NONBINARY BCH DECODING

by

Elwyn R. Berlekamp
University of North Carolina

Institute of Statistics Mimeo Series No. 502

December 1966

This research was supported in part by the National Science Foundation Grant no. GP-5790 and Army Research Office, Durham Grant no. DA-ARO-D-31-124-G670.

DEPARTMENT OF STATISTICS
University of North Carolina
Chapel Hill, N. C.

NONBINARY BCH DECODING

Abstract and Table of Contents:

This paper shows that the decoding of BCH codes readily reduces to the solution of a certain key equation. An iterative algorithm is presented for solving this equation over any field. Following a heuristic derivation of the algorithm, we give a complete statement of the algorithm and proofs of its principal properties.

Additional sections of this paper then discuss the relationship of this algorithm to the classical matrix methods, the simplification which the algorithm takes in the special case of binary codes, the generalization of the algorithm to BCH codes with a slightly different definition, the generalization of the algorithm to decode erasures as well as errors, and the extension of the algorithm to decode more than t errors in certain cases. Each of these final sections of the paper may be read independently of the others.

The Algorithm

- Introduction
- Heuristic Derivation of the Iterative Algorithm
- The Iterative Algorithm
- Iterative Algorithm Theorems
- Examples

Additional Results

- Beloved Historical Dregs
- Simplifications in the Binary Case
- Alternate BCH Codes
- Decoding Erasures as well as Errors
- Decoding More than t Errors.

Introduction

If the encoder transmits the codeword whose successive digits are the coefficients of the polynomial

$$c(z) = c_o + c_1 z + c_2 z^2 + \ldots + c_{N-1} z^{N-1} \; ; \quad c_i \in GF(q)$$

and the channel noise adds to this an error word given by

$$e(z) = e_o + e_1 z + e_2 z^2 + \ldots + e_{N-1} z^{N-1} \; ; \quad e_i \in GF(q)$$

then the decoder receives the word

$$r(z) = c(z) + e(z)$$

Let α be a primitive Nth root of unity over GF(q). If α^k is a root of $c(z)$, then $r(\alpha^k) = e(\alpha^k)$. If the error word consists of an error of value Y_1 at location $X_1 = \alpha^{j_1}$, an error of value Y_2 at location $X_2 = \alpha^{j_2}$, then $e(z) = \sum_i Y_i z^{j_i}$ and $e(\alpha^k) = \sum_i Y_i \alpha^{kj_i} = \sum_i Y_i X_i^k = S_k$. Here the X_i are called the <u>error locations</u>; the Y_i are called the <u>error values</u>; the S_k are called the <u>weighted power-sum symmetric functions</u>.

A t-error correcting BCH code is constructed in such a way that $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t}$ are roots of its generating polynomial*, and therefore, they are also roots of every codeword. Thus, as the first step of our BCH decoding procedure, we may calculate $S_k = e(\alpha^k)$ for $k = 1, 2, \ldots, 2t$.

In order to locate and evaluate the errors, it is helpful to consider two polynomials. The first, called the <u>error locator</u>, $\sigma(z)$, is the polynomial whose reciprocal roots are the error locations: $\boxed{\sigma(z) = \prod_i (1 - X_i z)}$

---

*The alternate, more general, definition $(\alpha^m, \alpha^{m+1}, \ldots, \alpha^{m+2t-1})$ for $m \neq 1$ is covered in a later section of this paper, "Alternate BCH Codes."

The second, called the error evaluator, $\omega(z)$, is defined by

$$\boxed{\omega(z) = \sigma(z) + \sum_i z\, X_i\, Y_i \prod_{j \neq i} (1 - X_j z)}$$

Since $\sigma(0) = 1$, the generating function for the quotient $\omega(z)/\sigma(z)$ is well defined. Its coefficients are given by

$$\frac{\omega}{\sigma} = 1 + \sum_i z\, X_i\, Y_i \,/\, (1 - X_i\, z)$$

$$= 1 + \sum_{k=1}^{\infty} \left(\sum_i Y_i\, X_i^k \right) z^k = 1 + S(z)$$

where we have defined the generating function $S(z)$ by $S(z) = \sum_{k=1}^{\infty} S_k\, z^k$.

