

NEGACYCLIC CODES FOR THE LEE METRIC

by

Elwyn R. Berlekamp

University of North Carolina

Institute of Statistics Mimeo Series No. 495

November 1966

Supported by National Science Foundation grant number
GP-5790.

To be presented at the
Symposium on Combinatorial
Mathematics and its Applications
April 10-14, 1967

DEPARTMENT OF STATISTICS
UNIVERSITY OF NORTH CAROLINA
Chapel Hill, N. C.

ABSTRACT

This paper presents a new class of codes over $GF(p)$, p a prime ≥ 5 , characterized by the property that if $c_0, c_1, c_2, \dots, c_{N-1}$ is a codeword, then $-c_{N-1}, c_0, c_1, \dots, c_{N-2}$ is also a codeword. These codes are shown to be useful for correcting errors measured in the Lee metric. According to this notion, the weight of a codeword is given by $\sum_{i=0}^{p-1} |c_i|$, where $0 \leq |c_i| \leq (p-1)/2$ and $c_i \equiv \pm c_i \pmod{p}$. For any given $t \leq (p-1)/2$, the principal theorem of this paper exhibits negacycle codes of certain block lengths capable of correcting any error pattern of Lee weight $\leq t$. The proof includes an efficient algebraic decoding algorithm.

After discussing the difficulties in extending this method to the construction of codes with Lee distances $> p$, we show that appropriately chosen low rate negacyclic codes actually have Lee distance $= (p^2-1) p^{m-1}/8$. Like the maximum length shift register cyclic codes, to which they are analagous, these low rate negacyclic codes are equidistant in both the Hamming metric and the Lee metric.

Introduction

One of the central problems in coding theory is the design of error correcting codes of given block length, N , over an alphabet of given size, q , and given information rate, $R = k/N$. Ideally, one would like to find the code with the lowest probability of error for a given memoryless channel. For certain types of highly symmetric channels, this problem can be greatly simplified by the introduction of an appropriate metric to measure the "distance" between different codewords. The most commonly used such metric is the Hamming metric, according to which the distance between two sequences is the number of positions in which they differ. For example, the Hamming distance between the two quintary sequences in Figure 1 is 3:

Figure 1: Two quintary codewords:

Sequence 1: 2,1,0,3,0,4,2,4,3,1

Sequence 2: 2,1,1,3,0,2,1,4,3,1

It is convenient to introduce an arithmetic structure on the alphabet of q symbols. In general, one may use the ring of integers mod q . If q is a prime power, one may instead take the structure to be the Galois Field, $GF(q)$. In either case, it is convenient to associate each sequence of N letters with a polynomial of degree $< N$ over the underlying ring (or field). For example, the two sequences in Figure 1 correspond to the polynomials

$$2 + x + 3x^3 + 4x^5 + 2x^6 + 4x^7 + 3x^8 + 1x^9$$

$$\text{and } 2 + x + x^2 + 3x^3 + 2x^5 + 1x^6 + 4x^7 + 3x^8 + 1x^9$$

whose difference is: $4x^2 + 2x^5 + x^6$

(Here we take the difference mod 5): After defining the Hamming weight of each digit, A, by

$$w_H(A) = \begin{cases} 0 & \text{if } A = 0 \\ 1 & \text{if } A \neq 0 \end{cases}$$

one may then define the Hamming weight of a codeword as the sum of the Hamming weights of the coefficients of the associated polynomial. The Hamming distance between two codewords is then defined as the Hamming weight of the difference. If the channel has the property that all errors are equally likely, then the probability of receiving one sequence when another sequence is sent is a monotonic decreasing function of the Hamming distance from the transmitted sequence to the received sequence. For this reason, the problem of finding a code with a low probability of error is approximated by the problem of finding a code which will correct all patterns of t or fewer errors. This latter problem has proved considerably more tractable than the former. Following the pioneering work of Shannon (1948), Hamming (1950), and Slepian (1956 and 1960); Hocquenghem (1959) and (independently) Bose and Chaudhuri (1960) presented a remarkable class of binary group codes, which have become known as BCH codes. Shortly thereafter, Peterson (1961) presented a decoding algorithm for these codes, and showed them to be a subset of the "cyclic codes" which had been first studied by Prange (1958). Gorenstein and Zierler (1962) then succeeded in generalizing the codes to the non-binary case. Additional properties of these codes were discovered by Mattson and Solomon (1962), and a refined decoding algorithm to include erasures was presented by Forney (1965). In 1966 Berlekamp discovered another decoding algorithm which greatly reduced the complexity of the decoder. Although research still continues on many un-

answered questions about the weight distributions of the codewords, the weight distributions of the coset leaders, the permutation groups which leave the codes invariant, and algorithms to decode more than t errors, the BCH codes have already proved themselves to be of considerable value in some situations. The applications of the BCH codes will no doubt increase as the knowledge of the latest decoding procedures spreads. Research also continues on the more general class of cyclic codes, of which the BCH codes are a proper subset.

All of this work has been based upon Hamming's notion of distance. For some channels, this notion is a reasonable representation of reality. For example, if the letters of the input alphabet correspond to the different members of a set of orthogonal or simplex signals which are transmitted through additive white Gaussian Noise, then all pairs of different letter-letter error transitions are equally likely. For other channels, however, the Hamming metric is a poor choice. For example, the letters of the input alphabet may correspond to different phases of a sinusoidal signal of fixed amplitude and frequency, to which white Gaussian noise is added. Such a channel has an inherent ordering of the letters of the input alphabet, and transitions between adjacent letters are much more likely than transitions between distant input letters. For such a channel, a much more useful notion is the Lee metric. The letters of the alphabet are taken to be the residue classes of integers mod q . The Lee weight of each letter of the alphabet is then defined by

$$w_L(A) = |A|, \text{ where } 0 \leq |A| \leq (q-1)/2 \text{ and either } \begin{array}{l} |A| \equiv A \pmod{q} \\ \text{or} \\ |A| \equiv -A \pmod{q} \end{array}$$

One then defines the Lee weight of a codeword as the sum of the Lee weights of the coefficients of the associated polynomial, and the Lee distance between two codewords as the Lee weight of the difference between the codewords. For example, mod 5 one has $|0| = 0$, $|1| = 1$, $|2| = 2$, $|3| = 2$, $|4| = 1$. The Lee distance between the two codewords in Figure 1 is seen to be $|4| + |2| + |1| = 1 + 2 + 1 = 4$. The reader should have no trouble in verifying that the Lee metric is nonnegative, transitive, and that it satisfies the triangle inequality. If $q = 2$ or $q = 3$, the Hamming metric and the Lee metric are identical; for larger q , these metrics differ.

