On Duursma Zeta Functions of Type IV Virtual Codes

Midn Sarah Catalano

4-21-2008

Contents

1 Motivation **1**

2 Introduction **2**
2.1 General Background 2
2.2 MacWilliams Identity 5

3 Duursma Zeta Function **7**
3.1 Definition 7
3.2 Extremal Virtual Self-Dual Weight Enumerators 11

4 Examples **12**

1 Motivation

1. Considerable work has been devoted to the study of self-dual codes. Iwan M. Duursma has written numerous papers on the matter (see [D1]-[D6]) and the greater part of this project is centered on his ground breaking work in the field. In 1999, Iwan M. Duursma defined the zeta function for a linear code as a generating function of its Hamming weight enumerator. The modest goal of this project is to go through Duursma's papers and evaluate its relevance for formally self-dual codes. Duursma's work in *Extremal Weight Enumerators and Ultraspherical Polynomials* will be extended to formally self-dual codes. More specifically, the project expands Duursma's work in this paper to zeta functions of formally self-dual codes of Type IV. (In fact, Duursma's work extends to the even broader class of virtual self-dual weight enumerators of Type IV. Theorems 9 and 11 below are extended to the weight enumerator case. See the remark before Definition 12 in §3 for details.)

2. The final and more ambitious goal of the project is to study the formulation for a Riemann hypothesis analog. The unsolved Riemann hypothesis has been a mystery since Riemann's work in the 1800's. The search for an analog for linear codes arose in the 1990's. This hypothesis deals with the nature of non-trivial zeros for zeta functions. The main result of this project, which deals with the Riemann hypothesis analog in a special case, is Theorem 16 below.

3. Examples of Duursma Zeta functions of self-dual codes of small length are computed in §4 with the help of the mathematical software program SAGE [S].

2 Introduction

Studying error correcting codes is necessary in advancing technology. The safe and reliable transfer of information depends on coding theory. The problem of transferring dependable information is important and this research project attempts to continue incremental progress in this field.

This project has value because the study of error correcting codes is relatively new. Profound advancement can be made in this field and is necessary for the transfer of reliable and safe information. The study of error correcting codes, as conducted in this research project, is specifically related to reliable communications. This is especially pertinent to Naval Reactors as nuke power begins to rely more heavily on computer communications. The need for such communication to be reliable is paramount. Error correcting codes are vital to systems, such as nuclear reactors, that depend upon computers and cannot be allowed to miscommunicate. More knowledge of this topic is crucial to the overarching goals of the Navy. This is especially true if the Navy continues with its plans for unmanned ships. The military depends upon reliable, fault–tolerant communication and this research project aims to continue research in the pertinent field of coding theory.

2.1 General Background

Let $\mathbb{F} = GF(q)$ denote a finite field with q elements, where q is a power of a prime. A **linear code** is a subspace of $\mathbb{F}^n$ for some $n > 1$. This integer n is called the **length** of C. Let C be a linear code of length n over $\mathbb{F}$. If $q = 2$

then the code is called **binary**. Similarly, if $q = 3$ then the code is called **ternary** and if $q = 4$ then the code is called **quaternary**. Throughout, assume that $\mathbb{F}^n$ has been given the standard basis $e_1 = (1, 0, ..., 0) \in \mathbb{F}^n$, $e_2 = (0, 1, 0, ..., 0) \in \mathbb{F}^n$, ..., $e_n = (0, 0, ..., 0, 1) \in \mathbb{F}^n$ and the usual dot product. The **dimension** of C is denoted k, so the number of elements of C is equal to q^k.

Another important parameter associated to the code is the number of errors which it can, in principle, correct. The Hamming metric is useful for quantifying such errors. For any two $x, y \in \mathbb{F}^n$, let $d(x, y)$ denote the number of coordinates where these two vectors differ:

$$d(x, y) = |\{0 \leq i \leq n \mid x_i \neq y_i\}|. \tag{1}$$

Define the **weight** wt of $v \in \mathbb{F}^n$ to be the number of non-zero entries of v. Note, $d(x, y) = \mathrm{wt}(x - y)$ because the vector $x - y$ has non-zero entries only at locations where x and y differ. The smallest distance between distinct codewords in a linear code C is the **minimum distance** of C:

$$d = d(C) = min_{c \in C,\, c \neq 0} d(0, c) \tag{2}$$

(for details see [HILL] Theorem 5.2). Call a linear code of length n, dimension k, and minimum distance d an $[n, k, d]$ **code**, or $[n, k]$ **code** if we wish to disregard the minimum distance. The **Singleton Bound** states that if an $[n, k, d]$ linear code over $\mathbb{F}$ exists, then $k \leq n - d + 1$. An **MDS Code**, or Maximum Distance Seperable, is one where the equality holds.