According to the above calculations, $\frac{\omega}{\sigma} = 1+S$, and

$$\boxed{(1+S)\sigma = \omega}$$

For the t-error-correcting BCH code, the decoder is able to calculate only $S_1, S_2, \ldots, S_{2t}$ at step I. Thus, only the first $2t$ coefficients of the generating function for $S(z)$ are known, and the decoder must attempt to find σ and ω from the equation

$$\boxed{(1+S)\sigma \equiv \omega \mod z^{2t+1}}$$

The solution of these equations for σ and ω, given $S \mod z^{2t+1}$, is the second step of the decoding procedure. We postpone the details until the next section.

Once the error location, $\sigma(z)$, is known, the decoder may locate the the errors by finding its reciprocal roots: $\sigma(z) = \prod_i (1 - X_i z)$. The solution of this equation for the X's, given $\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{\deg \sigma}$ (the

coefficients of $\sigma(z) = \Sigma \sigma_j z^j$) is the third step of the decoding procedure.

After the errors have been located, the decoder may evaluate the polynomial $\omega(z)$ at the point $z = X_i^{-1}$, obtaining

$$\omega(X_i^{-1}) = Y_i \prod_{j \neq i} (1 - X_j X_i^{-1})$$

Thus, as the fourth step of the decoding procedure, the decoder may evaluate the errors according to the formula

$$Y_i = \frac{\omega(X_i^{-1})}{\prod_{j \neq i}(1 - X_j X_i^{-1})} = \frac{X_i^{(\deg \sigma)} \omega(X_i^{-1})}{X_i \prod_{j \neq i} (X_i - X_j)} = \frac{\tilde{\omega}(X_i)}{X_i \prod_{j \neq i} (X_i - X_j)}$$

Thus, we may summarize our decoding procedure as follows:

Step I: Find the weighted power sum symmetric functions, $S_1, S_2, \ldots, S_{2t}$.

Step II: Knowing the generating function for $S(z) \bmod z^{2t+1}$, find the polynomials σ and ω from the equation $(1+S)\sigma \equiv \omega \bmod z^{2t+1}$.

Step III: Find the reciprocal roots of $\sigma(z)$, the error locations.

Step IV: Find the error values.

We now consider Step II in greater detail.

<u>Heuristic derivation of the iterative algorithm</u>

We wish to solve the equation

$$(1+S)\sigma \equiv \omega \bmod z^{2t+1}$$

for the polynomials $\sigma(z)$ and $\omega(z)$, given $S(z) \bmod z^{2t+1}$. The problem looks difficult, so we break it up into smaller pieces. We consider the sequence of equations

*1 $$(1+S)\,\sigma^{(n)} \equiv \omega^{(n)} \bmod z^{n+1}$$

For each $n = 0, 1, 2, \ldots, 2t$, we shall find polynomials $\sigma^{(n)} = \sum_k \sigma_k^{(n)} z^k$

and $\omega^{(n)} = \omega_k^{(n)} z^k$ which solve this equation. In general, these equations may have many solutions. Since the degree of σ is the number of errors, a good decoder must attempt to find a solution in which degree σ and degree ω are "small."

If we have a solution to *1 , then we might hope that this same pair of polynomials, $\sigma^{(n)}$ and $\omega^{(n)}$, might also solve the equation

$$(1+S)\ \sigma^{(n)} \stackrel{?}{\equiv} \omega^{(n)} \mod z^{n+2}$$

In general, we cannot expect to be so lucky. However we may write

*2: $$(1+S)\ \sigma^{(n)} \equiv \omega^{(n)} + \Delta_1^{(n)}\ z^{n+1} \mod z^{n+2}$$

where we define $\Delta_1^{(n)}$ as the coefficient of z^{n+1} in the product $(1+S)\ \sigma^{(n)}$. If $\Delta_1^{(n)} = 0$, then we may evidently proceed by taking $\sigma^{(n+1)} = \sigma^{(n)}$ and $\omega^{(n+1)} = \omega^{(n)}$. In order to define $\sigma^{(n+1)}$ in the case when $\Delta_1^{(n)} \neq 0$, we introduce the auxiliary polynomials $\tau^{(n)}$ and $\gamma^{(n)}$, which will be chosen so as to solve the <u>auxiliary equations</u>:

*3: $$(1+S)\ \sigma^{(n)} \equiv \gamma^{(n)} + z^n \mod z^{n+1}$$

Of course we would also like the degrees of $\tau^{(n)}$ and $\gamma^{(n)}$ to be "small." In terms of these auxiliary polynomials, we may define the successive σ's and ω's by

*4: $$\sigma^{(n+1)} = \sigma^{(n)} - \Delta_1^{(n)} z\ \tau^{(n)}$$

*5: $$\omega^{(n+1)} = \omega^{(n)} - \Delta_1^{(n)} z\ \gamma^{(n)}$$

It is readily seen that $\sigma^{(n+1)}$ and $\omega^{(n+1)}$ satisfy the equation

$$(1+S)\ \sigma^{(n+1)} \equiv \omega^{(n+1)} \mod z^{(n+1)+1}$$

if $\sigma^{(n)}$ and $\omega^{(n)}$ satisfy *1 and $\tau^{(n)}$ and $\gamma^{(n)}$ satisfy *3.

There are two obvious ways to define $\tau^{(n+1)}$ and $\gamma^{(n+1)}$:

Either

*6: $\tau^{(n+1)} = z\tau^{(n)}$ and $\gamma^{(n+1)} = z\,\gamma^{(n)}$

or

*7: $\tau^{(n+1)} = \dfrac{\sigma^{(n)}}{\Delta_1^{(n)}}$ and $\gamma^{(n+1)} = \dfrac{z\,\omega^{(n)}}{\Delta_1^{(n)}}$

Either choice will satisfy the equation

$$(1+S)\,\tau^{(n+1)} \equiv \gamma^{(n+1)} + z^{n+1} \mod z^{n+2}$$

if $\sigma^{(n)}$ and $\omega^{(n)}$ satsify *1 and $\tau^{(n)}$ and $\gamma^{(n)}$ satsify *3.

If $\Delta_1^{(n)} = 0$, then *7 is meaningless, and we are forced to define $\tau^{(n+1)}$ and $\gamma^{(n+1)}$ by *6. However, if $\Delta_1^{(n)} \neq 0$, then our choice between *6 and *7 must be based upon our desire to minimize the degrees of $\tau^{(n+1)}$ and $\gamma^{(n+1)}$. The degrees of $\sigma^{(n+1)}$, $\tau^{(n+1)}$, $\omega^{(n+1)}$, and $\gamma^{(n+1)}$ are given by

*8: $\deg \sigma^{(n+1)} =$

$\deg \sigma^{(n)}$ if $\Delta_1^{(n)} = 0$ or if $\deg \sigma^{(n)} > \deg \tau^{(n)}$

$1 + \deg \tau^{(n)}$ if $\Delta_1^{(n)} \neq 0$ and if $\deg \tau^{(n)} > \deg \sigma^{(n)} - 1$

$\leq$ either of above if $\Delta_1^{(n)} \neq 0$ and if $\deg \sigma^{(n)} = 1 + \deg \tau^{(n)}$

*9: $\deg \tau^{(n+1)} =$

$1 + \deg \tau^{(n)}$ if we use *6

$\deg \sigma^{(n)}$ if we use *7

$$*10:\ \deg\ \omega^{(n+1)} = \begin{cases} \deg\ \omega^{(n)} \text{ if } \Delta_1^{(n)} = 0 \text{ or if } \deg \omega^{(n)} > \deg \gamma^{(n)} \\ 1 + \deg\ \gamma^{(n)} \text{ if } \Delta_1^{(n)} \neq 0 \text{ and if } \deg \gamma^{(n)} > \deg \omega^{(n)} - 1 \\ \leq \text{either of the above if } \Delta_1^{(n)} \neq 0 \text{ and if } \deg \omega^{(n)} = 1 + \deg \gamma^{(n)} \end{cases}$$

$$*11:\ \deg\ \gamma^{(n+1)} = \begin{cases} 1 + \deg\ \gamma^{(n)} \text{ if we use } *6 \\ \deg\ \omega^{(n)} \text{ if we use } *7 \end{cases}$$