The Lee metric also proves useful in coding for the amplitude modulated channel, in which the q input letters represent different amplitude levels of the same basic signal to which noise^(*) is added. In many such examples, q may be rather large, say $q = 31$ or $q = 127$. In such cases, the Lee distance between two symbols near the middle of the alphabet is indeed a monotonic decreasing function of the probability of receiving one of them when the other is sent. However, the close Lee distance between the highest and lowest amplitudes belies the very small probability of confusing them. Nevertheless,

(*) We deliberately avoid restrictive statements concerning the noise distribution. The noise distribution most suitable for the Lee metric turns out to be the exponential distribution, according to which the noise assumes an average value between y and $y + dy$ with probability given by $1/2 \exp - |y| dy$. Many noise distributions, including the Gaussian, may be approximated by this distribution, and Lee metric codes can be used on such channels. Although the approximations may seem crude, the Lee metric codes may often be as good as any others which are known!

codes which correct all patterns of not more than t errors in the Lee metric still prove quite useful for such channels, because every sufficiently probable channel error pattern corresponds to an error pattern with low Lee weight. (The converse of this statement, as we have noted, is flagrantly false.) Although the same can be said for the Hamming metric, the approximation is much cruder.

To the author's knowledge, no significant work has been done in coding with the Lee metric except Lee's original paper in 1958. After defining the metric, most of that paper was concerned with assertions on the nonexistence of perfect codes. Rather than pursue that question (which remains unsolved in both the Hamming and Lee metrics), we devote the remainder of this paper to the construction of good algebraic codes and decoding algorithms.

Error location numbers and the error polynomial

For the remainder of this paper, we assume that the channel input alphabet consists of the elements of an odd prime field, $GF(p)$. The reader may at first think that this assumption is overly restrictive, since the Lee metric is defined for input alphabets of arbitrary size. Recall, however, that the Hamming metric is also defined for input alphabets of arbitrary size, yet no one has yet constructed any promising algebraic codes over the alphabet of six letters, nor over any alphabets whose size is not a prime-power. Unlike the Hamming metric, the Lee metric is defined in a modular way which forces the code-constructer to work in the ring of integers mod q , and not in some other structure, such as $GF(q)$. If the size of the input alphabet is not prime, this modular ring is not a field and our constructions fail.

Let us begin by considering the case of single errors. If the block length is N , then there are $2N$ possible single error patterns, since we may have a single Lee error of ± 1 in any of the N positions. Including the all-zero pattern of no errors, we see that there are $2N + 1$ different error patterns of Lee weight ≤ 1 . If we wish to correct all of these error patterns with a linear code containing r parity check digits, then each of these $2N+1$ error patterns must have a different pattern of parity check failures and $2N + 1 \leq p^r$, or $N \leq (p^r - 1)/2$.

Following the essential idea of the constructions of Bose-Chaudhuri and Hocquenghem, we next label each digit of the code with a nonzero element in some extension field of $GF(p)$. The previous inequality suggests that we label each digit of the code with two different error location numbers from $GF(p^r)$. Upon considering the definition of the Lee metric, and our desire to correct an error of ± 1 in each of the N positions, we assign the two location numbers $\pm \alpha^{j-1}$ to the j th digit of the code, where α is a primitive element of $GF(p^r)$. Since $\alpha^N = -1$, no two different positions have a common location number.

Figure 2: GF(25), in terms of α , a root of $X^2 + X + 2$

	Conjugate	Minimum Function	Order
$\alpha^0 = \alpha^{-24} = 0\alpha + 1$	α^0	$X-1$	1
$\alpha^1 = \alpha^{-23} = 1\alpha + 0$	α^5	X^2+X+2	24
$\alpha^2 = \alpha^{-22} = -1\alpha - 2$	α^{10}	X^2-2x-1	12
$\alpha^3 = \alpha^{-21} = -1\alpha + 2$	α^{15}	$X^2 - 2$	8
$\alpha^4 = \alpha^{-20} = -2\alpha + 2$	α^{20}	X^2-X+1	6
$\alpha^5 = \alpha^{-19} = -1\alpha - 1$	α^1	X^2+X+2	24
$\alpha^6 = \alpha^{-18} = 0\alpha + 2$	α^6	$X-2$	4
$\alpha^7 = \alpha^{-17} = 2\alpha + 0$	α^{11}	X^2+2x-2	24
$\alpha^8 = \alpha^{-16} = -2\alpha + 1$	α^{16}	X^2+X+1	3
$\alpha^9 = \alpha^{-15} = -2\alpha - 1$	α^{21}	$X^2 + 2$	8
$\alpha^{10} = \alpha^{-14} = 1\alpha - 1$	α^2	X^2-2x-1	12
$\alpha^{11} = \alpha^{-13} = -2\alpha - 2$	α^7	X^2+2x-2	24
$\alpha^{12} = \alpha^{-12} = 0\alpha - 1$	α^{12}	$X+1$	2
$\alpha^{13} = \alpha^{-11} = -1\alpha + 0$	α^{17}	X^2-X+2	24
$\alpha^{14} = \alpha^{-10} = 1\alpha + 2$	α^{22}	X^2+2x-1	12
$\alpha^{15} = \alpha^{-9} = 1\alpha - 2$	α^3	$X^2 - 2$	8
$\alpha^{16} = \alpha^{-8} = 2\alpha - 2$	α^8	X^2+X+1	3
$\alpha^{17} = \alpha^{-7} = 1\alpha + 1$	α^{13}	X^2-X+2	24
$\alpha^{18} = \alpha^{-6} = 0\alpha - 2$	α^{18}	$X+2$	4
$\alpha^{19} = \alpha^{-5} = -2\alpha + 0$	α^{23}	X^2-2x-2	24
$\alpha^{20} = \alpha^{-4} = 2\alpha - 1$	α^4	X^2-X+1	6
$\alpha^{21} = \alpha^{-3} = 2\alpha + 1$	α^9	$X^2 + 2$	8
$\alpha^{22} = \alpha^{-2} = -1\alpha + 1$	α^{14}	X^2+2x-1	12
$\alpha^{23} = \alpha^{-1} = 2\alpha + 2$	α^{19}	X^2-2x-2	24