A linear code C of length n and dimension k over $\mathbb{F}$ has a basis of k vectors of length n. If those vectors are arranged as rows of a matrix G, call the $k \times n$ matrix G a **generator matrix** for C.

The **dual code** of C is the vector space of all code words in $\mathbb{F}^n$ which are orthogonal to each codeword in C;

$$C^{\perp} = \{v \in \mathbb{F}^n \mid v \cdot c = 0 \ \forall \ c \in \ C\}.$$

C is **self-dual** if $C = C^{\perp}$.

Example 1 *Let*

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

be the generator matrix of a code C. This is a binary self-dual $[8,4,4]$ code. In fact, this is an extremal Type II code (these terms will be defined below).

Example 2 *Let*

$$G = \begin{pmatrix} 2 & 0 & 1 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 2 & 0 & 1 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 2 & 0 & 1 & 2 & 1 & 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 & 1 & 2 & 1 & 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 2 & 0 & 1 & 2 & 1 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 2 & 1 & 1 & 2 \end{pmatrix}$$

be the generator matrix of a code C. This is a ternary self-dual $[12,6,6]$ code. In fact, this is an extremal Type III code (these terms will be defined below).

Example 3 *Define the finite field of four elements as follows. Let z denote a root of the quadratic polynomial $x^2+x+1 \in GF(2)[x]$, where $GF(2)[x]$ denotes the polynomial ring in the indeterminate x. Let $GF(4) = \{0,1,z,z+1\}$. This set is a field of characteristic 2. Let*

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & z & z \\ 0 & 1 & 0 & z & 1 & z \\ 0 & 0 & 1 & z & z & 1 \end{pmatrix}$$

be the generator matrix of a code C. This is a quaternary self-dual $[6,3,4]$ code and is referred to as the **hexacode**. *In fact, this is an extremal Type IV code (these terms will be defined below). Note that this code is MDS.*

The dual code of C has parameters $[n, n-k]$. Moreover, denote the minimum distance of the dual code by $d^{\perp}$. For future reference, note that if $C = C^{\perp}$ then (equating dimensions) $k = n-k$, forcing n to be even and $k = n/2$. The **genus** of an $[n,k,d]$-code C is defined by

$$\gamma(C) = n+1-k-d.$$

This measures how "far away the code is from being MDS".

Lemma 4 *If C is a self-dual code then its genus satisfies $\gamma = n/2+1-d$.*

proof: It suffices to show that $k = n/2$ if $C = C^{\perp}$. But this was observed in the discussion above. $\square$

The **(Hamming) weight enumerator polynomial** A_C is defined by

$$A_C(x, y) = \sum_{i=0}^{n} A_i x^{n-i} y^i = x^n + A_d x^{n-d} y^d + \cdots + A_n y^n,$$

where

$$A_i = |\{c \in C \mid \mathrm{wt}(c) = i\}|$$

denotes the number of codewords of weight i. Let $W_C(z) = A_C(1, z)$, so therefore $A_C(x, y) = x^n W_C(y/x)$. The **support of** C is the set $\mathrm{supp}(C) = \{i \mid A_i \neq 0\}$. If $A_C(x, y) = A_{C^{\perp}}(x, y)$ then C is called a **formally self-dual code**. The **spectrum of** C is the list of coefficients of A_C:

$$spec(C) = [A_0, \ldots, A_n].$$

Two codes are **formally equivalent** if they have the same spectrum.

2.2 MacWilliams Identity

The goal of this section is to prove the MacWilliams identity (for simplicity, restricted to the binary case). This identity is necessary to verify the functional equation (4) for the Duursma Zeta Function. Several lemmas are needed to prove this identity. The proof given below follows Hill Ch. 13 [HILL].

