\right. \\ \left(\begin{array}{c} \cdot \\ 1 \end{array}\right. \\ \left(\begin{array}{c} \cdot \\ 1 \end{array}\right. \\ \left(\begin{array}{c} 1 \\ 1 \end{array}\right. \\ \left(\begin{array}{c} \alpha \\ 1 \end{array}\right. \\ \left(\begin{array}{c} \cdot \\ 1 \end{array}\right. \\ \left(\begin{array}{c} \cdot \\ 1 \end{array}\right. \\ \left(\begin{array}{c} 1 \\ 1 \end{array}\right. \\ \left(\begin{array}{c} \cdot \\ 1 \end{array}\right. \\ \left(\begin{array}{c} \cdot \\ 1 \end{array}\right. \\ \left(\begin{array}{c} \cdot \\ 1 \end{array}\right. \\ \left(\begin{array}{c} \alpha^{q-2} \\ 1 \end{array}\right. \end{array} \begin{array}{c} \binom{\alpha}{1} \\ 1 \\ \alpha \\ \alpha^2 \\ \cdot \\ \cdot \\ 1 \\ \alpha \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ \alpha^{q-2} \end{array} \begin{array}{c} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{array} \begin{array}{c} \binom{\alpha^{q-2}}{1} \\ 1 \\ \alpha^{q-2} \\ \alpha^{2(q-2)} \\ \cdot \\ \cdot \\ 1 \\ \alpha^{q-2} \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ \alpha^{(q-2)(q-2)} \end{array} \begin{array}{c} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{array} \begin{array}{c} \binom{1}{\alpha} \\ 1 \\ 1 \\ 1 \\ \cdot \\ \cdot \\ \alpha \\ \alpha \\ \cdot \\ \cdot \\ \alpha^{q-2} \\ \cdot \\ \cdot \\ \cdot \\ \alpha^{q-2} \end{array} \begin{array}{c} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{array} \begin{array}{c} \binom{1}{\alpha^{q-2}} \\ 1 \\ 1 \\ 1 \\ \cdot \\ \cdot \\ 1 \\ \alpha^{q-2} \\ \cdot \\ \cdot \\ \alpha^{q-2} \\ \cdot \\ \cdot \\ \cdot \\ \alpha^{(q-2)(q-2)} \end{array} \begin{array}{c} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{array} \begin{array}{c} \binom{\alpha^{q-2}}{\alpha^{q-2}} \\ 1 \\ \alpha^{q-2} \\ \alpha^{2(q-2)} \\ \cdot \\ \cdot \\ \alpha^{q-2} \\ \alpha^{(q-2)+(q-2)} \\ \cdot \\ \cdot \\ \alpha^{q-2} \\ \cdot \\ \cdot \\ \cdot \\ \alpha^{(q-2)(q-2)+(q-2)(q-2)} \end{array}$$

Taking $d = e = q-2$ in Theorem 2 of [DGV], we see that the rows of the matrix S are linearly independent and hence its columns are also linearly independent. We note that each vector in V appears as a column of S, and moreover, there are $n-k$ of them. Since $C_P^\perp$ has dimension $n-k$, we see that V is a basis for $C_P^\perp$. □

4 Generating Idempotents

4.1 Constructing an Idempotent Polynomial

Given an integer lattice point (a,b) in a polytope P in $\mathbb{R}^2$, we define a polynomial function $I_{i,j}(s,t)$ on $(\mathbb{F}_q^*)^2$ which corresponds to (a,b) and has the following property. If $(i,j) \equiv (-a,-b) \bmod q-1$, then

$$I_{i,j}(s,t) = \begin{cases} 1 & \text{for } (\alpha^i, \alpha^j), \\ 0 & \text{otherwise.} \end{cases}$$

Consider the following example.

Example 3. Given the polytope $P = \text{conv}(0, e_1, e_2, e_1 + e_2)$, the lattice points (a, b) correspond to the set

$$N = \{(1, 1), (1, \alpha^2), (\alpha^2, 1), (\alpha^2, \alpha^2)\} \in (\mathbb{F}_4^*)^2.$$

Let $I_{0,2}(s, t)$ be the polynomial

$$I_{0,2}(s, t) = \frac{(s - \alpha)(s - \alpha^2)(t - 1)(t - \alpha)}{(1 - \alpha)(1 - \alpha^2)(\alpha^2 - 1)(\alpha^2 - \alpha)}.$$

Note that each of the s factors in the formula corresponds to excluding the column of points in the field $(\mathbb{F}_4^*)^2$ where $s = \alpha$ and $s = \alpha^2$. Similarly each of the t factors in the formula corresponds to excluding the row of points in the field where $t = 1$ and $t = \alpha$ (see figure 1). Therefore, the only point in the entire field $(\mathbb{F}_4^*)^2$ that is not a zero is $(1, \alpha^2)$. Furthermore, at $(1, \alpha^2)$ the value of $I_{0,2}$ is 1. Similarly, for the other

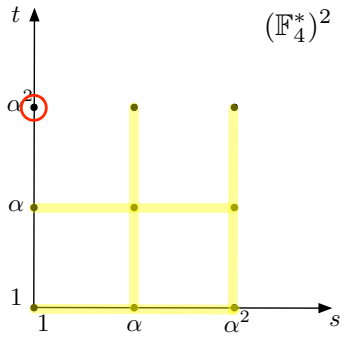


Figure 1: A graphical representation of the equation for $I_{0,2}(s, t)$. The terms $(s - \alpha), (s - \alpha^2), (t - 1), (t - \alpha)$ cover all the zeroes of $I_{0,2}(s, t)$.

elements of N, we have

$$\begin{aligned} I_{0,0}(s, t) &= \frac{(s - \alpha)(s - \alpha^2)(t - \alpha)(t - \alpha^2)}{(1 - \alpha)(1 - \alpha^2)(1 - \alpha)(1 - \alpha^2)} \\ I_{2,0}(s, t) &= \frac{(s - 1)(s - \alpha)(t - \alpha)(t - \alpha^2)}{(\alpha^2 - 1)(\alpha^2 - \alpha)(1 - \alpha)(1 - \alpha^2)} \\ I_{2,2}(s, t) &= \frac{(s - 1)(s - \alpha)(t - 1)(t - \alpha)}{(\alpha^2 - 1)(\alpha^2 - \alpha)(\alpha^2 - 1)(\alpha^2 - \alpha)}. \end{aligned}$$

A general formula for $I_{i,j}(s, t)$, that evaluates to 1 only at the point $(\alpha^i, \alpha^j) \in (\mathbb{F}_q^*)^2$ and evaluates to 0 for all other points in the field is given by $I_{i,j}(s, t) =$

$$\frac{(s - 1) \cdot \ldots \cdot \widehat{(s - \alpha^i)} \cdot \ldots \cdot (s - \alpha^{q-2}) \cdot (t - 1) \cdot \ldots \cdot \widehat{(t - \alpha^j)} \cdot \ldots \cdot (t - \alpha^{q-2})}{(\alpha^i - 1) \cdot \ldots \cdot \widehat{(\alpha^i - \alpha^i)} \cdot \ldots \cdot (\alpha^i - \alpha^{q-2}) \cdot (\alpha^j - 1) \cdot \ldots \cdot \widehat{(\alpha^j - \alpha^j)} \cdot \ldots \cdot (\alpha^j - \alpha^{q-2})}.$$

The following theorem gives us that the polynomial above is idempotent.

Theorem 3. *The polynomial $I_{i,j}(s,t)$ is idempotent.*

Proof. We prove that $I_{i,j}(s,t)$ is idempotent, which is equivalent to showing that $(I_{i,j}(s,t))^2 - I_{i,j}(s,t) \equiv 0 \bmod s^{q-1}-1, t^{q-1}-1$. First divide $(I_{i,j}(s,t))^2 - I_{i,j}(s,t)$ by $s^{q-1}-1$ and $t^{q-1}-1$:

$$(I_{i,j}(s,t))^2 - I_{i,j}(s,t) = q_1(s,t)(s^{q-1}-1) + q_2(s,t)(t^{q-1}-1) + r(s,t),$$

where $q_1(s,t)$, $q_2(s,t)$ are quotients and $r(s,t)$ is the remainder polynomial. Let $(\alpha^i, \alpha^j) \in (\mathbb{F}_q^*)^2$ and substitute for (s,t) in the equation above. Since we know that $I_{i,j}(s,t)$ evaluated for a point $(\mathbb{F}_q^*)^2$ is either 0 or 1 from the definition of $I_{i,j}(s,t)$, we know that $(I_{i,j}(s,t))^2 - I_{i,j}(s,t) = 0$. So we have

$$\begin{aligned} 0 &= q_1(\alpha^i,\alpha^j)((\alpha^i)^{q-1}-1) + q_2(\alpha^i,\alpha^j)((\alpha^j)^{q-1}-1) + r(\alpha^i,\alpha^j) \\ &= q_1(\alpha^i,\alpha^j)(1-1) + q_2(\alpha^i,\alpha^j)(1-1) + r(\alpha^i,\alpha^j) \\ &= 0 + 0 + r(\alpha^i,\alpha^j) \\ &= r(\alpha^i,\alpha^j), \end{aligned}$$

since we know that for any element $\beta \in \mathbb{F}_q^*$, $\beta^{q-1} = 1$. We have that the polynomial $r(s,t)$ is 0. Assume otherwise, that it is a non-zero remainder polynomial. We know that the degree of $r(s,t)$ in s,t must be strictly less than $q-1$ since $r(s,t)$ is the remainder polynomial after dividing by $(s^{q-1}-1)$ and $(t^{q-1}-1)$. Let s be some fixed $\alpha^i \in \mathbb{F}_q^*$ and let $\overline{r(t)} = r(\alpha^i, t)$ which has degree less than $q-1$. But $\overline{r(t)}$ has $q-1$ roots since by above we have that $\overline{r(\alpha^j)} = r(\alpha^i,\alpha^j) = 0$ for all $j = 0,\ldots,q-2$. So this is a contradiction and we have that $(I_{i,j}(s,t))^2 - I_{i,j}(s,t) \equiv 0 \bmod s^{q-1}-1, t^{q-1}-1$. □

Using the idempotent polynomial $I_{i,j}(s,t)$, we can construct a *generating idempotent* $I(s,t)$ for the toric code C_P. Define $I(s,t)$ by the following equation.

$$I(s,t) = \sum_{(a,b)\in P} I_{i,j}(s,t). \tag{1}$$

If q is a power of 2, it is clear that $I(s,t)$ is an idempotent since the sum of idempotents is idempotent. More generally, for any q, $I(s,t)$ is idempotent since the $I_{i,j}(s,t)$ are *pairwise orthogonal* in the sense that $I_{i,j}(s,t)I_{i',j'}(s,t) \equiv 0 \bmod s^{q-1}-1, t^{q-1}-1$ if $(i,j) \neq (i',j')$. We also know that $I(s,t)$ corresponds to a word in C_P by Corollary 1 since it contains the zeroes of C_P.

4.2 How the generating idempotent generates the code

In this section we prove that the idempotent $I(s,t)$ as defined above generates the code C_P. We begin with the following lemma.

Lemma 1. *Let $F(s,t)$ be an idempotent and $p(s,t)$ be an element of $\mathbb{F}_q[s,t]$. If $u(s,t) \equiv F(s,t)\cdot p(s,t) \bmod s^{q-1}-1, t^{q-1}-1$, then $u(a,b) = p(a,b)$ where $(a,b) \in (\mathbb{F}_q^*)^2$ is not a root of $F(s,t)$.*

Proof. Let $F(s,t)$ be an idempotent and $p(s,t)$ be any polynomial in $\mathbb{F}_q[s,t]$. Suppose $u(s,t) \equiv F(s,t) \cdot p(s,t) \bmod s^{q-1}-1, t^{q-1}-1$. Then

$$F(s,t) \cdot p(s,t) = (s^{q-1}-1) \cdot Q_1(s,t) + r_1(s,t),$$

for some $r_1, Q_1 \in \mathbb{F}_q[s,t]$, and

$$r_1(s,t) = (t^{q-1}-1) \cdot Q_2(s,t) + u(s,t),$$

for some $Q_2 \in \mathbb{F}_q[s,t]$. By substitution we obtain

$$F(s,t) \cdot p(s,t) = (s^{q-1}-1) \cdot Q_1(s,t) + (t^{q-1}-1) \cdot Q_2(s,t) + u(s,t).$$

Suppose $(a,b) \in (\mathbb{F}_q^*)^2$ is not a root of $F(s,t)$. Then,

$$\begin{aligned} F(a,b) \cdot p(a,b) &= (a^{q-1}-1) \cdot Q_1(a,b) + (b^{q-1}-1) \cdot Q_2(a,b) + u(a,b) \\ 1 \cdot p(a,b) &= (1-1) \cdot Q_1(a,b) + (1-1) \cdot Q_2(a,b) + u(a,b) \\ p(a,b) &= u(a,b). \end{aligned}$$

This gives the desired equality. □

Theorem 4. *Given a polytope $P \in \mathbb{R}^2$ and the field $\mathbb{F}_q$, $I(s,t)$ from (1) generates the toric code C_P.*

Proof. We show that there exist k linearly independent codewords $u_i(s,t)$ of the form $u_i(s,t) \equiv I(s,t) \cdot p_i(s,t) \bmod s^{q-1}-1, t^{q-1}-1$ for some $p_i(s,t) \in \mathbb{F}_q[s,t]$. Let $(a_1,b_1), \ldots, (a_k,b_k)$ be the integer lattice points of the polytope P in $\mathbb{R}^2$, and let $I_{i_l,j_l}(s,t)$ be the idempotent that corresponds to the point (a_l,b_l) using the construction in Section 4.1. Define $u_l(s,t) \equiv I(s,t) \cdot I_{i_l,j_l}(s,t) \bmod s^{q-1}-1, t^{q-1}-1$, where $I(s,t) = \sum_{(a_l,b_l) \in P} I_{i_l,j_l}(s,t)$. By Lemma 1, we know that $u_l(\alpha^{i_l}, \alpha^{j_l}) = I_{i_l,j_l}(\alpha^{i_l}, \alpha^{j_l})$, since $(\alpha^{i_l}, \alpha^{j_l})$ is not a root of $I(s,t)$.

Now suppose that

$$c_1 u_1(s,t) + c_2 u_2(s,t) + \cdots + c_k u_k(s,t) = 0$$

for $c_i \in \mathbb{F}_q$. If we evaluate the equation above at a given $(\alpha^{i_l}, \alpha^{j_l})$, we see that all terms are zero except for the l-th term. Therefore, the equation becomes

$$c_l u_l(\alpha^{i_l}, \alpha^{j_l}) = 0.$$

This implies that $c_l = 0$ since $u_l(\alpha^{i_l}, \alpha^{j_l}) = 1$. Thus, as l goes from 1 to k, we obtain that $c_l = 0$ for each $1 \leq l \leq k$ proving the linear independence of the polynomials $u_1(s,t), \ldots, u_k(s,t)$. Moveover, each u_l represents a codeword by Corollary 1. □

Example 4. We construct an idempotent $I(s,t)$ that generates the toric code C_P corresponding to the square polytope P with the following lattice points: (0,0), (0,1),

(1,0), (1,1) and evaluated over the field $\mathbb{F}_4$. We have the following correspondence between points (a, b) in P and idempotents $I_{i,j}$ given in Example 3.

$$(0,0) \leftrightarrow I_{0,0}(s,t)$$
$$(1,0) \leftrightarrow I_{2,0}(s,t)$$
$$(0,1) \leftrightarrow I_{0,2}(s,t)$$
$$(1,1) \leftrightarrow I_{2,2}(s,t)$$

Let $I(s,t) = \sum_{(a,b)\in P} I_{i,j}(s,t)$ and put

$$\begin{aligned} u_1(s,t) &\equiv I(s,t) \cdot I_{0,0}(s,t) \bmod s^{q-1}-1, t^{q-1}-1 \\ u_2(s,t) &\equiv I(s,t) \cdot I_{2,0}(s,t) \bmod s^{q-1}-1, t^{q-1}-1 \\ u_3(s,t) &\equiv I(s,t) \cdot I_{0,2}(s,t) \bmod s^{q-1}-1, t^{q-1}-1 \\ u_4(s,t) &\equiv I(s,t) \cdot I_{2,2}(s,t) \bmod s^{q-1}-1, t^{q-1}-1. \end{aligned}$$

Suppose that

$$c_1u_1(s,t) + c_2u_2(s,t) + c_3u_3(s,t) + c_4u_4(s,t) = 0$$

for some $c_i \in \mathbb{F}_4$. Then if we evaluate at the point $(1,1)$, we notice that as a consequence of Lemma 1, $u_1(1,1) = 1$, but $u_i(1,1) = 0$ for all $i \neq 1$. Thus, the equation becomes

$$c_1u_1(1,1) = 0.$$

This implies that c_1 must be zero. Similarly, for other carefully chosen values of $(\mathbb{F}_q^*)^2$, we obtain that each $c_i = 0$. Hence $\{u_i(s,t)\}_{i=1}^4$ is a set of linearly independent codewords in C_P. Since the dimension of C_P is 4, we have that $I(s,t)$ generates the code C_P.

5 Acknowledgments

This work was conducted during the 2009 Mathematical Sciences Research Institute Undergraduate Program (MSRI-UP) in Berkeley, CA. MSRI-UP is supported by the National Science Foundation (grant No. DMS-0754872) and the National Security Agency (grant No. H98230-09-0103). We would like to thank Dr. John B. Little, Dr. Herbert Medina, Dr. Emille Davie. Also special thanks to Candice Price, Ashley Wheeler, and the entire MSRI staff for their continued support throughout the MSRI-UP program.

References

[DGV] V. Diaz, C. Guevera, M. Vath, "Codes from n-Dimensional Cyclic Codes," *Proceedings of Summer Institute in Mathematics for Undergraduates (SIMU), 2001*, Humacao, Puerto Rico.

A Systematic Census of Generalized Toric Codes over $\mathbb{F}_4$, $\mathbb{F}_5$ and $\mathbb{F}_{16}$

James E. Amaya
The College of New Jersey

April J. Harry
Xavier University of Louisiana

Brian M. Vega
California State Polytechnic University, Pomona

July 2009

Abstract

Toric codes are a specific class of linear codes, originally introduced by J. Hansen [2]. In this report, we study generalized toric codes, which are generated by a set of points in $\mathbb{Z}_{q-1}^m$. The orbits of these sets of points determine equivalent codes. In order to find and distinguish the codes for a given blocklength and dimension, we used various Magma processes to compute minimum distances and weight distributions of codes. There is an online table that contains much of the existing knowledge about the minimum distances of linear codes with certain dimensions at `http://www.codetables.de`. In our analysis of codes, we sought to find codes over $\mathbb{F}_4$ and $\mathbb{F}_5$ that would have minimum distances that exceed the lower bound listed in the online table, and thus would be the best known codes in existence for given parameters. In the process, we also noticed an interesting property about the average weights of words in toric codes and found codes from $\mathbb{F}_5$ and $\mathbb{F}_{16}$ that we believe to be interesting.

1 Introduction

Toric codes were introduced in 1998 by J. Hansen [2], and are a generalization of the often-studied Reed-Solomon codes. We use some geometric concepts to explain how to construct a toric code.

Suppose that P is properly contained in the box $[0, q-2]^m$, denoted

$$\square_{q-1},$$

where q is a power of some prime. We define toric codes as follows:

Definition 1. Let $\mathbb{F}_q$ be a finite field with primitive element α. For $f \in \mathbb{Z}^m$ with $0 \leq f_i \leq q-2$ for all i, let $p_f = (\alpha^{f_1}, \ldots, \alpha^{f_m}) \in (\mathbb{F}_q^*)^m$. For any $e = (e_1, \ldots, e_m) \in P$, let x^e be the corresponding monomial and write

$$(p_f)^e = (\alpha^{f_1})^{e_1} \cdots (\alpha^{f_m})^{e_m}.$$

The *toric code* $C_p(\mathbb{F}_q)$ over the field $\mathbb{F}_q$ associated to P is the linear code of block length $n = (q-1)^m$ with generator matrix

$$G = ((p_f)^e),$$

where the rows are indexed by the $e \in P$, and the columns are indexed by the $p_f \in (\mathbb{F}_q^*)^m$.

Example 1. Let $P \subset \mathbb{R}^3$ be the vertices of the unit tetrahedron

$$P = \{(0,0,0), (1,0,0), (0,1,0), (0,0,1)\}.$$

We will find the toric code $C_P(\mathbb{F}_4)$. Begin by using the coordinates of each lattice point in P as exponents of x, y, and z: $1,\ x,\ y,\ z$. Let α be the primitive element in $\mathbb{F}_4$. Form ordered 3-tuples of the elements of $\mathbb{F}_4^* = \{1, \alpha, \alpha^2\}$:

$$(1,1,1), (1,1,\alpha), (1,1,\alpha^2), (1,\alpha,1), (1,\alpha,\alpha), (1,\alpha,\alpha^2), (1,\alpha^2,\alpha), (1,\alpha^2,\alpha^2), (\alpha,1,1),$$
$$(\alpha,1,\alpha), (\alpha,1,\alpha^2), (\alpha,\alpha,1), (\alpha,\alpha,\alpha), (\alpha,\alpha,\alpha^2), (\alpha,\alpha^2,\alpha), (\alpha,\alpha^2,\alpha^2), (\alpha^2,1,1),$$
$$(\alpha^2,1,\alpha), (\alpha^2,1,\alpha^2), (\alpha^2,\alpha,1), (\alpha^2,\alpha,\alpha), (\alpha^2,\alpha,\alpha^2), (\alpha^2,\alpha^2,\alpha), (\alpha^2,\alpha^2,\alpha^2).$$

Finally, we evaluate each monomial over the set of all 3-tuples to generate each row of a 4×27 matrix:

$$\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \alpha & \alpha & \alpha & \alpha & \alpha & \alpha & \alpha & \alpha & \alpha & \alpha^2 & \alpha^2 & \alpha^2 & \alpha^2 & \alpha^2 & \alpha^2 & \alpha^2 & \alpha^2 & \alpha^2 \\
1 & 1 & 1 & \alpha & \alpha & \alpha & \alpha^2 & \alpha^2 & \alpha^2 & 1 & 1 & 1 & \alpha & \alpha & \alpha & \alpha^2 & \alpha^2 & \alpha^2 & 1 & 1 & 1 & \alpha & \alpha & \alpha & \alpha^2 & \alpha^2 & \alpha^2 \\
1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2
\end{bmatrix}.$$

We now give some necessary background.

2 Background

In this section we establish a notion of equivalence for toric codes and the machinery needed to categorize them. First we need the following definitions:

Definition 2. An *invertible affine mapping* or *affine transformation* $T : \mathbb{Z}^m \to \mathbb{Z}^m$ has the form

$$T(x) = Ax + v$$

where A is an invertible integer matrix whose inverse is also an integer matrix, and v is an integer vector.

We will also need similar mappings from $\mathbb{Z}_{q-1}^m$ to itself.

Definition 3. The group of invertible affine mappings $T : \mathbb{Z}_{q-1}^m \to \mathbb{Z}_{q-1}^m$, denoted $AGL(m, \mathbb{Z}_{q-1})$ is the group of affine transformations

$$T(x) = Ax + v$$

where A is an $m \times m$ invertible matrix, and both A and v have entries from $\mathbb{Z}_{q-1}$.

These types of affine transformations can be thought of as permuting the integer lattice points in $\square_{q-1}$.

Definition 4. Let two toric codes C_1 and C_2 over $\mathbb{F}_q$ have generator matrices G_1 and G_2, respectively. If there exist a permutation matrix R and a diagonal matrix D with entries from $\mathbb{F}_{q-1}^*$ such that

$$G_1RD = G_2,$$

then C_1 and C_2 are said to be *monomially equivalent*.

We have the following consequences:

Theorem 1. *Two codes that are monomially equivalent have the same blocklength, dimension and weight distribution.*

Thus monomially equivalent codes are not considered distinct.

Definition 5. Let P and Q be subsets of $\mathbb{Z}_{q-1}^m$. If there exists an invertible affine transformation $T(x)$ in $AGL(m, \mathbb{Z}_{q-1})$ such that $T(P) = Q$, then P is $AGL(m, \mathbb{Z}_{q-1})$-*equivalent* to Q.

Theorem 2. *If P is $AGL(m, \mathbb{Z}_{q-1})$-equivalent to Q in $\mathbb{Z}_{q-1}^m$, then $C_P(\mathbb{F}_q)$ is monomially equivalent to $C_Q(\mathbb{F}_q)$.*

It follows immediately that the existence of an invertible affine transformation in $AGL(m, \mathbb{Z}_{q-1})$ between two sets of points in $\mathbb{Z}_{q-1}^m$ implies that their associated toric codes are equivalent.

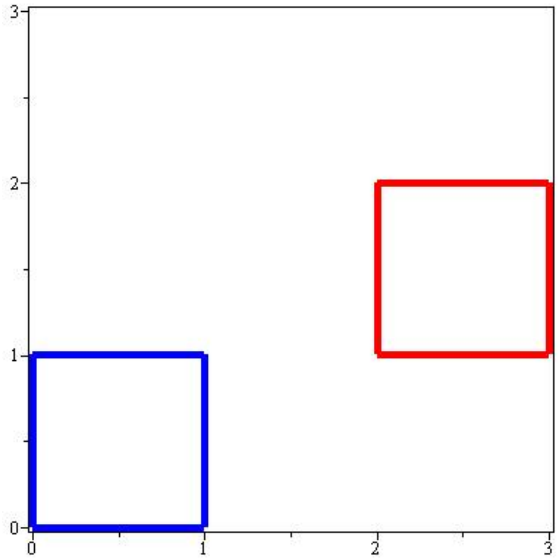


Figure 1: Transformation from Example 2

Example 2. Consider the vertices of the unit square. Let $P = \{(0,0), (1,0), (0,1), (1,1)\} \subset \mathbb{Z}_4^2$, and let $T(x) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 2 \\ 1 \end{bmatrix}$. So $T(P) = \{(2,1), (3,1), (2,2), (3,2)\}$

is the set of vertices of a translated unit square. It follows from the theorem that $C_P(\mathbb{F}_5)$ is monomially equivalent to $C_{T(P)}(\mathbb{F}_5)$. See Figure 1.

For a given k, there are $\binom{(q-1)^m}{k}$ subsets of $\mathbb{Z}_{q-1}^m$ having k elements. Any of these subsets could be used to construct a toric code of dimension k, so there are upwards of $\binom{(q-1)^m}{k}$ such codes. However, any two of these sets of points that are related by an affine mapping in $AGL(m, \mathbb{Z}_{q-1})$ will yield equivalent codes. The question arises as to how many distinct codes exist.

Definition 6. Let S be the collection of all k-element subsets of $\mathbb{Z}_{q-1}^m$. An *orbit* is a subset of S such that for all $v, w \in S$, there exists an affine mapping $T(x)$ in $AGL(m, \mathbb{Z}_{q-1})$ with $T(v) = w$.

The construction of orbits will partition the set S. Moreover, all elements in a given orbit yield toric codes that are monomially equivalent. However, this is not to say that two different orbits cannot yield toric codes that are monomially equivalent.

