Coding theory

Introduction to Coding Theory and Algebraic Geometry

Part I -- Coding Theory

Jacobus H. van Lint

1. Finite fields

In this chapter we collect (without proof) the facts from the theory of finite fields that we shall need in this course.

For more details we refer to Lidl and Niederreiter (1983), van Lint (1982), Mac Williams and Sloane (1977), Mc Eliece (1987).

A finite field with q elements is denoted by $\mathbb{F}_q$. (The notation is justified by the fact that two fields with q elements are isomorphic.) The notation $GF(q)$ is also used (Galois field). The easiest examples are the fields with p elements, where p is a prime number.

(1.1) $\mathbb{Z}/p\mathbb{Z}$ is a field if and only if p is a prime number. The other finite fields are residue class rings of $\mathbb{F}_p[x]$.

We observe:

(1.2) If $\mathbb{F}$ is a field then $\mathbb{F}[x]$ is a principal ideal ring.

The principal ideal generated by the polynomial $g(x)$ is denoted by $(g(x))$.

We shall need the following result.

(1.3) If $\mathbb{F}$ is a field then the residue class ring $\mathbb{F}[x]/(x^n-1)$ is a principal ideal ring and every ideal is generated by a divisor of (x^n-1).

For the construction of finite fields other than $\mathbb{F}_p$, we need polynomials $g(x)$ in $\mathbb{F}_p[x]$ that are *irreducible*. By the method of inclusion and exclusion one can show that if I_r denotes the number of monic irreducible polynomials of degree r in $\mathbb{F}_q[x]$, then

$$(1.4) \qquad \sum_{r \mid n} r I_r = q^n.$$

This shows that I_r is positive for all r.

(1.5) If p is prime and $g(x)$ is irreducible of degree r in $\mathbb{F}_p[x]$, then the residue class ring $\mathbb{F}_p[x]/(g(x))$ is a field with p^r elements.

It is an easy exercise to show that if $\mathbb{F}_q$ is a finite field, then q is a power of a prime p and $\mathbb{F}_p$ is a subfield. The number p is called the *characteristic* of the field.

Furthermore $(\mathbb{F}_q, +)$ is isomorphic to $(\mathbb{F}_p)^r$, where $q = p^r$. The multiplicative structure of $\mathbb{F}_q$ is also quite easy:

(1.6) The group $(\mathbb{F}_q \setminus \{0\},)$ is cyclic. A generator of this group is called a *primitive* element of the field.

This shows that the elements of $\mathbb{F}_q$ form the set of all solutions of the equation $x^q - x = 0$ (in the closure of $\mathbb{F}_p$, where $q = p^r$). This fact combined with (1.4) easily leads to the theorem that up to isomorphism there is only one field with q elements ($q = p^r$, p *prime*). However, for some applications in coding theory, the particular representation of the field can make a difference! (cf. Mac Williams and Sloane 1977, Ch. 10 § 5). Furthermore, an easy consequence of (1.6) is:

(1.7) $\mathbb{F}_{p^r}$ is a subfield of $\mathbb{F}_{p^s}$ if and only if r divides s.

A fact, sometimes referred to as the "freshman's dream" is the equation $(a+b)^p = a^p + b^p$ if a and b are elements in a field of characteristic p. A consequence of this fact is $a(x^q) = (a(x))^q$ if $a(x) \in \mathbb{F}_q[x]$. This leads to a result that we use quite often:

(1.8) If $0 \neq f(x) \in \mathbb{F}_q[x]$ and if $f(\alpha) = 0$, where $\alpha \in \mathbb{F}_{q^s}$, then $f(\alpha^q) = 0$.

The converse is also true:

(1.9) If $f(x) \in \mathbb{F}_{q^s}[x]$ is monic and has the property that $f(\alpha^q) = 0$ for every α for which $f(\alpha) = 0$, then $f(x) \in \mathbb{F}_q[x]$.

Let $q = p^r$ and let β be an element of $\mathbb{F}_q$. The *minimal polynomial* $m(x)$ of β is the monic irreducible polynomial in $\mathbb{F}_p[x]$ for which $m(\beta) = 0$. By (1.8) and (1.9) we have

$$m(x) = (x-\beta)(x-\beta^p)(x-\beta^{p^2}) \cdots (x-\beta^{p^{s-1}}),$$

where s is the smallest positive integer such that $\beta^{p^s} = \beta$.

(1.10) Example. Let α be a primitive element of $\mathbb{F}_{2^4}$. Denote the minimal polynomial of α^i by $m_i(x)$. Then

$$x^{15} - 1 = (x-1)\, m_1(x)\, m_3(x)\, m_5(x)\, m_7(x),$$

where e.g. $m_7(x) = (x-\alpha^7)(x-\alpha^{14})(x-\alpha^{13})(x-\alpha^{11})$.

Note that $m_5(x) = x^2 + x + 1$, the unique irreducible polynomial of degree 2 in $\mathbb{F}_2[x]$. Since α^3 is a fifth root of unity we must have $m_3(x) = x^4 + x^3 + x^2 + x + 1$. The other two factors are $x^4 + x + 1$ and $x^4 + x^3 + 1$. Usually one chooses α such that the first of these is $m_1(x)$.

There are two more well known results about polynomials over $\mathbb{F}_q$ that we shall use:

(1.11) If $f(x) \in \mathbb{F}_q[x]$ and α is a zero of $f(x)$ in some extension field of $\mathbb{F}_q$, then α is a multiple zero if and only if it is also a zero of $f'(x)$.

(1.12) If the polynomials $a(x)$ and $b(x)$ in $\mathbb{F}_q[x]$ have greatest common divisor 1, then there are polynomials $p(x)$ and $q(x)$ such that $a(x)\,p(x) + b(x)\,q(x) = 1$.

Finally, we mention the *trace* function Tr: $\mathbb{F}_q \to \mathbb{F}_p$. If $q = p^r$ then for $\xi \in \mathbb{F}_q$

$$Tr(\xi) := \xi + \xi^p + \xi^{p^2} + \cdots + \xi^{p^{r-1}}.$$

Note that by the freshman's dream this is a linear mapping. Since the function is not identically 0, it takes every value the same number of times.

2. Error-correcting codes

We shall not go into details concerning all the technical applications of error-correcting codes. These include satellite pictures, telephone messages via glass fibre using light, compact disc audio system. The idea is as follows. We consider "*information*" presented as a very long sequence of symbols from a finite set called the "*alphabet*". In this course the alphabet will be a finite field $\mathbb{F}_q$. In the sequence each symbol occurs with equal probability. This information is sent to a receiver over a so-called "noisy channel". In the model that we consider there is a fixed (small) probability p_e that a symbol, that is sent over the channel, is changed into one of the other symbols (again, all equally likely). Such an event is called a "*symbol-error*" and p_e is the symbol-error probability. As a result a fraction p_e of the transmitted symbols arrives incorrectly at the receiver end of the channel. The aim of coding theory is to lower the probability of error (considerably) at the expense of spending some of the transmission time or energy on *redundant* symbols. The idea is explained in one sentence as follows. When we read printed text we recognize a printing error in a word because in our vocabulary there is only one word that resembles (is "sufficiently close to") the printed word.

In *block coding* the message is split into parts of say k symbols. The "encoding" is an injective mapping from $\mathbb{F}_q^k$ to $\mathbb{F}_q^n$ (where $n > k$). In $\mathbb{F}_q^n$ we introduce so-called *Hamming-distance*:

$$d(\mathbf{x}, \mathbf{y}) := |\{1 \le i \le n, x_i \neq y_i\}|. \tag{2.1}$$

We define the *minimum distance* of the *code* C (i.e. the image of $\mathbb{F}_q^k$) by

$$d = \min\{d(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in C, \mathbf{y} \in C, \mathbf{x} \neq \mathbf{y}\}. \tag{2.2}$$

If $d = 2e + 1$ then C is an e-error-correcting code because if a received word has distance $\le e$ to some codeword, then it has distance $> e$ to all other codewords.

Note that the toll we pay for the possibility of correcting errors is that we have to send n symbols over the channel to convey k information symbols to the receiver. This is expressed by saying that the code C has *information rate* $R := k/n$. The general definition for any subset C of $\mathbb{F}_q^n$ is $R = n^{-1} \log_q |C|$.