The degree of $\sigma^{(n+1)}$ is seen to be subject to an "accidental" decrease if $\deg \sigma^{(n)} = 1 + \deg\ \tau^{(n)}$ and the leading coefficients of $\sigma^{(n)}$ and $\Delta_1^{(n)}$ $\tau^{(n)}$ happen to be equal. In order to circumcompute such accidents, we base our choice between *6 and *7 not on the actual degrees of $\sigma^{(n)}$, $\tau^{(n)}$, $\omega^{(n)}$, and $\gamma^{(n)}$, but on an upper bound, D(n), which is independent of such vagaries. We shall define this integral function, D(n), in such a way that

$$*12:\qquad \deg\ \sigma^{(n)} \leq D(n)$$

$$*13:\qquad \deg\ \tau^{(n)} \leq n - D(n)$$

From 8, 12, and 13, we are led to the recursive definition of D(n):

$$*14:\ D(n+1) = \begin{cases} D(n) \text{ if } \Delta_1^{(n)} = 0 \text{ or if } D(n) \geq \frac{n+1}{2} \\ n + 1 - D(n) \text{ if } \Delta_1^{(n)} \neq 0 \text{ and } D(n) \leq \frac{n+1}{2} \end{cases}$$

It is readily seen that if $\deg \sigma^{(n)} \leq \deg \tau^{(n)} \leq n - D(n)$, then $\deg \sigma^{(n+1)} \leq D(n+1)$. Similarly, if $\deg \omega^{(n)} \leq D(n)$ and $\deg\ \gamma^{(n)} \leq n - D(n)$, then $\deg \omega^{(n+1)} \leq D(n+1)$.

In order to ensure that $\deg \tau^{(n+1)} \leq (n+1) - D(n+1)$ and that $\deg \gamma^{(n+1)} \leq (n+1) - D(n+1)$, we must adopt the following rule for choosing between *6 and *7:

*15:
$$\begin{cases} \text{Use *6 if } \Delta_1^{(n)} = 0 \text{ or if } D(n) > (n+1)/2 \\ \text{Use *7 if } \Delta_1^{(n)} \neq 0 \text{ and } D(n) < (n+1)/2 \end{cases}$$

If $\Delta_1^{(n)} \neq 0$ and $D(n) = (n+1)/2$, then either *6 or *7 will give us polynomials $\tau^{(n+1)}$ and $\gamma^{(n+1)}$, each having degree $\leq n+1 - D(n+1)$. When in doubt, procrastinate ! We postpone the close decision between *6 and *7 in this case until we have looked at another consideration.

The initial equations are

$$(1+S)\,\sigma^{(0)} \equiv \omega^{(0)} \mod z$$
$$(1+S)\,\tau^{(0)} \equiv \gamma^{(0)} + 1 \mod z$$

The equations may be solved by the obvious initialization:

*16:
$$\sigma^{(0)} = \tau^{(0)} = \omega^{(0)} = 1; \quad \gamma^{(0)} = 0; \quad D(0) = 0$$

We notice that $\deg \sigma^{(0)} = \deg \tau^{(0)} = \deg \omega^{(0)} = 0 = D(0)$, but that $\deg \gamma^{(0)} = -\infty < D(0)$. Thus, at least initially, we may do even better than the restrictions

$$\deg \omega^{(n)} \leq D(n)$$
$$\deg \gamma^{(n)} \leq n - D(n)$$

We require that at least one of these expressions be satisfied with strict inequality. To this end, we introduce the Boolean function $B(n)$, with initial value $B(0) = 0$. (In general, either $B(n) = 0$ or $B(n) = 1$.) We then require that