Factorable quadratics over GF(5):

$$X^2 + 1 = (X + 2)(X - 2)$$

$$X^2 - 1 = (X + 1)(X - 1)$$

$$X^2 + X - 1 = (X + 2)^2$$

$$X^2 + 2X + 1 = (X + 1)^2$$

$$X^2 + 2X + 2 = (X + 1)(X + 2)$$

$$X^2 + X - 2 = (X + 1)(X + 2)$$

As one of the simplest nontrivial examples, we take $p = 5$, $r = 2$, $N = (5^2 - 1)/2 = 12$. The 24 nonzero elements of $GF(25)$ may be represented as the successive powers of α , where α is a root of the primitive quadratic $X^2 + X + 2$. Alternatively, every element of $GF(25)$ may be represented in the form $A_1\alpha + A_0$, where A_1 and $A_0 \in GF(5)$. The relations between these representations are given in Figure 2. We label the twelve digits of the code as follows:

Code digit position:	1	2	3	4	5	6	7	8	9	10	11	12
Positive location number:	α^0	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}
	0	1	-1	-1	-2	-1	0	2	-2	-2	+1	-2
	1	0	-2	2	2	-1	2	0	+1	-1	-1	-2
Negative location number:	α^{12}	α^{13}	α^{14}	α^{15}	α^{16}	α^{17}	α^{18}	α^{19}	α^{20}	α^{21}	α^{22}	α^{23}
	0	-1	1	1	2	1	0	-2	2	2	-1	2
	-1	0	2	-2	-2	1	-2	0	-1	1	1	2

We now claim that any error pattern of Lee weight t may be specified by giving the locations of the t different errors, or by giving the polynomial, $\sigma(z)$, whose reciprocal roots are these errors. As examples, we consider these error patterns:

$e(X) = X^4$	$X_1 = \alpha^4$	$\sigma(z) = 1 - \alpha^4 z$
$e(X) = -X^8$	$X_1 = \alpha^8 = \alpha^{20}$	$\sigma(z) = 1 - \alpha^{20} z$
$e(X) = X^3 + X^7$	$X_1 = \alpha^3, X_2 = \alpha^7$	$\sigma(z) = (1 - \alpha^3 z)(1 - \alpha^7 z)$
$e(X) = X^5 - X^9$	$X_1 = \alpha^5, X_2 = -\alpha^9 = \alpha^{21}$	$\sigma(z) = (1 - \alpha^5 z)(1 - \alpha^{21} z)$
$e(X) = 2X^6$	$X_1 = \alpha^6, X_2 = \alpha^6$	$\sigma(z) = (1 - \alpha^6 z)^2$
$e(X) = -X^7 - 2X^{10}$	$X_1 = -\alpha^7 = \alpha^{19}, X_2 = X_5 = -\alpha^{10} = \alpha^{22}$	$\sigma(z) = (1 - \alpha^{19} z)(1 - \alpha^{22} z)^2$

The crucial property of the error location numbers X_1, X_2, \dots is that

$$e(\alpha^j) = \sum_1 X_i^j = S_j \text{ for all odd } j.$$

The reader who is familiar with the Gorenstein-Zierler Hamming-metric extension of the BCH codes must be warned that no error values are used in our present formulation. A multiple error in some position is evidenced by a multiple root of the error polynomial. It is clear that

Each distinct error pattern of Lee weight t corresponds to a distinct error polynomial, $\sigma(z)$, whose degree is t .

With appropriate restrictions, the converse is also true:

An error polynomial, $\sigma(z)$ of degree t , corresponds to an error pattern of Lee weight t iff all reciprocal roots of $\sigma(z)$ are $2N$ th roots of unity, and no root has multiplicity greater than $(p-1)/2$, and no two reciprocal roots of $\sigma(z)$ sum to 0.

Double error correcting codes

Let us now construct a code of block length 12 over $GF(5^2)$ which corrects double errors in the Lee metric. Following the BCH argument in a heuristic manner, we take the first two rows of the parity check matrix as the positive digit location numbers, and the second two rows as the cubes of the first two rows:

$$\begin{bmatrix} 0 & 1 & -1 & -1 & -2 & -1 & 0 & 2 & -2 & -2 & 1 & -2 \\ 1 & 0 & -2 & 2 & 2 & -1 & 2 & 0 & 1 & -1 & -1 & -2 \\ 0 & -1 & 0 & -2 & 0 & 1 & 0 & 2 & 0 & -1 & 0 & -2 \\ 1 & 2 & 2 & -1 & -1 & -2 & -2 & 1 & 1 & 2 & 2 & -1 \end{bmatrix} = H$$