We shall always assume that the receiver uses so-called "*maximum-likelihood decoding*", i.e. a received word is "decoded" into a codeword that is closest (where a choice is made if this is not unique). Subsequently, the inverse of the encoding map yields the original information.

It is fairly obvious that we can make the probability of error after decoding as small as we like if we are willing to transmit at very low information rate. The reason that coding theory is interesting is given by Shannon's famous *channel coding theorem*. To explain this, we need a number called the *capacity* of the channel. This number depends on p_e and the size of the alphabet (i.e. q in our case). It lies between 0 and 1. (If $q = 2$ the capacity is $1 + p_e \log p_e + (1 - p_e) \log(1 - p_e)$, where logarithms are to the base 2.)

The theorem states that for any $\varepsilon > 0$ and for any R less than the capacity there is a code with information rate at least R, for which the probability of incorrect decoding (of a received word) is less than ε.

The reader should realize that we do not specify k or n but only restrict the value of k/n. The "good" code promised by the theorem will have very large *word length* n.

To describe the situation that we are interested in in this course, we need a few more definitions.

(2.3) C is an (n, M, d) code over $\mathbb{F}_q$ if C is a subset of $\mathbb{F}_q^n$ with minimum distance d and $|C| = M$.

(2.4) $A_q(n, d) := \max\{M \mid \text{there exists an } (n, M, d) \text{ code over } \mathbb{F}_q\}$.

A code that achieves the bound of (2.4) is called *optimal*. From Shannon's theorem we know that we should study long codes. However, if the channel has symbol-error probability p_e, then we should expect an average of $p_e n$ errors per received word. To correct these we need minimum distance more than $2p_e n$. So, if we increase n, then d should increase proportionally. We introduce the parameter $\delta := d/n$ and define

(2.5) $\alpha(\delta) := \limsup_{n\to\infty} n^{-1} \log_q A_q(n, \delta n)$.

Note that this function tells us something about the information rate of long codes with $d/n = \delta$. In the section on bounds on codes (Section 6) we shall describe the *Gilbert-Varshamov* bound, a lower bound for $\alpha(\delta)$ proved in 1952/1957 that was not improved until 1982. The new bound is obtained using methods from algebraic geometry and it is the purpose of this course to explain these methods and to give the necessary background on coding theory.

3. Linear codes

From now on we shall only consider linear codes.

(3.1) *Definition.* A q-ary linear code or $[n, k]$ code is a k-dimensional linear subspace of $\mathbb{F}_q^n$.

If the code has minimum distance d we shall write $[n, k, d]$ code. The information rate of such a code is k/n.

A linear code C is often described by a so-called *generator matrix* G. This is a matrix that has as its rows k basis vectors of C; (note that elements of $\mathbb{F}_q^n$ are called vectors or words). If G is a generator matrix for C, then $C = \{\mathbf{a}G \mid \mathbf{a} \in \mathbb{F}_q^k\}$, so encoding is multiplication by G. Since we are only interested in error correction and this does not depend on the order of the symbols in a word, we shall call two codes *equivalent* if one is obtained from the other by some permutation of the coordinate positions. Then we can assume w.l.o.g. that C has a generator matrix in so-called standard form: $G = (I_k P)$, where P is a k by $n-k$ matrix. In this case the first k symbols of a codeword are sometimes called information symbols and the remaining symbols are *parity check symbols*. (This name is due to the (historically) first example: a simple parity check code used on paper tape for computers. Here $q = 2$, $k = 5$, $n = 6$ and every codeword has an even number of ones, i.e. P is a column of ones).

(3.2) *Definition.* The *weight* $w(\mathbf{x})$ of a word is the number of nonzero symbols of $\mathbf{x}$. The *minimum weight* of a code C is the minimum of $w(\mathbf{c})$ over all nonzero codewords $\mathbf{c}$.

Note that for a linear code the minimum distance is equal to the minimum weight.

(3.3) *Definition.* If C is an $[n, k]$ code then we define the *dual code*

$C^\perp$ by

$$C^\perp := \{\mathbf{y} \in \mathbb{F}_q^n \mid \forall_{\mathbf{x} \in C} [\langle \mathbf{x}, \mathbf{y} \rangle = 0]\},$$

where $\langle \mathbf{x}, \mathbf{y} \rangle := \sum_{i=1}^{n} x_i y_i$ is the usual inner product.

Clearly $C^\perp$ is an $[n, n-k]$ code. Of special interest are *self-dual* codes, i.e. codes C for which $C^\perp = C$ (see Section 7).

A generator matrix H for the code $C^\perp$ is called a *parity check matrix* for C. The code C is given by

$$C = \{\mathbf{x} \in \mathbb{F}_q^n \mid \mathbf{x}H^\top = \mathbf{0}\}. \tag{3.4}$$

If $G = (I_k P)$ is in standard form, then $H = (-P^\top I_{n-k})$ is a parity check matrix (because $GH^\top = 0$). For the [6,5] binary single parity check code mentioned above, the equation in (3.4) is $x_1 + x_2 + \cdots + x_6 = 0$, i.e. the equation checks the parity of the received word. Note that this code cannot correct errors but it does detect the occurrence of a single error.

We mention a decoding method that is sometimes used in practice. For high-rate codes it is not too bad. The method is known as *syndrome decoding*. For any $\mathbf{x} \in \mathbb{F}_q^n$ the syndrome is defined as $\mathbf{x}H^\top$. For codewords the syndrome is $\mathbf{0}$. A received vector $\mathbf{x}$ with errors in it can be written as $\mathbf{x} = \mathbf{c} + \mathbf{e}$, where $\mathbf{c}$ is the transmitted word and $\mathbf{e}$ is known as the *error-vector*. If we pick a certain error-vector $\mathbf{e}$ and add it to all the codewords, the result is a coset of C in $\mathbb{F}_q^n$ and all the words in this coset have the same syndrome, namely $\mathbf{e}H^\top$.

This means that any vector in a coset is a candidate for the error-vector of a word in the same coset. By maximum likelihood decoding we should choose this vector so that it has minimum weight.
Decoding now goes as follows. For each coset of C we pick a member of minimal weight (often this element is unique). This is called the *coset leader*. We make a list of these coset leaders and their syndromes. When $\mathbf{x}$ is received, the syndrome is calculated, the leader is found by table lookup and $\mathbf{x}$ is decoded by subtracting the leader from $\mathbf{x}$. If, for example, we use a good binary code C of length 63 with $R = \frac{51}{63} = 0{,}8$, then C has over 2.10^{15} codewords but decoding involves only a check of a list of 4096 syndromes. Since this code can be chosen so that it has $d = 5$ (see 4.7), there are 63 coset leaders of weight 1 and 1953 of weight 2. One could list only these and their syndromes and in those cases that the syndrome of the received word is not in the list, the conclusion could be that more than two errors occurred.

Remark: If we transmit information over a binary symmetric channel with bit-error probability $p_e = 0.01$ using this code (with rate 0.8) we achieve an accuracy (after decoding) corresponding to a bit-error probability $p_e = 0.0005$.

We give one more definition.

(3.5) *Definition.* If C is a q-ary code of length n, then the *extended* code $\overline{C}$ is defined by

$$\overline{C} := \{(c_1, c_2, \cdots, c_n, c_{n+1}) \mid (c_1, c_2, \cdots, c_n) \in C, \sum_{i=1}^{n+1} c_i = 0\}.$$

(The symbol $c_n + 1$ is called *overall* parity check. It is 0 if and only if $c_1 + c_2 + \cdots + c_n = 0$.)

The best known examples of single error-correcting codes are the following codes. Let $n := (q^k - 1)/(q - 1)$. Since any nonzero column vector of length k has $q - 1$ nonzero multiples, it is possible to make a k by n matrix H in which no column is $\mathbf{0}$ and for which no two columns are linearly dependent. This implies that if $\mathbf{x}H^T = \mathbf{0}$, then $\mathbf{x}$ must have weight at least 3. Therefore H is the parity check matrix of a $[n, n-k, 3]$ code that is called a *Hamming* code.

4. Cyclic codes

We now consider linear codes with even more regularity.