Example 3. Consider the toric codes, $C_{P_1}(\mathbb{F}_5)$ and $C_{P_2}(\mathbb{F}_5)$ constructed using the sets of lattice points $P_1 = \{(0,0),(0,2),(2,0)\}$ and $P_2 = \{(0,0),(1,0),(2,0)\}$ over the field $\mathbb{F}_5$. P_1 and P_2 are in different orbits, which means that they are not $AGL(2, \mathbb{Z}_4)$-equivalent. However, $C_{P_1}(\mathbb{F}_5)$ and $C_{P_2}(\mathbb{F}_5)$ have the same weight enumerator, $x^{16} + 24x^8y^8 + 48x^4y^{12} + 52y^{16}$. This means that the two toric codes can be monomially equivalent. It suffices to show that if it is possible to make the generators of the codes equal by permuting columns, scaling columns, and performing row operations then $C_{P_1}(\mathbb{F}_5)$ is monomially equivalent to $C_{P_2}(\mathbb{F}_5)$. Note: a simplified notation can be used since the columns of the matrix can be divided evenly by four, where each block of columns has the same entry in each row.

$$G_1 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 4 & 4 & 0 & 0 & 4 & 4 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{Row}_1 + \text{Row}_2 \rightarrow$$

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

$$\text{Permute Col}_1 \text{ to Col}_{14},\ \text{Col}_4 \text{ to Col}_{14},\ \text{Col}_6 \text{ to Col}_9,\ \text{Col}_7 \text{ to Col}_{12} \rightarrow$$

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

$$\text{Simple Notation} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 3 & 3 & 3 & 3 \end{bmatrix}$$

$$\text{Simple Notation} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix} \text{Row}_2 + \text{Row}_3 \rightarrow \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

$$2\text{Row}_1 \rightarrow \begin{bmatrix} 2 & 0 & 0 & 2 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 3 \end{bmatrix} \text{Row}_3 + \text{Row}_1 \rightarrow \begin{bmatrix} 2 & 0 & 0 & 2 \\ 0 & 1 & 1 & 0 \\ 2 & 0 & 1 & 0 \end{bmatrix}$$

$$3\text{Col}_1, 3\text{Col}_3 \rightarrow \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \text{Swap Row}_2 \text{ and Row}_3 \rightarrow \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

This shows that $C_{P_1}(\mathbb{F}_5)$ and $C_{P_2}(\mathbb{F}_5)$ are monomially equivalent.

If we can find a representative element for each orbit then we would, essentially, have all the codes. In [1], Cameron refers to what is known as the cycle index polynomial, a very useful tool for computing the number of orbits.

Definition 7. Let G be a permutation group on a set Ω, where $|\Omega| = n$. For each element $g \in G$, we can decompose the permutation g into a product of disjoint cycles; let $c_i(g)$ be the number of i-cycles occurring in this decomposition. Now the *cycle index* of G is the polynomial $Z(G)$ in indeterminates $s_1, \ldots, s_n$ given by

$$Z(G) = \frac{1}{|G|} \sum_{g \in G} s_1^{c_1(g)} \cdots s_n^{c_n(g)}.$$

To find the generating function for the sequence giving the number of orbits of $AGL(m, \mathbb{Z}_{q-1})$ acting on subsets of all sizes $0 \leq k \leq (q-1)^m$, we use a version of this cycle index polynomial. In our version, we substitute $t^n + 1$ for each s_n. From this substitution we get

$$Q(G) = \sum_{k=0}^{(q-1)^m} B_k t^k.$$

Here B_k is the number of orbits which partition the set of order-k subsets of $\mathbb{Z}_{q-1}^m$.

Example 4. We will compute the cycle index polynomial for $m = 2$ over $\mathbb{F}_4$. From our *Perm*2 Magma procedure [see appendix] we get:

```
Permutation group acting on a set of cardinality 9
(2, 3)(5, 6)(8, 9)
(4, 5, 6)(7, 9, 8)
(2, 7, 3, 4)(5, 8, 9, 6)
(1, 2, 3)(4, 5, 6)(7, 8, 9)
(1, 4, 7)(2, 5, 8)(3, 6, 9)
```

We assign this permutation group to the variable G and use Magma to find its conjugacy classes:

```
Classes(G);

Conjugacy Classes of group G

---------------------------

[1]     Order 1       Length 1
        Rep Id(G)
[2]     Order 2       Length 9
        Rep (2, 3)(4, 7)(5, 9)(6, 8)
[3]     Order 2       Length 36
        Rep (2, 3)(5, 6)(8, 9)
[4]     Order 3       Length 8
        Rep (1, 2, 3)(4, 5, 6)(7, 8, 9)
[5]     Order 3       Length 24
        Rep (1, 9, 5)(2, 6, 7)
[6]     Order 3       Length 48
        Rep (1, 6, 7)(2, 4, 8)(3, 5, 9)
[7]     Order 4       Length 54
        Rep (2, 7, 3, 4)(5, 8, 9, 6)
[8]     Order 6       Length 72
        Rep (1, 4, 7)(2, 6, 8, 3, 5, 9)
[9]     Order 6       Length 72
        Rep (1, 6, 9, 7, 5, 2)(3, 8)
[10]    Order 8       Length 54
        Rep (1, 6, 7, 4, 5, 3, 8, 2)
[11]    Order 8       Length 54
        Rep (1, 3, 7, 2, 5, 6, 8, 4)
```

We form Cameron's cycle index from this information:

$$\begin{aligned} Z(G) &= \frac{1}{432}(s_1^9 + 9s_2^4 s_1 + 36s_2^3 s_1^3 + 8s_3^3 + 24s_3^2 s_1^3 + 48s_3^3 + \\ &\qquad 54s_4^2 s_1 + 72s_3 s_6 + 72s_6 s_2 s_1 + 54s_8 s_1 + 54s_8 s_1). \end{aligned}$$

We make the necessary substitutions and simplify:

$$Q(G) = t^9 + t^8 + t^7 + 2t^6 + 2t^5 + 2t^4 + 2t^3 + t^2 + t + 1.$$

From the Indexing Polynomial we know the information in Table 2.

3 Methods

As seen in the introduction, building the generator matrix by hand for a toric code is time consuming, and finding all the codewords is even more so. Also, creating the cycle index polynomial for larger Galois Fields requires a large amount of computation.

k	Number of Orbits
9	1
8	1
7	1
6	2
5	2
4	2
3	2
2	1
1	1
0	1

Table 1: Orbits of a given k with $m = 2$ over $\mathbb{F}_4$

In order to have a more efficient method of creating generalized toric codes, creating the cycle index polynomial, and generating representative toric codes from different orbits, we used a powerful software package called Magma.

3.1 Procedures for Toric Codes in Magma

Much of our research was made possible by David Joyner. He created procedures in Magma specifically for toric codes. His procedure, *toric_code*, created a toric code when a list of lattice points and a Galois Field was inputted. However, from there we created procedures which created the cycle index polynomial and generated representative toric codes from different orbits. The former is straightforward, while the latter can be approached several ways. Here is the main list of procedures:

Note: q =size of Galois Field, m =spatial dimension, k =dimension of toric code, N =number of random toric codes.

1. $CycleIndexPoly$ - Given q and m, return the cycle index polynomial.

2. $CodesForK$ - Given q, m, and k, return a representative toric code from each orbit, along with its minimum distance.

3. *random_toric_code_search* - Given k, N, and q, return a representative toric code from several orbits, not necessarily all, and its minimum distance.

4. *Stab* - Given a k, q, m, return a list of lists of lattice points which have a relatively large stabilizer.

5. $FixedAndRandom$ - Given a list of points, k, N, and q, return representative toric codes from "useful" orbits and their minimum distances.

3.2 Using Procedures

Once these procedures were ready for use, our research was underway. Our first procedure, $CycleIndexPoly$, was mostly used as a reference for the other procedures.

It shows us the number of representative toric codes we would need to find with a given dimension over a Galois Field of size q in m-space. For example, the cycle index polynomial for toric codes over the Galois Field of size 4 in 2-space is:

$$CycleIndexPoly(4,2) = t^9 + t^8 + t^7 + 2t^6 + 2t^5 + 2t^4 + 2t^3 + t^2 + t + 1.$$

This tells us that we need to find 2 representative toric codes with dimension 3 over the Galois Field of size 4 in 2-space.

Listed are the three methods used for finding representative toric codes:

Method 1: Use $CodesForK$, which was the ideal way of finding representative toric codes from each orbit and their minimum distances. However, due to the limitations on Magma, we could only use this procedure to find toric codes with a small block length and dimension. This is because $CodesForK$ finds codes by first finding all the orbits, and then choosing points from each orbit to create a representative toric code for each orbit. As k increased, the number of orbits grew too large for Magma to compute. Consequently, all the toric codes we were able to find with $CodesForK$ were already known.

Method 2: Use *random_toric_code_search*, programmed by John Little. This procedure is seen as an optimization of $CodesForK$. This was true except for the fact that it is hard for *random_toric_code_search* to find representative toric codes if it comes from a small orbit. We realized that if the lattice points used to construct a toric code have a big stabilizer, then the orbit will be small. Thus, we programmed *Stab*. However, we were once again prevented from running the procedure due to the limitations on Magma. This is because although Magma has a built-in stabilizer function, the permutation group that is isomorphic to the AGL, its GSet, and the element whose stabilizer is desired must also be input. The GSet is the set which the permutation group acts on. In order to make the GSet, $\binom{n}{k}$ subsets must be made, where $n = (q-1)^m$ is the blocklength of all toric codes over $\mathbb{F}_q$. For example, when $q = 8$, $m = 3$, and $k = 8$, GSet requires $\binom{64}{8} = 4426165368$ subsets, which Magma cannot compute in a reasonable amount of time. As a result, all the toric codes found with *random_toric_code_search* were already known.

Method 3: Use $FixedAndRandom$, which fixes the points inputted in the list and randomly adds more points to create a toric code of a certain dimension. This was used to find toric codes with large minimum distances, which in a sense makes the orbits more useful. For example, when fixing $\{(0,0,0),(0,0,1),(0,1,0),(1,0,0)\}$, $FixedAndRandom$ will find codes with a larger minimum distance than if the points $\{(0,0,0),(0,2,0),(0,1,0),(1,0,0)\}$ were used.

4 Main results

In some cases, we were able to find codes with minimum distances that equaled the existing lower bound for minimum distance for corresponding parameters. For $m = 2,\ q = 5,\ k = 6$, we found five codes which met the bound indicated by the online table. This raised the question of how to determine what the best code is in a group of codes that have the same minimum distance. We decided that taking an

average based on the weight distribution for each code would say something about the code's value. We call this average $HAVweight$, defined by

$$HAVweight = \sum_{i=d}^{n} \frac{iA_i}{q^k}$$

where the code is over $\mathbb{F}_q$, n is the blocklength, d is the minimum distance, and A_i is the number of words of weight i. For codes with the same q and m but different k, the average weight was still the same, and we conjecture that

$$HAVweight = \frac{(q-1)^{(m+1)}}{q}.$$

This agrees with part of the proof of the Plotkin bound in [3].

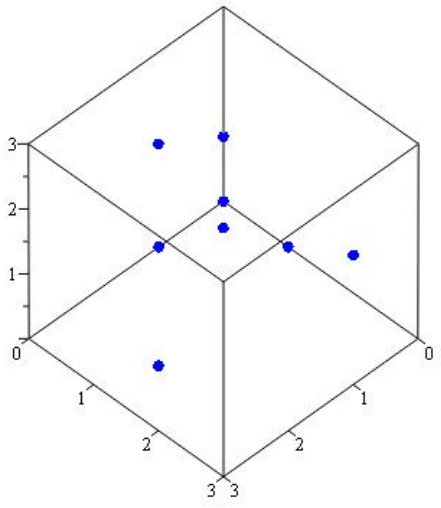


Figure 2: Points that generate code with $d = 42$

The online table at `http://www.codetables.de` contains information about the theoretical minimum distances for many n and k. Along with each theoretical upper bound is the largest minimum distance for a known code. In many cases, this known minimum distance meets the upper bound. However, there are some parameters for which the bound has not been met. Our method of combining fixed and random points led to the discovery of a code whose minimum distance fell within the bounds. The online code states that the largest known minimum distance for a code over $\mathbb{F}_5$ with $k = 8$ and $n = 64$ is 41, with an upper bound of 46. However, we were able to find that the code generated by the points

$$\{(0,0,0),(1,0,0),(0,1,0),(0,0,1),(1,1,1),(2,1,3),(1,3,2),(3,2,1)\}$$

has a minimum distance of 42. We also computed the weight distribution of this code (shown in Table 2). There are over 14,000 codewords with weight 58, which is impressive for a code with blocklength 64.

Weight (w)	Number of Words of Weight w
0	1
42	3840
43	2048
45	14336
46	1792
47	7168
48	57792
49	64512
50	14336
51	14336
52	53760
53	57344
54	50176
55	28672
56	1792
57	4352
58	14336
64	32

Table 2: Weight Distribution for Code with $d = 42$

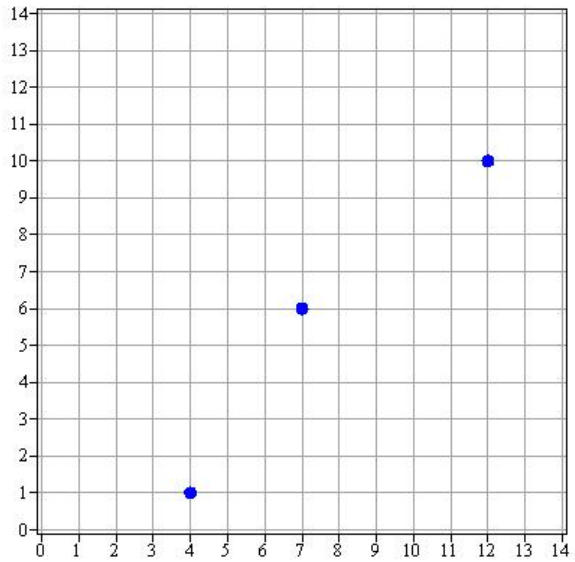


Figure 3: Points that generate code with $d = 210$

Though `http://www.codetables.de` does not offer any information about codes over $\mathbb{F}_{16}$, we were able to infer that one particular code over this field had a minimum distance worth noting. The code formed by the points

$$\{(7, 6), (12, 10), (4, 1)\}$$

with $m = 2$ and $k = 3$ has minimum distance 210. This is considerable, since the blocklength is 225. The weight distribution for this code is in Table 3.

Weight (w)	Number of Words of Weight w
0	1
210	675
211	3375
225	45

Table 3: Weight Distribution for Code with $d = 210$

5 Future Research

Though we were fortunate enough to improve an existing lower bound for a class of codes, the maximum theoretical minimum distance has not yet been reached. The same is true for other blocklengths and dimensions. We hope that improvements in the methods for searching for codes can fix this. Also, our progress was often held back by the limited number of Magma calculations that could be done within a reasonable time frame. A solution to this issue would mean that a census of codes could be extended to much larger parameters. We have hypothesized that the reason that codes with the largest possible minimum distances have not been found is because they may be in very small orbits that would be hard to find using random search. This theory could be confirmed if there was an efficient way of finding stabilizers for toric codes.

6 Acknowledgments

This work was conducted during the 2009 Mathematical Sciences Research Institute Undergraduate Program (MSRI-UP) in Berkeley, CA. MSRI-UP is supported by the National Science Foundation (grant No. DMS-0754872) and the National Security Agency (grant No. H98230-09-0103). We would like to thank the Mathematical Sciences Research Institute, our program director, Dr. Herbert Medina, our research leader, Dr. John Little, our group's graduate assistant, Ashley Wheeler, and all the other participants in MSRI-UP 2009.

References

[1] P. Cameron, *Polynomial aspects of codes, matroids and permutation groups*, notes available online at `http://www.maths.qmw.ac.uk/~pjc/csgnotes/cmpgpoly.pdf`.

[2] J. Hansen, "Toric surfaces and error-correcting codes," in *Coding Theory, cryptography and related areas (Guanajuanto, 1998)*, 132–142, Springer, Berlin, 2000.

[3] M. Plotkin, "Binary codes with specified minimum distance," *IRE Transactions on Information Theory*, 445–450, 1960.

7 Appendix

7.1 Other Results

$\mathbb{F}_q$	n	k	d	Known d
$\mathbb{F}_4$	27	7	9	15
		8	9	14
		9	9	13
		10	9	12
		11	9	12
		16	6	7
$\mathbb{F}_5$	16	5	8	9
		6	8	8
	64	5	43	48
		6	43	45
		7	43	44
		8	**42**	41
		9	32	40
		10	32	38
		11	32	36
		12	32	36
$\mathbb{F}_{16}$	225	3	**210**	-

Table 4: Best Minimum Distances Found vs. Known Minimum Distances with possible improvement

7.2 Magma Procedures

```
CyclePoly2:=function(q,m);
\\Finds the cycle index polynomial for toric codes over GF(q)
\\in 2-space. Only works for m=2.
P<t>:=PolynomialRing(Rationals());
num5:=(q-1)^m;
G:=GL(m,Integers(q-1));
G2:=GL(m+1,Integers(q-1));
num:=[1..#Generators(G)];
id:=Id(G);
id2:=Id(G2);
list:=[];
for i in num do
```

```
InsertBlock(~id2,G.i,1,1);
Append(~list,id2);
end for;
num2:=[1..m];
for j in num2 do
id22:=Id(G2);
InsertBlock(~id22,id,1,1);
id22[j,m+1]:=1;
Append(~list,id22);
end for;
l:=[];
for i in [0..q-2] do
for j in [0..q-2] do
M:=Matrix(Integers(q-1),m+1,1,[j,i,1]);
Append(~l,M);
end for;
end for;
blist:=[];
for j in [1..#list] do
brianslist:=[];
for i in [1..#l] do
flubb:=list[j]*l[i];
Append(~brianslist,flubb);
end for;
Append(~blist,brianslist);
end for;
brianslist3:=[];
for i in [1..#list] do
brianslist2:=[];
for j in [1..#l] do
newb:=ChangeRing(blist[i][j],Integers());
Append(~brianslist2,newb);
end for;
Append(~brianslist3,brianslist2);
end for;
unperm:=function(M);
pp:=M[1][1]+4*M[2][1]+1;
return pp;
end function;
unbrian:=[];
for i in [1..#list] do
unbri:=[];
for j in [1..#l] do
pp:=brianslist3[i][j][1][1]+(q-1)*brianslist3[i][j][2][1]+1;
Append(~unbri,pp);
```

```
end for;
Append(~unbrian,unbri);
end for;
G3:=PermutationGroup<num5|unbrian>;
class:=Classes(G3);
range:=[1..#class];
S:=[];
lengths:=[];
for len in range do
Append(~lengths,class[len][2]);
end for;
for i in range do
x:=CycleDecomposition(class[i][3]);
Append(~S,x);
end for;
rangetwo:=[];
for k in range do
y:=#S[k];
Append(~rangetwo,y);
end for;
powerstwo:=[];
for j in range do
powersone:=[];
for r in [1..rangetwo[j]] do
Append(~powersone,#S[j][r]);
end for;
Append(~powerstwo,powersone);
end for;
rangefour:=[1..#powerstwo];
rangefive:=[];
for rf in rangefour do
Append(~rangefive,#powerstwo[rf]);
end for;
polylist:=[];
for last in rangefour do
z:=1;
for rft in [1..rangefive[last]] do
z:=(1+t^powerstwo[last][rft])*z;
end for;
Append(~polylist,z);
end for;
polylistthree:=[];
for ti in range do
polylisttwo:=lengths[ti]*polylist[ti];
Append(~polylistthree,polylisttwo);
```

```
end for;
poly:=0;
for tf in range do
poly:=polylistthree[tf]+poly;
end for;
finalpoly:=poly div Order(G3);
X:=[*"This is for q=",q,"and m=",m,finalpoly*];
return X;
end function;

cart_prod_lists := function(R)
\\creates a list of cartesian products
N:=#R;
if N eq 1 then return R[1]; end if;
if N gt 1 then
L:=[];
R0:=[R[k]:k in [1..(N-1)]];
R1:=$$(R0);
for i in R1 do
for j in R[N] do
if N eq 2 then L:=Append(L,Append([i],j)); end if;
if N gt 2 then L:=Append(L,Append(i,j)); end if;
end for;
end for;
return L;
end if;
end function;

toric_points:=function(n,F)
\\creates the lattice points used to construct a toric
\\code over a finite field F
T:=[x : x in F | x ne 0];
L:=cart_prod_lists([T:i in [1..n]]);
return L;
end function;

toric_code := function(L,F)
\\constructs the toric code, given a set of lattice points L
\\and a finite field F
u:=L;
gens:=[];
d:=#L[1];
n:=#toric_points(d,F);
V:=VectorSpace(F,n);
Z:=Integers();
```

```
for v in u do
Append(~gens,[Z!v[i]: i in [1..d]]);
end for;
B0:={};
for e in gens do
Include(~B0, V![&*[t[i]^e[i] : i in [1..d]]: t in toric_points(d,F)]);
end for;
B:=SetToSequence(B0);
C1:=VectorSpaceWithBasis(B);
C:=LinearCode(C1);
return C;
end function;

Perm2:=function(q,m);
\\creates the affine general linear group for mxm matrices over GF(q).
G:=GL(m,Integers(q-1));
G2:=GL(m+1,Integers(q-1));
num:=[1..#Generators(G)];
num5:=(q-1)^m;
id:=Id(G);
id2:=Id(G2);
list:=[];
for i in num do
InsertBlock(~id2,G.i,1,1);
Append(~list,id2);
end for;
num2:=[1..m];
for j in num2 do
id22:=Id(G2);
InsertBlock(~id22,id,1,1);
id22[j,m+1]:=1;
Append(~list,id22);
end for;
l:=[];
for i in [0..q-2] do
for j in [0..q-2] do
M:=Matrix(Integers(q-1),m+1,1,[j,i,1]);
Append(~l,M);
end for;
end for;
blist:=[];
for j in [1..#list] do
brianslist:=[];
for i in [1..#l] do
flubb:=list[j]*l[i];
```

```
Append(~brianslist,flubb);
end for;
Append(~blist,brianslist);
end for;
brianslist3:=[];
for i in [1..#list] do
brianslist2:=[];
for j in [1..#l] do
newb:=ChangeRing(blist[i][j],Integers());
Append(~brianslist2,newb);
end for;
Append(~brianslist3,brianslist2);
end for;
unbrian:=[];
for i in [1..#list] do
unbri:=[];
for j in [1..#l] do
pp:=brianslist3[i][j][1][1]+(q-1)*brianslist3[i][j][2][1]+1;
Append(~unbri,pp);
end for;
Append(~unbrian,unbri);
end for;
perm:=PermutationGroup<num5|unbrian>;
return perm;
end function;

GenOrbits2 := function(q,m,k)
\\generates the orbits for codes of dimension k over GF(q)
\\in 2-space. only works for m=2.
n:=(q-1)^m;
GPerm:=Perm2(q,m);
points2:= [1..n];
points:=Seqset(points2);
S:= Subsets(points,k);
Sb:=GSet(GPerm,S);
orb:=Orbits(GPerm,Sb);
return orb;
end function;


IndexToPoint := function(index, dimension, modulus)
\\given an index representing a point, the dimension k of a
\\code, and a modulus, return a point.
count := [Integers()|0..(dimension-1)];
point := [];
```

```
for i in count do
value := Floor((index-1)/(modulus^i)) mod modulus;
point := Append(point, value);
end for;
return(point);
end function;


IndexlistToPointlist := function(indexlist, dimension, modulus)
\\given an indexlist representing points, the dimension k of
\\a code, and a modulus, return points.
returnlist := [];
for element in indexlist do
returnlist := Append(returnlist,
              IndexToPoint(element, dimension, modulus));
end for;
return(returnlist);
end function;


CodesForK2 := function(q,dimension,k)
\\returns all distinct toric codes of a given k over GF(q) in 2-space.
\\Only works for dimension=2.
field:=GF(q);
n:=#field;
orbitlist := GenOrbits2(q,dimension,k);
Y:=[];
for element in orbitlist do
repelement := IndexlistToPointlist(element[1], dimension, n-1);
T:=toric_code(repelement,field);
value := WeightDistribution(T);
X:=[*"This is for q=",q,"m=",dimension,"and k=",k,value,repelement*];
Append(~Y,X);
end for;
return Y;
end function;

CyclePoly3:=function(q,m);
\\Finds the cycle index polynomial for toric codes over GF(q) in 3-space.
\\Only works for m=3.
P<t>:=PolynomialRing(Rationals());
G:=GL(m,Integers(q-1));
G2:=GL(m+1,Integers(q-1));
num:=[1..#Generators(G)];
num5:=(q-1)^m;
```

```
id:=Id(G);
id2:=Id(G2);
list:=[];
for i in num do
InsertBlock(~id2,G.i,1,1);
Append(~list,id2);
end for;
num2:=[1..m];
for j in num2 do
id22:=Id(G2);
InsertBlock(~id22,id,1,1);
id22[j,m+1]:=1;
Append(~list,id22);
end for;
l:=[];
for i in [0..q-2] do
for j in [0..q-2] do
for k in [0..q-2] do
M:=Matrix(Integers(q-1),m+1,1,[k,j,i,1]);
Append(~l,M);
end for;
end for;
end for;
blist:=[];
for j in [1..#list] do
brianslist:=[];
for i in [1..#l] do
flubb:=list[j]*l[i];
Append(~brianslist,flubb);
end for;
Append(~blist,brianslist);
end for;
brianslist3:=[];
for i in [1..#list] do
brianslist2:=[];
for j in [1..#l] do
newb:=ChangeRing(blist[i][j],Integers());
Append(~brianslist2,newb);
end for;
Append(~brianslist3,brianslist2);
end for;
unbrian:=[];
for i in [1..#list] do
unbri:=[];
for j in [1..#l] do
```

```
pp:=brianslist3[i][j][1][1]+(q-1)*brianslist3[i][j][2][1]+
    ((q-1)^2)*brianslist3[i][j][3][1]+1;
Append(~unbri,pp);
end for;
Append(~unbrian,unbri);
end for;
G3:=PermutationGroup<num5|unbrian>;
class:=Classes(G3);
range:=[1..#class];
S:=[];
lengths:=[];
for len in range do
Append(~lengths,class[len][2]);
end for;
for i in range do
x:=CycleDecomposition(class[i][3]);
Append(~S,x);
end for;
rangetwo:=[];
for k in range do
y:=#S[k];
Append(~rangetwo,y);
end for;
powerstwo:=[];
for j in range do
powersone:=[];
for r in [1..rangetwo[j]] do
Append(~powersone,#S[j][r]);
end for;
Append(~powerstwo,powersone);
end for;
rangefour:=[1..#powerstwo];
rangefive:=[];
for rf in rangefour do
Append(~rangefive,#powerstwo[rf]);
end for;
polylist:=[];
for last in rangefour do
z:=1;
for rft in [1..rangefive[last]] do
z:=(1+t^powerstwo[last][rft])*z;
end for;
Append(~polylist,z);
end for;
polylistthree:=[];
```

```
for ti in range do
polylisttwo:=lengths[ti]*polylist[ti];
Append(~polylistthree,polylisttwo);
end for;
poly:=0;
for tf in range do
poly:=polylistthree[tf]+poly;
end for;
finalpoly:=poly div Order(G3);
X:=[*"This is for q=",q,"and m=",m,finalpoly*];
return X;
end function;

Perm3:=function(q,m);
\\see Perm2. Only works for m=3.
G:=GL(m,Integers(q-1));
G2:=GL(m+1,Integers(q-1));
num:=[1..#Generators(G)];
num5:=(q-1)^m;
id:=Id(G);
id2:=Id(G2);
list:=[];
for i in num do
InsertBlock(~id2,G.i,1,1);
Append(~list,id2);
end for;
num2:=[1..m];
for j in num2 do
id22:=Id(G2);
InsertBlock(~id22,id,1,1);
id22[j,m+1]:=1;
Append(~list,id22);
end for;
l:=[];
for i in [0..q-2] do
for j in [0..q-2] do
for k in [0..q-2] do
M:=Matrix(Integers(q-1),m+1,1,[k,j,i,1]);
Append(~l,M);
end for;
end for;
end for;
blist:=[];
for j in [1..#list] do
brianslist:=[];
```

```
for i in [1..#l] do
flubb:=list[j]*l[i];
Append(~brianslist,flubb);
end for;
Append(~blist,brianslist);
end for;
brianslist3:=[];
for i in [1..#list] do
brianslist2:=[];
for j in [1..#l] do
newb:=ChangeRing(blist[i][j],Integers());
Append(~brianslist2,newb);
end for;
Append(~brianslist3,brianslist2);
end for;
unbrian:=[];
for i in [1..#list] do
unbri:=[];
for j in [1..#l] do
pp:=brianslist3[i][j][1][1]+(q-1)*brianslist3[i][j][2][1]+
    ((q-1)^2)*brianslist3[i][j][3][1]+1;
Append(~unbri,pp);
end for;
Append(~unbrian,unbri);
end for;
perm:=PermutationGroup<num5|unbrian>;
return perm;
end function;

GenOrbits3 := function(q,m,k)
\\see GenOrbits2. only works for m=3
n:=(q-1)^m;
GPerm:=Perm3(q,m);
points2:= [1..n];
points:=Seqset(points2);
S:= Subsets(points,k);
Sb:=GSet(GPerm,S);
orb:=Orbits(GPerm,Sb);
return orb;
end function;

CodesForK3 := function(q,dimension,k)
\\see CodesForK2. Only works for dimension=3
field:=GF(q);
n:=#field;
```

```
orbitlist := GenOrbits3(q,dimension,k);
Y:=[];
for element in orbitlist do
repelement := IndexlistToPointlist(element[1], dimension, n-1);
T:=toric_code(repelement,field);
value := WeightDistribution(T);
X:=[*"This is for q=",q,"m=",dimension,"and k=",k,value,repelement*];
Append(~Y,X);
end for;
return Y;
end function;

random_toric_code_search2:=function(k,N,F)
     //
     // Generates any number N of random toric codes of
     // dimension k over the field F and "remembers" the
     // distinct weight distributions found in the set DistinctWDs,
     // plus the lists of points corresponding to the monomials for
     // each distinct one in the list DistinctMonoms.
     // This uses the cart_prod_lists and toric_code functions from
     // the toric code procedures by David Joyner distributed earlier.
     // This could be modified in several ways ....
     // Only works for 2-space.

     q:=#F;
     qminus1:=#[x : x in F | x ne 0];
     n:=(q-1)^2;
     DistinctWDs:={};
     DistinctMonoms:=[];
     L:=cart_prod_lists([ [i : i in [0..qminus1-1]],
                          [i : i in [0..qminus1-1]]]);
     for i:=1 to N do
        S:={};
        while #S lt k do
             Include(~S,Random(L));
        end while;
        C:=toric_code(SetToSequence(S),F);
        W:=WeightDistribution(C);
        if W notin DistinctWDs then
              Include(~DistinctWDs,W);
              Append(~DistinctMonoms,[*S,"MinDis=",W[2][1],
                 "DualMinDis=",MacWilliamsTransform(n,k,q,W)[2][1]*]);
        end if;
     end for;
     return [*"This is for m=2,q=",q,"and k=",k,
```

```
            "The number of orbits found are",#DistinctWDs,
            DistinctMonoms,"This is for m=2,q=",q,"and k=",k,
            "The number of orbits found are",#DistinctMonoms*];
     end function;

FixedAndRandom3:=function(k,N,F)
     //Fixes {@[0,0,0],[0,1,0],[0,0,1],[1,0,0]@} and adds random points
     //until that set has k elements.  Therefore, it finds "useful" toric
     //codes of dimension k over F. N is the number of random toric codes
     //generated.
     q:=#F;
     n:=(q-1)^3;
     qminus1:=#[x : x in F | x ne 0];
     DistinctWDs:={};
     DistinctMonoms:=[];
     L:=cart_prod_lists([ [i : i in [0..qminus1-1]],
        [i : i in [0..qminus1-1]], [i : i in [0..qminus1-1]]]);
     list:=[];
     list2:=[];
     for i:=1 to N do
        S:={@[0,0,0],[0,1,0],[0,0,1],[1,0,0]@};

     while #S lt k do
     x:=Random(L);
     Include(~S,x);
     end while;

        C:=toric_code(SetToSequence(S),F);
        W:=WeightDistribution(C);
        if W notin DistinctWDs then
              Include(~DistinctWDs,W);
          Append(~DistinctMonoms,[*S,"MinDis=",W[2][1],"DualMinDis=",
                MacWilliamsTransform(n,k,q,W)[2][1]*]);
    Append(~list2,S);
    Append(~list,W[2][1]);
        end if;
     end for;
     return [*"This is for m=3,q=",q,"and k=",k,
            "The number of orbits found are",
            #DistinctWDs,"The best minimum distance is",
            Maximum(list),list2,"This is for m=3,q=",q,"and k=",
            k,"The number of orbits found are", #DistinctMonoms,
            "The best minimum distance is",Maximum(list),L*];
     end function;
```

```
GSet2 := function(q,m,k)
//creates the set that AGL(2,Z_q-1) acts on for toric codes of
//dimension k.
n:=(q-1)^m;
GPerm:=Perm2(q,m);
points2:= [1..n];
points:=Seqset(points2);
S:= Subsets(points,k);
Sb:=GSet(GPerm,S);
return Sb;
end function;

GSet3 := function(q,m,k)
//see GSet2. Only works for m=3.
n:=(q-1)^m;
GPerm:=Perm3(q,m);
points2:= [1..n];
points:=Seqset(points2);
S:= Subsets(points,k);
Sb:=GSet(GPerm,S);
return Sb;
end function;

Stabilizer2:=function(q,m,k);
//finds all the stabilizers of all possible points that create
//distinct codes of dimension k over GF(Q) in 2-space.
n:=(q-1)^m;
G:=Perm2(q,m);
Y:=GSet2(q,m,k);
points2:= [1..n];
points:=Seqset(points2);
S:= Subsets(points,k);
x:=[];
for s in S do
stab:=#Stabilizer(G,Y,s);
Append(~x,[*s,stab*]);
end for;
return x;
end function;

Stabilizer3:=function(q,m,k);
//see Stabilizer2. Only works for 3-space.
n:=(q-1)^m;
G:=Perm3(q,m);
Y:=GSet3(q,m,k);
```

```
points2:= [1..n];
points:=Seqset(points2);
S:= Subsets(points,k);
x:=[];
for s in S do
stab:=#Stabilizer(G,Y,s);
Append(~x,[*s,stab*]);
end for;
return x;
end function;

AveWt:=function(weightdis,numwords);
//finds the average weight over a code given it's weight distribution
//and the number of words in the code.
P<x> := PolynomialRing(RealField());
list:=0;
numweight:=#weightdis;
for i in [1..numweight] do
list:=list+weightdis[i][1]*weightdis[i][2];
end for;
avewt:=x;
return [*Evaluate(avewt,list/numwords),list/numwords*];
end function;
```