The codewords are chosen to satisfy all four of these parity check equations. A codeword is transmitted and noise is added. From the first two parity check equations (the top two rows of the above matrix), the decoder may deduce the sum of the error locations, $S_1 = \sum X_i$; from the bottom two rows of the above equations, the decoder may deduce the sum of the cubes of the error locations, $S_3 = \sum X_i^3$. If there are no more than two errors, we have

$$S_1 = X_1 + X_2 \neq 0 \text{ unless } X_1 = X_2 = 0$$

$$S_3 = X_1^3 + X_2^3$$

$$\frac{S_3}{S_1} = X_1^2 - X_1 X_2 + X_2^2$$

$$\frac{S_3}{S_1} - S_1^2 = -3X_1 X_2$$

$$X_1 X_2 = \frac{S_1^3 - S_3}{3S_1}$$

$$\sigma(z) = 1 - S_1 z + \left(\frac{S_1^3 - S_3}{3S_1} \right) z^2$$

Thus, this code is capable of correcting double errors. To decode, we compute S_1 and S_3 from the parity check equations, and then perform the necessary arithmetic calculations in $GF(5^2)$ to find the error polynomial. If α^j is a reciprocal root of this polynomial, and $0 \leq j < N$, then there is an error of +1 in the $(j+1)$ st digit of the received word. If α^j is a reciprocal root of the error polynomial,

and $N \leq j < 2N$, then there is an error of -1 in the $(j + 1 - N)$ th digit of the received word. A double error in any position of the code manifests itself in a double reciprocal root of the error polynomial. For these double error correcting codes, the quadratic error polynomial

$$1 - S_1 z + \frac{(S_1^3 - S_3)}{3S_1} z^2$$

has repeated roots iff

$$S_1^2 = 4 \frac{(S_1^3 - S_3)}{3S_1} \quad \text{or} \quad 4S_3 = S_1^3$$

This condition also follows from the equations $S_1 = 2X_1$ and $S_3 = 2X_1^3$. Similarly, it may be seen that there is one error only if

$$\frac{S_1^3 - S_3}{3S_1} = 0 \quad \text{and zero errors only if } S_1 = 0.$$

The reader may wonder why we selected the bottom two rows of the parity check matrix to be the cubes of the first two rows, rather than the squares. If instead one selects the squares, then one does not get the desired equations:

$$X_1 + X_2 = S_1$$

$$X_1^2 + X_2^2 = S_2$$

but instead one gets the formidable-looking equations:

$$X_1 + X_2 = S_1$$

$$X_1 |X_1| + X_2 |X_2| = S_2$$

Here $X_i = \pm X_{i'}$, accordingly as $\log_{\alpha} X_i$ is between 0 and $N-1$, or between N and $2N-1$. The difficulty arises because $(+X_i)^2 = (-X_i)^2 \neq -(X_i)^2$. A similar problem arises if one includes any even power of the error location numbers as rows of the generator matrix. Although this may not necessarily represent a bad choice, it results in the formidable-looking equations which we prefer to avoid.

Our actual choice of the H matrix has another nice mathematical property which we shall now investigate. Since the successive columns of the first two rows of this matrix represent successive powers of α , the codeword polynomial, $c(x)$, of degree < 12 , satisfies these parity check equations iff $c(\alpha) = 0$. This can happen iff the code polynomial, $c(x)$ is a multiple of the minimum function of α , which is $x^2 + x + 2$. Similarly, the code polynomial satisfies the last two parity checks iff $c(\alpha^3) = 0$, which can happen iff $c(x)$ is a multiple of the minimum function of α^3 , which is $x^2 - 2$. Evidently, then, the polynomial $c(x)$ of degree < 12 represents a codeword iff $c(x)$ is a multiple of the product, $(x^2+x+2)(x^2-2)$. Since α is a primitive element of $GF(25)$, $\alpha^{24} = 1$ but $\alpha^k \neq 1$ for any k less than 24. Since $(\alpha^{12})^2 = 1$, and the only two square roots of 1 are ± 1 , it is evident that $\alpha^{12} = -1$. A similar argument reveals that $(\alpha^j)^{12} = -1$ for every odd j , including $j = 3$. Thus, both the minimum function of α (namely $x^2 + x + 2$) and the minimum function of α^3 (namely $x^2 - 2$) must be divisors of $x^{12} + 1$. Therefore, if $c(x)$ is a codeword, represented by

$$c(x) = c_0 + c_1 x + c_2 x^2 + \dots + c_{10} x^{10} + c_{11} x^{11},$$

then $c(x)$ is multiple of $(x^2 + x + 2)(x^2 - 2)$, and so is $xc(x) - c_{11}(x^{12} + 1)$

which is $-c_{11} + c_0 x + c_1 x^2 + \dots + c_9 x^{10} + c_{10} x^{11}$.

For this reason, we call this code a negacyclic code.

Negacyclic Codes

In general, we define

A Negacyclic code of block length N over $GF(p)$, (p an odd prime and N a nonmultiple of p) is the set of polynomials in an ideal of polynomials modulo $X^N + 1$ over $GF(p)$.

The monic polynomial of lowest degree in this ideal is called the generator polynomial, $g(x)$, and the quotient $(x^N + 1)/g(x)$ is called the check polynomial, $h(x)$.

Since $x^N + 1 = (x^{2N} - 1)/(x^N - 1)$, the roots of $x^N + 1$ are the roots of $x^{2N} - 1$ which are not roots of $x^N - 1$. If α is a primitive root of $x^{2N} - 1$, then the even powers of α are roots of $x^N - 1$ and the odd powers of α are roots of $x^N + 1$. We repeat, the roots of $x^N + 1$ are the odd powers of a primitive $2N$ th root of unity. Therefore, both the generator polynomial and the check polynomial of any negacyclic code of block length N may be conveniently described in terms of their roots, which are odd powers of a single primitive $2N$ th root of unity. For the double-error correcting code of block length 12 over $GF(5)$, which we discussed in the previous section, the roots of the generator polynomial are α, α^3 , and their quintary conjugates, α^5 and $(\alpha^3)^5 = \alpha^{15}$. The other eight odd powers of α , namely $\alpha^7, \alpha^9, \alpha^{11}, \alpha^{13}, \alpha^{17}, \alpha^{19}, \alpha^{21}$, and α^{23} are roots of the check polynomial.