(4.1) *Definition.* A linear code C is called *cyclic* if

$$\forall_{(c_0,c_1,\ldots,c_{n-1})\in C}\,[(c_{n-1}, c_0, \cdots, c_{n-2}) \in C].$$

From now on we make the convention $(n, q) = 1$.

To describe cyclic codes algebraically we observe that $\mathbb{F}_q^n$ as vector space is isomorphic to $\mathbb{F}_q[x]/(x^n-1)$, if we ignore the multiplication in this ring. We now identify the word $(a_0, a_1, \cdots, a_{n-1})$ with the corresponding polynomial $a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$. Observe that multiplication by x now is nothing but a cyclic shift of the word. Since a cyclic code is linear by definition, we have:

(4.2) *Theorem.* A linear code C in $\mathbb{F}_q^n$ is cyclic if and only if C is an *ideal* in $\mathbb{F}_q[x]/(x^n-1)$.

By (1.3) a cyclic code is a principal ideal generated by a polynomial $g(x)$, the *generator polynomial*, that divides $x^n - 1$. If $x^n - 1 = f_1(x) f_2(x) \cdots f_r(x)$ is the decomposition of $x^n - 1$ into irreducible factors we have 2^r choices for $g(x)$. (Some of these codes can be equivalent.)

The code M_i^- with generator $(x^n-1)/f_i(x)$ is called an *irreducible* cyclic code. Every cyclic code is a direct sum of irreducible cyclic codes. (This is an example of a well known structure theorem for ideals in semisimple algebras). An irreducible cyclic $[n, k]$ code is isomorphic to $\mathbb{F}_q^k$.

Note that (1.11) and the convention $(n, q) = 1$ ensure that $x^n - 1$ has no multiple zeros. So the factors $f_i(x)$ are distinct.

Let $(x^n-1) = g(x)\, h(x)$ in $\mathbb{F}_q[x]$. If $g(x) = g_0 + g_1 x + \cdots + g_{n-k} x^{n-k}$ and $h(x) = h_0 + h_1 x + \cdots + h_k x^k$, then

$$G = \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k} & & 0 & 0 & \cdots & 0 \\ 0 & g_0 & \cdots & & & g_{n-k} & 0 & \cdots & 0 \\ & & \ddots & & & & \ddots & & \\ & & & \ddots & & & & \ddots & \\ 0 & \cdots & \cdots & & g_0 & g_1 & \cdots & & g_{n-k} \end{bmatrix}$$

is a generator matrix for the code C with generator polynomial $g(x)$ and one easily checks that

$$H = \begin{bmatrix} 0 & 0 & \cdots & 0 & h_k & \cdots & h_0 \\ 0 & \cdots & \cdots & h_k & & h_0 & 0 \\ & & & & & & \\ h_k & & \cdots & h_0 & 0 & \cdots & 0 \end{bmatrix}$$

is a parity check matrix for C. We call $h(x)$ the *check polynomial*. Observe that the code with $h(x)$ as generator polynomial is equivalent to $C^\perp$ (namely: obtained by reversing the order of the n symbols). So $C^\perp$ has generator polynomial $x^k\, h(x^{-1})$. C has dimension n-degree $g(x)$.

Let C be a cyclic code with generator $g(x) = f_1(x) \cdots f_t(x)$. Let β_i be a zero of $f_i(x)$, $1 \le i \le t$. By (1.8) and (1.9) we know all the zeros of $g(x)$. We remind the reader that if β_i lies in the extension field IF_{q^m} of IF_q, then β_i can be interpreted as a column vector in $(IF_q)^m$. Now consider the t by n matrix over IF_{q^m}:

$$H := \begin{bmatrix} 1 & \beta_1 & \beta_1^2 & \cdots & \cdots & \beta_1^{n-1} \\ 1 & \beta_2 & \beta_2^2 & \cdots & \cdots & \beta_2^{n-1} \\ \vdots & \vdots & \vdots & & & \vdots \\ 1 & \beta_t & \beta_t^2 & \cdots & \cdots & \beta_t^{n-1} \end{bmatrix}.$$

This matrix can also be considered as a tm by n matrix over IF_q (where we assume that all β_i are in IF_{q^m}). In a sense H is a parity check matrix for the code C. Indeed $\mathbf{c} = (c_0, c_1, \cdots, c_{n-1})$ is in C if and only if $c_0 + c_1 \beta_i + c_2 \beta_i^2 + \cdots + c_{n-1} \beta_i^{n-1} = 0$ for $1 \le i \le t$, because $\mathbf{c}$ is in C if and only if $c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$ is divisible by $g(x)$. If we interpret H as a matrix over IF_q, then it is possible that the rows are not linearly independent, i.e. a parity check matrix for C can be obtained from H by deleting rows if necessary.

(4.3) *Example.* Let $n := 2^m - 1$ and let β be a primitive element of the field IF_{2^m}. The cyclic code C defined by $C := \{c(x) \mid c(\beta) = 0\}$ has the (binary) m by n parity check matrix $H = (1\ \beta\ \beta^2\ \cdots\ \beta^{n-1})$. Since all the columns of H are different and nonzero, this code is the (binary) $[n, n-m]$ Hamming code defined in Section 3.

We now come to a generalization of Hamming codes, the so-called *BCH* codes (discovered by Bose, Ray Chaudhuri and Hocquenghem).

(4.4) *Definition.* Let β be a primitive n^{th} root of unity in an extension field of IF_q. Let $g(x)$ be the least common multiple of the minimal polynomials of $\beta^l, \beta^{l+1}, \cdots, \beta^{l+t-2}$. The cyclic code of length n over IF_q with generator $g(x)$ is called a *BCH code* with *designed distance t.*

From now on we restrict ourselves to $l = 1$ (narrow-sense *BCH* codes). If $n = q^m - 1$, i.e. β is primitive in IF_{q^m}, the code is called a *primitive BCH* code.

(4.5) *Theorem.* The minimum distance of a *BCH* code with designed distance t is at least t. (This is called the *BCH bound.*)

Proof: As we saw earlier, a word $\mathbf{c} = (c_0, c_1, \cdots, c_{n-1})$ is in the code if and only if it has inner product 0 with every row of the matrix

$$(4.6) \qquad H := \begin{bmatrix} 1 & \beta & \beta^2 & \cdots & \beta^{n-1} \\ 1 & \beta^2 & \beta^4 & \cdots & \beta^{2(n-1)} \\ \vdots & & & & \vdots \\ 1 & \beta^{t-1} & \beta^{2(t-1)} & \cdots & \beta^{(n-1)(t-1)} \end{bmatrix}.$$

Any $t-1$ columns of H form a Vandermonde matrix. Since this matrix has determinant $\neq 0$, the columns are linearly independent. It follows that $\mathbf{c}$ cannot have weight less than t. □

(4.7) *Example.* Let $q = 2$, $n = 63$, β a primitive element of $\mathbb{F}_{2^6}$. We take $g(x) = m_1(x)\, m_3(x)$. By (1.8) $g(x)$ has as zeros β^i, where $i = 1, 2, 4, 8, 16, 32$ or $i = 3, 6, 12, 24, 48, 33$. Since we have four consecutive powers of β among the zeros, the code with generator $g(x)$ has minimum distance at least 5 (in fact it is 5). So, this yields a $[63, 51, 5]$ binary code. (This code was used as an example in Section 3.)

A special case of *BCH* codes is obtained if we take $n = q - 1$.

(4.8) *Definition.* A *Reed-Solomon* code (*RS* code) is a primitive *BCH* code of length $n = q - 1$ over $\mathbb{F}_q$. The generator of an *RS* code has the form $g(x) = \prod_{i=1}^{d-1} (x - \alpha^i)$, where α is primitive in $\mathbb{F}_q$.

By (4.5) this code has minimum distance at least d and by the Singleton bound (6.7) the distance cannot be larger. Therefore *RS* codes are *MDS* codes (also see Section 6), i.e. $[n, n-d+1, d]$ codes.

Sometimes one considers the extended code of length $q = n + 1$. A codeword that gets an overall parity check $c_{n+1} = 0$ has $x = 1$ as a zero, so by (4.5) it has weight at least $d + 1$. It follows that the extended code has minimum distance $d + 1$, i.e. it is also *MDS*. We remark that *RS* codes are used in the compact disc error correcting code.

