

Duursma zeta functions - a survey (11th draft)

David Joyner*

5-24-2008

Abstract

This is a purely expository survey paper on the Duursma zeta function of a linear block code. SAGE code is used to compute examples.

Contents

1	Introduction	2
1.1	Divisible codes	4
1.2	Some invariants	7
1.3	Virtual weight enumerators	10
2	The zeta polynomial	13
2.1	First definition	14
2.2	Second definition	20
2.3	Third definition	21
2.4	Analogies with curves	23
3	Properties	25
3.1	The functional equation	25
3.2	Puncturing preserves P	27
3.3	The RH	28

*wdjoyner@gmail.com; Math Dept, USNA, Annapolis, MD. These notes are licensed under the Attribution-ShareAlike Creative Commons license, <http://creativecommons.org/about/licenses/meet-the-licenses>.

4	Examples	31
4.1	Komichi’s example	31
4.2	The extremal case	32
4.3	“Random divisible codes”	34
4.4	A fsd $[26, 13, 6]_2$ -code	35
4.5	Extremal codes of short length	36
4.6	Non-self-dual examples	36
5	Chinen zeta functions	37
5.1	Hamming codes	41
5.2	Golay codes	43
5.3	Examples	43
6	Appendix: Proofs	47
6.1	MacWilliam’s identity	47
6.2	Mallows-Sloane-Duursma bounds	50

Let C be an $[n, k, d]_q$ code, ie a linear code over $GF(q)$ of length n , dimension k , and minimum distance d . Motivated by analogies with local class field theory, in [D1] Iwan Duursma introduced the *zeta function* $Z = Z_C$ associated to a linear code C over a finite field,

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

where $P(T)$ is a polynomial of degree $n + 2 - d - d^\perp$, called the *zeta polynomial*¹.

This paper will explore some of the properties of the zeta function and give examples using the software package **SAGE** [S].

1 Introduction

A linear code C is called a $[n, k, d]_q$ -code if it is a k -dimensional subspace of $GF(q)^n$ having minimum distance d ,

¹In general, if C is an $[n, k, d]$ -code then we use $[n, k^\perp, d^\perp]$ for the parameters of the dual code, C^\perp . It is a consequence of Singleton’s bound that $n + 2 - d - d^\perp \geq$, with equality when C is an MDS code.

$$d = \min_{c \in C, c \neq 0} \text{wt}(c),$$

where wt is the Hamming weight of a codeword. The dual code of C , denoted C^\perp , has parameters $[n, n - k, d^\perp]$, for some $d^\perp \geq 1$. If $C = C^\perp$ then the code is called *self-dual*. The *genus* of an $[n, k, d]_q$ -code C is defined by

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

This measures how “far away the code is from being MDS”. If C is an AG code constructed from the Riemann-Roch space of an algebraic curve over $GF(q)$ then it often is equal to the genus of the curve (see [TV] for details). Note that if C is a self-dual code then its genus satisfies $\gamma = n/2 + 1 - d$.

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

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

where

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

denotes the number of codewords of weight i . The *support* of C is the set $\text{supp}(C) = \{i \mid A_i \neq 0\}$. 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 :

$$\text{spec}(C) = [A_0, \dots, A_n].$$

We say two codes are *formally equivalent* if they have the same spectrum.

Example 1 *Here is a code C which is formally self-dual. Let*

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

and let G be the binary code generated by G . This code has spectrum $[1, 0, 0, 0, 15, 0, 15, 0, 0, 0, 1]$ and satisfies the “Riemann hypothesis” (see Definition 37 below for this term). Here is the SAGE code verifying this.

```

sage: MS = MatrixSpace(GF(2),5,10)
sage: G = MS([[1,1,1,1,0,0,0,0,0,0],[0,0,0,0,1,1,1,1,1,1],[1,0,0,0,1,1,1,1,0,0],\
...: [0,1,0,0,1,1,0,1,0,0],[0,0,1,0,1,0,1,0,1,0]])
sage: C = LinearCode(G)
sage: C
Linear code of length 10, dimension 5 over Finite Field of size 2
sage: C.spectrum()
[1, 0, 0, 0, 15, 0, 15, 0, 0, 0, 1]
sage: P = C.zeta_polynomial()
sage: P
2/7*T^4 + 2/7*T^3 + 3/14*T^2 + 1/7*T + 1/14
sage: RT = PolynomialRing(CC,"T")
sage: rts = RT(P).roots()
sage: [z[0].abs() for z in rts]
[0.707106781186548, 0.707106781186546, 0.707106781186548, 0.707106781186548]
sage: Cd = C.dual_code()
sage: Cd.spectrum()
[1, 0, 0, 0, 15, 0, 15, 0, 0, 0, 1]

```

Saying two codes are formally equivalent is stronger than saying that the codes are *isometric*, i.e., that there is a bijective linear transformation between them that preserves the weight function. In fact, it is known that two codes are permutation equivalent if and only if they are isometric (by a result of MacWilliams).

Open Question 1 *Given a homogeneous polynomial $F(x, y) = x^n + \sum_{i=1}^n f_i x^{n-i} y^i$ of degree n with non-negative integer coefficients, find necessary and sufficient conditions (short of enumerating all weight enumerators of linear codes with length n) which determine whether or not $F(x, y) = A_C(x, y)$ for some linear code C of length n .*

1.1 Divisible codes

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

Definition 2 Let C be a b -divisible code. If C and C^\perp both binary and contain the all-ones codeword then C is said to be *Type 2 divisible*. We say C is *Type 1 divisible* if C is not of Type 2.

Lemma 3 *If C is Type 1 divisible then*

$$d + bd^\perp \leq n + b(b + 1).$$

If C is Type 2 divisible then

$$2d + bd^\perp \leq n + b(b + 2).$$

proof: See Theorem 1 in Duursma [D3]. \square

The Gleason-Pierce Theorem² basically says that, other than a family of uninteresting examples, the formally self-dual divisible codes fall into one of the following four types.

Definition 4 Let C be a fsd b -divisible $[n, k, d]_q$ -code. We say C is *Type I* if $q = b = 2$, and n is even. We say C is *Type II* if $q = 2$, $b = 4$, and $8|n$. We say C is *Type III* if $q = b = 3$, and $4|n$. If $q = 4$, $b = 2$, and n is even then C is said to be *Type IV*.

For example, if C is a binary self-dual code, then it must be 2-divisible (since each codeword must be orthogonal to itself, hence have even weight). This implies that C^\perp contains the all-ones vector. But $C = C^\perp$, so C must be Type 2.

Lemma 5 (*upper bounds*) If C is sd then

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

proof: This is Theorem 9.3.1 in [HP]. \square

These upper bounds are sometimes referred to as the *Mallows-Sloane bounds*. In fact, the “Type I bound” even holds for formally self-dual codes (see Theorem 9.3.1 in [HP], §11.1 in [NRS]).

A code is called *optimal* if its minimum distance is maximal among all linear codes of that length and dimension. A code C is called *extremal* if the bound in Lemma 5 holds with equality.

²See Theorem 2.5.1 in [NRS], also Theorem 13 below.

Remark 1 (1) It is known that any two extremal codes (if they exist) have the same weight enumerator polynomial (in fact, they are essentially determined in Duursma [D3]).

(2) It is known that there exist only finitely many extremal codes (see §11.1 in [NRS] or Huffman and Pless [HP], p 345).

If C^\perp denotes the dual code of C , with parameters $[n, n - k, d^\perp]$, then the MacWilliams identity³ relates the weight enumerator of C^\perp to that of C :

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

In particular, C is formally self-dual if and only if $F = A_C$ satisfies the invariance condition

$$F(x, y) = F\left(\frac{x + (q - 1)y}{\sqrt{q}}, \frac{x - y}{\sqrt{q}}\right). \quad (2)$$

Example 6 The following examples are taken from Sloane [Sl]. The notation is as in [Sl] and will be used in the statement of Theorem 7 below.

1. $W_1(x, y) = x^2 + y^2$ is the weight enumerator of the Type I code $C = \{(0, 0), (1, 1)\}$.
2. $W_5(x, y) = x^8 + 14x^4y^4 + y^8$ is the weight enumerator of the Type II $[8, 4, 4]$ code C constructed by extending the binary $[7, 4, 3]$ Hamming code by a check bit. This is the smallest Type II code.
3. $W_6(x, y) = x^{24} + 759x^{16}y^8 + 2576x^{12}y^{12} + 759x^8y^{16} + y^{24}$ is the weight enumerator of the binary Golay code with parameters $[23, 12, 8]$.
4. $W_8(x, y) = x^{48} + 17296(x^{36}y^{12} + x^{12}y^{36}) + 535095(x^{32}y^{16} + x^{16}y^{32}) + 3995376(x^{28}y^{20} + x^{20}y^{28}) + 7681680x^{24}y^{24}$ is the weight enumerator of the extended binary quadratic residue code of associated to $p = 47$ with parameters $[48, 24, 16]$.
5. $W_9(x, y) = x^4 + 8xy^3$ is the weight enumerator of the Type III ternary code C with generator matrix

³This is proven in the appendix below.

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

and parameters $[4, 2, 3]$.

6. $W_{10}(x, y) = x^{12} + 264x^6y^6 + 440x^3y^9 + 24y^{12}$ is the weight enumerator of the Type III ternary Golay code with parameters $[12, 6, 6]$.
7. $W_{11}(x, y) = x^2 + 3y^2$ is the weight enumerator of the Type IV code $C = \{(0, 0), (1, 1), (\alpha, \alpha), (\alpha^2, \alpha^2)\}$, with parameters $[2, 1, 2]$. Here α is a generator of $GF(4)$, $\alpha^2 + \alpha + 1 = 0$.
8. $W_{12}(x, y) = x^6 + 45x^2y^4 + 18y^6$ is the weight enumerator of the Type IV code C with generator matrix

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

with parameters $[6, 3, 4]$. Here α is a generator of $GF(4)$, $\alpha^2 + \alpha + 1 = 0$.

1.2 Some invariants

The following result collects together several facts from §8.1 in Sloane [Sl].

Theorem 7 Assume C is a formally self-dual divisible code of Type I, II, III, or IV.

- I. If C is Type I then $A_C(x, y)$ is invariant under the group

$$G_I = \left\langle \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \right\rangle$$

of order 16. Moreover, $\mathbb{C}[x, y]^{G_I} = \mathbb{C}[W_1, W_5]$.

- II. If C is Type II then $A_C(x, y)$ is invariant under the group

$$G_{II} = \left\langle \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \right\rangle$$

of order 192. Moreover, $\mathbb{C}[x, y]^{G_{II}} = \mathbb{C}[W_5, W_6]$.

- III. If C is Type III then $A_C(x, y)$ is invariant under the group

$$G_{III} = \langle \sigma, \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix} \rangle, \quad \sigma = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 \\ 2 & -1 \end{pmatrix},$$

of order 48, where $\omega \in GF(9) - \{1\}$, $\omega^3 = 1$. Moreover, $\mathbb{C}[x, y]^{G_{III}} = \mathbb{C}[W_9, W_{10}]$.

- IV. If C is Type IV then $A_C(x, y)$ is invariant under the group

$$G_{IV} = \langle \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 3 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \rangle$$

of order 12. Moreover, $\mathbb{C}[x, y]^{G_{IV}} = \mathbb{C}[W_{11}, W_{12}]$.

Here are some computations illustrating the above theorem.

Example 8 Here is some SAGE code for computing the invariants of the group G generated by $g_1 = \begin{pmatrix} 1/\sqrt{q} & 1/\sqrt{q} \\ (q-1)/\sqrt{q} & -1/\sqrt{q} \end{pmatrix}$ with $q = 2$, $g_2 = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$, and $g_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$. The SAGE method `invariant_generators` calls Singular [GPS], which has methods implements (by Simon King) to compute group invariants.

```

SAGE
sage: F = CyclotomicField(8)
sage: z = F.gen()
sage: a = z+1/z
sage: a^2
2
sage: MS = MatrixSpace(F, 2, 2)
sage: b = -1
sage: g1 = MS([[1/a, 1/a], [1/a, -1/a]])
sage: g2 = MS([[1, 0], [0, b]])
sage: g3 = MS([[b, 0], [0, 1]])
sage: G = MatrixGroup([g1, g2, g3])
sage: G.invariant_generators()
[x1^2 + x2^2, x1^8 + 28/9*x1^6*x2^2 + 70/9*x1^4*x2^4 + 28/9*x1^2*x2^6 + x2^8]
```

It is not hard to check that this is equivalent with part I of Theorem 7.