Lemma 5 *Let C be a binary linear $[n, k]$ code.*

1. *Fix $y \in GF(2)^n - C^{\perp}$. As x ranges over the vector space C, the quantity $x \cdot y$ takes the value 0 and 1 equally often.*

2. *The following identity holds:*

$$\sum_{c \in C} (-1)^{c \cdot y} = \begin{cases} 2^k, & y \in C^{\perp}, \\ 0 & y \notin C^{\perp}. \end{cases}$$

proof: Part 1: Let $A = \{x \in C \mid x \cdot y = 0\}$ and $B = \{x \in C \mid x \cdot y = 1\}$. Let u be a codeword of C such that $u \cdot y = 1$. Let $u + A = \{u + a \mid a \in A\}$.

Then $u + A \subset B$, for if $a \in A$, then $(u + a) \cdot y = a \cdot y + u \cdot y = 0 + 1 = 1$. Similarly $u + B \subset A$. Hence, $|A| = |u + A| \leq |B| = |u + B| \leq |A|$. Thus, $|A| = |B|$.

Part 2: If $y \in C^{\perp}$, then $c \cdot y = 0$ for all $c \in C$, and so $\sum_{c \in C} (-1)^{c \cdot y} = |C| \cdot 1 = 2^k$. If $y \notin C^{\perp}$ then by Part 1, as x ranges over the vector space C, the quantity $x \cdot y$ takes the value 0 and 1 equally often, giving $\sum_{c \in C} (-1)^{c \cdot y} = 0$. $\square$

Lemma 6 *For each $x \in GF(2)^n$ the following polynomial identity holds:*

$$\sum_{y \in GF(2)^n} z^{\mathrm{wt}(y)} (-1)^{x \cdot y} = (1 - z)^{\mathrm{wt}(x)} (1 + z)^{n - \mathrm{wt}(x)}.$$

proof:

$$\begin{aligned}
\textstyle\sum_{y \in GF(2)^n} z^{\mathrm{wt}(y)} (-1)^{x \cdot y} &= \textstyle\sum_{y_1 \in \{0,1\}} \sum_{y_2 \in \{0,1\}} \cdots \sum_{y_n \in \{0,1\}} z^{y_1 + \cdots + y_n} (-1)^{x_1 y_1 + \cdots x_n y_n} \\
&= \textstyle\sum_{y_1 \in \{0,1\}} \cdots \sum_{y_n \in \{0,1\}} \left(\prod_{i=1}^n z^{y_i} (-1)^{x_i y_i} \right) \\
&= \textstyle\prod_{i=1}^n \left(\sum_{j \in \{0,1\}} z^j (-1)^{x_i j} \right) \\
&= (1 - z)^{\mathrm{wt}(x)} (1 + z)^{n - \mathrm{wt}(x)},
\end{aligned}$$

since $\sum_{j \in \{0,1\}} z^j (-1)^{rj} = 1 + z$, if $r = 0$, and $= 1 - z$, if $r = 1$. $\square$

Theorem 7 *(MacWilliams' identity): If C is a linear code over any finite field $\mathbb{F}$ of order q then*

$$A_{C^{\perp}}(x, y) = |C|^{-1} A_C(x + (q - 1)y, x - y).$$

This is the general statement of the MacWilliams identity. This proof will restrict to the binary case.

proof: Now, express the polynomial $f(z) = \sum_{c \in C} \sum_{y \in GF(2)^n} z^{\mathrm{wt}(y)} (-1)^{c \cdot y}$ in two ways.

On one hand, Lemma 6 implies

$$\begin{aligned}
f(z) &= \textstyle\sum_{c \in C} (1 - z)^{\mathrm{wt}(c)} (1 + z)^{n - \mathrm{wt}(c)} \\
&= (1 + z)^n \textstyle\sum_{c \in C} \left(\frac{1 - z}{1 + z} \right)^{\mathrm{wt}(c)} \\
&= (1 + z)^n W_C\left(\tfrac{1 - z}{1 + z} \right) = A_c(1 + z, 1 - z)
\end{aligned}$$