A Census of Two Dimensional Toric Codes over Galois Fields of Sizes 7, 8 and 9

Alejandro Carbonara
California Institute of Technology

Juan Pablo Murillo
Sonoma State University

Abner Ortiz
University of Puerto Rico at Humacao

July 2009

Abstract

J. Hansen introduced toric codes using the geometry of polygons in $\mathbb{R}^2$ and the convex polytopes P in $\mathbb{R}^m$ more generally. But collections of points more general than the sets $P \cap \mathbb{Z}^m$ can also be used to define *generalized* toric codes. In our research using the Magma computer algebra system, we search to find the generalized toric codes with the best parameters for some dimensions over the Galois fields of size 7, 8 and 9.

1 Background on toric codes

J. P. Hansen introduced toric codes with constructions from algebraic geometry [1]. According to Hansen's definition, toric codes are linear codes generated by lattice points within a convex polytope in $\mathbb{R}^m$ representing monomials over a Galois field $\mathbb{F}_q$, where q is a power of a prime [1].

The main focus of our research is on generalized toric codes, which means that we can use an arbitrary set of lattice points, not only ones that are in a polytope. To be more specific, the cases researched in this paper include toric codes with $q = 7, 8$ and 9 in a 2-dimensional space.

Definition 1. To obtain the **generator matrix of a toric code**, we start with k lattice points in $\mathbb{Z}_{q-1}^m$, and we let α be a primitive element of $\mathbb{F}_q$. For any point $p = (a_1, a_2, \ldots a_m) \in \mathbb{Z}_{q-1}^m$, the monomial associated to that point will be $F(x_1, x_2, \ldots x_m) = x_1^{a_1} x_2^{a_2} \ldots x_m^{a_m}$. To find the values of the monomials, we will evaluate them for all elements of $(\mathbb{F}_q^*)^m$. Let each of our lattice point monomials be associated to a row and each possible point in $(\mathbb{F}_q^*)^m$ be associated to column in the generator matrix, making it a $k \times (q-1)^m$ matrix. The $s_{(i,j)}$ position of our matrix would be the result of evaluating the monomial associated to the ith row over the point associated to the jth column. If P is the set of lattice points, the corresponding code is denoted C_P.

Example 1. Let $P = \{(2,0),(0,2),(1,2)\}$ be a set of 3 lattice points in $\mathbb{Z}_3^2$. These points give us the three monomials:

$$f_1(x,y) = x^2, f_2(x,y) = y^2, f_3(x,y) = xy^2.$$

We consider the coordinates of all points of $(\mathbb{F}_4^*)^2$, where α is a primitive element of $\mathbb{F}_4$:

$$\{(1,1),(\alpha,1),(\alpha^2,1),(1,\alpha),(\alpha,\alpha),(\alpha^2,\alpha),(1,\alpha^2),(\alpha,\alpha^2),(\alpha^2,\alpha^2)\}.$$

Each of the lattice points in P will determine a row of our generator matrix while each possible point in $(\mathbb{F}_4^*)^2$ will determine a column. We proceed to evaluate the monomials over the points of $(\mathbb{F}_4^*)^2$.

$$f_1(1,1) = 1, f_1(\alpha,1) = \alpha^2, f_1(\alpha^2,1) = \alpha, \ldots,$$

thus obtaining the matrix:

$$\begin{pmatrix} 1 & \alpha^2 & \alpha & 1 & \alpha^2 & \alpha & 1 & \alpha^2 & \alpha \\ 1 & 1 & 1 & \alpha^2 & \alpha^2 & \alpha^2 & \alpha & \alpha & \alpha \\ 1 & \alpha & \alpha^2 & \alpha^2 & 1 & \alpha & \alpha & \alpha^2 & 1 \end{pmatrix}.$$

We only need to look at a small part of $\mathbb{Z}^2$ to find all possible toric codes. It has been proven that over $\mathbb{F}_q$, so long as the lattice points are within $[0, q-2]^m$, the monomials generated by the lattice points will be linearly independent ([6], Theorem 2). In addition, since our monomials are over $\mathbb{F}_q$, the monomials $x^{a(q-1)+b}$ and x^b are equivalent. Lattice points that have terms greater than $q-2$ can be equivalent to lattice points within $\mathbb{Z}_{q-1}^m$. Therefore, we only need to consider lattice points within $\mathbb{Z}_{q-1}^m$ to consider every possible toric code. In fact, we will consider our lattice points to be in $\mathbb{Z}_{q-1}^m$ from now on.

2 Affine Transformations

Definition 2. An **affine transformation** $T : \mathbb{Z}_{q-1}^m \to \mathbb{Z}_{q-1}^m$ is a mapping $T(x) = Ax + v$ where A is an invertible $m \times m$ matrix with entries in $\mathbb{Z}_{q-1}$, and v is a vector in $\mathbb{Z}_{q-1}^m$.

Note that an affine transformation preserves the number of lattice points and the collinearity of points. Another important relationship between two codes is monomial equivalence.

Definition 3. Two codes C_1 and C_2 with the the same length and dimension, are **monomially equivalent** if there exist generator matrices G_1 for C_1, G_2 for C_2 and matrices P, D such that

$$G_2 = G_1 P D,$$

where P is permutation matrix and D diagonal matrix (see [3]).

This definition directly implies that if two codes are monomially equivalent, they have the same weight enumerator, so they have the same minimum distance.

Proposition 1. *If two codes are monomially equivalent, then they have the same weight enumerator.*

These two forms of equivalence are closely related.

Theorem 1. *Let P_1 and P_2 be subsets of $\mathbb{Z}_{q-1}^m$. If there exists an affine transformation $T : \mathbb{Z}_{q-1}^m \to \mathbb{Z}_{q-1}^m$ defined by $T(x) = Ax + v$, where A is an invertible matrix in $GL(m, \mathbb{Z}_{q-1}^m)$ and v is a vector in $\mathbb{Z}_{q-1}^m$, with $T(P_1) = P_2$, then the codes C_{P_1} and C_{P_2} are monomially equivalent.*

Proof. Suppose we have two $AGL(m, \mathbb{Z}_{q-1})$-equivalent set of points P_1 and P_2. Both P_1 and P_2 contain lattice points corresponding to monomials of the form x^e where $e \in \mathbb{Z}_{q-1}^m$ with $x = (x_1, x_2, \ldots x_n)$. By our hypothesis on P_1 and P_2 there exists an invertible affine transformation

$$T(x) = M(x) + \lambda$$

such that $T(P_1) = P_2$ and M is an element of $GL(m, \mathbb{Z}_{q-1}^m)$ so $\det(M)$ is invertible in $\mathbb{Z}_{q-1}$. Hence $\#(P_1) = \#(P_2)$. Let $P_1 \cap \mathbb{Z}_{q-1}^m = \{e(i) : i = 1, \ldots, \#(P_1)\}$ be the numbering of the points in P_1. So, the code C_{P_1} is spanned by $\mathrm{ev}(x^{e(i)})$ for $1 \leq i \leq n$ where ev is the evaluation, and similarly C_{P_2} is spanned by $\mathrm{ev}(x^{T(e(i))})$. Write α for a primitive element in $\mathbb{F}_q$. Let $e(i) \in P_1 \cap \mathbb{Z}_{q-1}^m$ and define $\alpha^f = (\alpha^{f_1}, \ldots, \alpha^{f_m}) \in (\mathbb{F}_q^*)^m$. The component of $\mathrm{ev}(x^{e(i)}) \in C_{P_1}$ corresponding to α^f is $\alpha^{\langle e(i), f \rangle}$, where $\langle e(i), f \rangle$ is the usual dot product. The corresponding entry in the codeword $\mathrm{ev}(x^{T(e(i))})$ in C_{P_2} is $\left\langle \alpha^{T(e(i)), f} \right\rangle$. This can be rewritten as

$$\alpha^{\langle Me(i) + \lambda, f \rangle} = \alpha^{\langle Me(i), f \rangle} \cdot \alpha^{\langle \lambda, f \rangle}.$$

The second term of the product is not dependent on $e(i)$. These nonzero scalars are the diagonal entries in the matrix D as in the definition of monomially equivalent codes. By a standard property of dot product,

$$\alpha^{\langle Me(i), f \rangle} = \alpha^{\left\langle e(i), M^t f \right\rangle}.$$

The transposed matrix M^t also defines a bijective mapping from $\mathbb{Z}_{q-1}^m$ to $\mathbb{Z}_{q-1}^m$ since $\det(M^t) = \det(M)$ is invertible in $\mathbb{Z}_{q-1}$. Now we must show that M^t induces a permutation of $\mathbb{Z}_{q-1}^m$. Suppose $M^t f \equiv M^t g \bmod q - 1$. Since $\det(M^t)$ is invertible mod $q - 1$, we know that M^t is invertible. So, we can multiply by $(M^t)^{-1}$ on the left. Hence $f \equiv g \bmod q - 1$ and M^t defines a permutation of the points α^f, as desired. Note that M^t permutes all of the codewords in the same way. This gives the permutation matrix P. Hence C_{P_1} is monomially equivalent to C_{P_2}. □

2.1 The Group of Affine Transformations

Now, consider the set of all affine transformations over a finite space. We can define composition as an operation over these transformations.

Theorem 2. *The set of invertible affine linear transformation over $\mathbb{Z}_{q-1}^m$ forms a group under composition.*

Proof. Consider two affine transformation $T(x) = Ax + v$ and $S(x) = Bx + u$ for invertible integer matrices A and B over $\mathbb{Z}_{q-1}$ and integer vectors u and v in $\mathbb{Z}_{q-1}^m$.

The composition of these two elements is

$$T(S(x)) = A(Bx + u) + v = (AB)x + (Au + v).$$

Since $Au + v$ is a vector in $\mathbb{Z}_{q-1}^m$ and AB is an invertible matrix, the composition of these two transformations is an affine transformation. Therefore, the set is closed under composition.

If we consider the affine transformation composed of the identity matrix and empty vector, we can easily see that it will map every point to itself. In effect, if we compose this with any other transformation, we will get that other transformation. This gives us an identity element.

Let A' be the inverse of A. Define $T'(x) = A'(x-v)$. Since $T(T'(x)) = T'(T(x)) = x$, we have constructed $T(x)$'s inverse, then the transformation is invertible.

Since this set is closed under composition, has an identity element, and is invertible, it must be a group. □

Now, let the group of affine transformations act upon the set of all possible subsets of points of $\mathbb{Z}_{q-1}^m$ in the natural way.We have established that if a set of points undergoes an affine transformation, the result is a monomially equivalent toric code.

Definition 4. Given a group action G on a set S, the **orbit** θ containing $s \in S$ is defined as

$$\theta = \{g \cdot s | g \in G\}.$$

The orbits partition the elements of S.

Thus under the action of $AGL(m, \mathbb{Z}_{q-1})$ on the k-sets of points in $\mathbb{Z}_{q-1}^m$, sets in the same orbit yield codes that are monomially equivalent.

Example 2. In $\mathbb{Z}_3^2$, the two sets of points $P_1 = \{(0,0),(0,1),(1,0)\}$ and $P_2 = \{(1,1),(1,2),(2,1)\}$ are in the same orbit of action of the affine transformation group $AGL(2,\mathbb{Z}_3)$ on triples of points. Over $\mathbb{F}_4$, the codes C_{P_1} and C_{P_2} have the same minimum distance 6, since we can translate the first set of points to get the second. However, affine transformations conserve collinearity, so $P_3 = \{(0,0),(1,0),(2,0)\}$ must be in a different orbit. In fact, the code C_{P_3} over $\mathbb{F}_4$ has distance 3.

3 Cycle Index Polynomial

Definition 5. Let G be a group, and let a, $b \in G$. Then a and b are **conjugate** if and only if there exists an element $g \in G$ such that $gag^{-1} = b$.

It is noteworthy to mention that conjugacy is an equivalence relation, and it partitions the group G into equivalence or conjugacy classes. Thus each element of G belongs to one conjugacy class only. The conjugacy class C containing the element a of G is defined as $C(a) = \{gag^{-1} : g \in G\}$.

Consider the group of all affine linear permutations acting on $\mathbb{Z}_{q-1}^m$. Now, find the conjugacy classes of that group. Naturally, you will have a number of classes, each with an order, a length, and a representative permutation for that group.

Definition 6. The elements of each conjugacy class C define permutations with the same cyclic structure. The length is the number of elements in each class. Let the length of the i^{th} class be l_i and let the number of j-cycles in the representative permutation be $x_{i,j}$. We can find a class polynomial by taking $g_i(x) = l_i \cdot \prod(1+t^j)^{x_{i,j}}$. Then, we define the **substituted cycle index polynomial** as $\frac{1}{|G|}\sum g_i(x)$, summed over all the classes.

Theorem 3 ([2])**.** *The coefficient of t^k in the substituted cycle index polynomial for a group G acting on a set S counts the number of distinct orbits of G acting on subsets of S of size k.*

Example 3. To find the substituted cycle index polynomial for $(\mathbb{F}_4^*)^2$, we first find all the conjugacy classes of the group $AGL(2, \mathbb{Z}_3)$ and their lengths. Each conjugacy class has a representative permutation. For each conjugacy class, we consider their respective permutation, and start forming the polynomial.

First, our set S consists of nine points, S={1,2,3,4,5,6,7,8,9}. We then find the generators for AGL(2,$\mathbb{Z}_3$) and the 13 conjugacy classes with their respective lengths and representative permutations.

Generators:

$$(2,9,4)(3,5,7), (1,2,3)(4,5,6)(7,8,9), (2,3)(5,6)(8,9)$$

Conjugacy Classes: (in Magma output format)

```
[1] Order 1       Length 1      Rep Id
[2] Order 2       Length 9      Rep (1, 6)(2, 5)(3, 4)(7, 9)
[3] Order 2       Length 36     Rep (1, 2)(4, 5)(7, 8)
[4] Order 3       Length 8      Rep (1, 2, 3)(4, 5, 6)(7, 8, 9)
...
[10] Order 8       Length 54     Rep (1, 5, 7, 4, 6, 2, 9, 3)
[11] Order 8       Length 54     Rep (1, 2, 7, 3, 6, 5, 9, 4)
```

Following from our definition (6), our substituted cycle index polynomial is

$$\begin{aligned} g(x) &= \frac{1}{432}((1+t)^9 + 9(1+t^2)^4(1+t) + 36(1+t)^3 + \ldots + 54(1+t^8)(1+t)) \\ &= t^9 + t^8 + t^7 + 2t^6 + 2t^5 + 2t^4 + 2t^3 + t^2 + t + 1. \end{aligned}$$

Note that the example of the substituted cycle index polynomial is symmetric. This can be shown by a relatively simple combinatorial proof.

Being able to know the number of orbits of a group is very important for estimating whether our procedures would take a reasonable time to run.

4 MacWilliams Identity

When we are testing codes, we are also interested in the dual code. One way of finding the distance of the dual code is through the MacWilliams identity.

Definition 7. Let A_w represent the number of codewords in a code C with weight w ranging from 0 to n. Then,

$$W_C(x, y) = \sum_{w=0}^{n} A_w x^w y^{n-w}$$

gives us the **weight enumerator polynomial**.

Theorem 4 (The MacWilliams Identity [4])**.** *Consider a code C and its dual code $C^\perp$. Then*

$$W_{C^\perp}(x, y) = \frac{1}{|C|} W_C(x + (q-1)y, x - y).$$

So, if we find the weight enumerator for a given code, we can get the weight enumerator of the dual code. We use this identity to derive information about dual codes.

5 Method of Search

To do the actual searching though codes we used the MAGMA computer algebra system. Magma had a lot of premade procedures to find orbits and minimum distances, allowing it to use the theory we have proven so far in our code.

5.1 Brute Force Method

As we have already shown, we can find the orbits of the set of affine transformations over points. The advantage to partitioning our points into orbits is that every set of points in an orbit generates a monomially equivalent code. Codes that are monomially equivalent have the same weight enumerator, so they have the same minimum distance and their dual codes have the same distance. So, to do a brute force search to find optimal codes, it is only necessary need to check one code in each orbit. Magma has a built in procedure that calculates the orbits for a given permutation group. To generate the field of affine transformations, we had Magma produce the generator matrices for the general linear group over our points. Then a couple of Maple procedures we made convert the generator matrices into generator permutations and create the additional translation permutations. So, we can then find the orbits.

The first procedure took one code from each orbit and found the minimum distance, then recorded the codes with the biggest distance. Ideally, this procedure

would quickly find the best toric code for given parameters. The problem is that the number of orbits became large very quickly. This meant that Magma took more time to calculate the orbits, and had to calculate more minimum distances. Since the Galois fields we dealt with were of sizes 7, 8, and 9, the points were in $(\mathbb{Z}_6)^2$, $(\mathbb{Z}_7)^2$, and $(\mathbb{Z}_8)^2$. Even for the smallest field the calculation took way too much time for a toric code with just 6 points. A faster algorithm was required.

5.2 Random Method

A simple random guess and check procedure turned up a lot of data. Since restricting how many trials the procedure did was easy, we were able to run the procedure for codes the brute force search could not handle. Unfortunately, this introduces a possibility for oversight. As the number of orbits increases, the chance that we miss the most optimal code increases. As a result, we have found that we can often find better codes if we rerun searches multiple times.

Originally, the procedure took every code with a unique weight enumerator to prevent the code from taking two codes from the same orbit. This quickly became too much data for larger codes. We then edited the procedure so that it only took codes that have a unique pair of distances for their main code and their dual code.

6 Code Discoveries

When looking for codes, we searched for the codes with greatest distance d. The distance of a code relates directly to how good that code is at error correction. Since all practical uses of codes relate to how good they can correct errors, finding new better distances is an important discovery. In our search, we compared our codes to the codes on the online code table [5]. The online code table compiles together the best known codes for each length and language size. Very often, there is a gap between the theoretically best code and the best known code. Many of the toric codes we found and their dual codes had distances as good the best known codes on the site.

Theorem 5. *A* $[49, 8, 34]$ *code exists.*

Proof. The codes over $\mathbb{F}_8$ generated by the two sets of points

$$\{(2,3),(2,0),(1,3),(6,0),(0,3),(5,1),(3,6),(0,6)\}$$

$$\{(6,2),(1,0),(2,1),(1,2),(0,2),(0,5),(3,5),(0,6)\}$$

are both [49,8,34] codes. Their generator matrices are size 8×49. These two codes have different weight enumerators. These are the respective weight enumerators of the codes above:

$$< 0, 1 >, < 34, 17493 >, < 35, 33663 >, < 36, 58310 >, \cdots$$

$$< 0, 1 >, < 34, 12348 >, < 35, 27342 >, < 36, 65856 >, \cdots$$

This means that they are distinct codes. □

Corollary 1. *Toric codes can generate codes with a larger distance than any other known method under certain parameters.*

Below is a table of all the optimal codes we found.