Example 9 Here is some SAGE code for computing the invariants of the group G generated by $g_1 = \begin{pmatrix} 1/\sqrt{q} & (q-1)/\sqrt{q} \\ 1/\sqrt{q} & -1/\sqrt{q} \end{pmatrix}$ with $q = 2$, $g_2 = \begin{pmatrix} i & 0 \\ 0 & 1 \end{pmatrix}$, and $g_3 = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$.

SAGE

```
sage: F = CyclotomicField(8)
sage: z = F.gen()
sage: a = z+1/z
sage: b = z^2
sage: MS = MatrixSpace(F,2,2)
sage: g1 = MS([[1/a,1/a],[1/a,-1/a]])
sage: g2 = MS([[1,0],[0,b]])
sage: g3 = MS([[b,0],[0,1]])
sage: G = MatrixGroup([g1,g2,g3])
sage: G.order()
192
sage: G.invariant_generators()
[x1^8 + 14*x1^4*x2^4 + x2^8,
 x1^24 + 10626/1025*x1^20*x2^4 + 735471/1025*x1^16*x2^8\
 + 2704156/1025*x1^12*x2^12 + 735471/1025*x1^8*x2^16\
 + 10626/1025*x1^4*x2^20 + x2^24]
```

The above group G which leaves invariant the weight enumerator of any self-dual doubly even binary code. The above result implies that any such weight enumerator must be a polynomial in $x^8 + 14x^4y^4 + y^8$ and $1025x^{24} + 10626x^{20}y^4 + 735471x^{16}y^8 + 2704156x^{12}y^{12} + 735471x^8y^{16} + 10626x^4y^{20} + 1025y^{24}$. Using SAGE's Gröbner bases algorithms, it is not hard to check that this is equivalent with part II of Theorem 7. The details are omitted.

Example 10 Here is some SAGE code for computing the invariants of the group G generated by $g_1 = \begin{pmatrix} 1/\sqrt{q} & 1/\sqrt{q} \\ (q-1)/\sqrt{q} & -1/\sqrt{q} \end{pmatrix}$ with $q = 3$, $g_2 = \begin{pmatrix} \omega & 0 \\ 0 & 1 \end{pmatrix}$, and $g_3 = \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix}$.

SAGE

```
sage: F = CyclotomicField(12)
sage: z = F.gen()
sage: a = z+1/z
sage: b = z^4
sage: a^2; b^3
3
```

```

1
sage: MS = MatrixSpace(F,2,2)
sage: g1 = MS([[1/a,1/a],[2/a,-1/a]])
sage: g2 = MS([[1,0],[0,b]])
sage: g3 = MS([[b,0],[0,1]])
sage: G = MatrixGroup([g1,g2,g3])
sage: G.order()
144
sage: G.invariant_generators()

[x1^12 + (-55/2)*x1^9*x2^3 + 231/16*x1^6*x2^6 + (-55/128)*x1^3*x2^9 + 61/1024*x2^12,
 x1^12 + 4*x1^9*x2^3 + 21/8*x1^6*x2^6 + 67/64*x1^3*x2^9 + (-1/512)*x2^12]

```

Example 11 Here is some SAGE code for computing the invariants of the group G generated by $g_1 = \begin{pmatrix} 1/\sqrt{q} & 1/\sqrt{q} \\ (q-1)/\sqrt{q} & -1/\sqrt{q} \end{pmatrix}$ with $q = 4$, $g_2 = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$, and $g_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$.

SAGE

```

sage: q = 4; a = 2
sage: MS = MatrixSpace(QQ, 2, 2)
sage: g1 = MS([[1/a,1/a],[(q-1)/a, -1/a]])
sage: g2 = MS([[1,0],[0,-1]])
sage: g3 = MS([[-1,0],[0,1]])
sage: G = MatrixGroup([g1,g2,g3])
sage: G.order()
12
sage: G.invariant_generators()
[x1^2 + 1/3*x2^2, x1^6 + 5/3*x1^4*x2^2 + 5/27*x1^2*x2^4 + 11/243*x2^6]

```

For the reader interested in more examples along these lines, we refer to Harada and Tagami [HT]. (We shall discuss this paper more below.)

1.3 Virtual weight enumerators

Definition 12 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* $\text{supp}(F) = \{i \mid f_i \neq 0\}$. If $F(x, y) = x^n + \sum_{i=d}^n A_i x^{n-i} y^i$ with $A_d \neq 0$ then we call n the *length* of F and d the *minimum distance* of F . Such an F of even degree satisfying (2), is called a *virtually self-dual weight enumerator* (or VSDWE for short) *over* $GF(q)$ having *genus*

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

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

The classification of non-trivial formally self-dual divisible codes into the four Types has a VSDWE analog. In other words, the Gleason-Pierce theorem has a strengthening where the hypothesis does not require the existence of a code, only a form which certain invariance properties.

Theorem 13 (*Gleason-Pierce-Assmus-Mattson*) *Let F be a b-divisible VS-DWE over $G(q)$.*

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

proof: The proof (or proofs - there are now two of them) is due to Assmus and Mattson. The easiest place to access the argument is in the survey paper Sloane [Sl]. The rough idea is as follows (for details, please see Sloane's paper).

Let G denote the subgroup of $GL(2, \mathbb{C})$ generated by the matrix of the "MacWilliams transform"

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

together with the diagonal matrices having b -th roots of unity on the diagonal (since $F(x, y) \mapsto F(\zeta x, y)$ and $F(x, y) \mapsto F(x, \zeta y)$ both fix F , if $\zeta \in F$ is any b -th roots of unity). Let G' denote its image in $PGL(2, \mathbb{C})$. Think of $F(x, y)$ as a function $f(z)$ of $z = x/y$ on \mathbb{P}^1 . Let m denotes the number of zeros of f (not counting multiplicity). By the invariance properties, $m = 1$ is impossible. If $m = 2$ then the invariance properties implies (V). If $m > 3$ then G' must be finite. The classification of finite subgroups of $PGL(2, \mathbb{C})$ results in the remaining possibilities (I), ..., (IV). \square

Next we give the virtual weight enumerator analog of Definition 4 above.

Definition 14 • Let $F(x, y)$ be a VSDWE. If $b > 1$ is an integer and $\text{supp}(F) \subset b\mathbb{Z}$ then F is called *b-divisible*.

- If F is a *b-divisible* VSDWE over $GF(q)$ then F is called

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

Theorem 15 (*Sloane-Mallows-Duursma*) If F is a *b-divisible* VSDWE with length n and minimum distance d then

$$d \leq \begin{cases} c \lfloor \frac{n}{c(c+1)} \rfloor + c, & \text{if } F \text{ is Type 1,} \\ c \lfloor \frac{n}{c(c+2)} \rfloor + c, & \text{if } F \text{ is Type 2.} \end{cases} \quad (3)$$

In particular,

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

proof: This is only stated for self-dual codes, but proof of Theorem 1 and the argument in §1.1 of Duursma [D3] hold more generally for VSDWEs. A complete proof is given in the appendix below. \square

A VSDWE F is called *extremal* if the bound in Theorem 15 holds with equality.

Remark 2 • Here is a more general definition. Let G be a subgroup of $GL(2, \mathbb{C})$ containing $\sigma = \frac{1}{\sqrt{q}} \begin{pmatrix} 1 & q-1 \\ 1 & -1 \end{pmatrix}$, acting on $\mathbb{C}[x, y]$ by $\sigma : F(x, y) \mapsto F(\sigma(x, y)^t)$, and $\chi : G \rightarrow \mathbb{C}^\times$ a character. Call a virtual weight enumerator F of length n a *formally χ -self-dual weight enumerator*, or a *VSDWE twisted by χ* , if⁴

⁴This “twisted” terminology is motivated by terminology in automorphic forms and arithmetical algebraic geometry for analogous objects.

$$F(x, y) = \chi(\sigma)F\left(\frac{x + (q-1)y}{\sqrt{q}}, \frac{x-y}{\sqrt{q}}\right).$$

The VSDWE definition above is the special case when χ is a trivial. This “twisted” definition also covers, for example, the case of Ozeki’s “formal weight enumerators” in [O]. For brevity, we call F a *twisted VSDWE* if it satisfies

$$F(x, y) = -F\left(\frac{x + (q-1)y}{\sqrt{q}}, \frac{x-y}{\sqrt{q}}\right). \quad (4)$$

Much of the theory of zeta functions for VSDWE’s also applies to twisted VSDWE’s. See Chinen [C1], [C2] and §5 below.

- Note that a virtual weight enumerator does not depend on a prime power q but a VSDWE does depend on q through (2).

Definition 16 *A virtual weight enumerator F is formally identified with an object we call a virtual code C subject only to the following condition: we formally extend the definition of $C \mapsto A_C$ to all virtual codes by $A_C = F$. Of course, if F is the weight enumerator of an actual code, C' say, then we have $A_C = F = A_{C'}$. In other words, a virtual code is only well-defined up to formal equivalence. If C_1 and C_2 are virtual codes then we define $C_1 + C_2$ to be the virtual code associated to the VWE $A_{C_1}(x, y) + A_{C_2}(x, y)$.*

Open Question 2 *Given a VSDWE, find necessary and sufficient conditions (short of enumeration) which determine whether or not it arises as the weight enumerator of some self-dual code C .*

2 The zeta polynomial

We shall give three definitions of the zeta polynomial, all due to Duursma.

2.1 First definition

Definition 17 A polynomial $P(T)$ for which

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

is called a *Duursma zeta polynomial of C* .

The *Duursma zeta function* is defined in terms of the zeta polynomial by means of (1) above.

Lemma 18 *The Duursma zeta polynomial $P = P_C$ exists and is unique, provided $d^\perp \geq 2$.*

proof: This is proven in the appendix to Chinen [C2]. Here is the rough idea. If we expand $\frac{(xT + y(1 - T))^n}{(1 - T)(1 - qT)}$ in powers of T , we find it is equal to

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

The Duursma polynomial is a polynomial of degree $n + 2 - d - d^\perp$. Provided $d^\perp \geq 2$, we can write the Duursma polynomial as $P(T) = a_0 + a_1T + \dots + a_{n-d}T^{n-d}$ and rewrite

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

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. \square

Example 19 Consider the self-dual code C of length $n = 6$, dimension $k = 3$, and minimum 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,T,x,y = var("q,T,x,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, we compute the coefficients and read off the matrix B :

```

SAGE
sage: aa = (F.coeff("T^4")).coffs("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, we compute the vector \vec{A} , and solve 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 function of C is given by $P(T) = \frac{1}{5} + 45T^4$.

Duursma has given several definitions (all equivalent of course) of $P(T)$. Before stating another one, we need the following definition and lemma.

Definition 20 Define c_j by

$$\frac{(xT + (1 - T)y)^n}{(1 - T)(1 - qT)} = \sum_{k=0}^{\infty} c_k(x, y)T^k.$$

Define $M_{n,\delta}$ by

$$M_{n,\delta}(x, y) = x^n + (q - 1)c_{n-\delta}(x, y).$$

This is called the *MDS virtual weight enumerator of length n and distance δ* .

It is not hard to see that

$$\frac{1}{(1 - T)(1 - qT)} = \sum_{j=0}^{\infty} \frac{q^{j+1} - 1}{q - 1} T^j,$$

and of course

$$(xT + (1 - T)y)^n = \sum_{i=0}^n \binom{n}{i} y^{n-i} (x - y)^i T^i.$$

Therefore,

$$c_k(x, y) = \sum_{i+j=k} \frac{q^{j+1} - 1}{q - 1} \binom{n}{i} y^{n-i} (x - y)^i.$$

Example 21 We use SAGE [S] to compute examples.

When $q = 2$,

$$M_{10,5}(x, y) = -34y^{10} + 220xy^9 - 585x^2y^8 + 840x^3y^7 - 630x^4y^6 + 252x^5y^5 + x^{10}$$

and when $q = 3$,

$$M_{12,5}(x, y) = -48y^{12} + 1152xy^{11} - 2376x^2y^{10} + 8360x^3y^9 - 7920x^4y^8 + 9504x^5y^7 - 3696x^6y^6 + 1584x^7y^5 + x^{12}.$$

The negative coefficients in these polynomials are consistent with the fact that for codes of dimension greater than 1, the length of an MDS code satisfies the bound $n \leq q + k - 1$ (see for example, pages 12-13 in [TV]). In the first example, a $[10, 6, 5]_2$ code must satisfy $10 \leq 2 + 6 - 1$ (so it doesn't exist) and, in the second example, a $[12, 8, 5]_3$ code must satisfy $12 \leq 3 + 8 - 1$ (so it doesn't exist).

On the other hand, when $q = 13$,

$$M_{12,5}(x, y) = 312177312y^{12} + 312178752xy^{11} + 143076384x^2y^{10} + 39755760x^3y^9 + 7436880x^4y^8 + 1007424x^5y^7 + 88704x^6y^6 + 9504x^7y^5 + x^{12}.$$

Indeed, according to SAGE's `ReedSolomonCode` command, there is an MDS code C having parameters $[12, 8, 5]_{13}$:

```

SAGE
sage: C = ReedSolomonCode(12,8,GF(13))
sage: C.spectrum()

[1,
 0,
 0,
 0,
 0,
 9504,
 88704,
 1007424,
 7436880,
 39755760,
 143076384,
 312178752,
 312177312]
```

This SAGE session tells us that

$$\text{spec}(C) = [1, 0, 0, 0, 0, 9504, 88704, 1007424, 7436880, 39755760, 143076384, 312178752, 312177312],$$

as the above (independently obtained) computation implies.

These virtual weight enumerators are computed using the following SAGE code

```

sage: R = PolynomialRing(QQ, 2, "xy")
sage: x, y = R.gens()
sage: f = lambda q, n, m : \
    (x*T+y*(1-T))^(n)*sum([T^i for i in range(m)]) \
    *sum([(q*T)^i for i in range(m)])
sage: M = lambda q, n, d, m : (f(q, n, m).list())[d]*(q-1)+x^n

```

As long as m is taken to be sufficiently large, this code will return the correct value of $M_{n,d}$.

A version of the following result is stated in Duursma's [D5] (see his equation (9)).

Lemma 22 *If F is a virtual weight enumerator of length n and minimum distance d then there are coefficients $c \in \mathbb{Q}$ and $a_i = a_j(F) \in \mathbb{Q}$ such that*

$$F(x, y) = cx^n + a_0M_{n,d}(x, y) + a_1M_{n,d+1}(x, y) + \cdots + a_rM_{n,d+r}(x, y), \quad (5)$$

for some r , $0 \leq r \leq n - d$. In fact, $c = 1 - a_0 - \cdots - a_r$.

proof: The functions $M_{n,d+i}(x, y) - x^n$ form a basis for the vector space $V = \{\sum_{i=d}^n b_i x^{n-i} y^i \mid b_i \in \mathbb{Q}\}$.

Consider the equation

$$F(x, y) - x^n = a_0(M_{n,d}(x, y) - x^n) + a_1(M_{n,d+1}(x, y) - x^n) + \cdots + a_r(M_{n,d+r}(x, y) - x^n).$$

If $r = \dim(V) - 1$ then one can solve for the a_0, \dots, a_r . Without loss of generality, we may take $r \geq 0$ to be as small as possible. We have then

$$F(x, y) = (1 - a_0 - \cdots - a_r)x^n + a_0M_{n,d}(x, y) + a_1M_{n,d+1}(x, y) + \cdots + a_rM_{n,d+r}(x, y).$$

□

Example 23 Duursma zeta function of the $[2^r - 1, 2^r - r - 1, 3]$ -Hamming code, $Ham(r, GF(2))$, can be computed using the following SAGE commands:

```

SAGE

sage: C = HammingCode(3,GF(2))
sage: C.zeta_function()
(2/5*T^2 + 2/5*T + 1/5)/(2*T^2 - 3*T + 1)
sage: C = HammingCode(4,GF(2))
sage: C.zeta_function()
(16/429*T^6 + 16/143*T^5 + 80/429*T^4 + 32/143*T^3 +
30/143*T^2 + 2/13*T + 1/13)/(2*T^2 - 3*T + 1)

```

In other words,

$$Z_{Ham(3,GF(2))}(T) = \frac{\frac{1}{5}(2T^2 + 2T + 1)}{2T^2 - 3T + 1},$$

and

$$Z_{Ham(4,GF(2))}(T) = \frac{\frac{1}{429}(16T^6 + 48T^5 + 80T^4 + 96T^3 + 90T^2 + 66T + 33)}{2T^2 - 3T + 1}.$$

Example 24 Duursma zeta function of the maximal binary linear self-dual doubly even code of length 8 can be computed using the following different SAGE commands:

```

SAGE

sage: MS = MatrixSpace(GF(2),4,8)
sage: G = MS([[1,1,1,1,0,0,0,0],[0,0,1,1,1,1,0,0],[0,0,0,0,1,1,1,1],[1,0,1,0,1,0,1,0]])
sage: C = LinearCode(G)
sage: C
Linear code of length 8, dimension 4 over Finite Field of size 2
sage: C.zeta_function()
(2/5*T^2 + 2/5*T + 1/5)/(2*T^2 - 3*T + 1)
sage: C.sd_zeta_polynomial()
2/5*T^2 + 2/5*T + 1/5
sage: C == C.dual_code()
True

```

In other words,

$$P_C(T) = (2T^2 + 2T + 1)/5.$$

2.2 Second definition

Here is Duursma's second definition of the zeta polynomial.

Definition 25 Let $F = A_C$ denote the weight enumerator of a $[n, k, d]_q$ -code C . Using the coefficients $a_j = a_j(F)$ of (5), define

$$P(T) = P_C(T) = a_0 + a_1T + \cdots + a_rT^r.$$

This $P(T)$ is the *Duursma zeta polynomial* of C .

More generally, if F is an virtual weight enumerator and the coefficients $a_j = a_j(F)$ are as in (5), define $P(T) = P_F(T) = a_0 + a_1T + \cdots + a_rT^r$.

Note that by comparing coefficients of x^n on both sides of (5), we see $a_0 + \cdots + a_r = 1$ is equivalent to $P(1) = 1$.

Example 26 Note that if C is a MDS code of length n and minimum distance d over $GF(q)$ then $A_C = M_{n,d}$ (this is proven as part of the discussion in §2 of Duursma [D2]). This forces $c = 0$, $a_0 = 1$ in (5), so⁵ $P(t) = 1$.

Remark 3 Note $[n, k, d]$ makes sense as parameters of a virtual weight enumerator is when F is a WE of an actual code C (so $F = A_C$) or when F is a VSDWE (so $\gamma = n/2 - d + 1$, where n and d are as in Definition 12) or a (virtual) MDS code (so $k = n + 1 - d$).

Lemma 27 The Duursma zeta function of Definition 17 is the same as the Duursma zeta function of Definition 25.

proof: By Definition 20, the zeta polynomial of Definition 17 associated to $F = A_C$ is T^r if you replace $F = A_C$ by $F = M_{n,d+j}$:

$$\frac{(xT + (1-T)y)^n}{(1-T)(1-qT)} T^j = \cdots + \frac{M_{n,d+j}(x,y) - x^n}{q-1} T^{n-d} + \cdots$$

Multiply by a_j and sum both sides over $j \in \{0, \dots, r\}$ to obtain Definition 25. Therefore, $P(T)$ satisfying Definition 17 also satisfies Definition 25. \square

⁵See also Duursma's Proposition 1 in [D5] and Chinen's Theorem 3.2 in [C3].

2.3 Third definition

In preparation for the third definition, which originated in §7 of Duursma [D1], we introduce some notation.

Let C be an $[n, k, d]_q$ code, let $S \subset \{1, 2, \dots, n\}$ be a subset, let C_S denote the subcode of C of codewords with support contained in S , and let $k_S = k_S(C)$ denote the dimension of C_S .

Lemma 28 *The dimension k_S satisfies*

$$k_S = \begin{cases} 0, & \text{for } 0 \leq |S| < d, \\ k - (n - |S|), & \text{for } n - d^\perp < |S| \leq n. \end{cases}$$

When $d \leq |S| \leq n - d^\perp$ then k_S depends on S and C in a more subtle way.

proof: It follows from the definition of the minimum distance d that $k_S = 0$ if $0 \leq |S| < d$. If C is $[n, k, d]$ then the dual code C^\perp is $[n, n - k, d^\perp]$, so $n - k + d^\perp \leq n + 1$, or $d^\perp \leq k + 1$. If $S^c = \{j \mid 1 \leq j \leq n, j \notin S\}$ then C_S is isomorphic to the code “shortened on S^c ”. The dimensions of such shortened codes is given in Theorem 1.5.7 in [HP]. In particular, if $|S^c| < d^\perp$ then we find $k_S = n - |S^c| - (n - k) = k - |S^c|$, as desired. \square

The *binomial moments* of C are the integers $B_0^1, B_1^1, B_2^1, \dots$ defined by

$$B_i^1 = B_i^1(C) = \sum_{\substack{S \\ |S|=i}} \frac{q^{k_S} - 1}{q - 1}.$$

Lemma 29 *The binomial moments satisfy*

$$B_i^1 = \begin{cases} 0, & \text{for } 0 \leq i < d, \\ \binom{n}{i} \frac{q^{i+k-n-1}}{q-1}, & \text{for } n - d^\perp < i \leq n. \end{cases}$$

proof: This is an easy corollary of the above lemma. \square

The numbers

$$b_i = b_i(C) = B_{d+i}^1 / \binom{n}{d+i} \tag{6}$$

are called the *normalized binomial moments* of C ($0 \leq i \leq n - d$). We extend this to all $i \in \mathbb{Z}$ by

$$b_i = b_i(C) = \begin{cases} 0, & \text{for } i < 0, \\ \frac{q^{i+d+k-n}-1}{q-1}, & \text{for } n - d^\perp - d < i. \end{cases}$$

Finally, we can give Duursma's third definition.

Definition 30 Define the *zeta function* of C to be the generating function of the normalized binomial moments of the code:

$$Z(T) = \sum_{i=0}^{\infty} b_i T^i.$$

This is a rational function (see Duursma [D1], §7),

$$Z(T) = \frac{P(T)}{(1-T)(1-qT)},$$

where

$$P(T) = p_0 + p_1 T + \cdots + p_{n+2-d-d^\perp} T^{n+2-d-d^\perp}$$

is the zeta polynomial, and

$$p_i = b_i - (q+1)b_{i-1} + qb_{i-2}. \quad (7)$$

Lemma 31 *The Duursma zeta function of Definition 30 is the same as the Duursma zeta function of Definition 17.*

proof: If

$$B^1(x, y) = \sum_{j=0}^n B_j^1 x^{n-j} y^j,$$

and $A_C(x, y) = x^n + (q-1)A^1(x, y)$ then it is known⁶ that $B^1(x, y) = A^1(x + y, y)$. Therefore, $\frac{A_C(x, y) - x^n}{q-1} = B^1(x - y, y)$ and

$$(zT + y)^n Z(T) = \cdots + B^1(z, y) T^{n-d} + \cdots$$

⁶This is proven in §9 of [D5]. See Theorem 1.1.26 and Exercise 1.1.27 in [TV] for a closely related result.

(where $z = x - y$) defines the Duursma zeta polynomial of C in the sense of Definition 17. Let us compare coefficients of $z^\ell T^{n-d}$ on both sides. On the right-hand side, it is $B_{n-\ell}^1$ and on the other side it is $\binom{n}{\ell} b_{n-d-\ell}$. We must verify that these are the same. However, this is the formula for the normalized binomial moment, so is, by definition, true. \square

As a corollary, we find that if the weight enumerator A_C is known, then

$$B^1(x, y) = \frac{A_C(x + y, y) - (x + y)^n}{q - 1} = \sum_{j=0}^n B_j^1 x^{n-j} y^j$$

is easy to compute and the coefficients of the zeta polynomial are given by (6) and (7). (In fact, this is what the SAGE command `zeta_polynomial` computes.)

SAGE