This method of describing negacyclic codes is important because it immediately reveals the equations which must be solved in order to find

the error polynomial. If α^j is a root of the generator polynomial, then the decoder may immediately compute the sum of the j th powers of the error locations by evaluating the received polynomial, $r(x) = c(x) + e(x)$ at $x = \alpha^j$, obtaining $r(\alpha^j) = 0 + e(\alpha^j) = \sum_i X_i^j = S_j$. Given these S_j , the decoder must then attempt to solve for the error polynomial, $\sigma(z)$, whose reciprocal roots give the error locations. The relation between $\sigma(z)$ and the S 's is given by Newton's identities, which may be readily derived as follows:

$$\text{Let } \sigma(z) = \prod_i (1 - X_i z) ; X_i \text{ not necessarily distinct}$$

$$\text{Then } \sigma'(z) = -\sum_i X_i \prod_{j \neq i} (1 - X_j z)$$

$$\begin{aligned} \text{and } \frac{-z \sigma'(z)}{\sigma(z)} &= \sum_i \frac{X_i z}{1 - X_i z} = \sum_i \sum_{k=1}^{\infty} (X_i z)^k \\ &= \sum_{k=1}^{\infty} \left(\sum_i X_i^k \right) z^k = \sum_{k=1}^{\infty} S_k z^k = S(z) \end{aligned}$$

We thus have Newton's Identities in generating function notation,

$$\boxed{S\sigma + z\sigma' = 0}$$

In negacyclic codes, all of the coefficients of the even powers of z in the generating function for S are initially unknown. For this reason, it is helpful to eliminate these terms from the equations by separating Newton's Identities into their even and odd parts:

$$\text{Letting } \hat{\sigma} = 1 + \sigma_2 z^2 + \sigma_4 z^4 + \dots$$

$$\check{\sigma} = \sigma_1 z + \sigma_3 z^3 + \dots$$

$$\hat{S} = S_2 z^2 + S_4 z^4 + \dots$$

$$\tilde{S} = S_1 z + S_3 z^3 + \dots$$

Newton's Identities become

$$(\hat{S} + \tilde{S}) (\tilde{\sigma} + \hat{\sigma}) + z(\tilde{\sigma} + \hat{\sigma})' = 0$$

which may be broken up into the two equations

$$\hat{S} \hat{\sigma} + \tilde{S} \tilde{\sigma} + z \hat{\sigma}' = 0$$

and
$$\hat{S} \tilde{\sigma} + \tilde{S} \hat{\sigma} + z \tilde{\sigma}' = 0$$

Subtracting $\hat{\sigma}$ times the latter equation from $\tilde{\sigma}$ times the former gives

$$\tilde{S} (\tilde{\sigma}^2 - \hat{\sigma}^2) + z(\tilde{\sigma} \hat{\sigma}' - \hat{\sigma} \tilde{\sigma}') = 0$$

Under sufficiently restrictive circumstances, these equations may be solved.

The major result is the following theorem:

If the roots of the generator polynomial of a negacyclic code over $GF(p)$ include $\alpha, \alpha^3, \alpha^5, \dots, \alpha^{2t-1}$, where $2t-1 < p$, then that negacyclic code is capable of correcting all error patterns of Lee weight $\leq t$.

Remarks: Before proving this theorem, we give a short table of parameters of some of the codes which this theorem promises. The codes which have relatively large t and a relatively small number of check digits, r , usually come in block lengths of the form $N = (p^m - 1)/2$. Nevertheless, there are also moderately good codes of this type having other block lengths; these codes are marked in the table by (*).

Error Correction:		1,	2,	3,	4,	5,			
p	N	r							
5	2	1,	2						
5	6*	2,	3						
5	12	2,	4						
5	62	3,	6						
5	312	4,	8						
7	3	1,	2,	3					
7	24	2,	4,	6					
7	171	3,	6,	9					
11	5	1,	2,	3,	4,	5			
11	15*	2,	3,	5,	7,	8			
11	60	2,	4,	6,	8,	10			
11	665	3,	6,	9,	12,	15			
17	8	1,	2,	3,	4,	5,	6,	7,	8
17	24*	2,	3,	5,	7,	8,	10,	12,	13
17	72*	2,	4,	6,	8,	9,	11,	13,	15
17	144	2,	4,	6,	8,	10,	12,	14,	16
127	63	1,	2,	3,	4,	...	62,	63	
127	8064	2,	4,	6,	8,	...	124,	126	

Proof Our proof is constructive; it consists of an efficient decoding procedure.

We begin with the equation

$$\tilde{S} (\hat{\sigma}^2 - \check{\sigma}^2) = z (\hat{\sigma} \check{\sigma}' - \check{\sigma} \hat{\sigma}')$$

Since $\hat{\sigma}(0) = 1$, we may divide through by $\hat{\sigma}^2$ to obtain

$$\tilde{S} \left(\left(\frac{\check{\sigma}}{\hat{\sigma}} \right)^2 - 1 \right) = z \left(\frac{\hat{\sigma} \check{\sigma}' - \check{\sigma} \hat{\sigma}'}{\hat{\sigma}^2} \right) = z \left(\frac{\check{\sigma}'}{\hat{\sigma}} \right)'$$

Introducing the generating function

$$R = \sum_{q \geq 1} r_q z^q, \quad \text{we have}$$

$$\tilde{S} (R^2 - 1) = z R'$$

$$\text{or } R = \int \frac{1}{z} \tilde{S} (-1 + R^2)$$

Although this equation may look formidable, its solution is, in fact, trivial, because each coefficient of R is specified in terms of certain coefficients of \tilde{S} and previously computed coefficients of R :

$$(R_1 z + R_3 z^3 + R_5 z^5 + \dots) = \int \frac{(S_1 z + S_3 z^3 + S_5 z^5 + \dots)}{z} (-1 + (R_1 z + R_3 z^3 + \dots)^2)$$

$$R_1 = -S_1$$

$$R_3 = \frac{-S_3 + R_1^2 S_1}{3}$$

$$R_5 = \frac{-S_5 + R_1^2 S_3 + 2 R_1 R_3 S_1}{5}$$

⋮

Difficulty is encountered if we attempt to compute the coefficient R_p , since this would require dividing by the integer p , which is zero in $\text{GF}(p)$.