On the other hand, reversing the order of summation gives

$$\begin{aligned} f(z) &= \textstyle\sum_{y\in GF(2)^n} z^{\mathrm{wt}(y)}\big(\sum_{c\in C}(-1)^{c\cdot y}\big) \\ &= \textstyle\sum_{y\in C^\perp} z^{\mathrm{wt}(y)}2^k \quad \text{(by Lemma 5, Part 2)} \\ &= 2^k W_{C^\perp}(z). \end{aligned}$$

Replace z by y/x in the above gives

$$A_c(1+y/x, 1-y/x) = 2^k \cdot A_(C^\perp)(1, y/x).$$

Since this polynomial is homogeneous in degree n, multiplying both sides by x^n gives the theorem in the binary case. □

If $C = C^\perp$, then $|C| = q^{n/2}$. Therefore, the MacWilliam's Identity can be rewritten in this case as

$$A_C(x,y) = q^{-n/2}A_C(x+(q-1)y, x-y) = A_C(\frac{x+(q-1)y}{\sqrt{q}}, \frac{x-y}{\sqrt{q}}),$$

where $q = 2$.

3 Duursma Zeta Function

3.1 Definition

The following definition generalizes the idea of the weight enumerator polynomial of a code.

Definition 8 A homogeneous polynomial $F(x,y) = x^n + \sum_{i=1}^n f_i x^{n-i}y^i$ of degree n with complex coefficients is called a **virtual weight enumerator** (or VWE) with **support** $\mathrm{supp}(F) = \{i \mid f_i \neq 0\}$. If $F(x,y) = x^n + \sum_{i=d}^n f_i x^{n-i}y^i$ with $f_d \neq 0$ then call n the **length** of F and d the **minimum distance** of F. Such an F of even degree satisfying the invariance condition

$$F(x,y) = F(\frac{x+(q-1)y}{\sqrt{q}}, \frac{x-y}{\sqrt{q}}),$$

is called a **virtual self-dual weight enumerator** (or VSDWE for short) *over* $\mathbb{F}$ having **genus**.

$$\gamma(F) = n/2 + 1 - d.$$

If $b > 1$ is an integer and $\mathrm{supp}(F) \subset b\mathbb{Z}$ then the VWE F is called **b-divisible**.

An example of a virtual weight enumerator $F(x, y)$ is the Hamming weight enumerator of a code $A_C(x, y)$. In fact, in case $F(x, y) = A_C(x, y)$ the length of F is the length of the code C and the minimum distance of F is the minimum distance of the code C. An example of a virtual self-dual weight enumerator is the Hamming weight enumerator of a self-dual code.

It is amazing that the b-divisible virtual self-dual weight enumerators can be classified.

Theorem 9 *(Gleason-Pierce-Assmus-Mattson) Let F be a b-divisible VSDWE over $GF(q)$.*

Then either

I. $q = b = 2$,

II. $q = 2$, $b = 4$,

III. $q = b = 3$,

IV. $q = 4$, $b = 2$,

V. *q is arbitrary, $b = 2$, and $F(x, y) = (x^2 + (q-1)y^2)^{n/2}$.*

For Assmus and Mattson's proof of this theorem, please see Sloane [Sl].

Next, in order to carefully define the problem that this paper addresses, the notion of Types of weight enumerators are introduced. Theorem 9 motivates the following definition.

Definition 10 If F is a b-divisible VSDWE over $\mathbb{F}$ then F is called

$$\begin{cases} \textbf{Type I}, & \text{if } q = b = 2,\ 2|n, \\ \textbf{Type II}, & \text{if } q = 2,\ b = 4,\ 8|n, \\ \textbf{Type III}, & \text{if } q = b = 3,\ 4|n, \\ \textbf{Type IV}, & \text{if } q = 4,\ b = 2,\ 2|n. \end{cases}$$

The divisibility condition is extremely restricting and, for example, forces the length n to be even. The next sections concentrate on Duursma zeta functions for certain weight enumerators of Type IV.

Theorem 11 *(Sloane-Mallows-Duursma) If F is a b-divisible VSDWE with length n and minimum distance d then*

$$d \leq \begin{cases} 2[n/8]+2, & \text{if } F \text{ is Type I,} \\ 4[n/24]+4, & \text{if } F \text{ is Type II,} \\ 3[n/12]+3, & \text{if } F \text{ is Type III,} \\ 2[n/6]+2, & \text{if } F \text{ is Type IV.} \end{cases}$$

For a proof, see Duursma [D3].