*17:
$$\deg \omega^{(n)} \leq D(n) - B(n)$$

*18:
$$\deg \gamma^{(n)} \leq n - D(n) - (1-B(n))$$

If we take the proper choice between *6 and *7 in the case when $\Delta_1^{(n)} \neq 0$ and $D(n) = (n+1)/2$, and if we define $B(n)$ carefully, then we may guarantee

that *17 and *18 hold for all n. By examining *10 and *11, the proper choice is seen to be:

*19:
$$\begin{cases} \text{Use *6 if } \Delta_1^{(n)} \neq 0,\ D(n) = (n+1)/2, \text{ and } B(n) = 0 \\ \text{Use *7 if } \Delta_1^{(n)} \neq 0,\ D(n) = (n+1)/2, \text{ and } B(n) = 1 \end{cases}$$

*20:
$$B(n+1) = \begin{cases} B(n) \text{ when using 6} \\ 1 - B(n) \text{ when using 7} \end{cases}$$

This completes the heuristic derivation of the recursive algorithm. To summarize, we start from the initial conditions *16. We then proceed recursively as follows: Define $\Delta_1^{(n)}$ by *2, $\sigma^{(n+1)}$ by *4, $\omega^{(n+1)}$ by *5, and D(n+1) by *14. According to *15 and *19, we then define $\tau^{(n+1)}$ and $\gamma^{(n+1)}$ by *6 or *7, and B(n+1) according to *20. The polynomials defined in this recursive manner are then seen to satisfy equations *1, *3, *12, *17, *18. We restate the algorithm explicitly as follows:

The Iterative Algorithm

Initially define $\sigma^{(0)} = 1$, $\tau^{(0)} = 1$, $\omega^{(0)} = 1$, $\gamma^{(0)} = 0$, $D(0) = 0$, $B(0) = 0$.

Proceed recursively as follows:

Define $\Delta_1^{(n)}$ as the coefficient of z^{n+1} in the product $(1+S)\ \sigma^{(n)}$,

Let

$$\sigma^{(n+1)} = \sigma^{(n)} - \Delta_1^{(n)}\ z\ \tau^{(n)}$$
$$\omega^{(n+1)} = \omega^{(n)} - \Delta_1^{(n)}\ z\ \gamma^{(n)}$$

If $\Delta_1^{(n)} = 0$, or if $D(n) > \frac{n+1}{2}$, or if $\Delta_1^{(n)} \neq 0$ and $D(n) = \frac{n+1}{2}$ and $B(n)=0$,

set

$$D(n+1) = D(n)$$
$$B(n+1) = B(n)$$
$$\tau^{(n+1)} = z\tau^{(n)}$$
$$\gamma^{(n+1)} = z\gamma^{(n)}$$

But, if $\Delta_1^{(n)} \neq 0$ and either $D(n) < \frac{n+1}{2}$ or $D(n) = \frac{n+1}{2}$ and $B(n) = 1$

set

$$D(n+1) = n+1 - D(n)$$
$$B(n+1) = 1-B(n)$$
$$\tau^{(n+1)} = \sigma^{(n)} / \Delta_1^{(n)}$$
$$\gamma^{(n+1)} = \omega^{(n)} / \Delta_1^{(n)}$$

Iterative Algorithm theorems (over any field)

(1) For each n,

1a: $\sigma^{(n)}(0) = \omega^{(n)}(0) = 1$

1b: $(1+S)\sigma^{(n)} \equiv \omega^{(n)} + \Delta_1^{(n)} z^{n+1} \mod z^{n+2}$