Best codes						
$\mathbb{F}_q$	[1]m	[2]n	[3]k	[4]d	[5]Best d	[6]Coordinates
$\mathbb{F}_7$	2	36	3	30	30	[5, 5], [2, 4], [3, 0]
	2	36	4	27	28	[1, 4],[4, 0],[3, 1],[2, 1]
	2	36	5	24	27	[2, 0],[2, 5],[4, 0],[3, 0],[1, 1]
	2	36	6	24	25	[3, 5],[5, 0],[2, 2],[4, 3],[0, 1],[2, 1]
	2	36	7	23	24	[2, 0],[1, 3],[0, 5],[4, 5],[3, 1],[2, 2],[3, 0]
	2	36	8	20	22	[2, 0],[0, 2],[1, 2],[5, 1],[2, 5],[4, 3],[1, 1],[0, 0]
	2	36	9	20	21	[1, 0],[2, 0],[1, 5],[5, 1],[2, 5],[4, 1],[3, 2],[0, 1],[3, 0]
	2	36	10	18	20	[4, 3],[2, 2],[2, 1],[5, 4],[1, 3],[3, 3],[1, 4],[3, 1],[0, 2],[3, 0]
$\mathbb{F}_8$	2	49	3	42	42	[5, 4],[0, 5],[6, 4]
	2	49	4	40	40	[3, 6],[1, 5],[5, 1],[3, 2]
	2	49	5	36	38	[6, 1],[3, 6],[3, 5],[1, 5],[5, 3]]
	2	49	6	36	36	[2, 0],[0, 4],[5, 0],[2, 3],[0, 1],[1, 1]]
	2	49	7	35	35	[1, 0],[0, 2],[1, 4],[5, 0],[4, 1],[2, 4],[4, 3]
	2	49	8	34	33	[6, 2],[1, 0],[2, 1],[1, 2],[0, 2],[0, 5],[3, 5],[0, 6]
	2	49	9	30	31	[4, 6],[4, 2],[1, 1],[5, 4],[5, 0],[0, 4],[3, 6],[0, 6],[6, 2]
	2	49	10	30	30	[6, 2],[1, 0],[2, 1],[1, 2],[0, 2],[0, 5],[3, 5],[0, 6],[4, 5],[1, 4]
$\mathbb{F}_9$	2	64	3	56	56	[4, 4], [1, 7], [5, 2]
	2	64	4	52	54	[3, 7],[4, 0],[3, 0],[2, 1]
	2	64	5	48	51	[1, 3],[6, 7],[3, 5],[1, 4],[4, 7]
	2	64	6	48	49	[3, 7],[7, 4],[1, 7],[2, 4],[3, 2],[2, 1]
	2	64	7	47	48	[5, 4],[4, 7],[5, 3],[7, 1],[4, 6], [3, 2], [3,0]
	2	64	8	45	46	[2, 0],[3, 7],[7, 4],[0, 4],[2, 5],[5, 2],[2, 3],[1, 1]
	2	64	9	40	44	[3, 6],[3, 4],[4, 4],[0, 7],[6, 4],[1, 6],[5, 3],[0, 1],[4, 2]
	2	64	10	40	43	[7, 5],[7, 3],[2, 4],[6, 1],[5, 4],[5, 7],[0, 0],[6, 6],[5, 2],[0, 7]

Below is the table of toric codes having the best dual codes which we found. Note that $[64, 55, 6]$ has a better distance than $[64, 54, 5]$, despite having more codewords. This is due to the large computational cost of finding the weight enumeration of $q = 9$, $k = 10$, matrices. Without a doubt a $[64, 54, 6]$ code exists, but we were not able to find it.

[1]m = Dimension of the space.
[2]n = The length of the codeword.
[3]k = The dimension of the code.
[4]d = Largest minimum distance we found for the given n, k.
[5]Best d = Optimal minimum distance found previously, as in [5].
[6]Coordinates = Lattice points.

Best Dual codes						
$\mathbb{F}_q$	m	n	k	d	Best d	Coordinates
$\mathbb{F}_7$	2	36	33	3	3	[5, 5], [2, 4], [3, 0]
	2	36	32	3	4	[1, 4],[4, 0],[3, 1],[2, 1]
	2	36	31	4	4	[2, 0],[2, 5],[4, 0],[3, 0],[1, 1]
	2	36	30	4	5	[3, 5],[5, 0],[2, 2],[4, 3],[0, 1],[2, 1]
	2	36	29	5	6	[2, 0],[1, 3],[0, 5],[4, 5],[3, 1],[2, 2],[3, 0]
	2	36	28	6	6	[1, 2],[0, 4],[5, 3],[4, 1],[3, 3],[4, 0],[3, 1],[3, 0]
	2	36	27	6	6	[1, 3],[2, 0],[1, 5],[4, 4],[5, 1],[5, 0],[4, 1],[4, 3],[3, 0]
	2	36	26	6	7	[4, 3],[2, 2],[2, 1],[5, 4],[1, 3],[3, 3],[1, 4],[3, 1],[0, 2],[3, 0]
$\mathbb{F}_8$	2	49	46	3	3	[5, 4],[0, 5],[6, 4]
	2	49	45	3	4	[3, 6],[1, 5],[5, 1],[3, 2]
	2	49	44	4	4	[6, 1],[3, 6],[3, 5],[1, 5],[5, 3]]
	2	49	43	4	5	[2, 0],[0, 4],[5, 0],[2, 3],[0, 1],[1, 1]]
	2	49	42	5	6	[3, 3],[1, 2],[6, 6],[1, 5],[5, 6],[0, 2],[3, 6]
	2	49	41	6	6	[6, 2],[1, 0],[2, 1],[1, 2],[0, 2],[0, 5],[3, 5],[0, 6]
	2	49	40	6	6	[4, 6],[4, 2],[1, 1],[5, 4],[5, 0],[0, 4],[3, 6],[0, 6],[6, 2]
	2	49	39	6	7	[6, 2],[1, 0],[2, 1],[1, 2],[0, 2],[0, 5],[3, 5],[0, 6],[4, 5],[1, 4]
$\mathbb{F}_9$	2	64	61	3	3	[4, 4], [1, 7], [5, 2]
	2	64	60	3	4	[3, 7],[4, 0],[3, 0],[2, 1]
	2	64	59	4	4	[1, 3],[6, 7],[3, 5],[1, 4],[4, 7]
	2	64	58	4	5	[3, 7],[7, 4],[1, 7],[2, 4],[3, 2],[2, 1]
	2	64	57	5	6	[4, 4],[6, 5],[2, 5],[1, 6],[2, 3], [3, 1], [1,1]
	2	64	56	5	6	[2, 0],[3, 7],[7, 4],[0, 4],[2, 5],[5, 2],[2, 3],[1, 1]
	2	64	55	6	6	[1, 0],[5, 5],[5, 7],[5, 6],[0, 7],[4, 4],[0, 6],[7, 1],[2, 2]
	2	64	54	5	7	[7, 5],[7, 3],[2, 4],[6, 1],[5, 4],[5, 7],[0, 0],[6, 6],[5, 2],[0, 7]

7 Conclusion

Finding the substituted cycle index polynomial enabled us to know the number of orbits each Galois field had for a distinct number of lattice points in the plane as well as a bound on how many codes exist in each orbit.

We used MacWilliams identities to find that the dual codes had the same weight distribution as its codes, and we created a procedure that would output both minimum distances together.

In our research, although our search was random due to the large number of orbits and the not so efficient memory of simple computers, we were able to find optimal codes mostly for $\mathbb{F}_8$. This leads us to think that there are better if not additional optimal codes to be found for each k number of lattice points in each Galois field. In addition, our findings prove that toric codes are a significant value to the field of Coding Theory, and therefore should be explored in further research.

8 Acknowledgments

This work was conducted during the 2009 Mathematical Sciences Research Institute Undergraduate Program (MSRI-UP) in Berkeley, CA. MSRI-UP is supported by the National Science Foundation (grant No. DMS-0754872) and the National Security Agency (grant No. H98230-09-0103). We thank Dr. Little, Dr. Medina and Ashley Wheeler for all their support and guidance.

References

[1] J. Hansen, "Toric Surfaces and error-correcting codes," *Coding theory, cryptography and related areas (Guanajuato, 1998)*, 132–142, Springer, Berlin, 2000.

[2] P. Cameron, *Polynomial aspects of codes, matroids and permutation groups*, notes available online at `http://www.maths.qmw.ac.uk/~pjc/csgnotes/cmpgpoly.pdf`.

[3] J. Little, J. Schwarz, "On toric codes and multivariate Vandermode matrices," *Appl. Alg. Engrg. Comm. Comput.* **18** (2007), 349–367.

[4] R. Roth, *Introduction to Coding Theory*, 1st ed., Cambridge University Press, 2006.

[5] M. Grassl, *Code Tables: Bounds on the parameters of various types of codes*, `http://www.codetables.de`.

[6] V. Diaz, C. Guevera, M. Vath, "Codes from n-Dimensional Cyclic Codes," *Proceedings of Summer Institute in Mathematics for Undergraduates (SIMU), 2001*, Humacao, Puerto Rico.

Indecomposable polyhedra and toric codes

Aileen Gaudinez Chapman University

Cheryl Outing Spelman College

Rachel Vega Concordia College

July 2009

Abstract

In this report we describe and classify indecomposable lattice polytopes in $\mathbb{R}^3$. This paper explores some indecomposable polyhedra not yet considered, and also introduces an idea of "family". We investigate the toric codes over the Galois Field $\mathbb{F}_8$ that are constructed by these polyhedra, following the original method of constructing toric codes from polytopes by Hansen. In addition, we conjecture equivalence relations between the members of a determined family and subsequently the toric codes they generate.

1 Introduction

A *polytope* is the convex hull of a finite set of points. For an m-dimensional polytope P, the *integer lattice points* are those points contained in $P \cap \mathbb{Z}^m$, referred to as lattice points. In the following we will consider only *lattice polytopes*, that is, polytopes that are convex hulls of finite sets of integer lattice points. Informally, a toric code is the code generated from the lattice points of a polytope. We offer a more precise definition below.

Definition 1. The *toric code* denoted $C_P(\mathbb{F}_q)$ over the field $\mathbb{F}_q$ associated to P is the linear code of block length $n = (q-1)^m$ with generator matrix

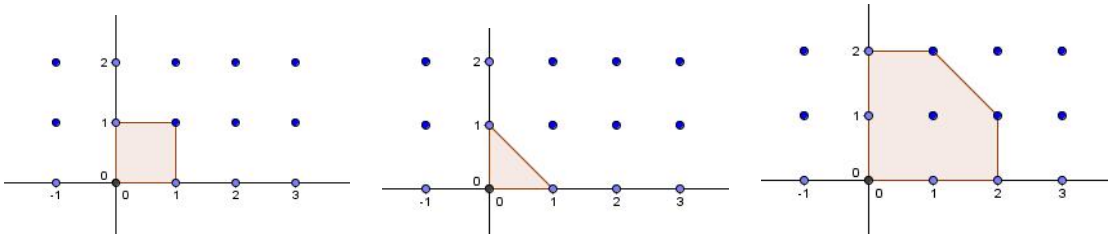
$$G = ((p_f)^e)$$

whose rows are indexed by $e \in P \cap \mathbb{Z}^m$ and columns indexed by $p_f \in (\mathbb{F}_q{}^*)^m$.

While trying to create these toric codes we look more closely at the polytopes that define them. Some essential terms include:

Definition 2. The *Minkowski sum* of two polytopes P_1 and P_2 is the set of vector sums of all lattice points in the two polytopes:

$$P_1 + P_2 = \{p_1 + p_2 \mid p_i \in P_i\}.$$



(a) Unit square (b) Unit triangle (c) Minkowski sum

Figure 1: Minkowski sum example

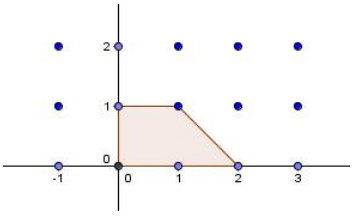


Figure 2: Polytope P

Definition 3. The *full Minkowski length* of a lattice polytope is defined as

$$l(P) = \max_{Q \subseteq P} \{j \mid Q = Q_1 + \cdots + Q_j \text{ for some lattice polytopes } Q_i\}.$$

Example 1. The full Minkowski length of P in Figure 2 is 2. To compute this we consider the Minkowski length of the entire polytope P and all of its subpolytopes. We observe two possible subpolytopes of P in Figure 3, with the maximum amount of lattice points.

Notice that there are no vectors whose Minkowski sum will produce a trapezoid, thus $l(P) = 1$. Also, $l(Q_1) = 1$. However, we can find two vectors, namely $(0, 1)$ and $(1, 0)$, whose sum results in the unit square, thus $l(Q_2) = 2$. The segment $\text{conv}\{(0, 0), (2, 0)\}$ also decomposes as a Minkowski sum with two terms. Therefore the full Minkowski length of our trapezoid P is 2.

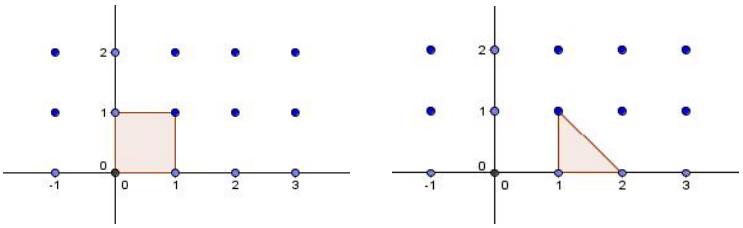


Figure 3: Subpolytopes Q_1 and Q_2 respectively

In using full Minkowski length, we find a negative correlation between the full Minkowski length and the parameters of the toric code. Example 2 shows this correlation with respect to the minimum distance of a toric code.

Example 2. Suppose we have four convex polytopes in 2 dimensions with toric codes over $\mathbb{F}_8$, each with four lattice points. Notice that since there are four lattice points that will index the rows of a generator matrix to the toric code, the dimension of the code $k = 4$. Let P_1, P_2, P_3, and P_4 be polytopes that create toric codes such that $P_1 = \text{conv}\{(0,0),(1,0),(2,0),(3,0)\}, P_2 = \text{conv}\{(0,0),(1,0),(2,0),(0,1)\}, P_3 = \text{conv}\{(0,0),(1,0),(0,1),(1,1)\}$, and $P_4 = \text{conv}\{(1,0),(0,1),(2,2)\}$. We remark that the fourth lattice point of P_4 is an interior point. Using MAGMA, we find that the minimum distances are $28, 35, 36$, and 42 respectively. Note that the full Minkowski length of P_1 is 3, the full Minkowski length of P_2 and P_3 is 2, and the full Minkowski length of P_4 is 1. We observe that polytopes with smaller full Minkowski length have bigger minimum distances (at least if the size of the field q is sufficiently large).

Definition 4. If a lattice polytope P has full Minkowski length $l(P) = 1$ then it is called *strongly indecomposable*. For instance, primitive segments are the only one dimensional strongly indecomposable polytopes.

In this paper our main goal is to extend the work of Soprunov and Soprunova [5], in which they looked at strongly indecomposable polygons in $\mathbb{R}^2$ and their corresponding toric codes. We extend this work by trying to identify new examples of strongly indecomposable 3-dimensional polytopes, *polyhedra* in $\mathbb{R}^3$. We are also interested in the toric codes generated by these polyhedra.

2 Monomially equivalent codes

While looking at indecomposable polytopes, we found that it would be easier to organize them into families. However, we first define relationships between members of a given family.

Definition 5. We will say that two integral convex polytopes P_1 and P_2 in $\mathbb{R}^m$ are *lattice equivalent* if there exists an invertible integer affine transformation T such that $T(P_1) = P_2$. We may say interchangeably that P_1 is $AGL(m,\mathbb{Z})$-equivalent to P_2.

There is a corresponding equivalence relation between codes:

Definition 6. Let C_1 and C_2 be two codes of block length n and dimension k over $\mathbb{F}_q$. Let G_1 be a generator matrix for C_1. Then C_1 and C_2 are said to be *monomially equivalent* if there is an invertible $n \times n$ diagonal matrix D and an $n \times n$ permutation matrix Π such that

$$G_2 = G_1 D \Pi$$

is a generator matrix for C_2.

Looking at lattice equivalent polytopes, a correlation between them and their respective toric codes is described in the theorem below found in [2]:

Theorem 1. *If two polytopes P_1 and P_2 are lattice equivalent, then the corresponding toric codes C_{P_1} and C_{P_2} are monomially equivalent.*

A family of indecomposable polyhedra was found by Reeve and presented in [3].

Theorem 2. *Let τ_r be a tetrahedron whose vertices are the lattice points $(0,0,0)$, $(1,0,0),(0,1,0),(1,1,r)$. Then for all $r \in \mathbb{Z}^+$, τ_r is included in a family of strongly indecomposable tetrahedra.*

Proof. Consider the tetrahedra τ_s and τ_t such that $s,t \in \mathbb{Z}^+$ and $s \neq t$. The vertices for the tetrahedra are $(0,0,0),(1,0,0),(0,1,0),(1,1,s)$ and $(0,0,0),(1,0,0),(0,1,0)$, $(1,1,t)$ respectively. It is obvious τ_s and τ_t are not lattice equivalent since the volumes of the tetrahedra are different. To show that both τ_s and τ_t belong to a family of strongly indecomposable tetrahedra we must show that each tetrahedron has full Minkowski length equal to 1. Since all of the subpolytopes of τ_s are indecomposable with Minkowski length equal to 1, τ_s has full Minkowski length equal to 1. Similarly, τ_t has full Minkowski length equal to 1. Therefore, τ_s and τ_t belong to a family of strongly indecomposable tetrahedra. □

3 Indecomposable Polytopes

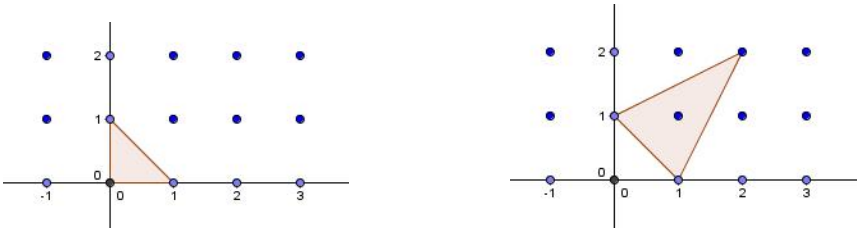
Following Reeve's example in [3], we seek other indecomposable polyhedra. In order to construct such polyhedra we first understand indecomposable polygons as discussed in [5].

Theorem 3 (Pick's Theorem)**.** *Let P be a lattice polygon in $\mathbb{R}^2$. Then the area of P is equal to*

$$A = I + \frac{B}{2} - 1,$$

where I in the number of interior lattice points and B is the number of boundary points in P.

Theorem 4 ([5], Theorem 1.4)**.** *Let P be a strongly indecomposable polygon. Then P is $AGL(2,\mathbb{Z})$-equivalent to either the 2-simplex Δ in Figure 4a or the "standard" exceptional triangle T_0 with vertices $(1,0),(0,1),(2,2)$, seen in Figure 4b, in its respective plane.*



(a) 2-simplex Δ (b) Exceptional T_0

Figure 4: Indecomposable Polygons

Proof. Let P be a strongly indecomposable polygon. We wish to show P must have 4 or fewer lattice points. Suppose x,$y \in P \cap \mathbb{Z}^2$ such that $x = (x_1, x_2)$ and $y = (y_1, y_2)$. If $x_i \equiv y_i \mod 2$, for $i = 1, 2$, then the lattice segment $[x, y]$ lies in P and is not primitive, so the full Minkowski length will be > 1. Because there exist only four pairs of remainders $\mod 2$, P will have at most four lattice points. Now let us consider the possible polygons P can be. Suppose P is a triangle with primitive sides. Let us consider the case where P has no interior lattice points. By Pick's Theorem, Theorem 3, we know that the area of this P will be $1/2$. This is the same as the area for our 2-simplex triangle Δ, so this triangle must be $AGL(2, \mathbb{Z})$-equivalent to Δ. Now let us consider the case where P is a triangle containing exactly one interior lattice point. Again using Formula 3, we can see that the area of this P will be $3/2$. Thus, this P is $AGL(2, \mathbb{Z})$-equivalent to the standard exceptional triangle T_0. Finally, suppose P is a quadrilateral. Then P has no interior lattice points and by Formula 3, the area of the quadrilateral is 1. So, this P will be $AGL(2, \mathbb{Z})$-equivalent to the unit square. But it is clear that the unit square is decomposable. Therefore, any strongly indecomposable polygon P must be $AGL(2, \mathbb{Z})$ - equivalent to either the standard 2-simplex Δ or the standard exceptional triangle T_0. □

We remark that there are far more than two indecomposable polytopes in 2 dimensions and can classify all of them as follows:

Definition 7. Any triangle that is lattice equivalent to the 2-simplex triangle is called a *unit triangle*. That is, the given triangle is $AGL(2, \mathbb{Z})$-equivalent to the 2-simplex triangle in its respective plane.

Definition 8. Any triangle that is lattice equivalent to the standard exceptional triangle T_0 with one interior point is called an *exceptional triangle*. Similarly to the above definition, the given triangle is $AGL(2, \mathbb{Z})$-equivalent to T_0 in its respective plane.

Lemma 1. *Each of the faces of a strongly indecomposable polyhedron P must either be an unit triangle or an exceptional triangle.*

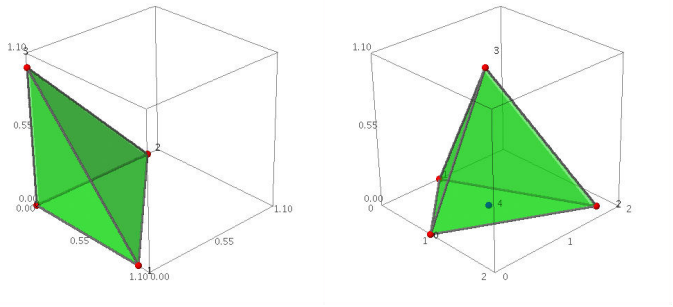
However, we must also consider polygons formed in the interior of the polyhedron, an issue addressed in the following corollary.

Corollary 1. *A 3-dimensional polyhedron P has full Minkowski length equal to 1 if all subsets of P with three lattice points are lattice equivalent to the standard 2-simplex Δ, and any four lattice points that are planar are lattice equivalent to the standard exceptional triangle T_0 in their respective planes.*

Proof. Let P be a 3-dimensional polyhedron with full Minkowski length equal to 1. It follows that the subsets of P have Minkowski length equal to 1. Clearly the cases of subsets with one or two points are both indecomposable. Assume $P_1 \subset P$ is a 2-dimensional polytope with three lattice points. Since P_1 has Minkowski length equal to 1, it must be $AGL(2, \mathbb{Z})$-equivalent to the standard 2-simplex Δ by Theorem 4. Now suppose $P_2 \subset P$ such that P_2 has four lattice points. P_2 cannot be a planar

quadrilateral because if it were it would have Minkowski length at least 2, which contradicts our hypothesis. So P_2 must be a 4-cycle with all the vertices not co-planar or $AGL(2,\mathbb{Z})$-equivalent to T_0. Consider the case where you have a non-planar 4-cycle. Each subset of the 4-cycle with three vertices must be $AGL(2,\mathbb{Z})$-equivalent to Δ. Therefore all subsets of P with with three or four lattice points are $AGL(2,\mathbb{Z})$-equivalent to the standard 2-simplex Δ or the T_0 respectively. □

4 Another Class of Tetrahedra



(a) (b)

Figure 5: Examples of indecomposable tetrahedra

Notice that Figure 5a is a member of the class of Reeve tetrahedra discussed in the previous section. Seeking other classes of indecomposable tetrahedra, we orient our standard exceptional triangle in the xy-plane and construct an indecomposable polyhedra around it. In Figure 5b we see such an indecomposable tetrahedra, whose lattice points are $(1,0,0),(0,1,0),(1,1,0),(2,2,0),(1,1,1)$. Note that the point $(1,1,0)$ is the interior point of the exceptional triangle.

Now that we have constructed a class of indecomposable polyhedra, we look at each of the member's corresponding toric code in $\mathbb{F}_8$. To do this we offer more specific definitions for equivalence relations between the polyhedra and their subsequent codes.

Definition 9. An *m-box* denoted $\square_{q-1}^m$ is an m-dimensional object where each dimension has $q-1$ units.

Within $\square_{q-1}^m$, determined by $\mathbb{F}_q$, a certain equivalence relation holds.

Definition 10. Let $\mathbf{v} \in (\mathbb{Z}_{q-1})^3$ be a vector and A be an invertible 3×3 matrix with integer entries *mod* $q-1$ such that $\det(A)$ is a unit *mod* $q-1$. Two subsets P_1, $P_2 \subset \square_{q-1}^m$ are *$AGL(3,\mathbb{Z}_{q-1})$-equivalent* if there exists a transformation $T : \mathbb{Z}_{q-1}^3 \to \mathbb{Z}_{q-1}^3$ such that $T(\mathbf{x}) = A\mathbf{x} + \mathbf{v}$ and satisfying $T(P_1) = P_2$.

As is the trend with our paper, we now discuss the correlation between $AGL(3,\mathbb{Z}_{q-1})$-equivalent polytopes and their corresponding codes.

Proposition 1. *If two subsets of $\square_{q-1}^m$, P_1 and P_2, are $AGL(m, \mathbb{Z}_{q-1})$-equivalent, then their toric codes C_1 and C_2 are monomially equivalent.*

Proof. Suppose we have two $AGL(m, \mathbb{Z}_{q-1})$-equivalent subsets of $\square_{q-1}^m$, P_1 and P_2. Both P_1 and P_2 contain integer lattice points corresponding to monomials of the form x^e where $e \in \mathbb{Z}_{q-1}^m$. By our hypothesis on P_1 and P_2, there exists an invertible transformation

$$T(\mathbf{x}) = A\mathbf{x} + \mathbf{v}$$

such that $T(P_1) = P_2$, where A is an element of $GL(m, \mathbb{Z}_{q-1})$ and $\det(A)$ is relatively prime to $q-1$. Let l be the number of lattice points in P_1, which equals the number of lattice points in P_2. Let $P_1 = \{e_i \mid i = 1, \ldots, l\}$. So, C_{P_1} is spanned by $ev(x^{e_i})$ for $1 \leq i \leq l$ and similarly C_{P_2} is spanned by $ev(x^{T(e_i)})$. Let α be a primitive element in ${\mathbb{F}_q}^*$. Define $\alpha^f = (\alpha^{f_1}, \ldots, \alpha^{f_m}) \in ({\mathbb{F}_q}^*)^m$. The component of $ev(x^{e_i}) \in C_{P_1}$ corresponding to α^f is $\alpha^{\langle e_i, f\rangle}$ where $\langle e_i, f\rangle$ is the usual dot product. The corresponding entry in the codeword $ev(x^{T(e_i)})$ in C_{P_2} is $\alpha^{\langle T(e_i), f\rangle}$. This can be written as

$$\alpha^{\langle Ae_i + \mathbf{v}, f\rangle} = \alpha^{\langle Ae_i, f\rangle} \cdot \alpha^{\langle \mathbf{v}, f\rangle}.$$

The second term of the product is not dependent on e_i. These nonzero scalars are the diagonal entries in the matrix D as in the definition of monomially equivalent codes. By a standard property of dot products,

$$\alpha^{\langle Ae_i, f\rangle} = \alpha^{\langle e_i, A^T f\rangle}.$$

The transposed matrix A^T also defines a bijective mapping from $\mathbb{Z}_{q-1}^m$ to $\mathbb{Z}_{q-1}^m$ since $\det(A^T) = \det(A)$. Now we must show that A^T induces a permutation of $({\mathbb{F}_q}^*)^m$. For some $g \in \mathbb{Z}_{q-1}^m$, suppose $A^T f \equiv A^T g \bmod q-1$. Since $\det(A^T)$ is a unit in $\mathbb{Z}_{q-1}$, we know that A^T is invertible and $(A^T)^{-1} \in GL(m, \mathbb{Z}_{q-1})$. So we can multiply by $(A^T)^{-1}$ on the left. Hence $f \equiv g \bmod q-1$ and A^T defines a permutation of the points α^f as we wanted. Note that A^T permutes all of the codewords in the same way. This gives the permutation matrix Π. Hence C_{P_1} is monomially equivalent to C_{P_2}. □

5 Classes of Hexahedra

In order to construct more indecomposable polyhedra, we adopt a different approach.