However, under the hypotheses of the theorem, $2t-1 < p$, and no such difficulty arises if we compute only $R_1, R_3, R_5, \dots, R_{2t-1}$.

Since R is an odd function of z , we may define the generating function T by the equation $T(z^2) = (1 + zR(z))^{-1} - 1$. It is evident that $T(0) = 0$, and that $(1+T(z^2)) = (1+zR(z))^{-1}$. Knowing the coefficients of $R_1, R_3, \dots, R_{2t-1}$, this equation enables us to compute recursively the coefficients T_1, T_2, \dots, T_t . Since $R = \frac{\check{\sigma}}{\hat{\sigma}}$ and $1+zR = \frac{\hat{\sigma} + z\check{\sigma}}{\hat{\sigma}}$

we define the polynomials

$$\omega(z^2) = \hat{\sigma}(z); \quad \xi(z^2) = \hat{\sigma}(z) + z\check{\sigma}(z)$$

It is evident that

$$(1+zR(z)) = \xi(z^2)/\omega(z^2)$$

and that

$$(1+T(z^2)) = \omega(z^2)/\xi(z^2)$$

so

$$(1 + T(z)) \xi(z) \equiv \omega(z) \pmod{z^{t+1}}$$

We further claim that if there are no more than t errors, then

ξ and ω satisfy the additional conditions

$\xi(0) = \omega(0) = 1$, $\deg \xi \leq \frac{1+t}{2}$, $\deg \omega \leq \frac{t}{2}$, and ξ and ω are relatively prime.

The last observation follows from the fact that no two reciprocal roots of σ may sum to zero. Therefore, σ may have no even factors of positive degree. This implies that $\check{\sigma}$ and $\hat{\sigma}$ are relatively prime, as are ξ and ω .

In view of these conditions, the equation

$$(1 + T) \xi \equiv \omega \pmod{z^{t+1}}$$

may be solved for ξ and ω by the iterative algorithm for decoding nonbinary BCH codes given by Berlekamp (1968). The solution is evidently given by

$$\xi = \xi^{(t)}, \quad \omega = \omega^{(t)}$$

where $\xi^{(t)}$ and $\omega^{(t)}$ are generated by the algorithm.

We may summarize the decoding procedure as follows:

- 1) Compute $R \pmod{z^{2t}}$ from the equation

$$R = \int \frac{S}{z} (R^2 - 1)$$

- 2) Compute $(1+T) \pmod{z^{t+1}}$ from the equation

$$(1 + T(z^2)) = (1 + z R(z))^{-1}$$

- 3) Use the iterative algorithm for decoding nonbinary BCH codes to find $\xi^{(0)}, \omega^{(0)}, T^{(0)}, \gamma^{(0)}, \xi^{(1)}, \omega^{(1)}, T^{(1)}, \gamma^{(1)}, \dots, \xi^{(t)}, \omega^{(t)}$,

which solve the equations, $(1+T) \xi^{(t)} \equiv \omega^{(t)} \pmod{z^{t+1}}$

- 4) Set $\hat{\sigma}(z) = \omega^{(t)}(z^2)$; $\check{\sigma}(z) = \frac{\xi^{(t)}(z^2) - \omega^{(t)}(z^2)}{z}$; $\sigma = \hat{\sigma} + \check{\sigma}$

- 5) Using a multiple Chien (1964) search, evaluate the polynomials $\hat{\sigma}(\alpha^{-j})$ and $\check{\sigma}(\alpha^{-j})$ for $j = 0, 1, \dots, N-1$, where α is the primitive $2N$ th root of unity whose successive powers give the code's successive location numbers. If $\hat{\sigma}(\alpha^{-j}) = -\check{\sigma}(\alpha^{-j})$, there is a positive error in the code's $(j-1)$ st position; if $\hat{\sigma}(\alpha^{-j}) = +\check{\sigma}(\alpha^{-j})$, there is a negative error in the code's $(j-1)$ st position. In either case, the multiplicity of the error may be determined by evaluating the derivatives of $\hat{\sigma}$ and $\check{\sigma}$.

If $\frac{d^{(i)} \hat{\sigma}}{dz^{(t)}}(\alpha^0) = \pm \frac{d \check{\sigma}^{(2)}(\alpha^j)}{dz^{(t)}}$ for $i < k$, but $\frac{d \hat{\sigma}^{(k)}}{dz^{(k)}} \neq \pm \frac{d \check{\sigma}^{(k)}(\alpha^j)}{dz^{(k)}}$

then the error has multiplicity k . In these equations, the negative signs are used to determine the multiplicity of a positive error, and vice versa.

If high speed is required and computing registers are plentiful, then one may perform multiple Chien searches on all of the polynomials $\hat{\sigma}, \check{\sigma}, \hat{\sigma}', \check{\sigma}', \hat{\sigma}'', \check{\sigma}'', \dots, \frac{d^{(t-1)} \check{\sigma}}{dz^{(t-1)}}$.

On the other hand, if computing registers are scarce but time is available, one should perform Chien searches on only polynomials $\hat{\sigma}$ and $\check{\sigma}$. By judicious programming it is possible to calculate the derivatives only when they are needed, without interrupting the Chien searches on $\hat{\sigma}(\alpha^{-j})$ and $\check{\sigma}(\alpha^{-j})$.