The original approach of Reed and Solomon was different. We take $n = q$. Number the elements of $\mathbb{F}_q$ as $\alpha_i := \alpha^i (0 \le i \le q-2)$, $\alpha_{q-1} = 0$, where α is primitive.

Let L be a set of polynomials of degree $< k$ in $\mathbb{F}_q[x]$. We define a code C by

$$C := \{(f(\alpha_0), f(\alpha_1), \cdots, f(\alpha_{q-1})) \mid f \in L\}.$$

Since a polynomial cannot have more than $k - 1$ zeros if its degree is less than k, the minimum weight (and hence the minimum distance) of C is at least $n - k + 1$. Since C has dimension k we see that C is an $[n, k, n-k+1]$ code, i.e. *MDS*. It is not difficult to see that this code is equivalent to an extended *RS* code as follows.

Let $f(x) = \sum_{j=0}^{k-1} a_j x^j$ and write $c_i := f(\alpha_i)$, $0 \le i \le q-2$.

Then if $1 \le l \le q - k - 1$ we have

$$\sum_{i=0}^{q-2} c_i (\alpha^l)^i = \sum_{j=0}^{k-1} a_j \sum_{i=0}^{q-2} (\alpha^{l+j})^i = 0,$$

since the inner sum is 0 because $1 \le l + j \le q - 2$. So, by (4.8) $\mathbf{c}$ is in the *RS* code with distance $q - k$.

As a preparation for the codes obtained from algebraic geometry we reformulate the second definition of *RS* codes. Let $\mathbb{P}$ be the projective line over $\mathbb{F}_q$. Let Q be the point (1,0). We consider the space $\mathbb{L}$ of rational functions defined on $\mathbb{P}$ that do not have poles, even if we consider $\mathbb{P}$ over the closure of $\mathbb{F}_q$, except possibly in Q and then with order less than k. Let $P_0, P_1, \cdots, P_{n-1}$ be the points of $\mathbb{P}$ different from Q. Then the code defined above is the set $\{(f(P_0), f(P_1), \cdots, f(P_{n-1})) \mid f \in \mathbb{L}\}$ because the functions in $\mathbb{L}$ clearly have the form $f(x, y) = \frac{a(x,y)}{y^l}$, where $a(x, y)$ is a homogeneous polynomial of degree l, where $l < k$. The points P_i are $P_i = (\alpha^i, 1)$, $0 \le i \le n-2$, $P_{n-1} = (0, 1)$.

We now generalize the idea of these codes a little more. We consider as alphabet $\mathbb{F}_{q^m}$ and take n distinct elements from this field, say $\alpha_1, \alpha_2, \cdots, \alpha_n$. Let $\mathbf{v} = (v_1, v_2, \cdots, v_n)$ be a vector of weight n over $\mathbb{F}_{q^m}$ and write $\mathbf{a} := (\alpha_1, \alpha_2, \cdots, \alpha_n)$.

(4.9) *Definition.* The *generalized Reed-Solomon* code $GRS_k(\mathbf{a}, \mathbf{v})$ has as codewords all $(v_1 f(\alpha_1), v_2 f(\alpha_2), \cdots, v_n f(\alpha_n))$, where f runs through the polynomials of degree $< k$ in $\mathbb{F}_{q^m}[x]$.

By the same arguments as used above, this code is *MDS*. From (4.8) we see that the dual of a Reed-Solomon code is again a Reed-Solomon code. It is not difficult to show that this is also true for generalized Reed-Solomon codes. (Hint: find a suitable basis for the polynomials of degree $< n-1$).

We now return to *BCH* codes. We take the point of view of (2.5) and fix a value of δ. We consider a sequence of primitive *BCH* codes over some fixed field $\mathbb{F}_q$, with wordlength $n_i := q^{m_i} - 1$, where $m_i \to \infty$. We require each code to have minimum distance at least δn_i and denote the information rate of the code with length n_i by R_i. Now we are in for a disappointment! One can prove that $R_i \to 0$ for $i \to \infty$.

So we see that for a given channel (i.e. fixed symbol error probability) one cannot hope to find a good code by looking at long primitive *BCH* codes; (these codes are *bad*, cf. Mac Williams and Sloane (1977) § 9.5).

Luckily *BCH* codes (and hence *RS* codes) also have a nice property, namely that they are easy to decode. We describe an algorithm that is used to decode *BCH* codes. It is a modification due to Massey and others of an algorithm that was designed by Berlekamp. Consider a *BCH* code of length n over $\mathbb{F}_q$ with zeros $\beta^1, \beta^2, \cdots, \beta^{2t}$, where β is a primitive n^{th} root of unity in $\mathbb{F}_{q^m}$. We use the following notation. A codeword $C(x)$ is transmitted and we receive $R(x) = R_0 + R_1 x + \cdots + R_{n-1} x^{n-1}$ and call $E(x) := R(x) - C(x) = E_0 + E_1 x + \cdots + E_{n-1} x^{n-1}$ the error-vector. The set $M := \{i \mid E_i \neq 0\}$ is the set of positions where an error has occurred and we assume that the number of errors $e := |M|$ is $\leq t$. Define

$\sigma(z) := \prod_{i \in M} (1 - \beta^i z)$, the so-called *error-locator* (because there is an error in position s if and only if $\sigma(\beta^{-s}) = 0$).

$\omega(z) := \sum_{i \in M} E_i \beta^i \prod_{j \in M \setminus \{i\}} (1 - \beta^j z)$, the *error-evaluator* (since $E_i = \omega(\beta^{-i}) / \sigma'(\beta^{-i})$).

Clearly $\sigma(z)$ is a polynomial of degree $e \leq t$ and $\omega(z)$ has degree less than e. If we know these polynomials, then we know M (by factoring $\sigma(z)$ or by substituting all possible values of z) and from $\omega(z)$ we can then find the values of the E_i by substituting $z = \beta^{-i}$. We now make a formal calculation.

$$\frac{\omega(z)}{\sigma(z)} = \sum_{i \in M} \frac{E_i \beta^i}{1 - \beta^i z} = \sum_{i \in M} E_i z^{-1} \sum_{l=1}^{\infty} (\beta^i z)^l =$$

$$= \sum_{l=1}^{\infty} z^{l-1} E(\beta^l).$$

The point of the algorithm is that the first $2t$ coefficients on the right-hand side are known, because $E(\beta^l) = R(\beta^l)$ for $1 \leq l \leq 2t$ by definition of the code. So, if we write $S(z) := \sum_{l=1}^{2t} R(\beta^l) z^{l-1}$, we now have to find the unknown polynomials $\sigma(z)$ and $\omega(z)$ about which we know that

(4.10) $\omega(z) \equiv \sigma(z)\, S(z) \pmod{z^{2t}}$.

We now perform Euclid's algorithm to calculate the g.c.d. of $S(z)$ and z^{2t}. This is a very efficient algorithm that involves easily performed calculations. One can show (cf. Mc Eliece 1977, § 8.5) that the first time that we find a remainder of degree less than t we are done. More precisely: the algorithm starts with $0.\, z^{2t} + 1.\, S(z) = S(z)$ and produces a sequence of equations

$$s_n(z).\, z^{2t} + t_n(z).\, S(z) = r_n(z)\,,$$

where the degree of $r_n(z)$ decreases until the g.c.d. is reached. Clearly the pair $r_n(z)$, $t_n(z)$ satisfies the congruence (4.10). When for the first time $r_n(z)$ has degree $< t$, we have found the required pair up to a constant factor (which is determined by the fact that $\sigma(0) = 1$).