```
sage: C = HammingCode(3,GF(2))
sage: C.zeta_polynomial()
2/5*T^2 + 2/5*T + 1/5
sage: C = best_known_linear_code(6,3,GF(2))
sage: C.minimum_distance()
3
sage: C.zeta_polynomial()
2/5*T^2 + 2/5*T + 1/5
```

2.4 Analogies with curves

Let X be a smooth projective curve of genus g^7 over a finite field $GF(q)$ defined by a polynomial equation $F(x, y) = 0$, where F is a polynomial with coefficients in $GF(q)$. Let N_k denote the number of solutions in $GF(q^k)$ and create the generating function

$$G(t) = N_1 t + N_2 t^2/2 + N_3 t^3/3 + \dots$$

⁷These terms will not be defined precisely here. Please see Tsafman-Vladut [TV], §2.3.2, or Schmidt [Sc] for a rigorous treatment.

Define the zeta function of X by the formal power series

$$\zeta(t) = \zeta_X(t) = \exp(G(t)) \tag{8}$$

so $Z(0) = 1$. It is known that⁸

$$\zeta_X(t) = \frac{P(t)}{(1-t)(1-qt)},$$

with $P(t)$ a polynomial, of degree $2g$ where g is the genus. This has a “functional equation” of the form

$$P(t) = q^g t^{2g} P\left(\frac{1}{qt}\right).$$

The logarithmic derivative of ζ_X is the generating function of the sequence of counting numbers $\{N_1, N_2, \dots\}$. The Riemann hypothesis for curves over finite fields states that the roots of P have absolute value $q^{-1/2}$. These roots can be interpreted in terms of the eigenvalues of a linear transformation⁹ on a vector space. In fact, there is a unitary symplectic $2g \times 2g$ matrix $\Theta = \Theta_X$ such that¹⁰

$$P(t) = \det(I - tq^{1/2}\Theta).$$

Open Question 3 *Let C be a self-dual code over $GF(q)$. When is there a curve $X/GF(q)$ for which the zeta function of the curve ζ_X is equal to the zeta function Z_C of the code?*

The answer to this question is “no” if q is “large” compared to the length of C (see Corollary 38).

Since the RH holds for ζ_X (this is a well-known theorem of André Weil), a necessary condition for Open Question 3 to hold is that the code must satisfy the RH. See Example 9.7 in [D6] for two (self-dual) codes for which this holds.

⁸This was first proved by Dwork using p -adic methods [Dw].

⁹In fact, it is possible to interpret $P(t)$ in terms of the characteristic function of “the Frobenius operator” acting on a cohomology space, though we shall omit details here.

¹⁰See Faifman and Rudnick [FR] for an interesting analysis of the “statistics” of the eigenvalues of Θ in the case when X is “hyperelliptic”.

Open Question 4 *Let C be a self-dual code over $GF(q)$. When is there a linear operator Φ on a “natural” rational vector space for which the zeta polynomial $P = P_C$ can be interpreted in terms of the characteristic function of Φ ?*

Open Question 5 *Let C be a self-dual code over $GF(q)$. Is there a “natural” interpretation of the coefficients of the logarithmic derivative of Z_C ?*

There is a “natural” interpretation of the coefficients of Z_C - see the construction in §2.3 above.

3 Properties

We survey some of the most remarkable properties, both conjectured and proven, of these zeta functions.

3.1 The functional equation

If $\gamma = \gamma(C)$ is the genus of C and if

$$z_C(T) = Z_C(T)T^{1-\gamma}$$

then the functional equation in [D1] can be written in the form

$$z_{C^\perp}(T) = z_C(1/qT).$$

If we let

$$\zeta_C(s) = Z_C(q^{-s})$$

and

$$\xi_C(s) = z_C(q^{-s})$$

then ζ_C and ξ_C have the same zeros but ξ_C is “more symmetric” since the functional equation expressed in terms of it becomes¹¹

¹¹This notation is inspired by analogous notation used for functions associated with the classical Riemann zeta function. See any book on the Riemann zeta function, or http://en.wikipedia.org/wiki/Riemann_zeta_function.

$$\xi_{C^\perp}(s) = \xi_C(1-s).$$

Abusing terminology, we call both Z_C and ζ_C the *Duursma zeta function* of C .

The analog of this for a VSDWE is as follows: let F denote a VSDWE with degree n and minimum distance d , so $\gamma = n + 1 - k - d = n/2 + 1 - d$ is the genus.

In fact, since Duursma's zeta function *only* depends on C via its weight enumerator $A_C(x, y)$ of C , for any virtual weight enumerator $F(x, y)$ there is an associated *zeta function* $Z = Z_F$ and *zeta polynomial* $P = P_F$. If we 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 32 *For any virtual weight enumerator F satisfying*

$$F(x, y) = a_0 M_{n,d}(x, y) + a_1 M_{n,d+1}(x, y) + \cdots + a_r M_{n,d+r}(x, y),$$

and for any q , the zeta function $Z = Z_F$ satisfies the functional equation

$$Z^\perp(T) T^{1-g^\perp} = Z\left(\frac{1}{qT}\right) \left(\frac{1}{qT}\right)^{1-g}. \quad (9)$$

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

$$P^\perp(T) = P\left(\frac{1}{qT}\right) q^g T^{g+g^\perp}, \quad (10)$$

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

Remark 4 (1) *Note that both P^\perp and P are polynomials of degree $n + 2 - d - d^\perp = g + g^\perp$, and g is the genus if $F = A_C$ is an actual weight enumerator.*

(2) *This proof is essentially the same as that of Proposition 9.2 in [D6]. This hypothesis here is slightly more general.*

proof: This is a consequence of Definition 25 and the MacWilliams identity.

By hypothesis, the coefficients $a_j = a_j(F)$ of (5) satisfy $a_0 + \dots + a_r = 1$. Therefore, $F^\perp = F \circ \sigma$ satisfies

$$F^\perp = a_0 M_{n,d} \circ \sigma + a_1 M_{n,d+1} \circ \sigma + \dots + a_r M_{n,d+r} \circ \sigma. \quad (11)$$

Recall that the dual of the MDS code with parameters $[n, k, \delta]$ is the MDS code with parameters $[n, k^\perp, \delta^\perp]$. By this and MacWilliams' identity, we have $M_{n,\delta} \circ \sigma = q^{n/2+1-\delta} M_{n,\delta^\perp} = q^{k-n/2} M_{n,\delta^\perp}$, where $k^\perp + \delta^\perp = n + 1$ and $k = n - \delta + 1$ is the dimension of the (virtual) MDS code of length n and minimum distance δ (for a proof of this, see Appendix A in Duursma [D5]). Thus, $M_{n,\delta} \circ \sigma = q^{n/2+1-\delta} M_{n,n-\delta+2}$, and it follows that

$$\begin{aligned} F^\perp &= \sum_{d \leq \delta \leq d+r} a_{\delta-d} q^{n/2+1-\delta} M_{n,n-\delta+2} \\ &= \sum_{n-d-r+2 \leq \delta' \leq n-d+2} a_{n-\delta'+2-d} q^{\delta'-1-n/2} M_{n,\delta'} \\ &= \sum_{0 \leq \delta'' \leq r} a_{r-\delta''} q^{n/2-d-r+1+\delta''} M_{n,n-d-r+2+\delta''}. \end{aligned}$$

This implies

$$\begin{aligned} P^\perp(T) &= a_0^\perp + a_1^\perp T + \dots + a_r^\perp T^r \\ &= a_r q^{n/2-r-d+1} + a_{r-1} q^{n/2-r-d+2} T + \dots + a_0 q^{n/2-d+1} T^r \\ &= a_r q^{n/2-r-d+1} + a_{r-1} q^{n/2-r-d+1} (Tq) + \dots + a_0 q^{n/2-r-d+1} (Tq)^r \\ &= q^{n/2-r-d+1} (a_r + a_{r-1} (Tq) + \dots + a_0 (Tq)^r) \\ &= q^{n/2-r-d+1} (Tq)^r (a_0 + a_1 (Tq)^{-1} + \dots + a_r (Tq)^{-r}) \\ &= q^{n/2-d+1} T^r P(1/qT). \end{aligned}$$

□

3.2 Puncturing preserves P

Suppose C is an $[n, k, d]$ code over $GF(q)$ and i is any integer satisfying $1 \leq i \leq n$. The *punctured code* $P_i(C)$ at the coordinate i is the code having length $n - 1$ obtained by projecting C onto the remaining coordinates. We denote The *shortened code* $S_i(C)$ at the coordinate i is the code having length $n - 1$ obtained by projecting the subcode

$$\{c = (c_1, \dots, c_n) \in C \mid c_i = 0\}$$

onto the remaining coordinates.

Lemma 33 *If C is a linear code of length n and i is an integer, $1 \leq i \leq n$, then*

$$P_i(C)^\perp = S_i(C^\perp).$$

A *check bit extension* \hat{C} is a code of length $n + 1$ of the form

$$\{(c_1, \dots, c_n, c_{n+1}) \in GF(q)^{n+1} \mid (c_1, \dots, c_n) \in C, c_{n+1} = c \cdot a\}$$

for some fixed vector $a \in GF(q)^n$.

To end this section, we recall that the zeta polynomial of a code C , P_C , remains the same if we replace C by (a) the averaged puncturing $P(C)$ of C , (b) the averaged shortening $S(C)$ of C , or (c) a check-bit extension \hat{C} of C . This provides two inductive formulas for computing the zeta polynomial.

Theorem 34 (Duursma [D5]) *If C is a linear code of length n , is*

$$F_{P(C)}(x, y) = \frac{1}{n} \sum_{i=1}^n A_{P_i(C)}(x, y)$$

denotes the averaged punctured weight enumerator, and

$$F_{S(C)}(x, y) = \frac{1}{n} \sum_{i=1}^n A_{S_i(C)}(x, y)$$

denotes the averaged shortened weight enumerator, then

$$P_C(T) = P_{F_{P(C)}}(T) = P_{F_{S(C)}}(T).$$

This is proven in §5 of Duursma [D5].

Open Question 6 *Is there a simple relationship between $P_C(T)$ and $P_{\hat{C}}(T)$?*

3.3 The RH

Knowledge of the zeros of $Z(T)$ could be very useful for understanding the possible values of the minimum distance.

Proposition 35 (Duursma) *If $\{\rho_1, \rho_2, \dots, \rho_r\}$, with $r \geq 1$, denote the zeros of the Duursma zeta function $P(T)$ of a linear code C and $[A_0, \dots, A_n]$ denotes the spectrum of C then*

$$d = q - \sum_i \rho_i^{-1} - \frac{A_{d+1}}{A_d} \frac{d+1}{n-d}.$$

In particular,

$$d \leq q - \sum_i \rho_i^{-1}.$$

The proof uses the assumption that C is a linear code, not a virtual code, and that $P(T) \neq 1$.

proof: For the first statement, see equations (5)-(6) of [D5] (also, (4.1) of [D4]). The second statement follows from the first since $\frac{A_{d+1}}{A_d} \geq 0$. \square

Corollary 36 *If C is any b -divisible code with $b \geq 2$ then*

$$d = q - \sum_i \rho_i^{-1}.$$

If C is a formally self-dual b -divisible code with $q < d$ then

$$d \leq \frac{n+2}{m+2} + q,$$

where $m = \min_i |\rho_i|$.

proof: The first statement follows from the definition of b -divisible. For the second statement, we have $d - q = -\sum_i \rho_i^{-1} \leq \frac{r}{m} = \frac{n+2-2d}{m}$. Multiply both sides by m and “solve” for d to get the result claimed. \square

If F is VSDWE then the zeros of the zeta function $\zeta_F(s)$ (or $\xi_F(s)$) occur in pairs about the “critical line” $Re(s) = \frac{1}{2}$.

Definition 37 *We say the zeta function ζ_F (or, by abuse of terminology, the VSDWE F) satisfies the Riemann hypothesis (RH) if all zeta zeros occur on the “critical line”.*

The following result is not best possible, but illustrates the idea that for “large” q , the RH is “often” false.

Corollary 38 *Let C be an $[n, k, d]$ code over $GF(q)$ with $q > n^2$, $2 \leq d$ and $d + d^\perp < n + 2$. If $n > 3$ then the Duursma zeta polynomial is not a constant and does not satisfy the RH.*

This is an easy consequence of the Proposition 35 (assume the RH is true and $q > n^2$, then show the hypothesis contradicts the trivial estimate $q - d \leq |\sum_i \rho_i^{-1}| \leq r\sqrt{q} = (n + 2 - d - d^\perp)\sqrt{q}$) and the proof is left to the reader.

Example 39 *It is clear from Example 26 above that the Duursma zeta function may have no zeros (i.e., may be constant). Indeed, this is true for all MDS codes, including some formally self-dual ones¹².*

Remark 5 *Let F denote a VSDWE as in Proposition 32 and let $r(T) = z_F(T/\sqrt{q})$. The functional equation implies $r(T)$ is a self-reciprocal function: $r(1/T) = r(T)$. The RH is the statement that all 2γ zeros of $r(T)$ lie on the “critical line” $|T| = 1$. If $r_0(\theta) = r(e^{i\theta})$ then the functional equation, and the fact that r has rational coefficients, implies*

$$r_0(\theta) = r_0(-\theta) = \overline{r_0(\theta)}.$$

In other words, $r_0(\theta)$ is real-valued.

Conjecture 40 (Duursma) *For all extremal virtual weight enumerators F , the zeta function $Z = Z_F$ satisfies the Riemann hypothesis.*

Lemma 41 *Let F denote a VSDWE of genus γ as above and let $P = P_F$ denote the associated zeta polynomial. It is known that $P(T^2/q) = T^{2\gamma}f(T + T^{-1})$, where $f \in \mathbb{R}[x]$ is a polynomial of degree 2γ with real coefficients.*

proof: See Duursma [D3], Theorem 7 and Lemma 10. \square

Remark 6 Lemma 2.1.1 in [DH] classifies those polynomials of even degree in $\mathbb{R}[x]$ which have all its roots on the unit circle. That result allows us to reformulate the RH (a statement about the complex roots of P) as a statement about the real roots of $f(x)$ in $-2 \leq x \leq 2$.

¹²Formally self-dual MDS codes exist - see Example 12 in [JKT], which gives a fsd $[42, 21, 22]$ -code over a very large extension of $GF(7)$. (In fact, this code even has A_5 as its permutation automorphism group.) Even better, in Kim and Lee [KL], a self-dual MDS code with parameters $[10, 5, 6]_{41}$ is constructed.

4 Examples

4.1 Komichi's example

In [HT], the authors mention an example which occurred in the master's thesis¹³ of A. Komichi. It is claimed that the Duursma zeta function of the code $C = H_8 \oplus H_8 \oplus H_8$, where H_8 is the self-dual extended Hamming [8, 4, 4]-code, violates the Riemann hypothesis. We verify this using SAGE .