Definition 11. Let P be a polyhedron with $\mathbf{x}_1, \mathbf{x}_2 \in P$ and $\mathbf{v} \in \mathbb{Z}^m$ be any primitive vector with entries relatively prime to one another. Then

$$\max_{\mathbf{x_1} \in P} \langle \mathbf{x}_1, \mathbf{v}\rangle - \min_{\mathbf{x_2} \in P} \langle \mathbf{x}_2, \mathbf{v}\rangle = \text{width in } \mathbf{v} \text{ direction},$$

so the *lattice width* is

$$width(P) = \min_{\mathbf{v} \in \mathbb{Z}^m} (\max\langle \mathbf{x}_1, \mathbf{v}\rangle - \min\langle \mathbf{x}_2, \mathbf{v}\rangle).$$

This is a key definition because we want restrictions that prohibit interior points. We require that our polyhedra have lattice width 1. For instance, we wish to remain between the xy and $z = 1$ planes while constructing our polyhedra.

Theorem 5. *If P is a triangle in a plane S and $P \cap \mathbb{Z}^3$ consists only of the vertices of P, then P is strongly indecomposable.*

Proof. Assume P is a triangle in a plane S and $P \cap \mathbb{Z}^3$ consists of only the three vertices of the polygon. There is no Minkowski sum of two or more polytopes that will result in the desired P or any subpolygon of P. Thus, the Minkowski length is 1. Therefore, by Definition 4 the triangle is strongly indecomposable. □

Theorem 6. *The hexahedron in Figure 6, based on the 2-simplex triangle Δ, is strongly indecomposable.*

Proof. We first look at the points of the 2-simplex hexahedron: $(0,0,0), (1,0,0)$, $(0,1,0), (0,0,1), (1,1,1)$. We separate the points into two groups: the points that lie in the xy-plane and the points that lie in the $z = 1$ plane. We take any set of four points and ensure that not all members of the set lie in one plane (thus creating a quadrilateral). In order to ensure no set of four is co-planar, we take all pairs of points in the xy-plane and find their xy-slopes to be $undefined$, 0 and -1. Then we compare those slopes to the xy-slope of the two points contained in the $z = 1$ plane, which equals 1. No two slopes are the same, so no four points lie in one plane. Therefore, any plane created with three points from this hexahedron will not contain any other points from the hexahedron. By Theorem 5 this hexahedron must be indecomposable. □

We say the indecomposable hexahedron in Figure 6 is "based on the 2-simplex triangle Δ" because we began its construction by orienting Δ in the xy-plane.

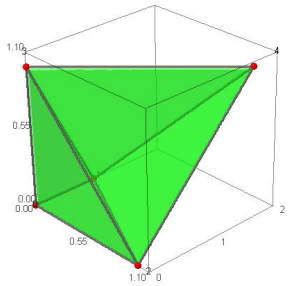


Figure 6: An indecomposable hexahedron based on the 2-simplex triangle

Figure 6 belongs to a class of hexahedra having lattice points that look like $(0,0,0), (1,0,0), (0,1,0), (0,0,1), (r,1,1)$, where $r \neq 0$, referred to as the unit triangle hexahedra class. The reason we prohibit r from being 0 is to avoid having four points in the same plane, resulting in a quadrilateral and thus a full Minkowski length > 1. In the same fashion, one could create more members of this class. That is, using triangles that are lattice equivalent to the 2-simplex triangle as bases.

Within this class, we see that under $\mathbb{F}_8$, there are three families of $AGL(3, \mathbb{Z}_{q-1})$-equivalent hexahedra. One of the families includes the unit triangle hexahedra where $r = 1, 3, 5$, the second contains $r = 2, 4$, and the last family contains $r = 6$. This implies that the codes generated from members within each family will be monomially equivalent to one another.

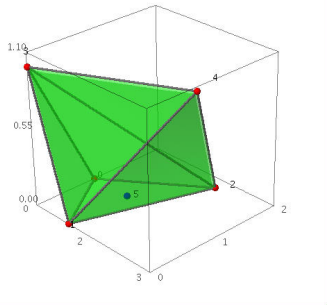


Figure 7: An indecomposable hexahedron based on the standard exceptional triangle

Theorem 7. *The hexahedron based on the standard exceptional triangle in Figure 7 is strongly indecomposable.*

Proof. Similar to the previous proof, we look at the lattice points of this hexahedron: $(0,1,0), (1,0,0), (1,1,0), (2,2,0), (0,0,1), (3,1,1)$ We already know there are four points in the xy-plane, but we know the polygon created is the exceptional triangle, therefore that face is indecomposable. We again separate the points into two groups: points that lie in the xy-plane and points that lie in the $z = 1$ plane and we look at the xy slopes created between any two points. The xy-slopes in the xy-plane are $undefined, -1, 0, 1/2, 1$ and 2. There are only two points in the $z = 1$ plane and the xy-slope between them is $1/3$. No two slopes are the same, so no four points lie in one plane. Therefore, any plane (excluding the xy plane) created with three points from this hexahedron will not contain any other points from the exceptional hexahedron. By Theorem 5 this hexahedron must be in decomposable. □

Figure 7 is included in the class of hexahedra based on an exceptional triangle, whose members' points resemble $(0,1,0), (1,0,0), (1,1,0), (2,2,0), (0,0,1), (r,1,1)$ where $r \neq 0, \pm 1, 2$ and $(r,1,1) \in \square_3$. For our specific figure $r = 3$.

Theorem 8. *There are no strongly indecomposable lattice pentahedra.*

Proof. Aiming for a contradiction, suppose we have a strongly indecomposable pentahedron Q with lattice points as vertices. There are only two combinatorially distinct ways to construct convex pentahedra: polyhedra with the same numbers of vertices, edges, and faces as a square pyramid or a triangular prism. We note that we must restrict the faces in order to have lattice points as vertices and exclude boundary points on the edges of Q. In both cases, Lemma 1 is not satisfied because at least one of the faces of each type is a quadrilateral. Therefore not all of the subpolytopes

of a given pentahedron Q will have Minkowski length 1, causing $l(Q) > 1$. This contradicts the definition of strongly indecomposable polytope. Hence there are no strongly indecomposable lattice pentahedra. □

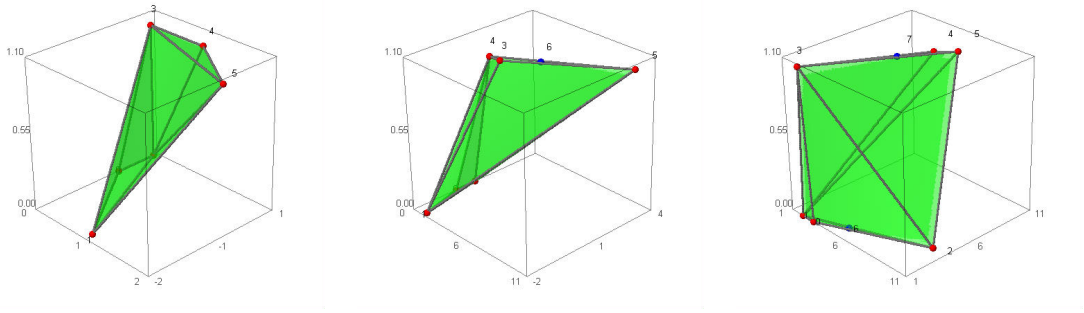
6 Heptahedra and Octahedra

Following the methods loosely outlined in the previous section, we seek even more strongly indecomposable polyhedra. Namely we are concerned with finding polyhedra with more than six faces. Note that a strongly indecomposable polyhedra with seven faces is not attainable as stated in the following:

Theorem 9. *There are no strongly indecomposable lattice heptahedra.*

Proof. In the same fashion as the proof of Theorem 8, suppose, by way of contradiction, that we have a strongly indecomposable heptahedron H with lattice points as vertices. There are thirty-four distinct types of heptahedra, not including mirror-images. In all of these types of heptahedra, Lemma 1 is not satisfied because all of them contain a quadrilateral, pentagon, or a hexagon as a face. Therefore not all of the subpolytopes of a given heptahedron H will have Minkowski length 1, causing $l(H) > 1$. This contradicts the definition of strongly indecomposable. Hence there are no strongly indecomposable lattice heptahedra. □

In Figure 8 we observe three indecomposable octahedra.



(a) An octahedron with two 2-simplex triangles

(b) An octahedron with a 2-simplex triangle and an exceptional triangle

(c) An octahedron with two exceptional triangles

Figure 8: Pictures of octahedra

Theorem 10. *The octahedron whose top and bottom bases are lattice equivalent to the 2-simplex triangle in Figure 8a is a strongly indecomposable polyhedron.*

Proof. We look at the points of this octahedron: $(0,0,0), (1,-2,0), (0,1,0), (0,1,1)$, $(1,1,1), (2,0,1)$. We separate the points into two groups: points in the xy-plane and points in the $z = 1$ plane. We look at the xy slopes created by any two points within each group. The xy-slopes in the xy-plane are $undefined$, -3, and -2. The

xy-slopes in the $z = 1$ plane are -1, $-1/2$ and 0. Since no two slopes are the same, no four points lie in one plane, including both the xy- and $z = 1$ planes since we already know there are three points in both of these. Therefore, any plane created with three points from this octahedron will not contain any other points from our 2-simplex octahedron. By Theorem 5 this octahedron must be indecomposable. □

Theorem 11. *The octahedron with one base lattice equivalent to the 2-simplex triangle and the other lattice equivalent to the standard exceptional triangle in Figure 8b is a strongly indecomposable polyhedron.*

Proof. We start by looking at the points of this octahedron: $(0, 0, 0), (1, -2, 0)$, $(0, 1, 0), (3, 1, 1), (2, 1, 1), (10, 4, 1), (5, 2, 1)$. We separate the points into two groups: points in the xy-plane and points in the $z = 1$ plane. We look at the xy-slopes created by any two points within each group. The xy-slopes in the xy-plane are $undefined$, -3 and -2. The xy-slopes in the $z = 1$ plane are $0, 3/8, 3/7, 1/3, 2/5$, and $1/2$. Since no two slopes are the same, no four points lie in the same plane, excluding the exceptional triangle we were already aware of in the $z = 1$ plane and we already know this face is indecomposable. Therefore, any plane created with three points from this octahedron (excluding the $z = 1$ plane) will contain no other points from the octahedron, except for the three points that create the vertices of our exceptional triangle. By Theorem 5 this octahedron must be indecomposable. □

Theorem 12. *The octahedron whose top and bottom bases are lattice equivalent to the standard exceptional triangle in Figure 8c is a strongly indecomposable polyhedron.*

Proof. We start by looking at the points of this octahedron: $(3, 1, 0), (2, 1, 0), (10, 4, 0)$, $(5, 2, 0), (1, 2, 1), (5, 9, 1), (6, 10, 1), (4, 7, 1)$. We separate the points into two groups: points in the xy-plane and points in the $z = 1$ plane. There are four points in both of these planes, but we already know they are indecomposable since they are lattice equivalent to the standard exceptional triangle. We then look at the xy-slopes created by any two points within each group. The xy-slopes in the xy-plane are $0, 3/8, 3/7, 1/3, 2/5$, and $1/2$. The xy-slopes in the $z = 1$ plane are $1, 3/2, 8/5, 5/3, 7/4$, and 2. Since no two slopes are the same, no four points lie in the same plane, excluding the xy-plane and $z = 1$ plane. Therefore, excluding the horizontal planes, any plane created with three points from this octahedron will not contain any other points from the exceptional octahedron besides those three. By Theorem 5 this octahedron must be indecomposable. □

7 Parameters of Toric Codes over $\mathbb{F}_8$

	Tetrahedra				**Hexahedra**			
	2-Simplex P_1		Exceptional P_2		2-Simplex P_3		Exceptional P_4	
r	C_P	$[n,k,d]$	C_P	$[n,k,d]$	C_P	$[n,k,d]$	C_P	$[n,k,d]$
1	C_{P_1}	[343, 4, 294]	C_{P_2}	[343, 5, 245]	$C_{P_{3a}}$	[343, 5, 291]		
2	C_{P_1}	[343, 4, 294]	C_{P_2}	[343, 5, 245]	$C_{P_{3b}}$	[343, 5, 283]		
3	C_{P_1}	[343, 4, 294]	C_{P_2}	[343, 5, 245]	$C_{P_{3a}}$	[343, 5, 291]	$C_{P_{4a}}$	[343, 6, 276]
4	C_{P_1}	[343, 4, 294]	C_{P_2}	[343, 5, 245]	$C_{P_{3b}}$	[343, 5, 283]	$C_{P_{4b}}$	[343, 6, 252]
5	C_{P_1}	[343, 4, 294]	C_{P_2}	[343, 5, 245]	$C_{P_{3a}}$	[343, 5, 291]	$C_{P_{4a}}$	[343, 6, 276]
6	C_{P_1}	[343, 4, 294]	C_{P_2}	[343, 5, 245]	$C_{P_{3c}}$	[343, 5, 252]	$C_{P_{4b}}$	[343, 6, 252]

Octahedra					
Two 2-Simplex P_5		2-Simplex and Exceptional P_6		Two Exceptional P_7	
C_P	$[n,k,d]$	C_P	$[n,k,d]$	C_P	$[n,k,d]$
C_{P_5}	[343, 6, 283]	C_{P_6}	[343, 7, 252]	C_{P_7}	[343, 8, 245]

8 Future work

We hope to expand our research by proving that families within the classes of unit triangle octahedra and exceptional triangle octahedra exist. We intend to do this by making generalizations about coordinates of the members' lattice point(s). Once we are able to show such families exist, we can make generalizations about octahedra, similar to that of our tetrahedra and hexahedra families. Furthermore, we would like to explore the toric codes of these octahedra and determine, if possible, any equivalence relations.

9 Acknowledgments

This work was conducted during the 2009 Mathematical Sciences Research Institute Undergraduate Program (MSRI-UP) in Berkeley, CA. MSRI-UP is supported by the National Science Foundation (grant No. DMS-0754872) and the National Security Agency (grant No. H98230-09-0103). We would like to thank our MSRI-UP director Dr. Herbert Medina, our research professor Dr. John Little, and most of all our amazing advisor Candice Price.

References

[1] V. Diaz, C. Guevera, M. Vath, "Codes from n-Dimensional Cyclic Codes," *Proceedings of Summer Institute in Mathematics for Undergraduates (SIMU), 2001*, Humacao, Puerto Rico.

[2] J. Little, R. Schwarz, "On toric codes and multivariate Vandermonde matrcies," *Appl. Alg. Engrg. Comm. Comput.* **18** (2007), 349–367.

[3] J. E. Reeve, "On the volume of lattice polyhedra," *Proc. London Math. Soc.* **(3)**, 7 (1957), 378–395.

[4] J. E. Reeve, "A further note on the volume of lattice polyhedra," *Proc. London Math. Soc.* **(34)** (1959), 57–62.

[5] I. Soprunov, J. Soprunova, "Toric surface codes and Minkowski length of polygons," *SIAM J. Discrete Math* **(23)** (2009), 384–400.

Multivariate Vandermonde determinants and toric codes

Leyda Almodóvar
University of Puerto Rico at Mayagüez

Eugene Cody
Phoenix College

Lourdes Morales
University of Puerto Rico at Río Piedras

July 2009

Abstract

The minimum distances of toric codes has been studied extensively for various forms of polytopes. In [2], the authors determine bounds for the minimum distance of toric codes for some polytopes $P \subseteq \mathbb{R}^m$ including the simplices of the form $\text{conv}(0, \ell e_1, \ldots, \ell e_n)$ using Vandermonde determinants. In this paper, we will derive lower and upper bounds to prove the exact minimum distance of some toric codes associated to the special polytopes $P = \text{conv}(0, \ell e_1, 2\ell e_2, 3\ell e_3) \subset \mathbb{R}^3$.

1 Introduction

Following Hansen, in [2], J. Little and R. Schwarz define toric codes using elements of algebraic geometry. A toric codeword is formed by evaluating monomials corresponding to integer lattice points in a convex polytope. In [2], a formal definition of a toric code is given:

Definition 1. Let $\mathbb{F}_q$ be a finite field with primitive element α. For $f \in \mathbb{Z}^m$ with $0 \leq f_i \leq q-2$ for all i, let $p_f = (\alpha^{f_1}, \ldots, \alpha^{f_m})$ in $(\mathbb{F}_q^*)^m$. For any $e = (e_1, \ldots, e_m) \in P \cap \mathbb{Z}^m$, let $x^e = x_1^{e_1} \ldots x_m^{e_m}$ be the corresponding monomial and write

$$(p_f)^e = (\alpha^{f_1})^{e_1} \cdots (\alpha^{f_m})^{e_m}.$$

The toric code $C_P(\mathbb{F}_q)$ over the field $\mathbb{F}_q$ associated to P is the linear code of block length $n = (q-1)^m$ with generator matrix

$$G = ((p_f)^e),$$

where the rows are indexed by the $e \in P \cap \mathbb{Z}^m$, and the columns are indexed by the $p_f \in (\mathbb{F}_q^*)^m$. In other words, letting $L = \text{Span}\{x^e : e \in P \cap \mathbb{Z}^m\}$, we define the

evaluation mapping

$$\begin{aligned} ev : L &\rightarrow \mathbb{F}_q^{(q-1)^m} \\ g &\mapsto (g(p_f) : p_f \in (\mathbb{F}_q^*)^m). \end{aligned}$$

Then, $C_P(\mathbb{F}_q) = ev(L)$. If the field is clear from the context, we will often omit it in the notation and simply write C_P. The matrix G will be called the standard generator matrix for the toric code.

The approach in [2] to finding the minimum distance of toric codes was based on studying the determinants of maximal square submatrices of the standard generator matrix G of the toric code. The authors provided a method based on multivariate generalization of Vandermonde determinants that applies equally to many toric codes associated to all integral convex polytopes $P \subseteq \mathbb{R}^m$ where $m \geq 2$. Their first step was to determine a suitable set S where the determinant for the Vandermonde matrix associated to P, where P is a rectangular solid or simplex, factored in a nice way to ensure a non-zero Vandermonde determinant. Their second step was to argue that there were enough suitable sets S to yield their desired conditions for determining the bounds for minimum distance. Our main goal in this paper is to extend the methods and results from [2] to more general families of simplices, specifically the family of simplices of the form $\ell P = \text{conv}\{0, \ell e_1, 2\ell e_2, 3\ell e_3\}$ for $\ell \geq 1$ in $\mathbb{R}^3$.

In our main results we begin with a theorem that gives the exact number of integer lattice points in the tetrahedron P. In the section §3.1 of our paper we offer a method to construct a suitable set S that yields the conditions for determining the minimum distance which leads to Proposition 5 and Proposition 7 in §3.2 that give lower bounds for the minimum distance of toric codes associated to the tetrahedra when $\ell = 1$ and $\ell = 2$. Furthermore, in §3.2 we use the monomials determined by the integer lattice points in the tetrahedron to formulate propositions that give an upper bound for the minimum distance of toric codes associated to tetrahedron for all ℓ. Finally we offer two theorems for the exact minimum distance of toric codes associated to the tetrahedron for $\ell = 1$ and $\ell = 2$.

In this paper we will use the following notations: Suppose that $P \subset \square_{q-1} \subset \mathbb{R}^m$ is an integral convex polytope for $m \geq 2$. We will write $\#(P)$ for the number of integer lattice points in P (that is, $\#(P) = |P \cap \mathbb{Z}^m|$)). We will write

$$P \cap \mathbb{Z}^m = \{e(i) : 1 \leq i \leq \#(P)\}$$

for the set of those integer lattice points. Also, for any set $A \subset \mathbb{R}^m$, conv(A) denotes the convex hull of A.

2 Preliminaries

2.1 Vandermonde Matrices

In [2], the authors apply the concept of determinants of Vandermonde matrices to study the minimum distance where the Vandermonde determinants give bounds for

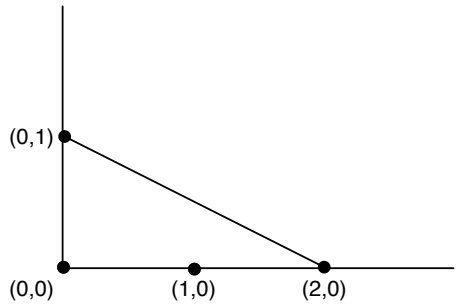


Figure 1: $P = \text{conv}\{(0,0),(2,0),(0,1)\}$ in $\mathbb{R}^2$

the minimum distance of toric codes associated to some polytopes $P \subseteq \mathbb{R}^m$ including the simplices of the form $\text{conv}(0, \ell e_1, \ldots, \ell e_n)$. We begin by defining the Vandermonde matrix for the univariate and multivariate cases.

Definition 2. A $n \times n$ univariate Vandermonde matrix has the form

$$V(a_1, a_2, \ldots, a_n) = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ a_1 & a_2 & \cdots & a_n \\ a_1^2 & a_2^2 & \cdots & a_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{n-1} & a_2^{n-1} & \cdots & a_n^{n-1} \end{pmatrix},$$

where the $a_1, a_2, \ldots, a_n$ are *distinct* elements of a field $\mathbb{F}$. The univariate Vandermonde determinant is of the form

$$\det V(a_1, a_2, \ldots, a_n) = \prod_{1 \le i < j \le n} (a_j - a_i).$$

Since the elements in $\mathbb{F}$ are distinct, $\det V(a_1, a_2, \ldots, a_n) \neq 0$.

Now we will introduce a multivariate generalization of these determinants. So using the notation introduced in §1, let P be an integral convex polytope, and suppose $P \cap \mathbb{Z}^m = \{e(i) : 1 \le i \le \#(P)\}$, listed in some order. Let $S = \{p_j : 1 \le j \le \#(P)\}$ be any set of $\#(P)$ points in $(\mathbb{F}_q^*)^m$, also ordered.

Definition 3. The multivariate Vandermonde matrix associated to P and S is the $\#(P) \times \#(P)$ matrix $V(P;S) = (p_j^{e(i)})$ where we use the standard multi-index notation $p_j^{e(i)}$ to indicate the value of the monomial $x^{e(i)}$ at the point p_j.

Also, notice that the multivariate Vandermonde determinant is determined by the exponent vectors $e \in P \cap \mathbb{Z}^m$ and the set $S \subset (\mathbb{F}_q^*)^m$ with $|S| = |\#(P)|$.

Here is an example of a simple multivariate Vandermonde matrix.

Example 1. Suppose $P = \text{conv}\{(0,0),(2,0),(0,1)\}$ in $\mathbb{R}^2$ (Figure 1) where $\#(P) = 4$, and $S = \{(x_j, y_j)\}$ is any set of 4 points in $(\mathbb{F}_q^*)^2$. The corresponding $\#(P) \times \#(P)$ multivariate Vandermonde matrix is

$$V(P;S) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ x_1^2 & x_2^2 & x_3^2 & x_4^2 \end{pmatrix}.$$

If x_1, x_2, x_3 are distinct, but $x_4 = x_1$ and $y_4 \neq y_1$, then it follows that $\det V(P;S) \neq 0$.

2.2 Minimum Distance via Vandermonde Matrices

In [2], the authors show that Vandermonde determinants may be used to bound the minimum distance of toric codes. The Proposition 2.1 from [2] gives the lower bound and is stated as follows:

Proposition 1. *Let $P \subset \mathbb{R}^m$ be an integral convex polytope. Let d be a positive integer and assume that in every set $T \subset (\mathbb{F}_q^*)^m$ with $\mid T \mid = (q-1)^m - d + 1$ there exists some $S \subset T$ with $\mid S \mid = \#(P)$ such that $\det V(P;S) \neq 0$. Then the minimum distance satisfies $d(C_P) \geq d$.*

Later we will use this proposition to prove the lower bound for the minimum distance and to prove the exact minimum distance for some polytopes.

2.3 The Ehrhart Polynomial

The problem of counting lattice points $\#(P)$ has been studied for various polytopes. The Vandermonde matrices in this paper rely on $\#(P)$, therefore we use a form of the Ehrhart polynomial to count the total number of lattice points. The Ehrhart polynomial is a method that gives the number of integer lattice points contained in the polytope ℓP, where P is a polytope and ℓ is a non-negative integer. The Ehrhart polynomial, $E_P(\ell)$, is given by

$$E_P(\ell) = V(P)\ell^3 + A(P)\ell^2 + a_1\ell + 1 \tag{1}$$

where $V(P)$ is the volume of the polytope, $A(P)$ is the sum of the lattice areas of plane faces in the polytope[1] and a_1 is some rational number (see [1]). It must be stated that equation (1) is valid *only* for polyhedra in $\mathbb{R}^3$.

[1]Pick's theorem states that for ordinary lattice points in $\mathbb{R}^2$, $A = \frac{1}{2}B + I - 1$ where A is area, B is the number of boundary lattice points and I is the number of interior lattice points. The area $A(P)$ used here is an extension of Pick's theorem from $\mathbb{R}^2$ to $\mathbb{R}^3$, hence $A(P)$ is the summation of lattice areas of plane faces in the polytope in $\mathbb{R}^3$ (see [3]).

3 Main results

As we studied toric codes associated to the tetrahedra ℓP for $P = \text{conv}\{0, e_1, 2e_2, 3e_3\}$ we determined lower bounds for the minimum distances by using the Vandermonde determinant approach and Proposition 1. Also,we were able to determine upper bounds for the minimum distances by looking at the codewords.

We begin this section with the following theorem.

Theorem 1. *For the polytope $P = \text{conv}\{0, e_1, 2e_2, 3e_3\}$,the number of integer lattice points is*

$$\#(\ell P) = (\ell + 1)^3.$$

Proof. We use the Ehrhart polynomial (1) defined in §2.3. The volume is found for the base polytope (where $\ell = 1$) through the formula $V = \frac{1}{3}hb$ where h is the height and b is the base to give $V(P) = 1$. Using a Maple algorithm (see §6.1), the number of lattice points in $P = \text{conv}\{0, \ell e_1, 2\ell e_2, 3\ell e_3\}$ was found to be 8 for $\ell = 1$ and 27 for $\ell = 2$. Substituting the corresponding values of $\#(P)$ in the Ehrhart polynomial (1) gives the system of equations

$$8 = V(P)(1)^3 + A(P)(1) + a_1 + 1 \tag{2}$$

$$27 = V(P)(2)^3 + A(P)(2) + a_1 + 1 \tag{3}$$

Solving for the variables $A(P)$ and a_1 by substituting $V(P) = 1$ in both (2) and (3), $\ell = 1$ in (2), and $\ell = 2$ in (3) gives the rational coefficients $A(P) = a_1 = 3$.