5. Classical Goppa codes

Let us recall that in (4.4) a *BCH* code was defined as the set of words $(c_0, c_1, \cdots, c_{n-1}) \in \mathbb{F}_q^n$ such that $c_0 + c_1(\beta^j) + c_2(\beta^j)^2 + \cdots + c_{n-1}(\beta^j)^{n-1} = 0$ where β is a primitive n^{th} root of unity and $1 \le j < d$. Here d is the designed distance. We can rewrite this as follows:

$$(z^n - 1)\sum_{i=0}^{n-1} \frac{c_i}{z - \beta^{-i}} = \sum_{i=0}^{n-1} c_i \sum_{k=0}^{n-1} z^k (\beta^{-i})^{n-1-k} =$$

$$= \sum_{k=0}^{n-1} z^k \sum_{i=0}^{n-1} c_i (\beta^{k+1})^i = z^{d-1} p(z),$$

i.e.

$$\sum_{i=0}^{n-1} \frac{c_i}{z - \beta^{-i}} = \frac{z^{d-1} p(z)}{z^n - 1}, \tag{5.1}$$

for some polynomial $p(z)$ and vice versa, i.e. $(c_0, c_1, \cdots, c_{n-1})$ is in the code if and only if the left-hand side of (5.1) written as a rational function $a(z)/b(z)$ has a numerator divisible by z^{d-1}. We now generalize this as follows.

(5.2) *Definition*: Let $g(z)$ be a monic polynomial over $\mathbb{F}_{q^m}$ and let $L := \{\gamma_0, \gamma_1, \cdots, \gamma_{n-1}\} \subseteq \mathbb{F}_{q^m}$ (here $n = |L|$). We require that $g(\gamma_i) \neq 0$, $0 \le i < n$. The *Goppa code* $\Gamma(L, g)$ with Goppa polynomial $g(z)$ is the set of words $(c_0, c_1, \cdots, c_{n-1})$ in $\mathbb{F}_q^n$ for which

$$\sum_{i=0}^{n-1} \frac{c_i}{z - \gamma_i} \equiv 0 \pmod{g(z)}. \tag{5.3}$$

Here (5.3) means that the numerator of the left-hand side, written as $a(z)/b(z)$, is divisible by $g(z)$. We can also make the convention that

$$\frac{1}{z-\gamma} := \frac{-1}{g(\gamma)} \left[\frac{g(z) - g(\gamma)}{z - \gamma} \right], \tag{5.4}$$

where the right-hand side is the unique polynomial $f(z) \bmod g(z)$ such that $(z-\gamma) f(z) \equiv 1 \pmod{g(z)}$.

From our introduction and (5.1) we see that if we take $g(z) = z^{d-1}$ and $L := \{\beta^{-i} \mid 0 \le i \le n-1\}$, where β is a primitive n^{th} root of unity, then the Goppa code $\Gamma(L, g)$ is the narrow sense *BCH* code of designed distance d. We remark that not all *BCH* codes are also Goppa codes.

We can also interpret (5.2) as follows. Consider the vector space of rational functions $f(z)$ with the following properties:

i) $f(z)$ has zeros in all the points where $g(z)$ has zeros, with at least the same multiplicity;

ii) $f(z)$ has no poles, except possibly in the points $\gamma_0, \gamma_1, \cdots, \gamma_{n-1}$ and then of order 1.

Consider the code over $\mathbb{F}_{q^m}$ consisting of all the words $(\mathrm{Res}_{\gamma_0} f, \mathrm{Res}_{\gamma_1} f, \cdots, \mathrm{Res}_{\gamma_{n-1}} f)$.

The Goppa code $\Gamma(L, g)$ is the "subfield subcode" consisting of all the words in the code with all coordinates in $\mathbb{F}_q$.

We shall now find a parity check matrix for $\Gamma(L, g)$. Let $g(z) = \sum_{i=0}^{t} g_i z^i$. Then $\frac{g(z)-g(x)}{z-x} = \sum_{k+j\le t-1} g_{k+j+1} x^j z^k$, so we have an easy expression for the polynomials on the right-hand side of (5.4). By (5.3) we must have, with $h_j := 1/g(\gamma_j)$,

$$\sum_{i=0}^{n-1} c_i h_i \sum_{k+j\le t-1} g_{k+j+1} (\gamma_i)^j z^k = 0,$$

i.e. the coefficient of z^k is 0 for $0 \le k \le t-1$. We see that **c** must have inner product 0 with the rows of the following matrix.

$$\begin{bmatrix} h_0 g_t & h_1 g_t & \cdots & h_{n-1} g_t \\ h_0(g_{t-1} + g_t \gamma_0) & h_1(g_{t-1} + g_t \gamma_1) & \cdots & h_{n-1}(g_{t-1} + g_t \gamma_{n-1}) \\ \vdots & & & \vdots \\ h_0(g_1 + g_2 \gamma_0 + \cdots + g_t \gamma_0^{t-1}) & \cdots & & h_{n-1}(g_1 + g_2 \gamma_{n-1} + \cdots + g_t \gamma_{n-1}^{t-1}) \end{bmatrix}.$$

Using elementary row operations we then find the following simple parity check matrix for $\Gamma(L, g)$:

$$H = \begin{bmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_0\gamma_0 & h_1\gamma_1 & \cdots & h_{n-1}\gamma_{n-1} \\ \vdots & \vdots & & \vdots \\ h_0\gamma_0^{t-1} & h_1\gamma_1^{t-1} & & h_{n-1}\gamma_{n-1}^{t-1} \end{bmatrix}. \qquad (5.5)$$

Note that if in (4.9) we take $\mathbf{v} := (h_0, h_1, \cdots, h_{n-1})$ and $\mathbf{a} := (\gamma_0, \gamma_1, \cdots, \gamma_{n-1})$, $k = t$, then the code $GRS_k(\mathbf{a}, \mathbf{v})$ has the matrix H of (5.5) as generator matrix. It follows that $\Gamma(L, g)$ is a subfield subcode of the dual of a certain Generalized Reed Solomon code, i.e. $\Gamma(L, g)$ is a subfield subcode of a Generalized Reed Solomon code!

Observe that in (5.5) we can again interpret each row as a set of m rows over $\mathbb{F}_q$. So we find (using (4.9)):

(5.6) *Theorem.* The Goppa code $\Gamma(L, g)$ has dimension $\ge n - mt$ and minimum distance $\ge t + 1$.

The fact that the minimum distance is at least $t+1$ follows directly from the definition (5.3). Since the code is linear, we can consider the weight of **c**. If this is w then the degree of the numerator $a(z)$ of the left-hand side of (5.3) is $w-1$ (in fact less if $\sum_{i=0}^{n-1} c_i = 0$). So $w-1$ is at least t. If $q = 2$ we can say a lot more.

Define $f(z) := \prod_{i=0}^{n-1} (z-\gamma_i)^{c_i}$. Then $\sum_{i=0}^{n-1} \frac{c_i}{z-\gamma_i} = f'(z)/f(z)$.

Since all exponents in $f'(z)$ are even, this is a perfect square. If we assume that $g(z)$ has no multiple zeros, then the fact that $g(z)$ divides $f'(z)$ implies that $g^2(z)$ divides $f'(z)$.

(5.7) *Theorem.* If $g(z)$ has no multiple zeros, then the binary Goppa code $\Gamma(L, g)$ has minimum distance at least $2t + 1$ (where $t :=$ degree $g(z)$).

We shall now show that the set of Goppa codes is a lot nicer than the *BCH* codes by showing that there are good long Goppa codes. (To appreciate what we mean by "good" the reader should first study Section 6.) We choose $n = q^m$, t, d and take $L = \mathbb{F}_{q^m}$. It remains to pick a Goppa polynomial $g(z)$ of degree t over $\mathbb{F}_{q^m}$ that is irreducible and such that $\Gamma(L, g)$ has minimum distance at least d. Suppose $\mathbf{c} = (c_0, c_1, \cdots, c_{n-1})$ is a word of weight $j < d$, i.e. a word that we do not allow in the code. As we saw before, the numerator of $\sum_{i=0}^{n-1} \frac{c_i}{z-\gamma_i}$ has degree $j-1$ and hence at most $\lfloor \frac{j-1}{t} \rfloor$ different polynomials of degree t can divide this numerator. Therefore we have to exclude at most $\sum_{j=1}^{d-1} \binom{n}{j} (q-1)^j \lfloor \frac{j-1}{t} \rfloor$ irreducible polynomials of degree t. This number is less than $\frac{d}{t} V_q(n, d)$ where (as in (6.1)) we use the notation $V_q(n, d) := \sum_{i=0}^{d} \binom{n}{i} (q-1)^i$. It is known that $\lim_{n\to\infty} n^{-1} \log_q V_q(n, \lfloor \delta n \rfloor) = H_q(\delta)$, where H_q is the entropy function (cf. (6.3)). A sufficient condition for the existence of the code we are looking for is that $\frac{d}{t} V_q(n, d)$ is less than the total number of irreducible polynomials of degree t over $\mathbb{F}_{q^m}$, which is known to be $\frac{1}{t} q^{mt}(1+o(1))$. (In fact this follows from (1.4).) So, we find as a sufficient condition (after taking logarithms, $d = \lfloor \delta n \rfloor$, $n \to \infty$):

$$H_q(\delta) + o(1) < \frac{mt}{n} + o(1), \quad (m \to \infty).$$

From (5.6) we know that the codes we are considering have rate $\geq 1 - \frac{mt}{n}$. So we have proved the following theorem.