```
----- SAGE -----
sage: MS = MatrixSpace(GF(2), 12, 24)
sage: G = MS([\
....: [ 1,1,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ],\
....: [ 0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ],\
....: [ 0,0,1,0,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ],\
....: [ 0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ],\
....: [ 0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,1,0,0,0,0,0,0 ],\
....: [ 0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0 ],\
....: [ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,1,0,0,0,0,0 ],\
....: [ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0 ],\
....: [ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0 ],\
....: [ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1 ],\
....: [ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1 ],\
....: ]\
....: ])
sage: C = LinearCode(G)
sage: Cd = C.dual_code(); C == Cd
True
sage: R = PolynomialRing(CC,"T")
sage: T = R.gen()
sage: C.zeta_polynomial()
512/253*T^18 + 512/253*T^17 + 256/253*T^16 - 148736/245157*T^14
- 66048/81719*T^13 - 185536/245157*T^12 - 49408/81719*T^11
- 43088/96577*T^10 - 1808/5681*T^9 - 21544/96577*T^8 - 12352/81719*T^7
- 23192/245157*T^6 - 4128/81719*T^5 - 4648/245157*T^4 + 2/253*T^2 + 2/253*T + 1/253
sage: f = R(C.zeta_polynomial())
sage: print [z[0].abs() for z in f.roots()]
[0.963950810639179, 0.707106781186546, 0.707106781186548,
0.707106781186546, 0.518698666447988, 0.707106781186548,
0.707106781186542, 0.707106781186548, 0.707106781186550,
0.707106781186551, 0.707106781186547, 0.707106781186546,
0.707106781186548, 0.707106781186544, 0.707106781186548,
0.707106781186549, 0.707106781186548, 0.707106781186549]
sage: P1 = list_plot([(z[0].real(),z[0].imag()) for z in f.roots()])
sage: t = var("t")
sage: pts = lambda t: [cos(t)/sqrt(2),sin(t)/sqrt(2)]
sage: P2 = parametric_plot(pts(t),0,2*pi,linestyle="--",rgbcolor=(1,0,0))
sage: show(P1+P2)
```

¹³This appears to be unpublished and I have not seen it myself.

The plot computed in the last line is shown below:

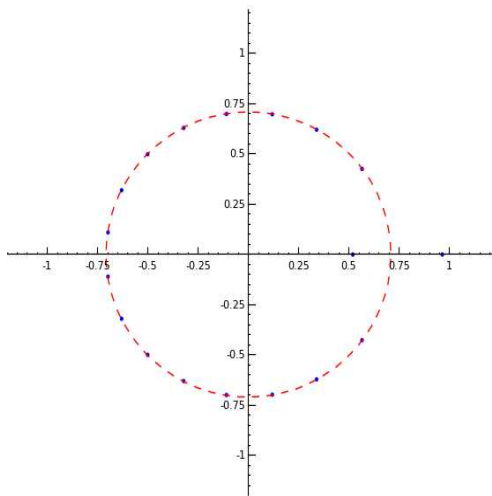


Figure 1: Roots of the zeta polynomial for a self-dual [24, 12, 4] binary code.

4.2 The extremal case

We shall summarize some results of Duursma [D3] and Harada and Tagami [HT] in this section.

If F is an extremal VSDWE then the zeta function $Z = Z_F$ can be explicitly computed. First, some notation. If F is a VSDWE of minimum distance d and $P = P_F$ is its zeta polynomial then define

$$Q(T) = \begin{cases} P(T), & \text{Type I,} \\ P(T)(1 - 2T + 2T^2), & \text{Type II,} \\ P(T)(1 + 3T^2), & \text{Type III,} \\ P(T)(1 + 2T), & \text{Type IV.} \end{cases}$$

Let $(a)_m = a(a+1)\dots(a+m-1)$ denote the *rising generalized factorial* and write $Q(T) = \sum_j q_j T^j$, for some $q_j \in \mathbb{Q}$. Let

$$\gamma_1(n, d, b) = (n-d)(d-b)_{b+1} A_d / (n-b-1)_{b+2},$$

and

$$\gamma_2(n, d, b, q) = (d-b)_{b+1} \frac{A_d}{(q-1)(n-b)_{b+1}},$$

where recall A_d denoted the coefficient of $x^{n-d}y^d$ in the virtual weight enumerator $F(x, y)$.

Theorem 42 (Duursma [D3]) *If F is an extremal VSDWE then the coefficients of $Q(T)$ are determined as follows.*

(a) *If F is Type I then*

$$\sum_{i=0}^{2m+2\nu} \binom{4m+2\nu}{m+i} q_i T^i = \gamma_1(n, d, 2) \cdot (1+T)^m (1+2T)^m (1+2T+2T^2)^\nu,$$

where $m = d - 3$, $4m + 2\nu = n - 4$, $b = q = 2$, $0 \leq \nu \leq 3$.

(b) *If F is Type II then*

$$\sum_{i=0}^{4m+8\nu} \binom{6m+8\nu}{m+i} q_i T^i = \gamma_1(n, d, 2) \cdot (1+T)^m (1+2T)^m (1+2T+2T^2)^m B(T)^\nu,$$

where $m = d - 5$, $6m + 8\nu = n - 6$, $b = 4$, $q = 2$, $0 \leq \nu \leq 2$, and $B(T) = W_5(1+T, T)$, where W_5 is as in Example 6.

(c) *If F is Type III then*

$$\sum_{i=0}^{2m+4\nu} \binom{4m+4\nu}{m+i} q_i T^i = \gamma_2(n, d, 3, 3) \cdot (1+3T+3T^2)^m B(T)^\nu,$$

where $m = d - 4$, $4m + 4\nu = n - 4$, $b = q = 3$, $0 \leq \nu \leq 2$, and $B(T) = W_9(1+T, T)$, where W_9 is as in Example 6.

(d) *If F is Type VI then*

$$\sum_{i=0}^{m+2\nu} \binom{3m+2\nu}{m+i} q_i T^i = \gamma_2(n, d, 2, 4) \cdot (1+2T)^m (1+2T+4T^2)^\nu,$$

where $m = d - 3$, $3m + 2\nu = n - 3$, $b = 2$, $q = 4$, and $0 \leq \nu \leq 2$.

It is easy to determine (especially with a computer algebra system such as SAGE) the coefficients q_j and p_j from these expressions.

Define the **ultraspherical polynomial** $C_n^m(x)$ on the interval $(-1, 1)$ by

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

Theorem 43 (Duursma [D3], section 5.2¹⁴) is due to If P is the Duursma zeta polynomial of an extremal Type IV virtual self-dual weight enumerator of length $n = 3m + 3$ and minimum distance $d = m + 3$ then

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

(Recall that, in this case, $Q(T) = P(T)(1 + 2T)$.)

It's known that all the roots of ultraspherical polynomials C_n^m lie on the interval $(-1, 1)$. The polynomial C_n^m is degree n and so there are n such roots. Replace T by $e^{i\theta}$ in the equation displayed in the Theorem above to obtain

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

Hence, all the roots of Q , therefore also P , lie on the circle of radius $1/\sqrt{q} = 1/2$. Indeed, the RH holds for all zeta functions associated to an extremal Type IV VSDWE (Duursma [D3]).

Using computer computations, Harada and Tagami [HT] (among other things) that RH holds for all zeta functions associated to extremal Type I, II, III VSDWEs of degree ≤ 200 .

4.3 “Random divisible codes”

Following Theorem 4 in Duursma [D5], we show that the Duursma zeta function of a “random divisible code” satisfies the RH.

Define the (virtual) weight enumerator of the $[n, k]_q$ random b -divisible code by

¹⁴A typo in [D3], §5.2, is corrected here.

$$F(x, y) = x^n + c \sum_{i=1}^{n/b} \binom{n}{bi} (q-1)^{bi} x^{n-bi} y^{bi},$$

where c is chosen so that $F(1, 1) = q^k$ and n is a multiple of b . Of course, by the classification of b -divisible codes (see Theorem 13), this weight enumerator may not correspond to an actual linear code.

Duursma shows that in the following cases the zeta function $Z_F(T)$ satisfies the RH: n is even, $k = n/2$ and

- $q = 2, b = 4,$
- $q = 3, b = 3,$
- $q = 4, b = 2.$

For details, see Duursma [D5], Theorem 4.

4.4 A fsd $[26, 13, 6]_2$ -code

Moreover, in this case the Riemann hypothesis is not valid for optimal codes (which may or may not be extremal) in general, as the following example illustrates.

Example 44 Consider the $[26, 13, 6]_2$ code with weight distribution

$$[1, 0, 0, 0, 0, 0, 39, 0, 455, 0, 1196, 0, 2405, 0, 2405, 0, 1196, 0, 455, 0, 39, 0, 0, 0, 0, 0, 1].$$

This is (by coding theory tables, as included in SAGE [S]) an optimal formally self-dual code. This code C has zeta polynomial

$$\begin{aligned} P(T) = & \frac{3}{17710} + \frac{6}{8855}T + \frac{611}{336490}T^2 + \frac{9}{2185}T^3 + \frac{3441}{408595}T^4 + \frac{6448}{408595}T^5 + \frac{44499}{1634380}T^6 \\ & + \frac{22539}{520030}T^7 + \frac{66303}{1040060}T^8 + \frac{22539}{260015}T^9 + \frac{44499}{408595}T^{10} + \frac{51584}{408595}T^{11} \\ & + \frac{55056}{408595}T^{12} + \frac{288}{2185}T^{13} + \frac{19552}{168245}T^{14} + \frac{768}{8855}T^{15} + \frac{384}{8855}T^{16}. \end{aligned}$$

Using SAGE, it can be checked that only 8 of the 12 zeros of this function have absolute value $\sqrt{2}$.

4.5 Extremal codes of short length

In this section, we give some examples using SAGE .

These do not satisfy $P(1) = 1$ but use the formulas in Theorem 42 above:

For the $[24, 12, 8]_2$ VSDWE:

$$P(T) = \frac{2}{969}T^{10} + \frac{2}{323}T^9 + \frac{10}{969}T^8 + \frac{4}{323}T^7 + \frac{197}{16796}T^6 + \frac{9}{988}T^5 + \frac{197}{33592}T^4 + \frac{1}{323}T^3 + \frac{5}{3876}T^2 + \frac{1}{2584}T + \frac{1}{15504}.$$

For the $[26, 13, 8]_2$ VSDWE:

$$P(T) = \frac{32}{13167}T^{12} + \frac{32}{4389}T^{11} + \frac{4}{323}T^{10} + \frac{496}{31977}T^9 + \frac{393}{24871}T^8 + \frac{31}{2261}T^7 + \frac{281}{27132}T^6 + \frac{31}{4522}T^5 + \frac{393}{99484}T^4 + \frac{62}{31977}T^3 + \frac{1}{1292}T^2 + \frac{1}{4389}T + \frac{1}{26334}$$

For the $[28, 14, 8]_2$ VSDWE:

$$P(T) = \frac{16}{5313}T^{14} + \frac{16}{1771}T^{13} + \frac{224}{14421}T^{12} + \frac{96}{4807}T^{11} + \frac{3469}{163438}T^{10} + \frac{291}{14858}T^9 + \frac{23}{1428}T^8 + \frac{622}{52003}T^7 + \frac{23}{2856}T^6 + \frac{291}{59432}T^5 + \frac{3469}{1307504}T^4 + \frac{6}{4807}T^3 + \frac{1}{14421}T^2 + \frac{1}{7084}T + \frac{1}{42504}$$

4.6 Non-self-dual examples

Consider the optimal binary code C having parameters $[6, 2, 4]$ and generator matrix

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

This has zeta polynomial $P(T) = (2T^2 + 2T + 1)/5$, as the following SAGE computation shows.