1c: $(1+S)\tau^{(n)} \equiv \gamma^{(n)} + z^n \mod z^{n+1}$

1d: $\deg \sigma^{(n)} \leq D(n)$ with equality if $B(n) = 1$

1e: $\deg \tau^{(n)} \leq n-D(n)$ with equality if $B(n) = 0$

1f: $\deg \omega^{(n)} \leq D(n)-B(n)$ with equality if $B(n) = 0$

1g: $\deg \gamma^{(n)} \leq n-D(n)-(1-B(n))$ with equality if $B(n) = 1$

(2) For each n,

$$\omega^{(n)} \tau^{(n)} - \sigma^{(n)} \gamma^{(n)} = z^n$$

(3) If σ and ω are any pair of polynomials which satisfy

$\sigma(0) = 1$ and $(1+S)\sigma \equiv \omega \mod z^{n+1}$, $D = \max\{\deg \sigma, \deg \omega\}$

then there exist polynomials U and V such that $U(0) = 1$, $V(0) = 0$, $\deg U \leq D - D(n)$, $\deg V \leq D-(n-D(n))$, and

$$\sigma = U\sigma^{(n)} + V\tau^{(n)}$$

$$\omega = U\omega^{(n)} + V\gamma^{(n)}$$

(4) If σ and ω are relatively prime, and $\sigma(0) = 1$ and $(1+S)\sigma \equiv \omega \mod z^{n+1}$, then

4a: Either $\deg \sigma \geq D(n) + 1-B(n) \geq D(n)$ or $\deg \omega \geq D(n)$ or both.

4b: If $\deg \sigma \leq \frac{n+1}{2}$ and $\deg \omega \leq \frac{n}{2}$, then $\sigma=\sigma^{(n)}$ and $\omega=\omega^{(n)}$

Proof of Theorem 1, excepting the equality clauses of 1d, 1e, 1f, and 1g.

These claims were proved in the heuristic derivation of the algorithm. Readers who prefer a direct proof may verify these claims by a straight forward but tedious induction on n.

Proof of Theorem 2:

According to Theorem 1,

$$(1+S)\ \sigma^{(n)} \equiv \omega^{(n)} \mod z^{n+1}$$

$$(1+S)\ \tau^{(n)} \equiv \gamma^{(n)} + z^n \mod z^{n+1}$$

Taking the product of these two congruences gives

$$(1+S)\ \tau^{(n)}\ \omega^{(n)} \equiv (1+S)\ \sigma^{(n)}\ (\gamma^{(n)} + z^n)$$

Dividing by (1+S) gives

$$\tau^{(n)}\ \omega^{(n)} \equiv \sigma^{(n)}\ \gamma^{(n)} + \sigma^{(n)}\ z^n$$

Since $\sigma^{(n)}\ z^n \equiv \sigma^{(n)}\ (0)\ z^n \equiv z^n \mod z^{n+1}$, this becomes

$$\tau^{(n)}\ \omega^{(n)} - \gamma^{(n)}\ \sigma^{(n)} \equiv z^n \mod z^{n+1}$$

According to Theorems 1e and 1f, $\deg \omega^{(n)} + \deg \tau^{(n)} \leq n$.

According to Theorems 1d and 1g, $\deg \sigma^{(n)} + \deg \gamma^{(n)} \leq n$.

Therefore, $\deg \left\{ \tau^{(n)}\ \omega^{(n)} - \sigma^{(n)}\ \gamma^{(n)} \right\} \leq n$

from which we conclude that

$$\boxed{\tau^{(n)}\ \omega^{(n)} - \sigma^{(n)}\ \gamma^{(n)} = z^n}$$

Proof of Equality clauses of 1d, 1e, 1f, and 1g:

If $B(n) = 0$, then

$$\deg \gamma^{(n)} \leq n-D(n)-1,$$

$$\deg \sigma^{(n)} \leq D(n),$$

$$\text{and} \quad \deg \left\{\sigma^{(n)} \gamma^{(n)}\right\} \leq n-1.$$

Since

$$\deg \left\{\omega^{(n)} \tau^{(n)} - \sigma^{(n)} \gamma^{(n)}\right\} = n, \text{ it follows that}$$

$$\deg \left\{\omega^{(n)} \tau^{(n)}\right\} = n \quad \text{and} \quad \deg \omega^{(n)} = D(n), \quad \deg \tau^{(n)} = n-D(n).$$

Similarly, if $B(n)=1$, then $\deg \omega^{(n)} \leq D(n)-1$, $\deg \tau^{(n)} \leq n-D(n)$, and $\deg \left\{\omega^{(n)} \tau^{(n)}\right\} \leq n-1$. It follows that $\deg \left\{\sigma^{(n)} \gamma^{(n)}\right\} = n$, $\deg \sigma^{(n)} = D(n)$, and $\deg \gamma^{(n)} = n-D(n)$.