Although this theorem gives us a powerful class of negacyclic codes, together with a practical decoding algorithm, there is a strong desire to remove the restriction that $t \leq (p-1)/2$. One may wish to correct more than $(p-1)/2$ errors. The obvious conjecture to do this is the code whose roots include further successive odd powers of α . If $\alpha, \alpha^3, \dots, \alpha^{p-2}, \alpha^p$ prove insufficient, then would you believe that $\alpha, \alpha^3, \alpha^5, \dots, \alpha^p, \alpha^{p+2}$ might work? Unfortunately, this attempt does not work, and the situation is even worse than this. We cannot correct that $(p+1)/2$ st error even if we use almost twice as much redundancy as we used to correct $(p-1)/2$ errors !

The theorem is as follows:

The negacyclic code of block length $N = (p^m - 1)/2$ over $GF(p)$, $m > 1$, whose generator polynomial is the product of the distinct minimum functions of $\alpha, \alpha^3, \alpha^5, \dots, \alpha^{2p-5}, \alpha^{2p-3}$ has codewords of Lee weight p .

Proof: We shall exhibit a set of p locations such that $S_j = \sum_{i=1}^p X_i^j = 0$ for all odd $j \leq 2p-3$. The word containing $(+)$ ones in these p locations must then be a codeword of Lee weight p .

For $i = 1, 2, \dots, p$, let $X_i = \xi + (i-1)$ where $\xi \notin GF(p)$, but $\xi \in GF(p^m)$. Then the X_i represent distinct locations, for $X_i \neq X_j$ unless $i = j$, and if $X_i = -X_j$ then $\xi + (i-1) = -(\xi + (j-1))$ and

$$\xi = -(i+j-2)/2 \in GF(p)$$

The "error" polynomial is

$$\sigma(z) = \prod_{i=1}^p (1 - X_i z) = \prod_{i=1}^p (1 - \xi z - z(i-1))$$

Recalling that

$$\prod_{i=1}^p (y - (i-1)) = y^p - y, \text{ for every } y$$

and that

$$\prod_{i=1}^p (y - z(i-1)) = y^p - z^{p-1} y$$

we deduce that

$$\begin{aligned} \sigma(z) &= (1 - \xi z)^p - z^{p-1} (1 - \xi z) \\ &= 1 - z^{p-1} - (\xi^p - \xi) z^p \end{aligned}$$

$$\sigma(z) = 1 - z^{p-1} - \psi z^p$$

where we have defined

$$\psi = \xi^p - \xi$$

Since

$$\sigma'(z) = z^{p-2}$$

Newton's Identities become

$$S(z) = \frac{-z \sigma'(z)}{1 - \sigma(z)} = \frac{-z^{p-1}}{1 - (z^{p-1} + \psi z^p)}$$

$$= - (z^{p-1} + z^{2p-2} + \psi z^{2p-1} + z^{3p-3} + 2\psi z^{3p-2} + \psi^2 z^{3p-1} + \dots)$$

Since $p-1$ is even, $S_j = 0$ for all odd $j \leq 2p-3$.

q.e.d.

The argument of the preceding theorem can be continued to show that certain other S 's must also vanish. This shows that certain other negacyclic codes, whose generators have these additional roots, still have codewords of weight p . We first continue the argument to derive an explicit formula for S_n :

We have

$$\begin{aligned} S(z) &= \frac{-z^{p-1}}{1 - (z^{p-1} + \psi z^p)} = -z^{p-1} \sum_{k=0}^{\infty} (z^{p-1})^k (1+\psi z)^k \\ &= -z^{p-1} \sum_{k=0}^{\infty} (z^{p-1})^k \sum_{I=0}^k \binom{k}{I} \psi^I z^I \end{aligned}$$

Letting $I = i(p-1) + j$

$$S(z) = -z^{p-1} \sum_{k=0}^{\infty} \sum_{i=0}^{\infty} \sum_{j=0}^{p-2} \binom{k}{i(p-1)+j} \psi^{i(p-1)+j} z^{(p-1)(i+k)+j}$$

Letting $k = i + K + 1$ gives

$$S(z) = - \sum_{k=1}^{\infty} \sum_{j=0}^{p-2} \psi^j \sum_{i=0}^{\infty} \binom{k-i-1}{i(p-1)+j} \psi^{(p-1)i} z^{(p-1)k+j}$$

The binomial coefficient vanishes unless $i(p-1) + j \leq k-i-1$, or

$i \leq \frac{k-j-1}{p}$, and hence

$$S_{k(p-1)+j} = -\psi^j \sum_{i=0}^{\lfloor \frac{k-i-1}{p} \rfloor} \binom{k-i-1}{i(p-1)+j} (\psi^{(p-1)})^i$$

From this argument it is obvious that codewords of weight p occur even if the roots of the generator include $\alpha, \alpha^3, \dots, \alpha^{2p-3}, \alpha^{2p-1}$,

$\alpha^{2p+3}, \dots, \alpha^{3p-4}, \alpha^{3p}, \alpha^{3p+2}, \dots, \alpha^{4p-5}, \alpha^{4p+1}, \dots$ In some cases, it might be possible to choose $\psi^{(p-1)}$ in such a way that certain additional S's also vanish.

Certain other classes of weak negacyclic codes may be discredited by examining the obvious low weight factors of $x^N + 1$. For example, over GF(5), one has $x^{12} + 1 = (x^6 + 2)(x^6 - 2)$, and $x^{12} + 1 = (x^4 + 1)(x^8 - x^4 + 1)$. From this one sees that any negacyclic code whose generator divides $(x^4 + 1)$ has distance ≤ 2 ; any code whose generator divides $x^6 + 2$ or $x^6 - 2$ or $x^8 - x^4 + 1$ has distance ≤ 3 .