```

SAGE
-----
sage: R_CC = PolynomialRing(CC, "T")
sage: n = 6; k = 2; q = 2
sage: C = best_known_linear_code(n, k, GF(q))
sage: C.zeta_polynomial()
2/5*T^2 + 2/5*T + 1/5

```

```

sage: [abs(z[0]) for z in R_CC(C.zeta_polynomial()).roots()]
[0.707106781186548, 0.707106781186548]
sage: C.weight_enumerator()
x^6 + 3*x^2*y^4
sage: Cd = C.dual_code()
sage: Cd.zeta_polynomial()
2/5*T^2 + 2/5*T + 1/5
sage: Cd.weight_enumerator()
x^6 + 3*x^4*y^2 + 8*x^3*y^3 + 3*x^2*y^4 + y^6
sage: n = 7; k = 4; q = 2
sage: C = best_known_linear_code(n,k,GF(q))
sage: C.zeta_polynomial()
2/5*T^2 + 2/5*T + 1/5
sage: C.weight_enumerator()
x^7 + 7*x^4*y^3 + 7*x^3*y^4 + y^7
sage: Cd = C.dual_code()
sage: Cd.zeta_polynomial()
2/5*T^2 + 2/5*T + 1/5
sage: Cd.weight_enumerator()
x^7 + 7*x^3*y^4
sage: n = 8; k = 4; q = 2
sage: C = best_known_linear_code(n,k,GF(q))
sage: C.zeta_polynomial()
2/5*T^2 + 2/5*T + 1/5
sage: C.weight_enumerator()
x^8 + 14*x^4*y^4 + y^8
sage: Cd = C.dual_code()
sage: Cd.zeta_polynomial()
2/5*T^2 + 2/5*T + 1/5
sage: Cd.weight_enumerator()
x^8 + 14*x^4*y^4 + y^8

```

Indeed, the optimal $[6, 2, 4]$ code has the same zeta polynomial as the Hamming $[7, 4, 3]$ code. This satisfies the RH even though it is not even formally self-dual. However, it does have the same zeta polynomial as the optimal self-dual $[8, 4, 4]$ code.

5 Chinen zeta functions

In the sections above, a virtual weight enumerator F is associated with a zeta function $Z = Z_F$. In this section, two related zeta functions were constructed

by Koji Chinen. First, he constructed a zeta function $Z = Z_F$, which we call a “twisted Chinen zeta function”, associated to a twisted VSDWE F . (What we call a “twisted VSDWE” he calls a “formal weight enumerator”.) Next, he constructed a zeta function associated to any code C , which we call a “Chinen zeta function”, which is essentially defined by combining the Duursma zeta function of C with that of its dual C^\perp (some care is required to insure that the functional equation leads to an extra symmetry property).

Here are the analogous result for Chinen zeta functions of the results above.

Let C be any $[n, k, d]$ code over $GF(q)$ and let $[n, n - k, d^\perp]$ denote the parameters of the dual code C^\perp . We assume they satisfy $d \geq 2$ and $d^\perp \geq 2$. Define the *invariant weight enumerator* by

$$\tilde{A}_C(x, y) = \frac{A_C(x, y) + q^{k-n/2} A_{C^\perp}(x, y)}{1 + q^{k-n/2}}.$$

Note that $\tilde{A}_C = \tilde{A}_{C^\perp} = \tilde{A}_C \circ \sigma_q$, by the MacWilliams identity. The *Chinen zeta polynomial* \tilde{P}_C is the zeta polynomial P_F associated to the virtual weight enumerator $F = \tilde{A}_C$. The *Chinen zeta function* is defined in terms of the zeta polynomial by means of the following equation.

$$\tilde{P}_C(T) = \frac{T^{\max(0, d-d^\perp)}}{1 + q^{k-n/2}} (P_C(T) + q^{n/2-d+1} T^{n-2d+2} P_C(1/qT)). \quad (12)$$

Theorem 45 (*K. Chinen [C3]*) *The Chinen zeta polynomial given by (12) above has degree $2\tilde{g} = n+2-2\min(d, d^\perp)$ and satisfies the functional equation*

$$\tilde{P}_C(T) = q^{\tilde{g}} T^{2\tilde{g}} \tilde{P}_C(1/qT).$$

By the functional equation, if $d > d^\perp$ then $\tilde{P}_C(T) = \frac{q^{k-n/2} P_{C^\perp}(T) + T^{d-d^\perp} P_C(T)}{1 + q^{k-n/2}}$; if $d < d^\perp$ then $\tilde{P}_C(T) = \frac{P_C(T) + q^{k-n/2} T^{d^\perp-d} P_{C^\perp}(T)}{1 + q^{k-n/2}}$; and if $d = d^\perp$ then $\tilde{P}_C(T) = \frac{P_C(T) + q^{k-n/2} P_{C^\perp}(T)}{1 + q^{k-n/2}}$.

Note that when $T = 1$, $P(1) = 1$ and (by the functional equation), $P(1/q) = q^{-g} = q^{d-1-n/2}$. This implies $\tilde{P}_C(1) = \frac{2}{1+q^{k-n/2}}$. It may be simpler is to use the “averaged” zeta function

$$P_C^*(T) = (P_C(T) + P_{C^\perp}(T))/2,$$

but this is *not* the Chinen zeta function.

Example 46 We use SAGE to compute the Chinen zeta polynomial of some small optimal codes. We shall normalize the Chinen zeta function so that $\tilde{P}_C(1) = 1$.

```

SAGE
sage: R_CC = PolynomialRing(CC, "T")
sage: n = 8; k = 2; q = 2
sage: C = best_known_linear_code(n,k,GF(q))
sage: P = C.chinen_polynomial()
sage: Cd = C.dual_code()
sage: Pd = Cd.chinen_polynomial()
sage: C.minimum_distance(); Cd.minimum_distance()
5
2
sage: P; P == Pd
2/5*t^6 + 9/35*t^5 + 4/35*t^4 + 2/35*t^3 + 2/35*t^2 + 9/140*t + 1/20
True
sage: [abs(z[0]) for z in R_CC(P*1.0).roots()]

[0.707106781186548,
 0.707106781186548,
 0.707106781186547,
 0.707106781186547,
 0.707106781186547,
 0.707106781186548]
sage: C.gen_mat()
[0 0 0 1 1 1 1 1]
[1 1 1 0 0 1 1 1]
sage: C0 = C.standard_form()[0]
sage: C0.gen_mat()
[1 0 1 1 0 1 1 1]
[0 1 0 0 1 1 1 1]

```

The RH is (apparently) true since the zeros have absolute value (approximately) $1/\sqrt{2}$.

```

SAGE
sage: C = HammingCode(3,GF(2))
sage: C.chinen_polynomial()
(2*sqrt(2)*t^3/5 + 2*sqrt(2)*t^2/5 + 2*t^2/5 + sqrt(2)*t/5 + 2*t/5 + 1/5)/(sqrt(2) + 1)

```

It can be easily shown that if C is formally self-dual then $\tilde{P}_C = P_C$. We say C (whether formally self-dual or not) *satisfies the RH* if its Chinen zeta polynomial has all its zeros on the “critical line”.

For example, if C is an MDS code then

$$\tilde{P}_C(T) = \frac{1}{1 + q^{k-n/2}}(1 + q^{n/2-d+1}T^{n-2d+2}).$$

If C is MDS and $n - 2d + 2 \neq 0$ then the RH holds for the Chinen zeta function.

Open Question 7 *Let C be any code over $GF(q)$. When is there a curve $X/GF(q)$ for which the zeta function of the curve ζ_X is equal to the Chinen zeta function Z_C of the code?*

Since the RH holds for ζ_X (this is a well-known theorem of André Weil), a necessary condition is that the code must satisfy the RH. See Example 9.7 in [D6] for two (self-dual) codes for which this holds.

Remark 7 *For the “twisted case”, including detailed proofs and numerous examples, please see Chinen [C2].*

Open Question 8 *Is the Chinen zeta function of a linear code C equal to the Duursma zeta function of some self-dual code C' ?*

If yes, then of course the set of Chinen zeta functions would be contained in the set of Chinen zeta functions.

Example 47 We use SAGE to compute the Chinen zeta polynomial of some indecomposable codes.

Consider codes which are generated by the matrix D_m (m even), defined as follows.

SAGE

```
def d_matrix(m):
    if not(is_even(m)):
        raise ValueError, "%s must be even and >2"%m
    M = int(m/2)
    A = [[0]*2*i+[1]*4+[0]*(m-4-2*i) for i in range(M-1)]
    MS = MatrixSpace(GF(2), M-1, m)
    return MS(A)
```

For example,

$$D_{14} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

and the binary code generated by this matrix is a $[14, 6, 4]$ code. You can see it's Chinen zeta function does not satisfy the RH:

```

SAGE
sage: n = 14; G = d_matrix(n); C = LinearCode(G); C
Linear code of length 14, dimension 6 over Finite Field of size 2
sage: C.spectrum()
[1, 0, 0, 0, 21, 0, 0, 0, 35, 0, 0, 0, 7, 0, 0]
sage: PT = PolynomialRing(CC,"T")
sage: PC = C.chinen_polynomial(); rts = PT(PC).roots()
sage: PC
64/39*t^12 - 32/429*t^10 - 32/429*t^9 - 160/1287*t^8 - 64/429*t^7 -
160/1287*t^6 - 32/429*t^5 - 40/1287*t^4 - 4/429*t^3 - 2/429*t^2 + 1/39
sage: [z[0].abs() for z in rts]