An **extremal** b-divisible virtual self-dual weight enumerator is one for which equality holds in the above theorem. The next section focuses on the Type IV extremal case. With Theorems 9 and 11, the foundations of Duursma's paper [D3] extend from self-dual codes to virtual self-dual weight enumerators. This is because the coding-theoretic versions of the Theorem 9 and 11, used by Duursma, in fact hold for virtual self-dual weight enumerators.

Definition 12 *(Duursma [D1])* Assume F is a virtual weight enumerator polynomial of length n and minimum distance d. A polynomial $P(T)$ of degree $n+2-d-d^{\perp}$ for which

$$\frac{(xT+(1-T)y)^n}{(1-T)(1-qT)}P(T) = \cdots + \frac{F(x,y)-x^n}{q-1}T^{n-d} + \dots \ .$$

is called a **Duursma zeta polynomial of** F.

Proposition 13 *If $d \geq 2$ and $d^{\perp} \geq 2$ then there exists a unique Duurzma zeta polynomial of degree $\leq n-d$.*

proof: This is proven in the appendix to Chinen [C2]. Here is the rough idea. Expand $\frac{(xT+y(1-T))^n}{(1-T)(1-qT)}$ in powers of T to get

$$b_{0,0}y^nT^0 + (b_{1,0}xy^{n-1}+b_{1,1}y^n)T^1 + (b_{2,0}x^2y^{n-2}+b_{2,1}xy^{n-1}+b_{2,2}y^n)T^2 + ...$$
$$+(b_{n-d,0}x^{n-d}y^d + b_{n-d,1}x^{n-d-1}y^{d+1} + ... + b_{n-d,n-d}y^n)T^{n-d} + ... \ ,$$

where b_{ij} are coefficients which may depend on q. The Duursma polynomial is a polynomial of degree $n+2-d-d^{\perp}$. Provided $d^{\perp} \geq 2$, the Duursma polynomial can be written as $P(T) = a_0 + a_1 T + ... + a_{n-d} T^{n-d}$. Now, rewrite the terms of degree $\leq n$

$$\frac{(xT + y(1-T))^n}{(1-T)(1-qT)} P(T) = ... + \frac{F(x,y) - x^n}{q-1} T^{n-d} + ...$$

by means of the matrix equation $B \cdot \vec{a} = \vec{A}$ given by

$$\begin{pmatrix} b_{n-d.0} & b_{n-d.1} & \dots & b_{n-d.n-d} \\ 0 & b_{n-d-1.0} & \dots & b_{n-d-1.n-d-1} \\ 0 & 0 & b_{n-d-2.0} & \dots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & b_{0,0} \end{pmatrix} \begin{pmatrix} a_{n-d} \\ a_{n-d-1} \\ \vdots \\ a_0 \end{pmatrix} = \begin{pmatrix} A_n/(q-1) \\ A_{n-1}/(q-1) \\ \vdots \\ A_d/(q-1) \end{pmatrix}.$$

The diagonal entries of this matrix are binomial coefficients, hence are non-zero. Therefore the matrix is invertible and the existence is established.

To establish uniqueness, suppose that

$$\frac{(xT + y(1-T))^n}{(1-T)(1-qT)} P_1(T) = ... + \frac{F(x,y) - x^n}{q-1} T^{n-d} + ...$$

and

$$\frac{(xT + y(1-T))^n}{(1-T)(1-qT)} P_2(T) = ... + \frac{F(x,y) - x^n}{q-1} T^{n-d} + ...$$

hold. Subtracting these gives

$$\frac{(xT + y(1-T))^n}{(1-T)(1-qT)} (P_1(T) - P_2(T)) = 0.$$

This forces $P_1 = P_2$. $\square$

An example will be given in §4.

The **Duursma zeta function** of F is defined in terms of the zeta polynomial by means of $A_c(1+z, 1-z)$

$$Z(T) = \frac{P(T)}{(1-T)(1-qT)}. \tag{3}$$

In case of ambiguity denote this function by Z_F. Define the **Riemann hypothesis** to be the following statement: all (complex) zeros of $Z(T)$ satisfy

$|T| = 1/\sqrt{q}$. This is the analog for linear codes of the still unsolved conjecture regarding the Riemann zeta function.