(5.8) *Theorem.* There exists a sequence of Goppa codes over $\mathbb{F}_q$ that have information rate tending to $1 - H_q(\delta)$, i.e. the rate tends to the Gilbert-Varshamov bound.

We remark that the decoding method that we discussed for *BCH* codes in Section 4 can be generalized to also decode Goppa codes. As in Section 4 we call the received word $R = C + E$. Using a similar notation we define

$$S(z) := \sum_{i=0}^{n-1} \frac{E_i}{z-\gamma_i} \quad \text{(using the convention of (5.4))}.$$

By (5.3) we can calculate $S(z)$ from R. Now we again define an error locator and error evaluator by

$$\sigma(z) := \prod_{i \in M} (z-\gamma_i), \quad \omega(z) := \sum_{i \in M} E_i \prod_{j \in M \setminus \{i\}} (z-\gamma_j).$$

Then clearly

$$S(z)\,\sigma(z) \equiv \omega(z) \pmod{g(z)}$$

and we are again in the situation of (4.10).

6. Bounds on codes

We now return to the problem of finding bounds on codes and the study of the function $\alpha(\delta)$ defined in (2.5). We need a few definitions and lemmas. If we consider the set of words in $\mathbb{F}_q^n$ that have distance at most d to a fixed word, then the cardinality of this set is

$$V_q(n, d) := \sum_{i=0}^{d} \binom{n}{i} (q-1)^i. \tag{6.1}$$

We define the *entropy function* H_q on $[0, \frac{q-1}{q}]$ by

$$H_q(0) := 0, \tag{6.2}$$
$$H_q(x) := x \log_q(q-1) - x \log_q x - (1-x) \log_q(1-x), \ 0 < x < \frac{q-1}{q}.$$

The following lemma can easily be proved using Stirling's formula (cf. van Lint 1982, 5.16):

(6.3) *Lemma.* For $0 \le \delta \le \frac{q-1}{q}$ we have

$$\lim_{n\to\infty} n^{-1} \log_q V_q(n, \lfloor \delta n \rfloor) = H_q(\delta).$$

Suppose C is a code of length n over $\mathbb{F}_q$ with minimum distance d and suppose that it is not possible to find a word not in C that has distance at least d to all codewords in C. Then clearly $|C| V_q(n, d-1) \ge q^n$. This simple argument is the proof of the Gilbert-Varshamov bound.

(6.4) *Theorem.* $A_q(n, d) \ge q^n / V_q(n, d-1)$.

If we take $d = \lfloor \delta n \rfloor$ and use (2.5) and (6.3), then we find the asymptotic Gilbert bound:

(6.5) *Theorem.* $\alpha(\delta) \ge 1 - H_q(\delta)$.

Suppose that we now consider only linear codes in $\mathbb{F}_q^n$. We claim that we find a result just as good as (6.4)!

(6.6) *Theorem:* If $q^n / V_q(n, d-1) > q^{k-1}$ then there exists an $[n, k, d]$ code over $\mathbb{F}_q$.

Proof: For $k = 0$, the assertion is trivial. Suppose the inequality holds and that we have a $[n, k-1, d]$ code C_{k-1}. By the proof of (6.4) there is a word $\mathbf{x} \in \mathbb{F}_q^n$ that has distance at least d to all the words of C_{k-1}. If $a \in \mathbb{F}_q$ and $\mathbf{c} \in C_{k-1}$, then $w(a\mathbf{x} + \mathbf{c}) = w(\mathbf{x} + a^{-1}\mathbf{c}) = d(\mathbf{x}, -a^{-1}\mathbf{c}) \ge d$.
Hence C_{k-1} and $\mathbf{x}$ span a linear code C_k with minimum distance d. □

As we already remarked in Section 2 the lower bound (6.5) was not improved until recently.
The bound that we shall find from algebraic geometry (cf. alg. geom. 5.5) is

$$\alpha(\delta) + \delta \ge (\sqrt{q} - 1)^{-1}. \tag{*}$$

To see whether this improves (6.5) we first calculate the tangent to the curve (6.5) that has the same slope as (*). By differentiating (6.2) we find the equation

$$\log_q(q-1) - \log_q \delta + \log_q(1-\delta) = 1,$$

with solution $\delta_0 = (q-1)/(2q-1)$.
Therefore the line given by (*) intersects the curve (6.5)
if: $1 - H_q(\delta_0) < 1 - (\sqrt{q}-1)^{-1} - \delta_0$, i.e. $1 + (\sqrt{q}-1)^{-1} < \log_q(2q-1)$.
This is true for $q \geq 43$ but since in (*) q must be a square, the smallest value of q for which an improvement of (6.5) is found is $q = 49$.
It was pointed out by Manin (1982) that upper bounds for $A_q(n, \delta)$ could be used to prove theorems on algebraic curves as follows. The equation (*) is true for all q if we replace the right-hand side by $\gamma_q := \liminf g/n$ where we consider curves over $\mathbb{F}_q$ with n rational points ($n \to \infty$) and genus g. However, the line (*) must remain under the known upper bounds. At the time he wrote this, the bound for γ_2 could be improved using the best known upper bound for $A_2(n, \delta)$. At present the best known bounds for γ_q are better than what we can find using coding theory.
For the sake of completeness we now treat a number of upper bounds for $\alpha(\delta)$.

(6.7) *Theorem.* (Singleton bound.) $A_q(n, d) \leq q^{n-d+1}$.

Proof: If C is a code with distance d, then deleting the last $d-1$ coordinates of each word in C yields a code of length $n-d+1$ in which all the words are still different. □

(6.8) *Corollary.* A $[n, k, d]$ code has $d \leq n-k+1$.

Note that the proof of (6.7) implies that if equality holds in (6.8) then on any k positions the codewords take all possible q^k values (i.e. these k positions could be taken as "information positions"). Such a code is usually called a *maximum distance separable* code (*MDS* code). Note that if G is the generator matrix of such an *MDS* code, then any k columns of G are independent. This implies that the dual code has minimum distance at least $k+1$. Therefore this distance is $k+1$ (by (6.8)) and we see that the dual of an *MDS* code is again *MDS*.
One of the best known upper bounds is fairly obvious and asymptotically bad. This is the *sphere-packing bound*:

(6.9) *Theorem*: If $d = 2e+1$ then $A_q(n, d) \leq q^n / V_q(n, e)$.

Proof: The "spheres" of radius e around codewords are disjoint. □

In order to prove a better bound we now consider a code C over $\mathbb{F}_q$ with M words of length n and distance d. We make a list of these words (as a matrix). We number the elements of $\mathbb{F}_q$ from 0 to $q-1$. Consider column i of the matrix. Let the j^{th} symbol of the alphabet occur m_j^i times in this column.
We calculate in two ways the sum of the distances of all ordered pairs of codewords. Taking all pairs of rows we find at least $M(M-1)d$. By looking at the columns we find (using Cauchy-Schwarz):

$$M(M-1)d \leq \sum_{i=1}^{n} \sum_{j=0}^{q-1} m_j^i (M - m_j^i) = \sum_{i=1}^{n} (M^2 - \sum_{j=0}^{q-1} (m_j^i)^2) \leq$$

$$\leq \sum_{i=1}^{n} (M^2 - q^{-1} (\sum_{j=0}^{q-1} m_j^i)^2) = n \frac{q-1}{q} M^2.$$

It follows that $M \le \dfrac{d}{d - n\dfrac{q-1}{q}}$ if $d > n\dfrac{q-1}{q}$.