[0.707106781186548,
 0.707106781186548,
 0.707106781186548,
 0.707106781186547,
 0.707106781186548,
 0.707106781186548,
 0.707106781186549,
 0.707106781186548,
 0.707106781186547,
 0.707106781186548,
 0.814795710093010,
 0.613650751723920]
```

In particular, the RH for the Chinen zeta function is not true for all indecomposable codes.

5.1 Hamming codes

Chinen [C3] computed the zeta polynomial of the Hamming codes. Consider the Hamming code $C = C_{r,q}$ having parameters $[n = \frac{q^r-1}{q-1}, n - r, 3]$ over $GF(q)$, with $r \geq 3$. (When $r = 2$ the Hamming code is MDS and so has already been computed.)

The Duursma zeta polynomial of the dual code is given by

$$P_{C^\perp}(T) = c \cdot [1 + \sum_{j=1}^{n-d-1} \left(\binom{j+d-1}{d-1} - q \binom{j+d-2}{d-1} \right) T^j],$$

where the constant $c = c_{r,q}$ is chosen so that $P(1) = 1$. This is Proposition 4.4 in [C3].

The Chinen zeta polynomial of the Hamming codes $C_{r,q}$ ($r \geq 3$, $q \geq 2$) is given by

$$\tilde{P}_C(T) = \frac{c}{1 + q^{r-n/2}} (F_1(T) - qF_2(T)), \quad (13)$$

where

$$F_1(T) = \sum_{j=0}^{n-d-1} \binom{n-i-2}{d-1} q^{i+2-n/2} T^i + \sum_{j=d-3}^{n-4} \binom{i+2}{d-1} T^i,$$

and

$$F_2(T) = \sum_{j=0}^{n-d-2} \binom{n-i-3}{d-1} q^{i+2-n/2} T^i + \sum_{j=d-2}^{n-4} \binom{i+1}{d-1} T^i.$$

This is Theorem 4.5 in [C3].

Example 48 Here is the Chinen zeta polynomial of the Hamming $[7, 4, 3]$ code:

SAGE

```
sage: C = HammingCode(3,GF(2))
sage: C.chinin_polynomial()
(2*T^2/5 + 2*sqrt(2)*T*(T^2/5 + T/5 + 1/10) + 2*T/5 + 1/5)/(sqrt(2) + 1)
```

Theorem 49 (*Chinen*) *The Chinen zeta polynomial of the Hamming codes $C_{r,q}$ ($r \geq 3$, $q \geq 4$) satisfies the RH.*

This theorem is also true when $r = 2$ ($q \geq 2$), as a corollary to equation (3.3) in [C3], since then C is MDS.

Chinen’s proof of this theorem is too beautiful to omit at least a brief sketch. We need the following remarkable result.

Lemma 50 (*Chinen*) *If $f(T)$ is a degree m polynomial of “decreasing symmetric form”*

$$f(T) = a_0 + a_1T + \cdots + a_kT^k + a_kT^{m-k} + a_{k-1}T^{m-k-1} + \cdots + a_0T^m,$$

with $a_0 > a_1 > \cdots > a_k > 0$ then all roots of $f(T)$ lie on the unit circle $|T| = 1$.

To prove Theorem 49, Chinen explicitly computes the coefficients a_i of a normalized Chinen zeta polynomial f of $C = C_{r,q}$ and proves that it has the above decreasing symmetric form. This implies the RH, as desired. The proof of the above lemma and the explicit computation of the coefficients are carefully worked out in [C3], which we refer to for details.

5.2 Golay codes

This section summarizes some of the results in Chinen [C3], §7.

The Chinen zeta polynomial of the [11, 6, 5] Golay code C over $GF(3)$ is

$$\tilde{P}_C(T) = \frac{\sqrt{3}-1}{14}(\sqrt{3}T+1)(3T^2+3T+1).$$

Chinen also presents an explicit expression but complicated expression for the Chinen zeta polynomial of the [23, 12, 7] Golay code C over $GF(2)$. He also shows that both of these Chinen zeta functions satisfy the “Riemann hypothesis”. The proof is by explicitly computing zeros, verifying the RH numerically.

5.3 Examples

We begin with a random example:

SAGE

```
sage: RT = PolynomialRing(CC,"T")
sage: MS = MatrixSpace(GF(2), 3, 8)
sage: G = MS([[1,0,0,1,0,1,1,0],[0,1,0,1,0,0,0,1],[0,0,1,0,1,1,1,0]])
sage: C = LinearCode(G)
sage: C.minimum_distance()
3
sage: Cd = C.dual_code(); Cd.minimum_distance()
2
sage: f = RT(C.chinen_polynomial())
sage: print [z[0].abs() for z in f.roots()]
[0.707106781186548, 0.707106781186548, 0.707106781186548,
 0.707106781186548, 0.707106781186547, 0.707106781186547]
sage: C.gen_mat()
[1 0 0 1 0 1 1 0]
[0 1 0 1 0 0 0 1]
[0 0 1 0 1 1 1 0]
sage: C.spectrum()
[1, 0, 0, 1, 3, 2, 0, 1, 0]
sage: Cd.spectrum()
[1, 0, 3, 10, 7, 4, 5, 2, 0]
sage: C.chinen_polynomial()
2/7*t^6 + 4/21*t^5 + 13/70*t^4 + 17/105*t^3 + 13/140*t^2 + 1/21*t + 1/28
sage: C.zeta_polynomial()
3/7*T^5 + 3/14*T^4 + 11/70*T^3 + 17/140*T^2 + 17/280*T + 1/56
sage: f = RT(C.zeta_polynomial())
sage: print [z[0].abs() for z in f.roots()]
[0.644472635143760, 0.644472635143761, 0.458731710756610,
 0.476718789722295, 0.458731710756610]
```

This next example is also random:

SAGE

```
sage: C = RandomLinearCode(8,3,GF(2)); C.minimum_distance()
3
sage: Cd = C.dual_code(); Cd.minimum_distance()
2
sage: C.spectrum()
[1, 0, 0, 1, 3, 2, 0, 1, 0]
sage: Cd.spectrum()
[1, 0, 3, 6, 11, 8, 1, 2, 0]
sage: C.chinen_polynomial()
2/7*t^6 + 4/21*t^5 + 13/70*t^4 + 17/105*t^3 + 13/140*t^2 + 1/21*t + 1/28
sage: C.gen_mat()
[1 0 0 1 1 0 0 1]
[0 1 0 0 0 1 1 0]
[0 0 1 1 0 0 1 1]
sage: C.zeta_polynomial()
3/7*T^5 + 3/14*T^4 + 11/70*T^3 + 17/140*T^2 + 17/280*T + 1/56
```

The next example concerns a code which is formally self-dual but not self-dual.

```

SAGE
sage: RT = PolynomialRing(CC,"T")
sage: MS = MatrixSpace(GF(2), 4, 8)
sage: G = MS([[1,0,0,0,0,0,1,1,0],[0,1,0,0,1,1,1,0],
              [0,0,1,0,1,1,1,1],[0,0,0,1,0,0,1,0]])
sage: C = LinearCode(G)
sage: C.minimum_distance()
2
sage: Cd = C.dual_code(); Cd.minimum_distance()
2
sage: f = RT(C.chinen_polynomial())
sage: print [z[0].abs() for z in f.roots()]
[0.707106781186549, 0.707106781186547, 0.707106781186547,
 0.707106781186546, 0.707106781186547, 0.707106781186547]
sage: C.gen_mat()
[1 0 0 0 0 1 1 0]
[0 1 0 0 1 1 1 0]
[0 0 1 0 1 1 1 1]
[0 0 0 1 0 0 1 0]
sage: C.chinen_polynomial()
2/7*t^6 + 2/7*t^5 + 11/70*t^4 + 3/35*t^3 + 11/140*t^2 + 1/14*t + 1/28
sage: C.spectrum()
[1, 0, 1, 4, 3, 4, 3, 0, 0]
sage: Cd = C.dual_code(); Cd.minimum_distance()
2
sage: Cd.spectrum()
[1, 0, 1, 4, 3, 4, 3, 0, 0]
sage: list_plot([z[0].real(),z[0].imag() for z in f.roots()])

```

The last command gives a plot of the roots:

Our last example is one for which the RH is false.

```

SAGE
sage: RT = PolynomialRing(CC,"T")
sage: MS = MatrixSpace(GF(2), 4, 8)
sage: G = MS([[1,1,0,0,0,0,1,1],[0,0,1,0,0,1,0,1],[0,0,0,1,0,1,1,0],[0,0,0,0,1,1,1,1]])
sage: C = LinearCode(G)
sage: C.chinen_polynomial()
1/7*t^6 + 1/7*t^5 + 39/140*t^4 + 17/70*t^3 + 39/280*t^2 + 1/28*t + 1/56
sage: C.spectrum()
[1, 0, 0, 4, 6, 4, 0, 0, 1]
sage: Cd = C.dual_code(); Cd.minimum_distance()
2
sage: Cd.spectrum()
[1, 0, 1, 0, 11, 0, 3, 0, 0]
sage: C.minimum_distance()
3
sage: Cd = C.dual_code(); Cd.minimum_distance()
2

```

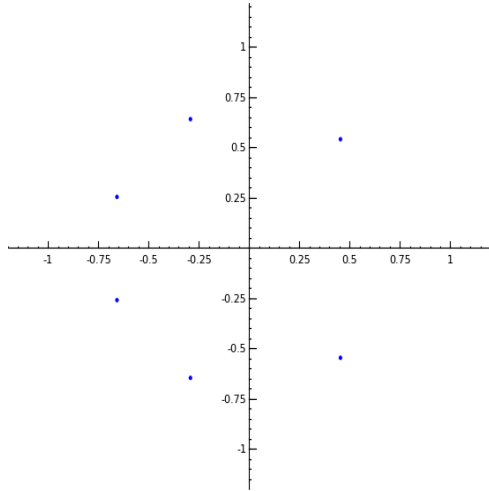


Figure 2: Roots of the Chinen zeta polynomial for a formally self-dual $[8, 4, 2]$ binary code.