The Duursma zeta function satisfies an analog of the functional equation for the Riemann zeta function. But before stating the functional equation, new notation is needed.

Define $F^\perp$ by $F^\perp = F \circ \sigma$, where

$$\sigma = \frac{1}{\sqrt{q}} \begin{pmatrix} 1 & q-1 \\ 1 & -1 \end{pmatrix}$$

Then there is a functional equation relating Z and $Z^\perp = Z_{F^\perp}$ (and hence also P and $P^\perp = P_{F^\perp}$). Note that even though F may not depend on q, $F^\perp$ (and hence $Z^\perp$) does.

Proposition 14 *The Duursma zeta function satisfies the functional equation*

$$Z^\perp(T)T^{1-g^\perp} = Z(\frac{1}{qT})(\frac{1}{qT})^{1-g}. \tag{4}$$

Analogously, the zeta polynomial $P = P_F$ satisfies the functional equation

$$P^\perp(T) = P(\frac{1}{qT})q^g T^{g+g^\perp}, \tag{5}$$

where $g = n/2 + 1 - d$ and $g^\perp = n/2 + 1 - d^\perp$.

This paper concerns the zeros of the zeta function in the case where F is an extremal virtual b-divisible self-dual weight enumerator of type IV.

3.2 Extremal Virtual Self-Dual Weight Enumerators

Following Duursma [D3], define the **ultraspherical polynomial** $C_n^m(x)$ on the interval $(-1, 1)$ by

$$C_n^m(\cos\theta) = \sum_{\substack{0 \le k, \ell \le n \\ k+\ell = n}} \binom{m+k}{k}\binom{m+\ell}{\ell} \cos((k-\ell)\theta).$$

The following theorem[1] is due to Duursma [D3], section 5.2.

[1]Be careful of serious typos in section 5.2 of Duursma, which are corrected below.

Theorem 15

$$Q(T^2/2) = \frac{m!^2}{(3m)!} T^m C_m^{m+1}(\frac{T+T^{-1}}{2})$$

Where $Q(T) = P(T)(1+2T)$ and P is the Duursma zeta polynomial of an extremal Type IV virtual self-dual weight enumerator of length $n = 3m+3$ and minimum distance $d = m+3$.

The main result is stated below.

Theorem 16 *The Duursma zeta function of an extremal self-dual weight enumerator of Type IV with length divisible by 3 satisfies the Riemann hypothesis.*

proof: It's a known fact [Sz] that all the roots of ultraspherical polynomials C_n^m lie on the interval $(-1, 1)$. This polynomial is degree n and so there are n such roots. In the theorem above, replacing T by $e^{i\theta}$ gives

$$Q(e^{2i\theta}/2) = \frac{m!^2}{(3m)!} e^{i\theta m} C_m^{m+1}(\cos\theta).$$

Therefore, all the roots of the degree m polynomial Q, hence the roots of P, lie on the circle of radius $1/\sqrt{q} = 1/2$. According to Duursma [D3], §4.4, all other Type IV extremal virtual self-dual weight enumerators have length of the form $3m+1$ or $3m+2$. This verifies the Riemann hypothesis in the case with length divisible by 3. □

4 Examples

The first example below computes a Duursma zeta function "by hand" in a simple case.

Example 17 Consider the binary self-dual code C of length $n = 6$, dimension $k = 3$, and minium distance $d = 2$. This is unique up to equivalence and has weight enumerator $W(x, y) = x^6 + 3x^4y^2 + 3x^2y^4 + y^6$. The SAGE commands

SAGE

```
sage: q = var("q")
sage: T = var("T")
```

```
sage: x = var("x")
sage: y = var("y")
sage: f1 = lambda q,T,N: sum([ sum([q^i for i in range(k+1)])*T^k for k in range(N)])
sage: f2 = lambda x,y,T,n: sum([ binomial(n,j)*(x-y)^j*y^(n-j)*T^j for j in range(n+1)])
sage: a0,a1,a2,a3,a4 = var("a0,a1,a2,a3,a4")
sage: F = expand(f1(2,T,6)*f2(x,y,T,6)*(a0+a1*T+a2*T^2+a3*T^3+a4*T^4))
```

compute the first 6 terms (as a power series in T) of the series $\frac{(xT+y(1-T))^n}{(1-T)(1-qT)}P(T)$ when $q=2$, $n=6$, $k=3$, and $d=2$. Next, SAGE computes the coefficients and read off the matrix B:

SAGE

```
sage: aa = (F.coeff("T^4")).coeffs("x")
sage: v = [expand(aa[i][0]/y^(6-i)) for i in range(5)]
sage: B0 = [v[0].coeff("a%s"%str(i)) for i in range(5)]
sage: B1 = [v[1].coeff("a%s"%str(i)) for i in range(5)]
sage: B2 = [v[2].coeff("a%s"%str(i)) for i in range(5)]
sage: B3 = [v[3].coeff("a%s"%str(i)) for i in range(5)]
sage: B4 = [v[4].coeff("a%s"%str(i)) for i in range(5)]
sage: B0.reverse(); B1.reverse(); B2.reverse(); B3.reverse(); B4.reverse()
sage: B = matrix([B0,B1,B2,B3,B4])
sage: B

[  1  -3   4  -2   1]
[  0   6 -12  12   0]
[  0   0  15 -15  15]
[  0   0   0  20   0]
[  0   0   0   0  15]
```

Note that the diagonal entries are binomial coefficients.

Finally, the vector $\vec{A}$ is determined by solving the equation $B \cdot \vec{a} = \vec{A}$:

SAGE

```
sage: Wmx6 = 3*x^4*y^2+3*x^2*y^4+y^6
sage: c = [Wmx6(1,y).coeff("y^%s"%str(i)) for i in range(2,7)]
sage: c.reverse()
sage: cc = vector(c)
sage: (B^(-1)*cc).list()
[4/5, 0, 0, 0, 1/5]
```

This implies that the zeta polynomial of C is given by $P(T) = \frac{1}{5} + \frac{4}{5}T^4$.

The next example illustrates the computation of the Duursma zeta function for a quaternary code.

Example 18 *The hexacode is an MDS code. In general, it is true that the Duursma zeta function of any MDS code is* $P(T) = 1$.

Here is a more interesting example. Let z *denote the same element as was defined in Example 3. Let*

$$G = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & z+1 & 1 & 1 & z & 1 & 1 & z+1 & z \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & z+1 & z+1 & 0 & z & 0 & 1 & z & z+1 & z+1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & z+1 & 1 & 0 & z+1 & z+1 & z+1 & z & 0 & z \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & z+1 & 1 & 0 & z+1 & z+1 & z+1 & z & z \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & z & 1 & 1 & z+1 & z+1 & 1 & 1 & z & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & z & z+1 & z+1 & z+1 & 0 & 1 & z+1 & 0 & z \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & z & z+1 & z+1 & z+1 & 0 & 1 & z+1 & z \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & z+1 & z & 1 & 0 & z & 0 & z+1 & z+1 & z+1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & z+1 & 1 & 1 & z & 1 & 1 & z+1 & 1 & z
\end{pmatrix}$$

be a generator matrix of a code C. *This is an extremal Type IV code over a field with four elements. According to* SAGE, *the zeta polynomial for this code is* $P(T) = \frac{48}{143}T^4 + \frac{48}{143}T^3 + \frac{32}{143}T^2 + \frac{12}{143}T + \frac{3}{143}$. *It can be checked directly, using* SAGE, *that this satisfies the Riemann hypothesis.*