Thus, the Ehrhart polynomial $E_P(\ell) = \ell^3 + 3\ell^2 + 3\ell + 1$ is factored to give $(\ell+1)^3$.

$$\therefore \#(\ell P) = (\ell + 1)^3. \qquad \square$$

3.1 Constructing the set S

Here we have two examples of how we construct a set S such that the $\det V(P; S)$ $\neq 0$ for the tetrahedron $P = \text{conv}\{0, e_1, 2e_2, 3e_3\}$ and $2P = \text{conv}\{0, 2e_1, 4e_2, 6e_3\}$ by looking at the planes that form the tetrahedra.

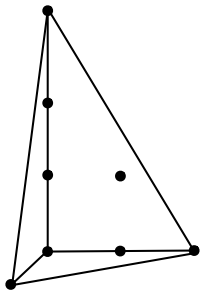


Figure 2: $P = \text{conv}\{0, e_1, 2e_2, 3e_3\}$.

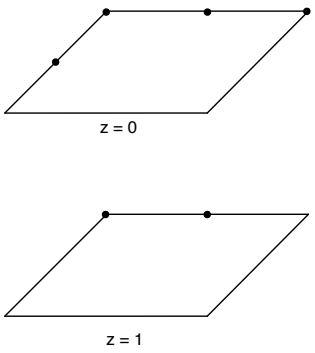


Figure 3: Planes $z = 0$ and $z = 1$ for $P = \text{conv}\{0, e_1, 2e_2, 3e_3\}$.

Example 2. Let $P = \text{conv}\{0, e_1, 2e_2, 3e_3\}$ in $\mathbb{R}^3$ (see Figure 2). Then, $P \bigcap \mathbb{Z}^3 = \{(0,0,0), (0,1,0), (0,2,0), (1,0,0), (0,1,1), (0,0,1), (0,0,2), (0,0,3)\}$.

On the plane $z = 0$ we have four points (see Figure 3):

$$P_0 = \{(0,0,0), (0,1,0), (0,2,0), (1,0,0)\}.$$

As we can see there are three points on the vertical line $x = 0$, and one on the vertical line $x = 1$. This means that there are two different first coordinates and three different second coordinates in this plane, so the first four points in S are $S_0 = \{(x_1, y_1, z_1), (x_1, y_2, z_1), (x_1, y_3, z_1), (x_2, y_1, z_1)\}$, where $x_1 \neq x_2$, and $y_1 \neq y_2 \neq y_3$.

On the plane $z = 1$ we have two points (see Figure 3): $P_1 = \{(0,0,1), (0,1,1)\}$. This means that there are two different second coordinates in this plane, so the next two points in S are $S_1 = \{(x_3, y_4, z_2), (x_3, y_5, z_2)\}$, where $y_4 \neq y_5$.

On the plane $z = 2$ we have one point: $P_2 = \{(0,0,2)\}$ and on the plane $z = 3$ we have one point: $P_3 = \{(0,0,3)\}$.

So the set S will end up looking like $S = \{(x_1, y_1, z_1), (x_1, y_2, z_1), (x_1, y_3, z_1), (x_2, y_1, z_1), (x_3, y_4, z_2), (x_3, y_5, z_2), (x_4, y_6, z_3), (x_5, y_7, z_4)\}$, where $z_1 \neq z_2 \neq z_3 \neq z_4$, $x_1 \neq x_2$, $y_1 \neq y_2 \neq y_3$ and $y_4 \neq y_5$.

Proposition 2. *Let S be the set stated above. Then* $\det V(P; S) \neq 0$.

Proof. Notice that the determinant for

$$V(P;S) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ y_1 & y_2 & y_3 & y_1 & y_4 & y_5 & y_6 & y_7 \\ y_1^2 & y_2^2 & y_3^2 & y_1^2 & y_4^2 & y_5^2 & y_6^2 & y_7^2 \\ x_1 & x_1 & x_1 & x_2 & x_3 & x_3 & x_4 & x_5 \\ y_1 z_1 & y_2 z_1 & y_3 z_1 & y_1 z_1 & y_4 z_2 & y_5 z_2 & y_6 z_3 & y_7 z_4 \\ z_1 & z_1 & z_1 & z_1 & z_2 & z_2 & z_3 & z_4 \\ z_1^2 & z_1^2 & z_1^2 & z_1^2 & z_2^2 & z_2^2 & z_3^2 & z_4^2 \\ z_1^3 & z_1^3 & z_1^3 & z_1^3 & z_2^3 & z_2^3 & z_3^3 & z_4^3 \end{pmatrix}.$$

is $-(z_1 - z_2)^2(y_2 - y_3)(y_1 - y_3)(y_1 - y_2)(z_2 - z_4)(-z_4 + z_1)(z_2 - z_3)(-z_3 + z_1)(z_3 - z_4)$ $(y_4 - y_5)(-x_2 + x_1)$. So the $\det V(P; S) \neq 0$ since $z_1 \neq z_2 \neq z_3 \neq z_4$, $x_1 \neq x_2$, $y_1 \neq y_2 \neq y_3$ and $y_4 \neq y_5$. □

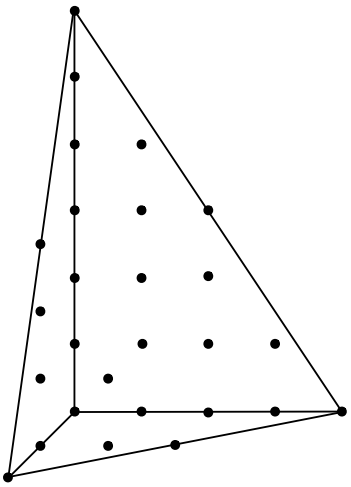


Figure 4: $2P = \text{conv}\{0, 2e_1, 4e_2, 6e_3\}$

Example 3. Let $2P = \text{conv}\{0, 2e_1, 4e_2, 6e_3\}$ in $\mathbb{R}^3$ (see Figure 4).

Then, $P \cap \mathbb{Z}^3 = \{(0,0,0), (0,0,1), (0,0,2), (0,0,3), (0,0,4), (0,0,5), (0,0,6),$
$(0,1,0), (0,1,1), (0,1,2), (0,1,3), (0,1,4), (0,2,0), (0,2,1),$
$(0,2,2), (0,2,3), (0,3,0), (0,3,1), (0,4,0), (1,0,0), (1,0,1),$
$(1,0,2), (1,0,3), (1,1,0), (1,1,1), (1,2,0), (2,0,0)\}$

and $|S| = 27$.

On the plane $z = 0$ we have nine points (see Figure 5): $P_0 = \{(0,0,0), (0,1,0), (0,2,0), (0,3,0), (0,4,0), (1,0,0), (1,1,0), (1,2,0), (2,0,0)\}$. As we can see there are five points on the vertical line $x = 0$, three points on the vertical line $x = 1$ and one on the vertical line $x = 3$. This means that there are three different first coordinates and five different second coordinates in this plane, so the first nine points in S are $S_0 = \{(x_1, y_1, z_1), (x_1, y_2, z_1), (x_1, y_3, z_1), (x_1, y_4, z_1), (x_1, y_5, z_1), (x_2, y_1, z_1), (x_2, y_2, z_1), (x_2, y_3, z_1), (x_3, y_1, z_1)\}$, where $x_1 \neq x_2 \neq x_3$, and $y_1 \neq y_2 \neq y_3 \neq y_4 \neq y_5$.

On the plane $z = 1$ we have six points (see Figure 5): $P_1 = \{(0,0,1), (0,1,1), (0,2,1), (0,3,1), (1,0,1), (1,1,1)\}$. As we can see there are four points on the vertical line $x = 0$ and two on the vertical line $x = 1$. This means that there are two different first coordinates and four different second coordinates in this plane, so the next six points in S are: $S_1 = \{(x_4, y_6, z_2), (x_4, y_7, z_2), (x_4, y_8, z_2), (x_4, y_9, z_2), (x_5, y_6, z_2), (x_5, y_7, z_2)\}$, where $x_4 \neq x_5$, and $y_6 \neq y_7 \neq y_8 \neq y_9$.

On the plane $z = 2$ we have four points (see Figure 5): $P_2 = \{(0,0,2), (0,1,2), (0,2,2), (1,0,2)\}$. As we can see there are three points on the vertical line $x = 0$ and one on the vertical line $x = 1$. This means that there are two different first coordinates and three different second coordinates in this plane, so the next four points in S are: $S_2 = \{(x_6, y_{10}, z_2), (x_6, y_{11}, z_2), (x_6, y_{12}, z_2), (x_7, y_{10}, z_2)\}$, where $x_6 \neq x_7$, and $y_{10} \neq y_{11} \neq y_{12}$.

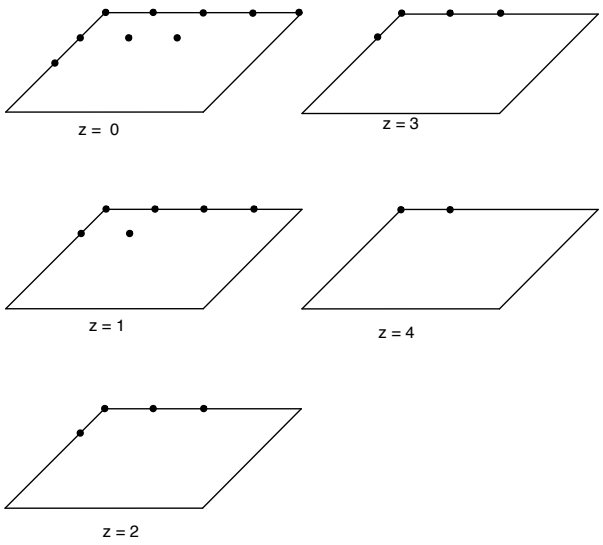


Figure 5: Planes $z = 0, z = 1, z = 2, z = 3$ and $z = 4$ for $2P = \text{conv}\{0, 2e_1, 4e_2, 6e_3\}$

On the plane $z = 3$ we have four points (see Figure 5): $P_3 = \{(0,0,3), (0,1,3), (0,2,3), (1,0,3)\}$. As we can see there are three points on the vertical line $x = 0$ and one on the vertical line $x = 1$. This means that there are two different first coordinates and three different second coordinates in this plane, so the next four points in S are: $S_3 = \{(x_8, y_{13}, z_3), (x_8, y_{14}, z_3), (x_8, y_{15}, z_3), (x_9, y_{13}, z_3)\}$, where $x_8 \neq x_9$, and $y_{13} \neq y_{14} \neq y_{15}$.

On the plane $z = 4$ we have two points (see Figure 5): $P_4 = \{(0,0,4), (0,1,4)\}$. This means that there are two different second coordinates in this plane, so the next two point in S are: $S_4 = \{(x_{10}, y_{16}, z_4), (x_{11}, y_{17}, z_4)\}$, where $y_{16} \neq y_{17}$.

On the plane $z = 5$ we have one point: $P_5 = \{(0,0,5)\}$ and on the plane $z = 6$ we have one point: $P_6 = \{(0,0,6)\}$.

So, the set S will end up looking like this:

$$\begin{aligned} S \;=\; & \{(x_1,y_1,z_1),(x_1,y_2,z_1),(x_1,y_3,z_1),(x_1,y_4,z_1),(x_1,y_5,z_1),(x_2,y_1,z_1),\\ & (x_2,y_2,z_1),(x_2,y_3,z_1),(x_3,y_1,z_1),(x_4,y_6,z_2),(x_4,y_7,z_2),(x_4,y_8,z_2),\\ & (x_4,y_9,z_2),(x_5,y_6,z_2),(x_5,y_7,z_2),(x_6,y_{10},z_2),(x_6,y_{11},z_2),(x_6,y_{12},z_2),\\ & (x_7,y_{10},z_2),(x_8,y_{13},z_3),(x_8,y_{14},z_3),(x_8,y_{15},z_3),(x_9,y_{13},z_3),(x_{10},y_{16},z_4),\\ & (x_{10},y_{17},z_4),(x_{11},y_{18},z_5),(x_{12},y_{19},z_6)\} \end{aligned}$$

where $z_1 \neq z_2 \neq z_3 \neq z_4 \neq z_5 \neq z_6 \neq z_7$, $x_1 \neq x_2 \neq x_3$, $x_{14} \neq x_{15}$, $x_6 \neq x_7$, $x_8 \neq x_9$, $y_1 \neq y_2 \neq y_3 \neq y_4 \neq y_5$, $y_6 \neq y_7 \neq y_8 \neq y_9$, $y_{10} \neq y_{11} \neq y_{12}$, $y_{13} \neq y_{14} \neq y_{15}$ and $y_{16} \neq y_{17}$.

Proposition 3. *Let S be the set stated above. Then* $\det V(P;S) \neq 0$

Proof. We constructed a Vandermonde matrix where the rows are indexed by the corresponding monomials for each of the integer lattice points in $P \cap \mathbb{Z}^3$ and the columns are indexed by the points in S.

By doing row operations we were able to find a block of zeroes in the lower left using the fact that the first 9 points had the same z value. Then, the determinant was the product of the determinant of a 9 by 9 matrix and the determinant of a 18 by 18 matrix, and some common factors of the columns. We found another block of zeroes in the 18 by 18 matrix using the fact that the first 6 points had the same z value. We kept doing the same process until we were left with the product of several smaller determinants and the common factors we pulled out each time.

Consequently,

$$\begin{aligned}
\det V(P;S) \;=\; & \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x_1 & x_1 & x_1 & x_1 & x_1 & x_2 & x_2 & x_2 & x_3 \\ x_1^2 & x_1^2 & x_1^2 & x_1^2 & x_1^2 & x_2^2 & x_2^2 & x_2^2 & x_3^2 \\ x_1y_1 & x_1y_2 & x_1y_3 & x_1y_4 & x_1y_5 & x_2y_1 & x_2y_2 & x_2y_3 & x_3y_1 \\ x_1y_1^2 & x_1y_2^2 & x_1y_3^2 & x_1y_4^2 & x_1y_5^2 & x_2y_1^2 & x_2y_2^2 & x_2y_3^2 & x_3y_1^2 \\ y_1 & y_2 & y_3 & y_4 & y_5 & y_1 & y_2 & y_3 & y_1 \\ y_1^2 & y_2^2 & y_3^2 & y_4^2 & y_5^2 & y_1^2 & y_2^2 & y_3^2 & y_1^2 \\ y_1^3 & y_2^3 & y_3^3 & y_4^3 & y_5^3 & y_1^3 & y_2^3 & y_3^3 & y_1^3 \\ y_1^4 & y_2^4 & y_3^4 & y_4^4 & y_5^2 & y_1^4 & y_2^4 & y_3^4 & y_1^4 \end{vmatrix} \cdot \\
& \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ x_4 & x_4 & x_4 & x_4 & x_5 & x_5 \\ y_6 & y_7 & y_8 & y_9 & y_6 & y_7 \\ y_6^2 & y_7^2 & y_8^2 & y_9^2 & y_6^2 & y_7^2 \\ y_6^3 & y_7^3 & y_8^3 & y_9^3 & y_6^3 & y_7^3 \\ x_4y_6 & x_4y_7 & x_4y_8 & x_4y_9 & x_5y_6 & x_5y_7 \end{vmatrix} \cdot \\
& \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_6 & x_6 & x_6 & x_7 \\ y_{10} & y_{11} & y_{12} & y_{10} \\ y_{10}^2 & y_{11}^2 & y_{12}^2 & y_{10}^2 \end{vmatrix} \cdot \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_8 & x_8 & x_8 & x_9 \\ y_{13} & y_{14} & y_{15} & y_{12} \\ y_{13}^2 & y_{14}^2 & y_{15}^2 & y_{13}^2 \end{vmatrix} \\
& (y_1-y_5)(y_1-y_4)(y_1-y_2)^2(-x_3+x_2)(-x_3+x_1)(x_1-x_2)^3 \cdot \\
& (y_8-y_9)(y_7-y_9])(y_7-y_8)(-y_9+y_6)(y_6-y_8)(y_6-y_7)^2 \cdot \\
& (x_4-x_5)^2(y_{11}-y_{12})(y_{10}-y_{12})(y_{10}-y_{11})(x_6-x_7)(y_{14}-y_{15}) \cdot \\
& (y_{13}-y_{15})(y_{13}-y_{14})(x_8-x_9)(y_{17}-y_{16})(z_2-z_1)^6(z_3-z_1)^4 \cdot \\
& (z_4-z_1)^4(z_5-z_1)^2(z_6-z_1)(z_7-z_1)(z_3-z_2)^4(z_4-z_2)^4(z_5-z_2)^2 \cdot \\
& (z_6-z_2)(z_7-z_2)(z_4-z_3)^4(z_5-z_3)^2(z_6-z_3)(z_7-z_3)(z_5-z_4)^2 \cdot \\
& (z_6-z_4)(z_7-z_4)(z_6-z_5)(z_7-z_5)(z_7-z_6)
\end{aligned}$$

So, $\det V(P;S) \neq 0$ since $z_1 \neq z_2 \neq z_3 \neq z_4 \neq z_5 \neq z_6 \neq z_7$, $x_1 \neq x_2 \neq x_3$, $x_4 \neq x_5$, $x_6 \neq x_7$, $x_8 \neq x_9$, $y_1 \neq y_2 \neq y_3 \neq y_4 \neq y_5$, $y_6 \neq y_7 \neq y_8 \neq y_9$, $y_{10} \neq y_{11} \neq y_{12}$, $y_{23} \neq y_{14} \neq y_{15}$ and $y_{16} \neq y_{17}$. □

Now, we will give a general definition for the set S for all ℓ.

Definition 4. Let $L = P \cap \mathbb{Z}^3$, where the polytope $P = \text{conv}\{0, \ell e_1, 2\ell e_2, 3\ell e_3\}$. Let P_i be the intersection of the plane $z = i$ and the set L. Say $|P_i| = m_i$. The set S

in $(\mathbb{F}_q^*)^3$ of size $|L|$ is chosen to be combinatorially similar to the set L in $\mathbb{Z}^3$ in the following sense.

Let the set S_i in $(\mathbb{F}_q^*)^3$ correspond to P_i in $\mathbb{Z}^3$ by choosing it to contain m_i points in $(\mathbb{F}_q^*)^3$ with the same third coordinate. Furthermore, if there are k_j lattice points on the same line $x = j$ in P_i, we require that k_j of these m_i points in S_i have the same first coordinate. We note that this implies that these k_j points will have distinct second coordinate.

Now put $S = \bigcup_{i=0}^{3\ell} S_i$. A set S constructed in this manner is said to be an *ℓP-configuration.*

Conjecture 1. $\det V(\ell P; S) \neq 0$ for all $\ell \geq 1$ and all ℓP configurations S.

We believe that it is always possible to find some $S \subset T$ with $|S| = \#(P)$ such that $\det V(P; S) \neq 0$ for every ℓ because there seems to be a correspondance between the distribution of points in S contained in each plane of the form $z = z_i \in \mathbb{F}_q^*$ and the factorization of the determinant of the Vandermonde matrix $V(P; S)$.

3.2 Minimum Distance

Proposition 4. *Let C_P be the toric code over the field $\mathbb{F}_q$ from the tetrahedron $P = \text{conv}\{0, e_1, 2e_2, 3e_3\}$. Then C_P has minimum distance less than or equal to $(q-1)^3 - 3(q-1)^2$.*

Proof. To show that $d(C_P) \leq (q-1)^3 - 3(q-1)^2$ we must look at the codewords. The codewords are $\text{ev}(a + bx + cy + ey^2 + fyz + gz + hz^2 + iz^3)$. So, to find the minimum distance we need to find the largest number of zero entries a codeword can have:

If $b, c, e, f = 0$ and we assume that $a + gz + hz^2 + iz^3 = \gamma(z - \beta^n)(z - \beta^o)(z - \beta^p)$, then we would have zeroes in any location where $z = \beta^n$, $z = \beta^o$ and $z = \beta^p$ for $\gamma = \textit{constant}$, and x and y can take any value which implies that we have $q - 1$ choices for x and $q - 1$ choices for y. Thus, since the weight of a codeword is the length of the codeword minus the number of zero entries in the codeword, and we have some codewords with $3(q-1)^2$ zero entries, then $d(C_P) \leq (q-1)^3 - 3(q-1)^2$. □

Proposition 5. *Let C_P be the toric code over the field $\mathbb{F}_q$ from the tetrahedron $P = \text{conv}\{0, e_1, 2e_2, 3e_3\}$. Then C_P has a minimum distance greater than or equal to $(q-1)^3 - 3(q-1)^2$.*

Proof. To show that $d(C_P) \geq (q-1)^3 - 3(q-1)^2$, we need to show that for all $T \subset (\mathbb{F}_q^*)^3$, with $|T| = (q-1)^3 - [(q-1)^3 - 3(q-1)^2] + 1 = 3(q-1)^2 + 1$, there is some $S \subset T$ which looks like $S = \{(x_1, y_1, z_1), (x_1, y_2, z_1), (x_1, y_3, z_1), (x_2, y_1, z_1), (x_3, y_4, z_2), (x_3, y_5, z_2), (x_4, y_6, z_3), (x_5, y_7, z_4)\}$, where $z_1 \neq z_2 \neq z_3 \neq z_4$, $x_1 \neq x_2$, $y_1 \neq y_2 \neq y_3$ and $y_4 \neq y_5$. Then the $\det V(P; S) \neq 0$ by Proposition 2.

Since we have $3(q-1)^2 + 1$ points in the set T and there are $(q-1)$ horizontal planes of the form $z = z_i \in \mathbb{F}_q^*$, by the Pigeonhole Principle we know that there exists one plane $z = z_1$ which has at least $3(q-1) + 1$ points. Since we have at most $q - 1$ points in each horizontal line of the form $y = y_i \in \mathbb{F}_q^*$ in each plane and $q \geq 5$,we are

guaranteed three different values for y and we have $2(q-1)+1$ points left on the plane that are not on that line.

Let $T' = T - \{(x_1, y_1, z_1), (x_1, y_2, z_1), (x_1, y_3, z_1), (x_2, y_1, z_1)\}$. T' has $3(q-1)^2 - 3$ points and $3(q-1)^2 - 3 \geq (q-1)^2 + 1$ if $q \geq 5$, which implies there is at least one plane that has at least 2 points of the set T' on it that are different from $(x_1, y_1, z_1), (x_1, y_2, z_1), (x_1, y_3, z_1), (x_2, y_1, z_1)$.

Let $T'' = T' - \{(x_3, y_4, z_2), (x_3, y_5, z_2)\}$. The set T'' has $3(q-1)^2 - 6$ points and $3(q-1)^2 - 6 \geq (q-1)^2$, which implies there are more planes left that have at least one point each which is different from any of these six points.

Therefore, by Proposition 1, this means that $d(C_P) \geq (q-1)^3 - 3(q-1)^2$. □

Theorem 2. *Let C_P be the toric code over the field $\mathbb{F}_q$ from the tetrahedron $P = \text{conv}\{0, e_1, 2e_2, 3e_3\}$. Then C_P has minimum distance equal to $(q-1)^3 - 3(q-1)^2$.*

Proof. By Proposition 4 and Proposition 5, we can see that this is true for $\ell = 1$. □

Proposition 6. *Let C_{2P} be the toric code over the field $\mathbb{F}_q$ from the tetrahedron $2P = \text{conv}\{0, 2e_1, 4e_2, 6e_3\}$. Then C_{2P} has minimum distance less than or equal to $(q-1)^3 - 6(q-1)^2$.*

Proof. To prove that $d(C_{2P}) \leq (q-1)^3 - 6(q-1)^2$ we need to find the largest possible number of zero entries in each codeword:

If all the coefficients of the monomials containing two or more variables are assumed to be zero, then the remaining equation of the linear combinations is as follows:

$$a + bz + cz^2 + dz^3 + ez^4 + fz^5 + gz^6 + hy + iy^2 + jy^3 + ky^4 + nx + px^2.$$

Now, if we let $h = i = j = k = n = p = 0$ and suppose that

$$a + bz + cz^2 + dz^3 + ez^4 + fz^5 + gz^6 = \gamma(z - \beta^r)(z - \beta^s)(z - \beta^t)(z - \beta^u)(z - \beta^v)(z - \beta^w),$$

then we would have zeros in any location where $z = \beta^r, z = \beta^s, z = \beta^t, z = \beta^u, z = \beta^v$, and $z = \beta^w$ for $\gamma = constant$, and x and y can take any value which implies that we have $q-1$ choices for x and $q-1$ choices for y. This implies that the number of zeros a codeword can have is at most $6(q-1)^2$.

Thus, the minimum distance of a codeword is the length of the codeword minus the number of zero entries in the codeword, $d(C_{2P}) \leq (q-1)^3 - 6(q-1)^2$. □

Proposition 7. *Let C_{2P} be the toric code over the field $\mathbb{F}_q$ from the tetrahedron $2P = \text{conv}\{0, 2e_1, 4e_2, 6e_3\}$. Then C_{2P} has a minimum distance greater than or equal to $(q-1)^3 - 6(q-1)^2$.*

Proof. To show that $d(C_{2P}) \geq (q-1)^3 - 6(q-1)^2$ we need to show that for all $T \subset (\mathbb{F}_q^*)^3$ with $|T| = (q-1)^3 - [(q-1)^3 - 6(q-1)^2] + 1 = 6(q-1)^2 + 1$ there is some $S \subset T$ which looks like:

$$\begin{aligned} S \;=\; & \{(x_1, y_1, z_1), (x_1, y_2, z_1), (x_1, y_3, z_1), (x_1, y_4, z_1), (x_1, y_5, z_1), (x_2, y_1, z_1), (x_2, y_2, z_1), \\ & (x_2, y_3, z_1), (x_3, y_1, z_1), (x_4, y_6, z_2), (x_4, y_7, z_2), (x_4, y_8, z_2), (x_4, y_9, z_2), (x_5, y_6, z_2), \\ & (x_5, y_7, z_2), (x_6, y_{10}, z_3), (x_6, y_{11}, z_3), (x_6, y_{12}, z_3), (x_7, y_{10}, z_3), (x_8, y_{13}, z_4), \\ & (x_8, y_{14}, z_4), (x_8, y_{15}, z_4), (x_9, y_{13}, z_4), (x_{10}, y_{16}, z_5), (x_{10}, y_{17}, z_5), (x_{11}, y_{18}, z_6), \\ & (x_{12}, y_{19}, z_7)\} \end{aligned}$$

then $\det V(2P;S) \neq 0$ by Proposition 3.

Since we have $6(q-1)^2+1$ points in the set T and there are $q-1$ planes of the form $z = z_i \in \mathbb{F}_q^*$, by the Pigeonhole Principle we will have $6(q-1)+1$ points on some plane of the form $z = z_1$, and since $q \geq 8$ and we have $q-1$ vertical lines of the form $x = x_i \in \mathbb{F}_q^*$ on a plane, we are guaranteed to have 9 points on that plane such that 5 of them are on the same line $x = x_1$.

Since we have at most $q-1$ points on one line, then we would have $5(q-1)+1$ points left that are not on $x = x_1$, so we are guaranteed to have a line $x = x_2$ with 3 points on it and we still have left $4(q-1)+1$ points that are not on $x = x_1$ or $x = x_2$.