```

sage: f = RT(C.chinen_polynomial())
sage: print [z[0].abs() for z in f.roots()]
[1.19773471696883, 1.19773471696883, 0.707106781186547,
 0.707106781186547, 0.417454710894058, 0.417454710894058]
sage: print [z[0] for z in f.roots()]
[0.0528116723604142 + 1.19656983895421*I,
 0.0528116723604137 - 1.19656983895421*I,
 -0.571218487412783 + 0.416784644196318*I,
 -0.571218487412783 - 0.416784644196317*I,
 0.0184068150523700 + 0.417048707955401*I,
 0.0184068150523701 - 0.417048707955401*I]
sage: C.gen_mat()
[1 1 0 0 0 0 1 1]
[0 0 1 0 0 1 0 1]
[0 0 0 1 0 1 1 0]
[0 0 0 0 1 1 1 1]
sage: C.chinen_polynomial()
1/7*t^6 + 1/7*t^5 + 39/140*t^4 + 17/70*t^3 + 39/280*t^2 + 1/28*t + 1/56
sage: list_plot([z[0].real(),z[0].imag()] for z in f.roots())

```

The last command gives a plot of the roots:

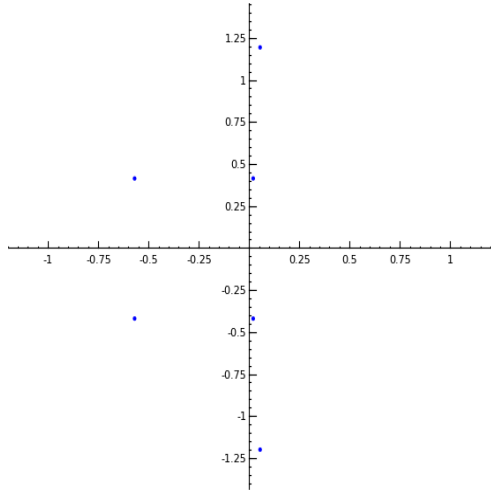


Figure 3: Roots of the Chinen zeta polynomial for a $[8, 4, 3]$ binary code violating the RH.

6 Appendix: Proofs

6.1 MacWilliam's identity

Theorem 51 (*MacWilliams' identity*): *If C is a linear code over $GF(q)$ then*

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

Before proving this we need a few definitions. Index the finite field $GF(q)$ in some fixed way: $GF(q) = \{\omega_0 = 0, \omega_1, \dots, \omega_{q-1}\}$. The **composition** of $v = (v_1, \dots, v_n) \in GF(q)^n$ is defined by

$$\text{comp}(v) = (s_0, \dots, s_{q-1}),$$

where $s_j = s_j(v)$ denotes the number of components of v equal to ω_j . Clearly, $\sum_{i=0}^{q-1} s_i(v) = n$, for each $v \in GF(q)^n$. For $s = (s_0, \dots, s_{q-1}) \in \mathbb{Z}^q$, let $T_C(s)$ denote the number of codewords $c \in C$ with $\text{comp}(c) = s$. Define the *complete weight enumerator* by

$$W_C(z_0, \dots, z_{q-1}) = \sum_{c \in C} z_0^{s_0(c)} \dots z_{q-1}^{s_{q-1}(c)} = \sum_{s \in \mathbb{Z}^q} T_C(s) z_0^{s_0} \dots z_{q-1}^{s_{q-1}}.$$

Sometimes, when it is convenient, we identify the variables z_i with the variables z_{ω_i} . This enumerator is related to the Hamming weight enumerator as follows:

$$A_C(x, y) = W_C(x, y, \dots, y).$$

Let $\{1, \alpha, \alpha^2, \dots, \alpha^{m-1}\}$ denote a power basis of $GF(q)/GF(p)$. Let $\zeta = \zeta_p = e^{2\pi i/p}$ denote a p -th root of unity. If $\beta, \gamma \in GF(q)$ are written $\beta = \beta_0 + \beta_1\alpha + \dots + \beta_{m-1}\alpha^{m-1}$ and $\gamma = \gamma_0 + \gamma_1\alpha + \dots + \gamma_{m-1}\alpha^{m-1}$ ($\beta_i \in GF(p)$, $\gamma_j \in GF(p)$), then we define the character $\chi_\beta : GF(q) \rightarrow \mathbb{C}^\times$ by

$$\chi_\beta(\gamma) = \zeta^{\beta_0\gamma_0 + \dots + \beta_{m-1}\gamma_{m-1}}.$$

Each character χ of $GF(q)$ is of this form, $\chi = \chi_\beta$, for some unique $\beta \in GF(q)$. In particular,

$$\chi_1(\gamma) = \zeta^{\gamma_0},$$

which is for all $\gamma \in GF(q)$ (hence also for $\gamma \in GF(p)$). For $u, v \in GF(q)^n$, define

$$\chi_u(v) = \chi_1(u \cdot v),$$

and define the *Fourier transform* by

$$\hat{f}(u) = \sum_{v \in GF(q)^n} \chi_u(v) f(v),$$

for any function f on $GF(q)^n$. If $u = (u_1, \dots, u_n) \in GF(q)^n$, $v = (v_1, \dots, v_n) \in GF(q)^n$, then $\chi_u(v) = \prod_{i=1}^n \chi_{u_i}(v_i)$. This means that if $f(u) = \prod_{i=1}^n f_i(u_i)$ is a “factorizable” function then

$$\hat{f}(u) = \prod_{i=1}^n \hat{f}_i(u_i).$$

Lemma 52 (*Poisson’s summation formula*) *If C is an $[n, k]$ code over $GF(q)$ then*

$$\sum_{c \in C^\perp} f(c) = \frac{1}{|C|} \sum_{c \in C} f(c).$$

Now we can start with proof of the MacWilliams identity. Define

$$f(u) = z_0^{s_0(u)} \cdots z_{q-1}^{s_{q-1}(u)},$$

so

$$f(u) = z_0^{s_{0,i}} \cdots z_{q-1}^{s_{q-1,i}} = \prod_{i=1}^n f_i(u_i),$$

where $s_{k,i} = 1$ if $u_i = \omega_k$ and $= 0$ otherwise. Another way to define $f(u)$ is as follows:

$$f(u) = \prod_{i=1}^n z_{u_i} = \prod_{i=1}^n f_i(u_i).$$

We have then

$$\begin{aligned} \hat{f}(u) &= \sum_{v \in GF(q)^n} \chi_u(v) f(v) \\ &= \prod_{i=1}^n \hat{f}_i(u_i) \\ &= \prod_{i=1}^n (F_q \cdot (z_0, \dots, z_{q-1}))_i, \end{aligned}$$

where F_q is a $q \times q$ circulant matrix of elements of $GF(q)$ and $(F_q \cdot z)_i$ denotes its i -th component:

$$(F_q \cdot (z_0, \dots, z_{q-1}))_i = \sum_{\omega \in GF(q)} \chi_{\omega_i}(\omega) z_\omega = \sum_{\omega \in GF(q)} \chi_1(\omega_i \omega) z_\omega.$$

Poisson's summation formula implies

$$W_{C^\perp}(z_0, \dots, z_{q-1}) = \frac{1}{|C|} W_C(F_q \cdot (z_0, \dots, z_{q-1})).$$

Let $x = z_0$ and $y = z_1 = \cdots = z_{q-1}$. It remains to note

$$\sum_{b=1}^{q-1} \chi_a(b) = \begin{cases} q-1, & \text{if } a = 0, \\ -1, & \text{if } a \neq 0, \end{cases}$$

□

6.2 Mallows-Sloane-Duursma bounds

We sketch a proof of Theorem 15 following Duursma [D3]. We shall restrict to the Type 1 case for simplicity. (The Type 2 case is similar, but follow modifications as indicated in [D3], §2.) We shall also assume that F contains the term y^n (the analog of the assumption that C contains the all 1's codeword).

Some notation. If $p(x, y) \in \mathbb{C}[x, y]$ then we define

$$p(x, y)(D) = p\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right).$$

If σ is any invertible 2×2 matrix and $(u, v) = (x, y)\sigma$ then

$$p((u, v)\sigma^t)(D)F(u, v) = p(x, y)(D)F((x, y)\sigma). \quad (14)$$

Lemma 53 *Fix a homogeneous function $f(x, y)$. For all i with $1 \leq i \leq e$, let $a_i \neq 0$, $b_i, c_i \neq$, and d_i be complex numbers satisfying $(a_i x + b_i y)^m | (c_i x + d_i y)(D)f(x, y)$, for some integer $m > 1$. Then*

$$\prod_{i=1}^e (a_i x + b_i y)^{m-e+1} | \left(\prod_{i=1}^e c_i x + d_i y \right) (D)f(x, y).$$

Note: This implies the result that Duursma claims in [D3], (5) (where his equation, in his special case, has $m = d^\perp - 1$ and $e = c$).

proof: Using (14), we may assume without loss of generality that $d_i = 0$, for all i , after making a suitable linear change of coordinates. Under the coordinate transformation $z = x/y$, $f(x, y)$ may be regarded as a polynomial $F(z)$ on \mathbb{P}^1 . If $f(x, y) = x^k y^{n-k}$ then $F(z) = z^k$, and an explicit calculation allows one to check that $D_x f(x, y)$ corresponds to $D_z F(z)$.

The hypothesis $(a_i x + b_i y)^m | (c_i x)(D)f(x, y)$ can be rephrased as saying that the derivative of F under has roots of multiplicity m at certain points. Therefore the e -th order derivative of F gives a function which has zeros at these points of order at least $m - e + 1$. \square

Let F be any VSDWE of length n and minimum distance d .

Note that if F is as above then

$$y^{d-1} | y(D)F(x, y). \quad (15)$$

These equations (14) and (15) imply

$$(u - v)^{d-1} | ((q - 1)u - v)(D)F(u, v).$$

Taking $u = x$ and $v = \zeta y$, we have

$$(x - \zeta y)^{d-1} | ((q - 1)x - \zeta y)(D)F(x, y).$$

By Lemma 53, this implies

$$(x^b - y^b)^{d-b} | ((q - 1)^b x^b - y^b)(D)F(x, y). \quad (16)$$

Using equations (15) and (16) and reasoning similar to that used in the proof of Lemma 53, we obtain

$$a(x, y) | p(x, y)(D)F(x, y),$$

where $a(x, y) = y^{d-b-1}(x^b - y^b)^{d-b-1}$ and $p(x, y) = y((q - 1)^b x^b - y^b)$. Comparing highest order terms in y in $p(x, y)(D)F(x, y)$ (recall we assumed F contains y^n), we obtain $d - b - 1 + b(d - b - 1) \leq n - b - 1$. From this, (3) follows in the Type 1 case. This is the first part of Theorem 15. The second part follows from the first using properties of the greatest integer function in a straightforward way. \square

Acknowledgements: I am very grateful to Thann Ward for the reference to [S1], Koji Chinen for many interesting emails about his work, and to Cary Huffman and Iwan Duursma for very interesting conversations on this topic.

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>

- [DH] S. DiPippo, E. Howe, *Real polynomials with all roots on the unit circle and abelian varieties over finite fields*, J. Number Theory 78(1998)426-450.
Available: <http://arxiv.org/abs/math/9803097>
- [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.
- [Dw] B. Dwork, *On the rationality of the zeta function of an algebraic variety*, Amer. J. Math., vol. 82 (1960), pp. 631-648.
- [FR] D. Faifman, Z. Rudnick, *Statistics of the zeros of zeta functions in families of hyperelliptic curves over a finite field*,
Available: <http://front.math.ucdavis.edu/0803.3534>
- [G] The GAP Group, **GAP – Groups, Algorithms, and Programming**, Version 4.4.9; 2006. <http://www.gap-system.org>.
- [GPS] G.-M. Greuel, G. Pfister, and H. Schönemann. **SINGULAR 3.0**. A Computer Algebra System for Polynomial Computations. Centre for Computer Algebra, University of Kaiserslautern (2005). <http://www.singular.uni-kl.de>.

- [HT] T. Harada, M. Tagami, *A Riemann hypothesis analogue for invariant rings*, Discrete Mathematics, vol. 307, pp. 2552-2568, 2007.
- [HP] W. C. Huffman and V. Pless, **Fundamentals of error-correcting codes**, Cambridge Univ. Press, 2003.
- [JKT] D. Joyner, A. Ksir, W. Traves, *Automorphism Groups of Generalized Reed-Solomon Codes* in **Advances in coding theory and cryptology**, (edited by T Shaska, W C Huffman, D Joyner & V Ustimenko) Series on Coding Theory and Cryptology - Vol. 3, World Scientific, 2007.
- [KL] J.-L. Kim and Y. Lee, *Euclidean and Hermitian self-dual MDS codes over large finite fields*, J. Combinatorial Theory, Ser. A, 105 (2004) pp. 79-95.
- [NRS] G. Nebe, E. Rains, N. Sloane, **Self-dual codes and invariant theory**, Springer-Verlag, 2006.
- [O] M. Ozeki, *On the notion of Jacobi polynomials for codes*, Math. Proc. Cambridge Phil. Soc. 121(1997)15-30.
- [S] The SAGE Group, **SAGE : Mathematical software**, version 2.10.
<http://www.sagemath.org/>
- [Sc] W. Schmidt, **Equations over finite fields: an elementary approach**, 2nd ed., Kendrick Press, 2004.
- [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.
- [TV] M. A. Tsfasman and S. G. Vladut, **Algebraic-geometric codes**, Mathematics and its Applications, Kluwer Academic Publishers, Dordrecht 1991.

Index

- binomial moments
 - normalized, 20
 - of a code, 20
- check bit extended code, 26
- Chinen zeta functions, 36
- Chinen zeta polynomial, 36
- code
 - divisible, 4
 - extremal, 5
 - formally equivalent, 3
 - formally self-dual, 3
 - isometric, 3
 - optimal, 5
 - punctured, 26
 - random divisible, 33
 - self-dual, 3
 - shortened, 26
 - Type I s.d., 11
 - Type II s.d., 11
 - Type III s.d., 11
 - Type VI s.d., 11
- composition, 44
- divisible
 - code, 4
 - weight enumerator, 11
- Duursma zeta function, 13
- Fourier transform, 45
- functional equation, 25
- genus of a linear code, 3
- Gleason-Pierce Theorem, 4
- MacWilliams identity, 5, 44
- Mallows-Sloane bounds, 5
- Poisson's summation formula, 45
- Riemann hypothesis, 28, 38
- rising generalized factorial, 31
- spectrum, 3
- support, 3
- Type 1 divisible code, 4
- Type 2 divisible code, 4
- weight enumerator
 - complete, 44
 - divisible, 10
 - extremal formally self-dual, 11
 - genus, 10
 - invariant, 36
 - length, 10
 - minimum distance, 10
 - polynomial (Hamming), 3
 - twisted virtually self-dual, 12
 - virtual MDS, 15
 - virtually self-dual, 10
- zeta function of a code, 20
- zeta polynomial of a code, 13, 18, 21