SAGE

```
sage: F.<z> = GF(4,"z")
sage: MS = MatrixSpace(F, 9, 18)
sage: G = MS([
....: [1, 0, 0, 0, 0, 0, 0, 0, 0, 1, z^2, 1, 1, z, 1, 1, z^2, z],\
....: [0, 1, 0, 0, 0, 0, 0, 0, 0, z^2, z^2, 0, z, 0, 1, z, z^2, z^2],\
....: [0, 0, 1, 0, 0, 0, 0, 0, 0, z^2, 1, 0, z^2, z^2, z^2, z, 0, z],\
....: [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, z^2, 1, 0, z^2, z^2, z^2, z, z],\
....: [0, 0, 0, 0, 1, 0, 0, 0, 0, z, 1, 1, z^2, z^2, 1, 1, z, 1],\
....: [0, 0, 0, 0, 0, 1, 0, 0, 0, z, z^2, z^2, z^2, 0, 1, z^2, 0, z],\
....: [0, 0, 0, 0, 0, 0, 1, 0, 0, z, z^2, z^2, z^2, 0, 1, z^2, z],\
....: [0, 0, 0, 0, 0, 0, 0, 1, 0, z^2, z, 1, 0, z, 0, z^2, z^2, z^2],\
....: [0, 0, 0, 0, 0, 0, 0, 0, 1, z^2, 1, 1, z, 1, 1, z^2, 1, z]])
sage: C = LinearCode(G)
sage: print C.spectrum()
[1, 0, 0, 0, 0, 0, 0, 0, 2754, 0, 18360, 0, 77112, 0, 110160, 0, 50949, 0, 2808]
sage: R.<T> = PolynomialRing(CC,"T")
sage: P = C.sd_zeta_polynomial(4)
sage: P
48/143*T^4 + 48/143*T^3 + 32/143*T^2 + 12/143*T + 3/143
sage: rts = R(P).roots()
sage: [abs(r[0]) for r in rts]
[0.500000000000000, 0.500000000000000, 0.500000000000000, 0.500000000000000]
```

Background Information: SAGE is a computer algebra program whose open source kernel is written in the Python programming language.

Acknowledgements: I thank the readers of this honors project for their helpful suggestion that improved this presentation. The SAGE examples are

due to my advisor. I also thank Prof. Philippe Gaborit for the generator matrix of the last example and Prof. Thann Ward for the reference to Sloane [Sl].

References

[C1] K. Chinen, *Zeta functions for formal weight enumerators and the extremal property*, Proc. Japan Acad. Ser. A Math. Sci. vol. 81, Number 10 (2005), 168-173.

[C2] ——, *Zeta functions for formal weight enumerators and an analogue of the Mallows-Sloane bound*, `http://arxiv.org/pdf/math/0510182`, `http://front.math.ucdavis.edu/math.NT/0510182`

[C3] ——, "An abundance of invariant polynomials satisfying the Riemann hypothesis," `http://arxiv.org/abs/0704.3903`

[D1] I. Duursma, *Combinatorics of the two-variable zeta function*, in **Finite fields and applications**, 109–136, Lecture Notes in Comput. Sci., 2948, Springer, Berlin, 2004.

[D2] ——, *Results on zeta functions for codes*, Fifth Conference on Algebraic Geometry, Number Theory, Coding Theory and Cryptography, University of Tokyo, January 17-19, 2003.

[D3] ——, *Extremal weight enumerators and ultraspherical polynomials*, Discrete Mathematics, vol. 268, no. 1-3, pp. 103-127, July 2003.

[D4] ——, *A Riemann hypothesis analogue for self-dual codes*, In: **Codes and Association schemes**, Eds. Barg and Litsyn, AMS Dimacs Series, vol. 56, pp. 115-124, 2001.

[D5] ——, *From weight enumerators to zeta functions*, in **Discrete Applied Mathematics**, vol. 111, no. 1-2, pp. 55-73, 2001.

[D6] ——, *Weight distributions of geometric Goppa codes*, Transactions of the AMS, vol. 351, pp. 3609-3639, September 1999.

[HILL] R. Hill **A First Course In Coding Theory**, Oxford University Press. (1986).

[HP] W. C. Huffman and V. Pless, **Fundamentals of error-correcting codes**, Cambridge Univ. Press, 2003.

[MS] F. J. MacWilliams and N. J. A. Sloane. **The Theory of Error-correcting Codes**. North-Holland. (1983)

[S] The SAGE Group, SAGE : *Mathematical software*, version 2.11
`http://www.opensourcemath.org/sage/`
`http://www.sagemath.org/`

[Sl] N. J. A. Sloane, *Self-dual codes and lattices*, in **Relations Between Combinatorics and Other Parts of Mathematics**., Proc. Symp. Pure Math., Vol. 34, American Mathematical Society, Providence, RI, 1979, pp. 273-308.

[Sz] G. Szegö, **Orthogonal Polynomials** Vol XXIII, American Mathematical Society, 4th Edition, Colloqium Publications, Providence, RI, 1975.