Let $T' = T - \{(x_1,y_1,z_1),(x_1,y_2,z_1),(x_1,y_3,z_1),(x_1,y_4,z_1),(x_1,y_5,z_1),(x_2,y_1,z_1),(x_2,y_2,z_1),(x_2,y_3,z_1),(x_3,y_1,z_1)\}$. Then, $|T'| = 6(q-1)^2 - 8 \geq 5(q-1)^2+1$, which implies we have $5(q-1)+1$ points on a different plane $z = z_2$, so we are guaranteed to have 6 points on that plane such that 4 of them are on the same line $x = x_3$. Since we have at most $q-1$ points on one line, we have $4(q-1)+1$ points left that are not on $x = x_3$.

Now, let $T'' = T' - \{(x_4,y_6,z_2),(x_4,y_7,z_2),(x_4,y_8,z_2),(x_4,y_9,z_2),(x_5,y_6,z_2),(x_5,y_7,z_2)\}$. Then, $|T''| = 6(q-1)^2 - 14 \geq 5(q-1)^2+1$, which implies we have $5(q-1)+1$ points on a different plane $z = z_3$, so we are guaranteed to have 4 points on that plane such that 3 of them are on the same line $x = x_4$ and there are $4(q-1)+1$ points left that are not on $x = x_4$.

Let $T''' = T'' - \{(x_6,y_{10},z_3),(x_6,y_{11},z_3),(x_6,y_{12},z_3),(x_7,y_{10},z_3)\}$. Then, $|T'''| = 6(q-1)^2 - 18 \geq 5(q-1)^2+1$ so by the same reasoning we are guaranteed to have a different plane $z = z_4$ with 4 points on it, such that 3 of them are on the same line $x = x_5$.

Let $T^{(4)} = T''' - \{(x_8,y_{13},z_4),(x_8,y_{14},z_4),(x_8,y_{15},z_4),(x_9,y_{13},z_4)\}$. Then, $|T^{(4)}| = 6(q-1)^2 - 22 \geq 5(q-1)^2+1$, which implies we have a different plane $z = z_5$ with $5(q-1)^2+1$ points on it, such that 2 of them are on the same line $x = x_6$.

Let $T^{(5)} = T^{(4)} - \{(x_{10},y_{16},z_5),(x_{10},y_{17},z_5)\}$. Then, $|T^{(5)}| = 6(q-1)^2 - 24 \geq 5(q-1)^2+1$, which guarantees we have enough points left that are not on any of these planes.

Therefore, by Proposition 1, this means that $d(C_{2P}) \geq (q-1)^3 - 6(q-1)^2$. □

Theorem 3. *Let C_{2P} be the toric code over the field $\mathbb{F}_q$ from the tetrahedron $2P = \text{conv}\{0, 2e_1, 4e_2, 6e_3\}$. Then C_{2P} has minimum distance equal to $(q-1)^3 - 6(q-1)^2$.*

Proof. By Proposition 6 and Proposition 7, we can see that this is true for $\ell = 2$. □

Proposition 8. *Let $C_{\ell P}$ be the toric code over the field $\mathbb{F}_q$ from the tetrahedron $\ell P = \text{conv}\{0, \ell e_1, 2\ell e_2, 3\ell e_3\}$. Then $C_{\ell P}$ has minimum distance less than or equal to $(q-1)^3 - 3\ell(q-1)^2$.*

Proof. The codewords are formed by taking linear combinations of the monomials corresponding to the integer lattice points, and finding the minimum distance is achieved by seeking codewords that can have most zeros. So, if all the coefficients of the terms containing the x and y variables are assumed to be zero, the remaining equation is a polynomial with variable z of degree 3ℓ.

The best possible case is a perfect factorization of the polynomial in such a way that it's equal to $\gamma(z-\beta_1)(z-\beta_2)\dots(z-\beta_{3\ell})$ where γ and β_i are constants. Thus, there would be at best 3ℓ choices for the z, $(q-1)$ choices for the x variable and $(q-1)$ choices for the y variable that would make that remaining polynomial equal to zero. Hence, a codeword can have at most $3\ell(q-1)^2$ zeros.

Therefore, since the minimum distance is the length of the codeword minus the maximum amount of zeros, $d(C_P) \leq (q-1)^3 - 3\ell(q-1)^2$. □

4 Future Work

Conjecture 2. Let $C_{\ell P}$ be the toric code over the field $\mathbb{F}_q$ from the tetrahedron $P = \text{conv}\{0, \ell e_1, 2\ell e_2, 3\ell e_3\}$. Then $C_{\ell P}$ has minimum distance greater than or equal to $(q-1)^3 - 3\ell(q-1)^2$.

One of our future goals is to work on proving Conjecture 1 to be able to prove Conjecture 2. Since we already established the upper bound for the minimum distance which is less than or equal to $(q-1)^3 - 3\ell(q-1)^2$, then proving Conjecture 2 will give us the lower bound and we would be able to conclude that the minimum distance is equal to $(q-1)^3 - 3\ell(q-1)^2$.

Our other goals for the future are to work on other examples of more general classes of polytopes and to look for ways to analyze the minimum distance for the codes associated to those polytopes:

1. The polytope $\text{conv}\{e_1, e_2, 2e_1 + 2e_2\}$.
2. *zonotopes* P in $\mathbb{R}^n$ with $g \geq n$ generators.
3. Cartesian products, Minkowski sums, etc. of polytopes of these types.
4. Other classes of polytopes.

5 Acknowledgments

This work was conducted during the 2009 Mathematical Sciences Research Institute Undergraduate Program (MSRI-UP) in Berkeley, CA. MSRI-UP is supported by the National Science Foundation (grant No. DMS-0754872) and the National Security Agency (grant No. H98230-09-0103). We would like to thank Dr. John Little, Dr. Herbert Medina, Dr. Emille Davie and the entire MSRI-UP staff for their support throughout the program.

6 Appendix: Maple Algorithms

We developed several Maple algorithms to compute data that confirms our results found in this article.

6.1 Computing Lattice Points of the Special Polytope $P = \text{conv}\{0, \ell e_1, 2\ell e_2, 3\ell e_3\}$ in $\mathbb{R}^3$

This algorithm finds all the integer lattice points for $P = \text{conv}\{0, \ell e_1, 2\ell e_2, 3\ell e_3\}$ in $\mathbb{R}^3$:

```
P := {};
ell := an integer greater than or equal to one;
for x from 0 to ell do
  for y from 0 to 2*\ell do
    for z from 0 to 3*\ell do
      pt := [x, y, z];
      c := 6x+3y+2z$;
      if c <= 6*ell$ then
        P := P union {pt};
      end if:
    end do:
  end do:
end do:
print(P);
nops(P);
```

6.2 Computing Monomials

This algorithm computes monomials for given lattice points in $\mathbb{R}^m$. The monomials used in Example 1 were calculated by writing all the integer lattice points in P:

```
P :=[[0,0],[0,1],[1,0],[2,0]];
nP:=nops(P);
monoms:=[ ];
for i to nP do
   mono := x^P[i][1]* y^P[i][2];
   monoms := [op(monoms), mono];
end do:
print(monoms);
```

The output for Example 1 is

$$[1, x, x^2, y].$$

6.3 Constructing a Vandermonde Matrix and Computing its Determinant

This algorithm constructs a Vandermonde matrix and computes the determinant for a polytope in $\mathbb{R}^3$ given an S:

```
with(linalg);
S := [write the appropriate S]:
VPS := matrix(nP, nP);
for i to nP do
  for j to nP do
     VPS[i, j] := S[j][1]^P[i][1]*S[j][2]^P[i][2]*S[j][3]^P[i][3]:
  end do:
end do:
eval(VPS);
factor(det(VPS));
```

References

[1] R. Diaz, S. Robins, "The Ehrhart polynomial of a lattice n-simplex," *Electronic Research Announcements of the the American Mathematical Society* 2: 1–6, 1996.

[2] J. Little, R. Schwarz, "On toric codes and multivariate Vandermonde matrcies," *Appl. Alg. Engrg. Comm. Comput.* **18** (2007), 349–367.

[3] A. Liu, "Lattice points and Pick's Theorem," *Mathematics Magazine*, Vol. 52, No. 4, 232–235, Sept. 1979.

List Decoding Algorithms for Reed-Solomon Codes and their Maximum Decoding Radii

Kimberly Heu
University of Hawaii at Manoa

Caitlyn Parmelee
Nazareth College of Rochester

July 2009

Abstract

Reed-Solomon codes are linear, cyclic codes that can be used to ensure that a correct message is received provided that there are at most a specific number of errors. One of the methods to deal with the decoding of received messages is the Guruswami-Sudan list decoding algorithm. This paper will present several of the properties of list decoding, including a detailed analysis of list decoding with lists of size one and the optimal benefits of list decoding.

1 Introduction

There exist various types of algebraic decoders for linear codes, and for Reed-Solomon (RS) codes in particular. Many of these decoders will return exactly one unique code word. In particular, for bounded distance decoding, for any code C with minimum distance $d = 2t + 1$ or $d = 2t + 2$, then for any error with weight at most t, there is a unique closest codeword to the received word, and algorithms like the Euclidean Algorithm (Sugiyama) decoder or the Berlekamp-Massey algorithm decoder will return that unique closest codeword. These algorithms have a limited number of errors that can be corrected, because this number is constrained by the minimum distance of the code. List decoding algorithms were developed to make it possible to correct more than t errors.

Guruswami and Sudan introduced a list decoding algorithm for Reed-Solomon codes in their 1999 paper [2]. Their algorithm returns a list of codewords within a given Hamming distance, τ, of the received word, which is typically greater than the $t = \lfloor \frac{d-1}{2} \rfloor$ errors that can be corrected using unique codeword decoding.

We will first consider the algebraic basis of list decoding methods using the special case of decoding with lists of size one, detailing how the method works and demonstrating that the decoding results from this case are equivalent to using a bounded distance decoder. We then move on to considering lists of higher degrees.

One of the key concepts relating to the list decoding algorithm is multiplicity. We will extend the case of lists of size one to get lists of higher degrees, which in turn

will give us larger decoding radii. However, there is a maximum decoding radius, which will be achieved at a finite multiplicity. In his dissertation, Eriksson introduces graphs of the multiplicity generating the maximal decoding radius for a given n and k. The second portion of this paper examines some of the properties relating to this multiplicity, using graphs similar to Eriksson's.

2 Background

We will now give some necessary background for our paper. List decoding algorithms are designed for Reed-Solomon Codes, so we begin with a brief overview of these codes. Then, we will discuss working with polynomials of two variables, which is necessary for some of the mathematics used in list decoding. Finally, we will define list decoding and give some basic information on the process.

2.1 Reed-Solomon Codes

Since the list decoding algorithm we are using deals exclusively with Reed-Solomon codes, we define these codes now.

Definition 1. Let α be a primitive element for the field $\mathbb{F}_{p^r}$. A *Reed-Solomon code* $\mathrm{RS}(p^r, \delta)$ is a cyclic code of length $n = p^r - 1$ over $\mathbb{F}_{p^r}$ whose generator polynomial has the form

$$g(x) = (x - \alpha^{m+1})(x - \alpha^{m+2}) \cdots (x - \alpha^{m+\delta-1}) \tag{1}$$

for some m, $m \geq 0$.

The encoding function, $E(f(x))$, for a word $f(x)$, is defined as the following:

$$E(f(x)) = f(x) - r(x), \tag{2}$$

where $f(x)$ is our word, and $r(x)$ is the remainder from the division of $f(x)$ by the generating polynomial $g(x)$.

2.2 Alternate Construction of Reed-Solomon Codes

Let α be a primitive element, so that the powers of α represent distinct nonzero elements of the field. Let $L_k = \mathrm{Span}\{1, t, t^2, \ldots, t^{k-1}\} \subset \mathbb{F}_{p^r}[t]$. To construct a code with dimension k, we evaluate polynomials $f \in L_k$ to get the entries in our codeword:

$$\begin{aligned} \mathrm{ev} : L_k &\longrightarrow \mathbb{F}_{p^r}^{p^r-1} \\ f &\longmapsto (f(1), f(\alpha), f(\alpha^2), \ldots, f(\alpha^{p^r-2})). \end{aligned}$$

This process produces the same code as constructed in Equation 1 with $m = 0$.

2.3 Polynomials

To understand list decoding, one must have a grasp of concepts relating to polynomials of two variables. One of the key concepts needed is monomial weight, which gives a weight to every monomial, ensuring that the terms of polynomials can be ordered. This is done by fixing a weight vector, $\mathbf{w} = (u, v)$, and using that to calculate the weight of a monomial.

Definition 2. The ***w**-degree* of a monomial, $x^i y^j$ is $\deg_{\mathbf{w}} x^i y^j = ui + vj$.

To break ties in weights, we will use reverse lexicographic order, which is one of the possible lexicographic orders and is a convenient ordering for list decoding.

Definition 3. In ***w**-reverse lexicographic* (**w**-revlex) order, if $ui_1 + vj_1 = ui_2 + vj_2$, we say that $x^{i_1} y^{j_1} < x^{i_2} y^{j_2}$ if $i_1 > i_2$.

It is useful to note that under **w**-revlex order, monomials are ranked with increased powers of y. Also, for list decoding, we will always use $\mathbf{w} = (1, k-1)$.

Definition 4. Define *C(v,l)* to be the number of monomials of weighted $(1, v)$-degree less than or equal to l.

Every monomial can be given a unique index number, r,which is not the degree, such that $r \geq 0$, when all monomials are ordered under a given **w**-revlex ordering. Any polynomial $Q(x, y)$ will have some R that is the maximum of the r's of its monomials. This R is called the rank of $Q(x, y)$.

Another concept that is important for list decoding is the derivative of a function. Consider the polynomial $Q(x) = \sum_{i=0}^{n} a_i x^i \in \mathbb{F}[x]$. For any $\alpha \in \mathbb{F}$ we can expand $Q(x)$ to be

$$Q(x + \alpha) = \sum_{r=0}^{n} Q_r(\alpha) x^r. \tag{3}$$

Definition 5. The rth Hasse derivative of Q at α is defined to be $Q_r(x)$

Definition 6. The polynomial $Q(x)$ has a zero of *order* or *multiplicity* m at α if and only if $Q(\alpha) = Q_1(\alpha) = \cdots = Q_{m-1}(\alpha) = 0$, but $Q_m(\alpha) \neq 0$.

When we increase list size, we also need to consider polynomials of more than one variable, so it is also useful to consider $Q(x, y) = \sum_{i,j} a_{i,j} x^i y^j$ and its expansion:

$$Q(x + \alpha, y + \beta) = \sum_{r=0}^{n} \sum_{s=0}^{m} Q_{r,s}(\alpha, \beta) x^r y^s. \tag{4}$$

Definition 7. The polynomial $Q(x, y)$ has a zero of *multiplicity* m at (α, β) if and only if $Q_{r,s}(\alpha, \beta) = 0$ for all (r, s) with $r + s < m$ and $Q_{r,s}(\alpha, \beta) \neq 0$ for some (r, s) with $r + s = m$.

2.4 List Decoding

Definition 8. Given a received word, a *list decoding* algorithm will return all codewords within a given distance of the received word.

List decoding algorithms are designed to return lists containing up to a specified number of codewords, all of which will be within a given distance of the received word. By increasing the list size we can find additional possible words, and therefore have the ability to increase the maximum number of errors we can correct. These algorithms increase the size of the lists by increasing the multiplicity required. In turn, this forces us to extend the polynomial considered by our key equation, increasing the possible number of codewords to be recovered.

For a given multiplicity m, the number of different derivatives we consider is $\binom{m+1}{2}$, so this gives us a total of $n\binom{m+1}{2}$ equations, and we need to pick an appropriately large number of monomials to include in our $Q(x, y)$ and, in turn, a large enough list size. Section 4 will give a specific process for the decoding.

3 Lists of Size One

The list decoding method for Reed-Solomon codes that we will consider has its origins in a unique decoding algorithm found by Welch and Berlekamp. The key equation for this algorithm is set up in such a way that the factors of the equation represent the error locations and the error values of each error in our received word. Suppose the codeword corresponding to $f(x) \in L_{k-1}$ is sent, and $r(x)$ is received. Let E be the set of error locations. Consider the polynomial $Q(x, y) = w(x) \cdot (y - f(x))$ where $w(x) = a\prod_{i\in E}(x + \alpha^i)$ where a is a nonzero constant. We know that the value of $Q(x, y)$ will be zero when any nonzero field element α^i is substituted for x and the corresponding coefficient r_i from the received polynomial is substituted for y.

Now consider the coefficients in a polynomial $Q(x, y) = u(x)y + v(x)$ where the degree of $u(x)$ is less than or equal to t, and the degree of $v(x)$ is less than or equal to $t + k - 1$ as unknowns.

Definition 9. The *interpolation equations* are defined as

$$Q(\alpha^i, r_i) = u(\alpha^i)r_i + v(\alpha^i) = 0 \ \forall \ i = 0, 1, \ldots, p^r - 2 \tag{5}$$

where r_i is the ith entry of r.

Using the interpolation equation, we can generate a system of equations from a received polynomial. All of the solutions to this system of equations can be factored into polynomials, one of which will be the polynomial used to generate the transmitted codeword. The following example is a walk-through of the decoding process for an $RS(2^4, 9)$ code.

Example 1. We will use an $RS(2^4, 9)$ code, with α a root of the irreducible polynomial $z^4 + z + 1$. Since $\delta = 9 = 2t + 1$ we have $t = 4$ and can correct up to 4 errors.

We can encode the word given by $x^{14} + \alpha x^{13} + x^{12} + \alpha x^{11} + \alpha^2 x^9 + x^8$, using (2) with the generating polynomial as given in (1). We get:

$$\begin{aligned} b(x) &= x^{14} + \alpha x^{13} + x^{12} + \alpha x^{11} + \alpha^2 x^9 + x^8 + (\alpha^2+1)x^7 + (\alpha^2+1)x^6 + x^5 \\ &\quad +(\alpha^3+\alpha^2+\alpha+1)x^4 + \alpha^2 x^3 + \alpha^2 x^2 + (\alpha^3+\alpha^2)x + (\alpha^3+1). \end{aligned}$$

Introduce an error, $e(x) = (\alpha^2+\alpha+1) + x^3 + (\alpha^2+\alpha)x^8 + (\alpha+1)x^{12}$, so $r(x) = b(x) + e(x)$, our received word, is:

$$\begin{aligned} r(x) &= x^{14} + \alpha x^{13} + \alpha x^{12} + \alpha x^{11} + \alpha^2 x^9 + (\alpha^2+\alpha+1)x^8 + (\alpha^2+1)x^7 \\ &\quad +(\alpha^2+1)x^6 + x^5 + (\alpha^3+\alpha^2+\alpha+1)x^4 + (\alpha^2+1)x^3 + \alpha^2 x^2 \\ &\quad +(\alpha^3+\alpha^2)x + (\alpha^3+\alpha^2+\alpha). \end{aligned}$$

The $Q(x, y)$ that we will use to build our system of equations is:

$$\begin{aligned} Q(x,y) &= (c_0 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4)y + d_0 + d_1 x + d_2 x^2 + d_3 x^3 + d_4 x^4 \\ &\quad +d_5 x^5 + d_6 x^6 + d_7 x^7 + d_8 x^8 + d_9 x^9 + d_{10} x^{10}. \end{aligned}$$

The powers of x were chosen because we know that we have to have at least 4 errors, so $\deg(c(x)) = 4$ and we had to pick $d(x)$ so that we would have more unknown variables than equations, to guarantee a non-zero solution. We have an equation for each power of x in our received word, so this means that we need at least 16 unknown variables.

Using Maple to put the coefficients of the equation into a matrix and finding the nullspace of the matrix, we get:

$$\begin{aligned} Q(x,y) &= (\alpha+1) + \alpha^2 x + \alpha^3 x^2 + x^3 + (\alpha^3+\alpha^2+\alpha)x^4)y + 1 + (\alpha^2+1)x \\ &\quad +(\alpha^3+\alpha^2)x^2 + (\alpha^2+1)x^3 + (\alpha^3+\alpha)x^4 + \alpha^2 x^5 + (\alpha+1)x^6 \\ &\quad +(\alpha^3+\alpha^2+\alpha)x^7 + (\alpha^3+\alpha^2+\alpha+1)x^8 + x^9. \end{aligned}$$

We know that $Q(x, y) = w(x)(y - f(x))$, so solving for $f(x)$ we get

$$\begin{aligned} f(x) &= (\alpha+1)x^5 + (\alpha^2+\alpha+1)x^4 + (\alpha^2+\alpha)x^3 + (\alpha^3+\alpha^2)x^2 \\ &\quad +(\alpha^3+1)x + (\alpha^3+\alpha^2+\alpha). \end{aligned}$$

So using our $ev(f(x))$ function, we can recover the original codeword:

$$\begin{aligned} b(x) &= x^{14} + \alpha x^{13} + x^{12} + \alpha x^{11} + \alpha^2 x^9 + x^8 + (\alpha^2+1)x^7 + (\alpha^2+1)x^6 + x^5 \\ &\quad +(\alpha^3+\alpha^2+\alpha+1)x^4 + \alpha^2 x^3 + \alpha^2 x^2 + (\alpha^3+\alpha^2)x + (\alpha^3+1). \end{aligned}$$

Theorem 1. *Given an RS(p^r, δ) code where $\delta = 2t+1$, a transmitted word, $ev(f(x))$, and a received word $r = ev(f(x)) + e$, where e is the error vector and $wt(e) \leq t$, there exists a nonzero solution, $u(x), v(x)$, to the system of linear equations given by*

$$Q(\alpha^i, r_i) = u(\alpha^i) r_i + v(\alpha^i) = 0 \ \forall \ i = 0, 1, \ldots, p^r - 2, \tag{6}$$

where r_i is the ith entry of r, the degree of $u(x)$ is less than or equal to t, and the degree of $v(x)$ is less than or equal to $t + k - 1$.

Furthermore, if $u(x)$ is of minimal degree, then (up to a constant multiple) $u(x) = \prod_{i \in E}(x + \alpha^i)$ and $v(x) = f(x) \prod_{i \in E}(x + \alpha^i)$, where E is the set of error locations.

Proof. Under these hypotheses the system of interpolation equations has at least one solution corresponding to the error locator and the transmitted codeword. Since we can rewrite that $Q(x, y)$ as $Q(x, y) = w(x)y - w(x)f(x)$, then $u(x) = w(x)$ and $v(x) = w(x)f(x)$ is one of the solutions to the system of equations. Note that the degree of $w(x) \leq t$ because we are assuming t or fewer errors and the degree of $w(x)f(x) \leq t + k - 1$ because $w(x)$ has degree at most t and $f(x)$ has degree $k - 1$ based on the length of our word. We can also generate additional solutions to the system of equations by letting $u(x) = w(x)p(x)$ for some polynomial $p(x)$ whose degree is less than or equal to $t - wt(e)$. Of these, all $u(x)$ have a degree greater than or equal to $w(x)$, with $p(x) = 1$ being the only solution whose degree is equal to $\deg(w(x))$, so $w(x)$ is the solution with the smallest degree. The basis for all possible solutions of this form consists of $t - wt(e) + 1$ basis elements. We would now like to show that this form of solution is the only form for solutions to M, our coefficient matrix.

Consider that $Q(x, y)$ has the form:

$$Q(x, y) = (c_0 + c_1x + \cdots + c_tx^t)y + (d_0 + d_1x + \cdots + d_{t+k-1}x^{t+k-1}). \tag{7}$$

Let us consider the number of equations and unknown variables we are working with while solving the system of equations generated by the key equation, as presented in (6). First, for the number of equations, we will create an equation for each i, as defined in (6), so $p^r - 1$ equations. Next, we have an unknown for each of the coefficients for all of the terms involved in our equation. Breaking this down, we have $\deg(u(x)y) = \deg(u(x)) = t$ and $\deg(v(x)) = t + k - 1 = p^r - t - 2$, from the fact that $k = n - d + 1$, and $d = 2t + 1$, and $n = p^r - 1$.

This means that we have a total of $(t + 1) + (p^r - t - 2 + 1) = p^r$ unknowns. Since we have more equations than unknown variables, we are guaranteed to have a solution in which the coefficients are not all zero.

We can represent the system of equations as the matrix multiplication:

$$\begin{pmatrix} r_0 \cdot (\alpha^0)^0 & r_0 \cdot (\alpha^0)^1 & \cdots & r_0 \cdot (\alpha^0)^t & (\alpha^0)^0 & (\alpha^0)^1 & \cdots & (\alpha^0)^{t+k-1} \\ r_1 \cdot (\alpha^1)^0 & r_1 \cdot (\alpha^1)^1 & \cdots & r_1 \cdot (\alpha^1)^t & (\alpha^1)^0 & (\alpha^1)^1 & \cdots & (\alpha^1)^{t+k-1} \\ r_2 \cdot (\alpha^2)^0 & r_2 \cdot (\alpha^2)^1 & \cdots & r_2 \cdot (\alpha^2)^t & (\alpha^2)^0 & (\alpha^2)^1 & \cdots & (\alpha^2)^{t+k-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{p^r-2} \cdot (\alpha^{p^r-2})^0 & r_{p^r-2} \cdot (\alpha^{p^r-2})^1 & \cdots & r_{p^r-2} \cdot (\alpha^{p^r-2})^t & (\alpha^{p^r-2})^0 & (\alpha^{p^r-2})^1 & \cdots & (\alpha^{p^r-2})^{t+k-1} \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_t \\ d_0 \\ \vdots \\ d_{t+k-1} \end{pmatrix}.$$

We can calculate the nullspace of the coefficient matrix M, and this will be a basis consisting of one or more non-zero vectors. We are guaranteed to have at least one nonzero vector in the nullspace since we know that we have at least one nonzero solution to our system of equations.

To show that we have covered all possible solutions for our system of equations, we will to show that the dimension of the null space of M is exactly equal to $t - wt(e) + 1$, using the fact that $\text{nullity}(M) + \text{rank}(M) = \text{number of columns in } M$. We know that M has $2t + k + 1$ columns, so it will suffice to show that the rank of M is $t + k + wt(e)$. We can do this by using row and column operations on M to row-reduce it, and counting the number of linearly independent, nonzero rows.

$$M = \begin{pmatrix} C_1 & D_1 \\ C_2 & D_2 \end{pmatrix}$$

Figure 1: The breakdown of M

For discussion purposes, we will be denoting sections of M as in Figure 1, where the dimensions of C_1 are $(t+k) \times (t+1)$, the dimensions of C_2 are $(t) \times (t+1)$, the dimensions of D_1 are $(t+k) \times (t+k)$ and the dimensions of D_2 are $(t) \times (t+k)$. The submatrices C_1 and C_2 together make up the coefficients corresponding to the c_i variables, and we will call this $(2t+k) \times (t+1)$ submatrix C. Similarly, D_1 and D_2 together make up the coefficients corresponding to the d_i variables, and we will call this $(2t+k) \times (t+k)$ submatrix D.