This does not look very useful because we do not expect d to be so large. However we already have a result for $\alpha(\delta)$ from this inequality, namely

(6.10) $\qquad \alpha(\delta) = 0 \quad \text{for } \dfrac{q-1}{q} \le \delta \le 1.$

To make use of the inequality for smaller values of δ we define the length n' by $n' := \lfloor \dfrac{q(d-1)}{q-1} \rfloor$; note that $n' < n$. We consider the last $n - n'$ symbols of all the codewords. There is a subset of M' codewords ending in the same $n - n'$ symbols, where $M' \ge q^{n'-n} M$.
For this subset the inequality derived above also holds, i.e.

$$q^{n'-n} M \le M' \le \frac{d}{d - n'\dfrac{q-1}{q}} \le d.$$

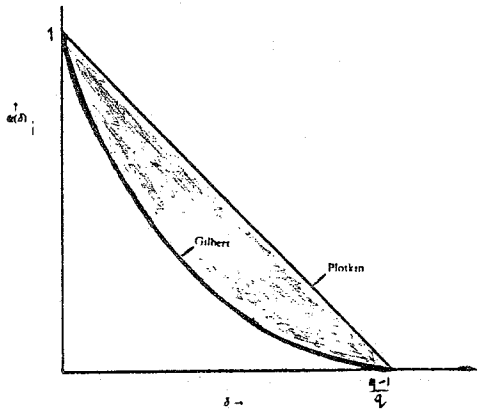
Taking $d = \delta n$, $n \to \infty$ we find the following theorem.

(6.10)*Theorem.* (Plotkin bound)

$$\alpha(\delta) \le 1 - \frac{q\delta}{q-1} \quad \text{for } 0 \le \delta \le \frac{q-1}{q},$$

$$\alpha(\delta) = 0 \quad \text{for } \frac{q-1}{q} \le \delta \le 1.$$

This leaves the shaded region in the following figure for possible values of $\alpha(\delta)$.



We remark that there are several sharper upper bounds than (6.10) (cf. Mac Williams and Sloane 1977).

7. Self-dual codes

A linear code C is called *self-dual* if $C = C^{\perp}$. Clearly the rate of such a code is ½. Many authors have studied such codes and discovered interesting connections with invariant theory and with lattice sphere packings (cf. Mac Williams and Sloane, 1977, Ch. 19). Recently there has been interest in geometric Goppa codes that are self-dual. For examples see alg.geom. § III. ref. [5], [6]. Here we give some theorems about self-dual codes.

A simple example of a self-dual code is the binary extended [8, 4, 4] Hamming code with generator matrix

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

As a second example we consider $q = 2^m$ and then construct the *RS* code C of length $n = q - 1$, dimension $k = \frac{1}{2}q$ and minimum distance $d = n - k + 1$ as in (4.8). The generator polynomial is $\prod_{i=1}^{\frac{1}{2}q-1}(x-\alpha^i)$, where α is primitive in IF_q. As we saw in Section 4, the dual $C^{\perp}$ has generator polynomial $\prod_{i=0}^{\frac{1}{2}q-1}(x-\alpha^i)$.

It follows that $C^{\perp}$ is the subcode of C consisting of all the words $(c_0, c_1, \cdots, c_{n-1})$ in C for which $c_0 + c_1 + \cdots + c_{n-1} = 0$. Therefore the extended code $\bar{C}$ is a $[q, \frac{1}{2}q, \frac{1}{2}q+1]$ self-dual code. Since this is an *MDS* code we should consider it as a good code. It is natural to ask the question whether it is possible to find self-dual codes that are "good" in the asymptotic sense.

This means that, given the fact that we must have rate = ½, we should find out what can be said about d (or better $\delta = d/n$) if $n \to \infty$. The following theorems will provide us with an answer. (We consider only binary codes).

(7.1) *Theorem.* Let C be an $[n, k]$ binary code, where $n = 2t$, $\mathbf{1} \in C$, $C \subset C^{\perp}$. Then the number of $[n, t]$ self-dual codes that contain C is $\prod_{i=1}^{t-k}(2^i+1)$.

Proof. For $k \le m \le t$ we shall count the number a_m of binary $[n, m]$ codes D such that $C \subseteq D \subseteq D^{\perp}$. Clearly $a_k = 1$. If D is such a code $(m < t)$ then $D^{\perp}$ is the union of 2^{n-2m} cosets of D. Since $\mathbf{1} \in C$, each of these cosets contains only vectors of even weight and hence the union of D with any other coset is a code D' of dimension $m+1$ such that $C \subset D' \subset (D')^{\perp}$.

Exactly the same argument applied to C and D' shows that D' contains $2^{m+1-k} - 1$ subcodes of dimension m that also contain C.

Therefore $a_{m+1} = (2^{n-2m} - 1)/(2^{m-k+1} - 1)\, a_m \quad (k \le m < t)$. □

(7.2) *Corollary*:

a) There are $\prod_{i=1}^{\frac{1}{2}n-1}(2^i+1)$ self-dual codes of length n (n even),

b) If $\mathbf{x}$ is a vector of even weight, $\mathbf{x} \notin \{\mathbf{0}, \mathbf{1}\}$, then there are $\prod_{i=1}^{\frac{1}{2}n-2}(2^i+1)$ self-dual codes that contain $\mathbf{x}$.

Proof: a) Every self-dual code contains $\{\mathbf{0}, \mathbf{1}\}$.

b) Apply (7.1) to the code C generated by $\mathbf{1}$ and $\mathbf{x}$.

□

Now suppose n is even and $d = \delta n$. The number of even-weight vectors $\mathbf{x}$ with weight less than d is less than $V_2(n, d)$. Therefore (7.2) implies that a self-dual binary code of length n with minimum distance at least d exists if $V_2(n, d) < (2^{\frac{1}{2}n-1}+1)$. From Lemma 6.3 we see that if $H_2(\delta) < \frac{1}{2}$, then a sequence of such codes with $n \to \infty$ exists. The Gilbert-Varshamov bound (6.5) states that for $H_2(\delta) = ½$ we have $\alpha(\delta) \geq \frac{1}{2}$. We have proved:

(7.3) *Theorem.* There exists a sequence of binary self-dual codes that meets the Gilbert bound.

8. Codes from curves

We use the notation of alg. geom. § III.

1) *Reed-Solomon codes and BCH codes.*

Let β be a primitive n^{th} root of unity in $\mathbb{F}_{q^m}$ (m minimal).

We consider $\mathbb{P}^1 / \mathbb{F}_{q^m}$. Let $P_0 = (0,1), P_\infty = (1,0)$ and define the divisor $D := \sum_{j=1}^{n} P_j$, where $P_j := (\beta^j, 1)$, $1 \le j \le n$. We define the divisor G by $G := a P_0 + b P_\infty$, $a \ge 0$, $b \ge 0$. In this case $L(G)$ consists of the rational functions $\frac{b(x)}{a(x)}$ over $\mathbb{F}_{q^m}$ with degree $a(x) \le a$, degree $b(x) \le b$. It follows that $L(G)$ has dimension $a + b + 1$ and as basis the functions x^i ($-a \le i \le b$). A generator for the code $C(D, G)$ has as rows $(\beta^i \beta^{2i} \cdots \beta^{ni})$ where $-a \le i \le b$. If $(c_1, c_2, \cdots, c_n)$ is a codeword, then $\sum_{j=1}^{n} c_j (\beta^l)^j = 0$ if $a + 1 \le l \le n - b - 1$. Therefore the code is a Reed-Solomon code (in the sense of (4.9)). The subfield subcode found by restriction to $\mathbb{F}_q$ is a *BCH* code with designed distance $n - (a+b)$. This is the bound we also find for the distance of $C(D, G)$.