Nevertheless, there do exist negacyclic codes having very large minimum distances - if one is willing to transmit information at sufficiently low rates. We shall prove the following theorem:

Let $N = (p^m - 1)/2$ and let α be a primitive $2N$ th root of unity over GF(p). The negacyclic code whose check polynomial is the minimum function of α is equidistant; every nonzero codeword has Lee weight given by

$$\omega_L = (p^2 - 1) p^{m-1} / 8.$$

Proof: This code has $p^m - 1 = 2N$ nonzero codewords, each of which is of the form $c(x) = M(x) g(x)$, where $\deg M < \deg h$. If some nonzero codeword had only k distinct negacyclic shifts, then

$$(x^k - 1)c(x) \equiv 0 \pmod{x^N + 1}$$

or
$$(x^k - 1)c(x) = (x^k - 1)M(x)g(x) = \frac{(x^k - 1)M(x)(x^N + 1)}{h(x)} \equiv 0 \pmod{x^N + 1}$$

which implies that h divides $(x^k - 1)M$. Since h is irreducible and $\deg M < \deg h$, h must divide $x^k - 1$. Since h is primitive, $k = 2N$ or some multiple thereof. This proves that all $2N$ negacyclic shifts of any

nonzero codeword are distinct. Since there are only $2N$ nonzero codewords in the entire code, it follows that every nonzero codeword is a negacyclic shift of every other nonzero codeword, and that every nonzero codeword has the same Lee weight.

We may compute the average Lee weight of all of the codewords in any nontrivial linear code over $GF(p)$ as follows: We list each codeword as a row of a matrix, containing N columns and as many rows as there are codewords. Since the code is linear, every nontrivial column (excluding trivial columns containing all zeroes) must contain equal numbers of each of the p symbols. The average weight of each column is therefore

$$\frac{2}{p} \sum_{k=1}^{(p-1)/2} k = (p^2-1)/4p, \text{ and the average weight}$$

of the entire code is $N(p^2-1)/4p$. Applying this result to the present case gives $(p^m-1)(p^2-1)/8p$ for the average weight of all of the p^m codewords, and $p^{m-1}(p^2-1)/8$ for the average weight of the $p^m - 1$ nonzero codewords.

q.e.d.

This theorem shows that certain of the codes guaranteed by our main theorem are actually much better than claimed. For example, the code with $p = 127$, $N = 63$, having 1 message digit and 62 check digits actually has Lee distance $32 \cdot 63$, so that it is capable of correcting almost 16 times as many errors as promised by the theorem !

These low-rate equidistant negacyclic codes are quite analagous to the low rate maximum length shift register cyclic codes. It may be possible to use these low rate negacyclic codes as the raw material from which shorter, higher rate, non-negacyclic codes may be manufactured via algebraic puncturing a la Solomon-Stiffler (1966).

Similarly, the high-rate codes of the main theorem are analagous to the high-rate BCH codes. The single error correcting negacyclic codes, like the Hamming codes, are "perfect" in that they satisfy a volume bound with equality. It may be possible to show that the double-Lee-error correcting negacyclic codes, like the double error correcting cyclic BCH codes, are quasi-perfect, although at present this is only speculation.

It appears that the Lee weights of all of the codewords in large classes of negacyclic codes can be enumerated, and in some cases it may also be possible to obtain some results on the more important problem of enumerating the weights of the coset leaders. Perhaps it may also be possible to find other "good" classes of negacyclic codes. For example, do there exist negacyclic codes analagous to the quadratic residue codes ?

Many problems of this sort remain to be solved. I hope that some of them may prove of interest to the reader.

REFERENCES

- Berlekamp, E. R., "Practical BCH Decoders." PGIT , 1967.
- Berlekamp, E. R. Constructive Coding Theory. McGraw-Hill, 1968.
- Bose, R. C. and Ray-Chaudhuri, D. K., "On a Class of Error-Correcting Binary Codes." Information & Control 3, 1960, pp. 68-79.
- Bose, R. C. and Ray-Chaudhuri, D. K., "Further Results on Error-Correcting Binary Codes." Information & Control 3, 1960, pp. 279-290.
- Chien, R. T., "Cyclic Decoding Procedures for Bose-Chaudhuri-Hocquenghem Codes." PGIT 10, 1964, pp. 357-362.
- Forney, G. D., "On Decoding BCH Codes." PGIT 11, 1965, pp. 549-557.
- Gorenstein, D. and Zierler, N., "A Class of Error-Correcting Codes in p^m symbols." JSIAM 9, June 1961, pp. 207-214.
- Hamming, R. W., "Error Detecting and Error-Correcting Codes." BSTJ 29, 1950, pp. 47-160.
- Hocquenghem, A., "Codes Correcteurs d'Erreurs." Chiffres 2, Sept. 1959, pp. 147-156.
- Lee, C. Y., "Some Properties of Nonbinary Error-Correcting Codes." PGIT 4, 1958, pp. 77-82.
- Mattson, H. C. and Solomon, G., "A New Treatment of Bose-Chaudhuri Codes." JSIAM 9, # 4, 1961, pp. 654-669.
- Peterson, W. W., Error-Correcting Codes. MIT Press and John Wiley & Sons, 1961.
- Prange, E., Cyclic Error-Correcting Codes in Two Symbols. Air Force Cambridge Research Center, Cambridge, Massachusetts, AFCRC-TN-57-103, 1957.
- Solomon, G. and Stiffler, J. J., "Algebraically Punctured Cyclic Codes." Information & Control 8, 1965, pp. 170-179.
- Shannon, C. E. and Weaver, W., A Mathematical Theory of Communication. University of Illinois Press, Urbana, Illinois, 1949.
- Slepian, D., "A Class of Binary Signaling Alphabets." BSTJ 35, 1956, pp. 203-234.
- Slepian, D., "Some Further Theory of Group Codes." BSTJ 39, 1960, pp. 1219-1252.