Let $f(x) = b_0 + b_1 x + b_2 x^2 + \cdots + b_{k-1} x^{k-1}$. Consider the jth column of our coefficient matrix written as:

$$\begin{pmatrix} (b_0 + b_1\alpha^0 + \cdots + b_{k-1}(\alpha^0)^{k-1} + e_0) \cdot 1 \\ (b_0 + b_1\alpha^1 + \cdots + b_{k-1}(\alpha^1)^{k-1} + e_1) \cdot \alpha^j \\ (b_0 + b_1\alpha^2 + \cdots + b_{k-1}(\alpha^2)^{k-1} + e_2) \cdot \alpha^{2j} \\ \vdots \\ (b_0 + b_1\alpha^{q-2} + \cdots + b_{k-1}(\alpha^{q-2})^{k-1} + e_{q-2}) \cdot \alpha^{(q-2)j} \end{pmatrix}.$$

(Note: $0 \leq j \leq t$.) If we look at all of the terms with coefficient b_l where $0 \leq l \leq k-1$, notice that this is exactly b_l multiplied by the column in D corresponding to $\alpha^{i(j+l)}$. Then each of these can be reduced using column operations to all zeroes in the entries where the received word agrees with the codeword. For example, to get rid of the entries with coefficient b_0, we multiply b_0 by the first column in D (corresponding to α^0) and subtract this from our column in C. To get rid of the entries with coefficient b_1, we multiply by the column in D corresponding to α, and similarly for all the b_l . When we do this for every b_l, we are left with all zeroes in that column of C except for in the rows corresponding to the error locations because the $e_i \cdot \alpha^{i(j+1)}$ term remains for all $0 \leq i \leq q-2$ where $e_i \neq 0$.

We can do this for all the columns in C because the lowest term we need to get rid of is the b_0 term in the first column of C, which can be reduced by multiplying the column in D corresponding to α^0 by b_0 and subtracting it from the column in C. The highest term we need to get rid of is the b_{k-1} term in the j-th column of C where $j = t$, which can be reduced by multiplying the column in D corresponding to α^{t+k-1} by b_{k-1} and subtracting it from the column in C. Now after reducing every column in C, we are left with the same number of nonzero rows as we have errors. After column reduction, we move all the rows corresponding to the error locations to the rows starting at the $(t+k+1)$-th row of our matrix. Since we have t or fewer errors, these rows will always fit in the bottom t rows of our matrix where they will remain unaffected by the row reduction that follows.

Next, let us consider D. Any $(t+k) \times (t+k)$ submatrix of D is a Vandermonde matrix of the form:

$$\begin{pmatrix} (\alpha^{i_1})^0 & (\alpha^{i_1})^1 & \cdots & (\alpha^{i_1})^{t+k-1} \\ (\alpha^{i_2})^0 & (\alpha^{i_2})^1 & \cdots & (\alpha^{i_2})^{t+k-1} \\ \cdots & \cdots & \ddots & \cdots \\ (\alpha^{i_{t+k}})^0 & (\alpha^{i_{t+k}})^1 & \cdots & (\alpha^{i_{t+k}})^{t+k-1} \end{pmatrix},$$

where $\alpha^{i_j} \neq \alpha^{i_k}$ provided $j \neq k$. Note that this will be a Vandermonde matrix regardless of the order of the a_i's, as long as they are all distinct, which is the case for D, even after the row operations. Since this is a Vandermonde matrix, we know that this set of $t+k$ rows is linearly independent. Therefore, we can use row operations on M to simplify D to a $(t+k) \times (t+k)$ identity matrix followed by t rows of zeroes below. These row operations will not affect any values within C because we will be adding linear combinations of the first $t+k$ rows of M to each other and to the last t rows of M, but all of the first $t+k$ rows in C have already been simplified to all zeroes.

Finally, we have to consider the remaining t rows at the bottom of the matrix, some of which may consist of all zeroes. We have $wt(e)$ rows remaining. All of these rows are linearly independent. To see this, consider any two rows, say i and j. The first term in these rows is exactly r_i and r_j. Now, these are both constants so $r_i = a \cdot r_j$ for some nonzero field element a, and we can add a times row j to row i and cancel out the first term in row i. Looking at the second term in each, this means that we added $a \cdot r_j \alpha^{2j}$ to $r_i \alpha^{2i} = (a \cdot r_j)\alpha^{2i}$, but $a \cdot r_j \alpha^{2j} \neq a \cdot r_j \alpha^{2i}$ unless $i = j$, which we know is not the case. Therefore, the remaining $wt(e)$ nonzero rows in C_2 are linearly independent.

In summary, we have $(t+k) + wt(e)$ linearly independent rows in matrix M. Therefore, the rank of M is $t+k+wt(e)$. This means that nullity$(M) = (2t+k+1) - (t+k+wt(e)) = t+1-wt(e)$. Since the dimension of the basis of our nullspace is exactly the number of basis elements we already know of, our nullspace consists of exactly those vectors of the form $Q(x,y) = w(x)p(x)y - w(x)p(x)f(x)$ and the element of minimal degree out of all possible solutions is $Q(x,y) = w(x)y - w(x)f(x)$. □

The result of Theorem 1 tells us that when we find the nullspace of our coefficient matrix, we can factor any element of the nullspace and still recover the original $f(x)$ from among the factors.

We will now extend the idea behind the key equation to cases where we may have more than one candidate $f(x)$.

4 Lists of Size Greater than One

So far, we have considered only the special case of list sizes equal to one. We would now like to consider cases of list sizes that are greater than one. We can use the Guruswami-Sudan list decoding method to increase the desired multiplicity for a polynomial root of our $Q(x,y)$, which will not decrease our decoding radius. At some point, the multiplicity will continue to increase, but the decoding radius, having

reached its maximum, will remain the same. We are concerned with the multiplicity at which the maximum radius is first reached.

For any set values of n, k, and multiplicity m, we can calculate the corresponding decoding radius, t_m using the following process:

1. Pick l so that $C(k-1,l) > n\binom{m+1}{2}$.
2. Pick K_m so that $K_m = \min\{K : C(k-1, mK-1) > n\binom{m+1}{2}\}$.
3. If $mK_m > l$, then $t_m = n - K_m$.

For any n and k, the value of t_m will increase up to a certain point, denoted t_∞. As given in [5], the value of K_∞ for which K_m will no longer decrease is given by the formula:

$$K_\infty = \lfloor \sqrt{vn} + 1 \rfloor. \tag{8}$$

Therefore, using $v = k - 1$, we have:

$$t_\infty = n - 1 - \lfloor \sqrt{(k-1)n} \rfloor. \tag{9}$$

To understand how the procedures for calculating t_m and t_∞ for a given n, k, and m, consider the following example.

Example 2. Let $n = 15$ and $k = 6$. For $m = 2$, we have the following calculations for K_2 and t_2:

$$\begin{aligned} l &= 18, \text{ because } C(5,18) = 46 \text{ and } 15\binom{3}{2} = 45, \\ K_2 &= 10, \text{ because } C(5,19) = 50 \text{ while } C(5,17) = 42, \\ t_2 &= 15 - 10 = 5. \end{aligned}$$

We can also achieve a better decoding radius, as can be seen by the values of K_∞ and t_∞:

$$\begin{aligned} K_\infty &= \lfloor \sqrt{(6-1)15} \rfloor + 1 = 9, \\ t_\infty &= 15 - 1 - \lfloor \sqrt{(6-1)15} \rfloor = 6. \end{aligned}$$

As a note, it can be shown that this maximum decoding radius will first be achieved when $m = 6$.

Example 3. We will use an $RS(2^4, 13)$ code with multiplicity $m = 1$. The code parameters give $n = 15$, $k = 3$. The first thing we will need to do is to calculate the values of l and t_1 so that we will know what polynomial to use for $Q(x, y)$ and the number of errors we can expect to correct.

First, we construct a table showing l and $C(2, l)$ for a range of values of l, as seen in Figure 2.

We need $C(2,l) > 15\binom{2}{2} = 15$, so we pick $l = 6$. Then, we need the minimum K such that $C(2, 1 \cdot K - 1) > 15$, so we have K_1=7. Since we meet the condition of

l	0	1	2	3	4	5	6	7	8	9	10
$C(2,l)$	1	2	4	6	9	12	16	20	25	30	36

Figure 2: Table of $C(2,l)$

$mK_m > l$ because $7 > 6$, we will have the decoding radius $t_1 = 15 - 7 = 8$. Note that this is an improvement over the number errors we would expect to correct with one of the standard decoding methods as we have $13 = 2t + 1$ so $t = 6$.

Since $k - 1 = 2$ and $l = 6$, our $Q(x, y)$ polynomial will include all monomials with degree at most 6 using a $\mathbf{w} = (1, 2)$ weighted degree. Therefore, we have the following $Q(x, y)$:

$$\begin{aligned} Q(x,y) &= c_1y^3 + (c_2 + c_3x + c_4x^2)y^2 + (c_5 + c_6x + c_7x^2 + c_8x^3 + c_9x^4)y \\ &\quad +(c_{10} + c_{11}x + c_{12}x^2 + c_{13}x^3 + c_{14}x^4 + c_{15}x^5 + c_{16}x^6). \end{aligned}$$

Letting α be a primitive element of the field $\mathbb{F}_{2^4}$, we can begin the transmission and decoding of an actual codeword. First, let our codeword, $b(x)$, be:

$$\begin{aligned} b(x) &= x^{14} + \alpha x^{13} + x^{12} + (\alpha^3 + \alpha^2 + \alpha + 1)x^{11} + (\alpha^3 + \alpha^2)x^9 + (\alpha + 1)x^8 \\ &\quad +\alpha x^7 + (\alpha^3 + \alpha^2)x^6 + (\alpha^3 + \alpha^2 + \alpha)x^5 + (\alpha + 1)x^4 \\ &\quad +(\alpha^3 + \alpha^2 + \alpha + 1)x^3 + (\alpha^3 + \alpha^2 + 1)x^2 + (\alpha^3 + \alpha^2 + 1)x. \end{aligned}$$

We will introduce the error, $e(x)$ as:

$$e(x) = x^{14} + x^{13} + x^{12} + x^{11} + x^3 + x^2 + x + 1$$

This gives us the received word, $r(x)$:

$$\begin{aligned} r(x) &= (\alpha + 1)x^{13} + (\alpha^3 + \alpha^2 + \alpha)x^{11} + (\alpha^3 + \alpha^2)x^9 + (\alpha + 1)x^8 + \alpha x^7 \\ &\quad +(\alpha^3 + \alpha^2)x^6 + (\alpha^3 + \alpha^2 + \alpha)x^5 + (\alpha + 1)x^4 + (\alpha^3 + \alpha^2 + \alpha)x^3 \\ &\quad +(\alpha^3 + \alpha^2)x^2 + (\alpha^3 + \alpha^2)x + 1. \end{aligned}$$

Using $r(x)$ to create a system of equations using $Q(x, y)$, we create a coefficient matrix and find the nullspace. The nullspace of this system has dimension exactly four, of which we will consider one in detail. Using one and plugging in the values for the coefficients in $Q(x, y)$ and factoring $Q(x, y)$, we get the following:

$$\begin{aligned} Q(x,y) &= (\alpha^3 + \alpha^2 + 1)(y + \alpha x^2 + (\alpha^3 + \alpha^2)x + \alpha^3 + \alpha^2 + \alpha + 1) \\ &\quad \times y(y + \alpha x^2 + (\alpha^3 + \alpha^2)x + \alpha^3 + \alpha^2 + \alpha). \end{aligned}$$

This gives us the following as the possible polynomials to use in our $ev(f(x))$ mapping:

$$\begin{aligned} f_1(x) &= \alpha x^2 + (\alpha^3 + \alpha^2)x + \alpha^3 + \alpha^2 + \alpha + 1, \\ f_2(x) &= 0, \\ f_3(x) &= \alpha x^2 + (\alpha^3 + \alpha^2)x + \alpha^3 + \alpha^2 + \alpha. \end{aligned}$$

So, using the $ev(f(x))$ mapping, we get the three candidate codewords:

$$\begin{aligned}
b_1(x) &= (\alpha+1)x^{13} + (\alpha^3+\alpha^2+\alpha)x^{11} + x^{10} + (\alpha^3+\alpha^2+1)x^9 + \alpha x^8 \\
&\quad +(\alpha+1)x^7 + (\alpha^3+\alpha^2+1)x^6 + (\alpha^3+\alpha^2+\alpha+1)x^5 + \alpha x^4 \\
&\quad +(\alpha^3+\alpha^2+\alpha)x^3 + (\alpha^3+\alpha^2)x^2 + (\alpha^3+\alpha^2)x + 1, \\
b_2(x) &= 0, \\
b_3(x) &= x^{14} + \alpha x^{13} + x^{12} + (\alpha^3+\alpha^2+\alpha+1)x^{11} + (\alpha^3+\alpha^2)x^9 + (\alpha+1)x^8 \\
&\quad +\alpha x^7 + (\alpha^3+\alpha^2)x^6 + (\alpha^3+\alpha^2+\alpha)x^5 + (\alpha+1)x^4 \\
&\quad +(\alpha^3+\alpha^+\alpha+1)x^3 + (\alpha^3+\alpha^2+1)x^2 + (\alpha^3+\alpha^2+1)x.
\end{aligned}$$

Both $b_1(x)$ and $b_3(x)$ are codewords within a Hamming distance of 8 from our received codeword, so they are on the list. Note that $b_3(x)$ is the original codeword that was transmitted. Also, it can be checked that using any of the other elements in the nullspace would yield lists containing both $b_1(x)$ and $b_3(x)$ as candidates for the transmitted word.

This is an example of Theorem 7.19 from [5], which guarantees that the factors $(y - f(x))$ appear in $Q(x, y)$.

Theorem 2. *Let $Q(x, y)$ be an interpolating polynomial of $(1, v)$-weighted degree $\leq l$ such that $D_{r,s}Q(x_i, y_i) = 0$ for $i = 1, 2, \ldots, n$ and for all $r + s < m$. (That is, each(x_i, y_i) is interpolated up to order m.) Let $p(x)$ be a polynomial of degree at most v such that $y_i = p(x_i)$ for at least K_m values of i in $\{1, 2, \ldots, n\}$. If $mK_m > l$, then $(y - p(x)) \mid Q(x, y)$.*

5 Multiplicity

Expanding on the work done in [1], we created multiplicity graphs for $4 \leq n \leq 127$. These graphs take a given n and calculate the multiplicity, $m(k)$, as defined below, for each $2 \leq k \leq n$.

Definition 10. Given a blocklength, n, the value $m(k)$ represents the multiplicity for which the maximum decoding radius, t_∞ is first reached for a given k-value.

First we compared the overall shape of the graphs. Each graph had a similar pattern of peaks and valleys as seen in Figure 4a-c. For every $n = 4s$ where $s \in \mathbb{N}$ the smoother curve normally occurring near $k = \frac{1}{4}n + 1$ is replaced by two higher peaks as shown in Figure 4d. Also, note that for $k \geq n - 1$ the code cannot correct any errors.

Then we considered the maximum point of these graphs. In all the cases that Eriksson looked at the maximum was unique, so we studied whether or not this was always true.

Proposition 1. *The value of k for which $m(k)$ is maximal is not necessarily unique.*

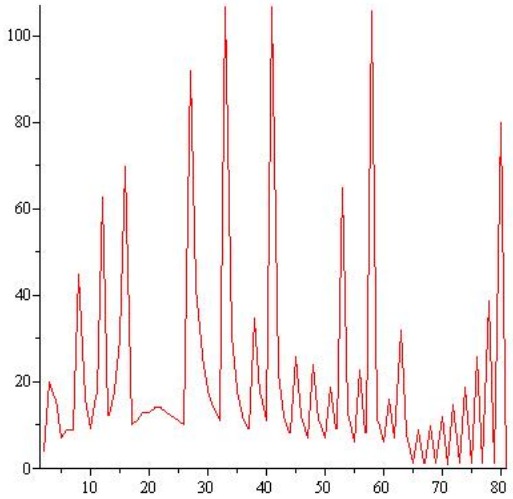


Figure 3: $m(k)$ as a function of k for $n = 81$.

Proof. Consider the graph of $m(k)$ for $n = 81$. As seen in Figure 3, we have $m(33) = m(41) = 107$. Also, note that the value of $m(58) = 106$, so it is not a third maximum. Therefore, the value of k for which $m(k)$ is maximal is not unique. It is still unknown whether three absolute maxima can exist. □

Then we looked at the n-values for which $n = q - 1$, where q is the size of the field. These values represent the most commonly studied blocklengths. The results can be seen in Figure 5.

Codes for other n also exist and are called shortened Reed-Solomon codes. We also considered these cases. The following graph shows the relationship between every n and the k where the maximum $m(k)$ occurs. The upper bound is the line $k = n - 1$ because $k \not\geq n$ by definition. (Note: There are two k-values for $n = 81$.)

Taking a closer look at Figure 6, we conjecture that for every $n = 4p^s$ where p is prime and $s \in \mathbb{N}$, $k = \frac{1}{4}n + 2$. Also, note that for $n > 32$ this appears to be a lower bound for the graph.

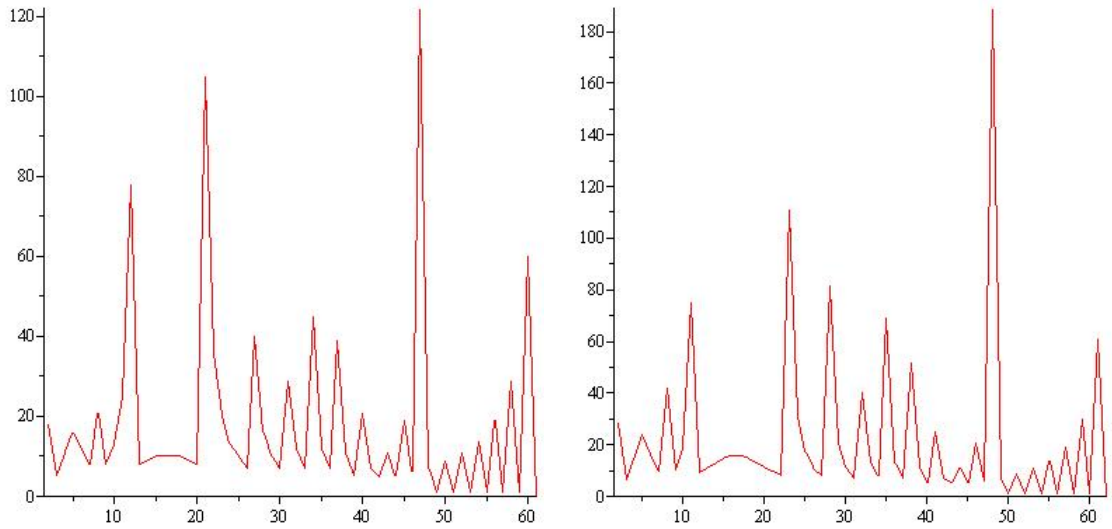
Returning to our original graphs, we then studied the two peaks appearing at $k = \frac{1}{4}n$ and $k = \frac{1}{4}n + 2$ for any n that is a multiple of 4. We noticed that the multiplicity attained at these points was dependent on n. For $k = \frac{1}{4}n$, $m(k)$ is of the form

$$m(k) = \frac{1}{4}n\left(\frac{1}{2}n - 1\right),$$

and for $k = \frac{1}{4}n + 2$, $m(k)$ is of the form

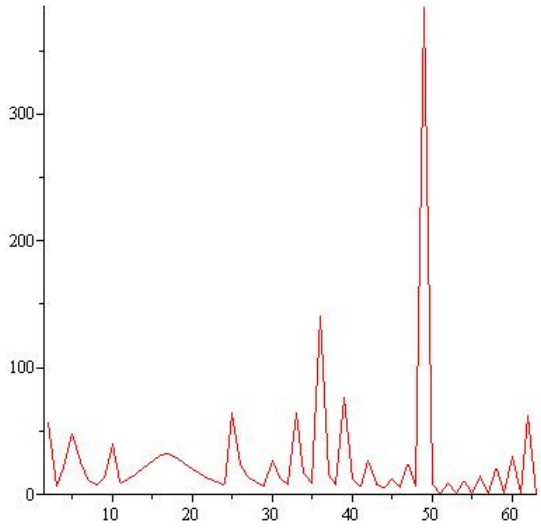
$$m(k) = \frac{1}{4}n\left(\frac{1}{2}n + 1\right).$$

We also noted that the point between the peaks at $k = \frac{1}{4}n + 1$ had a very low multiplicity. The multiplicity at this point also has a closed form, $m(k) = \lfloor \frac{1}{2}k \rfloor$. (Note: It is not known whether or not these formulas hold for $n > 127$.)

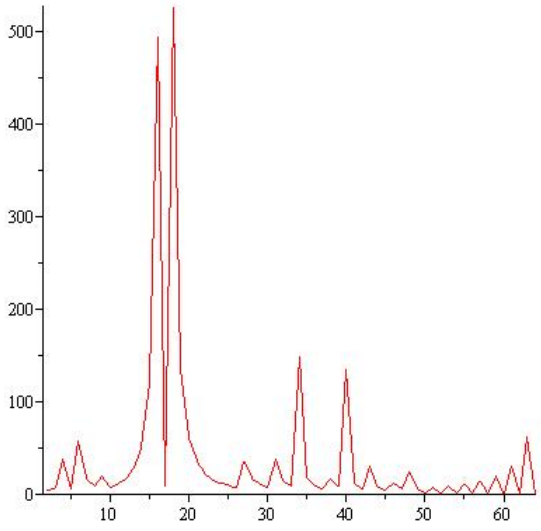


(a) $n = 61$

(b) $n = 62$



(c) $n = 63$



(d) $n = 64$

Figure 4: Graphs of k vs. $m(k)$ for several values of n.

q	n	k	$m(k)$
4	3	2	2
8	7	6	6
16	8	9	33
32	31	18	69
64	63	49	385
128	127	98	777
256	255	113	9633

Figure 5: Table for $n = q - 1$

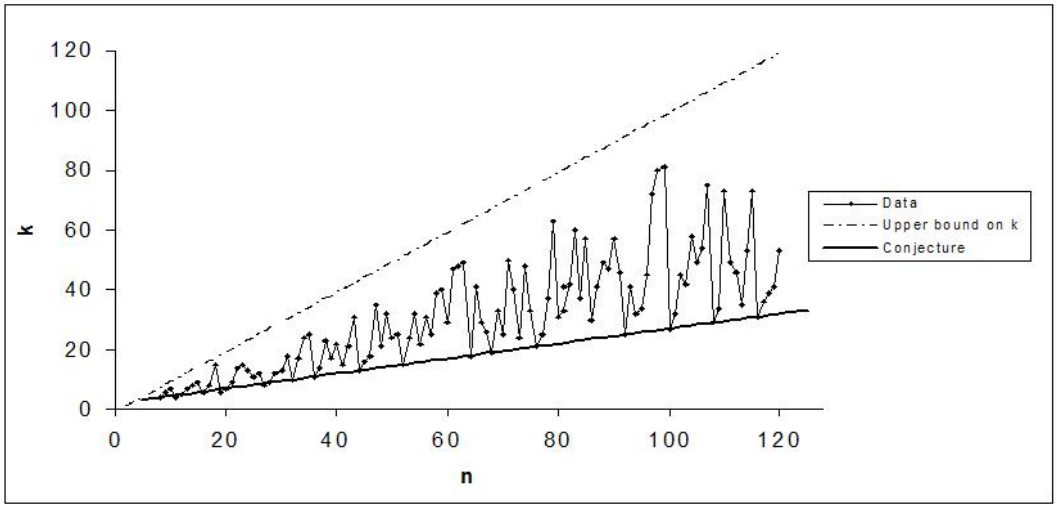


Figure 6: Data for which k produces the maximum $m(k)$ given n.

6 Further Research

Looking at the data we collected, we made many interesting observations, but did not have time to explore them all. The first of which is proving our equations dealing with the two peaks for all $n = 4s$ and not just $n \leq 127$.

Another intriguing observation that we noted was that given any $n \leq 127$, the multiplicity never exceeds

$$\frac{1}{3}\left(\frac{4}{9}n^2 - 1\right),$$

and this bound is reached by every n that is a multiple of 3, but not a multiple of 9.

We would also like to further explore the reason for the special properties that occur at $n = 4s$ for $s \in \mathbb{N}$.

We would like to extend our collection of graphs past $n = 127$. This will help to demonstrate our conjectured bounds further and maybe lead to proofs of these bounds.

7 Conclusions

In this paper, we discussed the Guruswami-Sudan method for list decoding algorithms, including the special case of lists of length one. We adapted the Welch-Berlekamp decoding method for Reed-Solomon codes to decode with lists of length exactly one. We proved that the modified version of the Welch-Berlekamp key equation will always give a system of equations whose solutions are made up of polynomials whose roots always include the polynomial that was used to generate the transmitted codeword.

We also looked at lists with length greater than one and examined the effects of increasing the multiplicity used for decoding with fixed code parameters n and k. For any values of n and k, increasing the desired multiplicity increases the decoding radius, up to a certain value of m, called $m(k)$, at which point $t_m = t_\infty$ for all higher multiplicities m, such that $m \geq m(k)$. We generated and analyzed the data contained in the results for $4 \leq n \leq 127$ and the corresponding values $2 \leq k \leq n-2$. We showed that there is not necessarily a value of k for a given n that would require a uniquely high multiplicity to attain the maximum decoding radius, and also presented several other conjectures on patterns and bounds relating combinations of n, k, $m(k)$, and $t_{m(k)}$. We finished by presenting several of the patterns and bounds in the graphs that we did not have time to fully investigate.

8 Acknowledgments

This work was conducted during the 2009 Mathematical Sciences Research Institute Undergraduate Program (MSRI-UP) in Berkeley, CA. MSRI-UP is supported by the National Science Foundation (grant No. DMS-0754872) and the National Security Agency (grant No. H98230-09-0103). We would like to thank our advisor, Dr. John Little and the MSRI-UP 2009 director, Dr. Herbert Medina for their support and direction. We would also like to thank Dr. Emille Davie, Candice Price and Ashley Wheeler for their guidance throughout the program. Finally, we would like to thank Dr. Erik Guentner, Dr. Matt Koetz, and Dr. Yousuf George for encouraging us to pursue this opportunity.

References

[1] Eriksson, Jonas, "Aspects of List-of-Two Decoding," Dissertation, Linköpings universitet (No. 1010), 2006.

[2] Guruswami, V.; Sudan, M., "Improved Decoding of Reed-Solomon Codes and Algebraic Geometry Codes," *IEEE Trans. Info. Theory*, vol. 45, no. 6., pp. 1757–1767, Sept. 1999.

[3] Hankerson, D.C.; G. Hoffman; D.A. Leonard; et. al., *Coding Theory and Cryptography: The Essentials*, 2nd Ed., Marcel Dekker, New York, 2000.

[4] Little, John, *MSRI-UP 2009 Coding Theory*, MSRI-UP text, Berkeley, CA, 2009.

[5] Moon, Todd K., *Error Correction Coding*, Wiley-Interscience, Hoboken, 2005.

[6] Sudan, M., "Decoding of Reed-Solomon Codes Beyond the Error-Correction Bound," *J. Complexity*, vol. 13, pp. 180–193, 1997.