2) *Codes from Hermitean curves*

We consider the alphabet $\mathbb{F}_q$, where $q = r^2$ (r a power of p). Consider the so-called *hermitean curve* X in $\mathbb{P}^2$ over $\mathbb{F}_q$ given by

(8.1) $$x^{r+1} + y^{r+1} + z^{r+1} = 0.$$

By the Plücker formula (alg.geom. § II. 4) the curve X has genus $g = \frac{r(r-1)}{2}$, i.e. $g = \frac{1}{2}(q - \sqrt{q})$. As an exercise we actually calculate the number of rational points of X. If one of the coordinates is 0, we may take another to be 1 and then we have $r + 1$ solutions for the third coordinate, since $x^{r+1} = 1$ has $r + 1$ solutions in $\mathbb{F}_q$. So, there are $3(r+1)$ points with $xyz = 0$. If $xyz \neq 0$, we may take $z = 1$ and we can choose the value of y^{r+1} to be any element in $\mathbb{F}_r \setminus \{0, 1\}$. This again leaves us with $r + 1$ choices for x. In this way we find $(r-2)(r+1)^2$ solutions. The total number of rational points of X is therefore $1 + q\sqrt{q}$.

Now let Q be the point $(0, 1, 1)$ and define the divisors $G := m Q$, $D :=$ sum of all the other rational points. We take $q - \sqrt{q} < m < q\sqrt{q}$. We find the geometric Goppa code $C(D, G)$ with length $n = q\sqrt{q}$, dimension $k = m - g + 1$ and minimum distance $d \ge n - m = n - k - g + 1$.

Again as an exercise, we treat a very simple example. Take $q = 4$ an write $\mathbb{F}_4 = \{0, 1, \omega, \bar{\omega}\}$ where $\bar{\omega} = \omega^2 = \omega + 1$. In this case $g = 1$. If we take $m = 2$ then $C(D, G)$ has dimension 2. As basis we need two functions belonging to $L(2Q)$. Clearly $f(x, y, z) = 1$ is one of these. We claim that $x/(y+z)$ is another. This follows from the fact that on X

$$\frac{x}{y+z} = \frac{x(y^2+yz+z^2)}{(y+z)(y^2+yz+z^2)} = \frac{y^2+yz+z^2}{x^2},$$

i.e. Q is indeed a pole of order 2 (x is a local parameter in Q).
Substituting the coordinates of the eight rational points $\neq Q$ we find a generator matrix with a row of ones and a second row with each of the elements $0, 1, \omega, \bar{\omega}$ two times. Obviously this code has distance 6 as the theorem promises. If we take $m = 3$, we must find one more basis function, now with a pole of order 3 in Q. We leave it as an exercise to find such a function and to check by hand that d is 5. If we then take $m = 4$ we can add the function $(x/(y+z))^2$ to our basis. The resulting code is an [8, 4, 4] code that is *self-dual*.

Now, let us use the code $C(D, G)$ of length $n = 64$ and rate ½ over $\mathbb{F}_{16}$ for comparison. We assume that we have a very poor channel with $p_e = 0.04$. We compare the code with a Reed-Solomon code over the same alphabet. Since this code has length 16 we shall consider four words of the second code as one message. The code $C(D, G)$ has $m = 37$ and $d = 27$. It is already far better than a *BCH* code over this alphabet (it has $d \leq 18$). The error probability for a word (of 64 letters from $\mathbb{F}_{16}$) for $C(D, G)$ is 2.10^{-7} as compared to 8.10^{-4} for the *RS* code.
We remark that it is easy to find a basis for $L(m\,Q)$, i.e. to find a generator matrix for the code. Consider the functions $f(x, y, z) = \dfrac{x^i y^j}{(y+z)^l}$, where $0 \leq i \leq 4$, $j \geq 0$, $i + j = l$. Using (8.1) with $r = 5$ we can replace $(y+z)^{-l}$ by $(y^4 + y^3 z + y^2 z^2 + y z^3 + z^4)^l x^{-5l}$. Therefore $f(x, y, z)$ has a pole of order $5l - i$ in Q. Clearly these functions are independent. For $5l - i \leq 37$ there are exactly 32 triples (i, j, l) satisfying the conditions.

3) *New bounds for binary codes.*
We consider an example that was considered before (cf. alg. geom. 1.9, 3.12)

Let X be the Klein quartic over $\mathbb{F}_8$ (genus $g = 3$):

(8.2) $\qquad x^3 y + y^3 z + z^3 x = 0.$

The points over $\mathbb{F}_8$ are easily found. Let α be a primitive element satisfying $\alpha^3 + \alpha + 1 = 0$. Clearly the three points (0, 0, 1), (0, 1, 0) and (1, 0, 0) are on X. If $xyz \neq 0$ we take $z = 1$, $y = \alpha^i$ $(0 \leq i \leq 6)$. Writing $x = \alpha^{3i}\,\xi$ we find $\xi^3 + \xi + 1 = 0$, i.e. $\xi \in \{\alpha^i, \alpha^2, \alpha^4\}$. So X has 24 points. Take $Q := (0, 0, 1)$, $G = 10\,Q$, D the sum of the 23 other points. Then the code $C := C(D, G)$ has length 23, distance 23 - *deg* $G = 13$, dimension = $10 - g + 1 = 8$. Since $\mathbb{F}_8 \cong (\mathbb{F}_2)^3$ we can consider codewords in C as 3 by 23 matrices over $\mathbb{F}_2$. We now extend the code by adding a fourth row as "parity row" (making a 4 by 23 matrix over $\mathbb{F}_2$ with column of even weight). It is obvious that we have constructed a binary [92, 24, 26] code.
By leaving out one bit we find a binary [91, 24, 25] code. This example (due to Barg et al 1987) beats the best known code for $n = 91$, $d = 25$.

4) *A geometric MDS code* (example due to R. Pellikaan)
Consider the curve X with equation

(8.3) $$x^2y + \omega y^2 z + \bar{\omega} z^2 x = 0$$

over $\mathbb{F}_4 := \{0, 1, \omega, \bar{\omega}\}$. The curve X has genus 1 and is nonsingular. The nine rational points of X are:

	P_1	P_2	P_3	P_4	P_5	P_6	Q_1	Q_2	Q_3
x	1	0	0	1	1	1	ω	1	1
y	0	1	0	ω	$\bar{\omega}$	1	1	ω	1
z	0	0	1	$\bar{\omega}$	ω	1	1	1	ω

The line $x + y + \bar{\omega} z = 0$ is tangent to X at Q_1 and also intersects X in Q_2. Let $G := 2Q_1 + Q_2$ and $D := P_1 + \cdots + P_6$.

To describe the code $C(D, G)$ we use as basis for $L(G)$ the functions $x/(x+y+\bar{\omega}z)$, $y/(x+y+\bar{\omega}z)$ and $\bar{\omega}z/(x+y+\bar{\omega}z)$. This gives as generator matrix

$$\begin{pmatrix} 1 & 0 & 0 & 1 & \omega & \omega \\ 0 & 1 & 0 & \omega & 1 & \omega \\ 0 & 0 & 1 & \omega & \omega & 1 \end{pmatrix}$$.(Note that the code is equivalent to its dual).

Our bounds show that this [6, 3] code has $d \geq 3$ but in fact this is a [6, 3, 4] code, i.e. it is an *MDS* code. This contradicts an assertion of Driencourt and Michon (1986) that none of the $[q+2, q-1, 4]$ codes is elliptic.

References

- A.M. Barg, S.L. Katsman and M.A. Tsfasman (1987), Algebraic Geometric Codes from Curves of Small Genus, Probl. of Information Transmission **23** (1987), 34-38.

- Y. Driencourt and J.F. Michon (1986), Rapport sur les Codes Géométriques, Univ. Aix-Marseille II et Université Paris 7.

- R. Lidl and H. Niederreiter (1983), Finite Fields, Addison-Wesley, Reading-Mass.

- J.H. van Lint (1982), Introduction to Coding Theory, Springer Verlag, New York.

- F.J. Mac Williams and N.J.A. Sloane (1977), The Theory of Error-Correcting Codes, North Holland, Amsterdam.

- Y.I. Manin (1982), What is the maximal number of points on a curve over $I\!F_2$? J. Fac. Sci. Univ. Tokyo, Sec. Ia, 28, No 3, 715-720.

- R.J. Mc Eliece (1977), The Theory of Information and Coding, Encyclopedia of Math. and its Applic. Vol. 3, Addison-Wesley, Reading-Mass.

- R.J. Mc Eliece (1987), Finite Fields for Computer Scientists and Engineers, Kluwer, Boston.