Remarks on Theorem 3:

This theorem gives us the form of the general solution (of any degree) of the equations $\sigma(0) = 1$, $(1+S)\sigma \equiv \omega \mod z^{n+1}$. In view of theorem 2 and theorem 1a, it is evident that $\sigma^{(n)}$ and $\tau^{(n)}$ are relatively prime. Hence any polynomial, f, may be expressed in the form

$$f = U \sigma^{(n)} + V \tau^{(n)}$$

Theorem 3 asserts that if σ and ω are a solution to the equations $\sigma(0) = 1$, $(1+S)\sigma \equiv \omega \mod z^{n+1}$, then the same U and V hold for both expressions $\sigma = U\sigma^{(n)} + V\tau^{(n)}$ and $\omega = U\omega^{(n)} + V\gamma^{(n)}$. Furthermore, theorem 3 asserts that the degrees of U and V are small.

Proof of Theorem 3: By hypothesis

$$(1+S)\ \sigma \equiv \omega \mod z^{n+1}$$

Multiplying by theorem 1b gives

$$(1+S)\quad \sigma^{(n)}\omega \equiv (1+S)\sigma^{\omega^{(n)}}$$

$$\sigma^{(n)}\omega \equiv \sigma^{\omega^{(n)}}$$

*21: $$\sigma^{(n)}\omega - \sigma^{\omega^{(n)}} = -z^n V, \text{ where } V(0)=0 \text{ and } \deg V \leq D(n)+D-n$$

Multiplying the hypothesis by theorem 1c gives

$$(1+S)\ \tau^{(n)}\omega \equiv (1+S)\sigma^{(\gamma^{(n)} + z^n)}$$

$$\tau^{(n)}\omega \equiv \sigma(\gamma^{(n)} + z^n) \equiv \sigma^{\gamma^{(n)}} + z^n$$

*22 $$\tau^{(n)}\omega - \sigma^{\gamma^{(n)}} = U z^n, \text{ where } U(0)=1 \text{ and } \deg U \leq D-D(n)$$

Subtracting $\tau^{(n)}$ times *21 from $\sigma^{(n)}$ times *22 gives

$$(\tau^{(n)}\omega^{(n)} - \sigma^{(n)}\gamma^{(n)})\sigma = z^n(U\sigma^{(n)} + V\tau^{(n)})$$

Using Theorem 2, this becomes

$$\boxed{\sigma = U\sigma^{(n)} + V\tau^{(n)}}$$

Similarly, subtracting $\gamma^{(n)}$ times *21 from $\omega^{(n)}$ times *22 gives

$$(\omega^{(n)}\tau^{(n)} - \sigma^{(n)}\gamma^{(n)})\omega = z^n(U\omega^{(n)} + V\gamma^{(n)})$$

Using Theorem 2, this becomes

$$\boxed{\omega = U\omega^{(n)} + V\gamma^{(n)}}$$

Proof of Theorem 4:

Theorem 4a is an immediate consequence of *22, 1e, and 1g.

Consequently, if $\deg \sigma \leq (n+1)/2$ and $\deg \omega \leq n/2$, then $D(n) \leq (n+1)/2$, with strict inequality if $B(n) = 0$. According to Theorems 1d and 1f, we may deduce that

$$\deg \sigma^{(n)} \leq (n+1)/2,\ \deg \omega^{(n)} \leq n/2,\ \deg (\sigma^{(n)}\omega) \leq n + \tfrac{1}{2} < n+1 ;$$

$$\deg (\sigma\omega^{(n)}) \leq n+\tfrac{1}{2} < n+1,\quad \deg (-z^{n}V) = \deg(\sigma^{(n)}\omega - \sigma\omega^{(n)}) < n+1,$$

$\deg V < 1$. Since $V(0) = 0$, this implies $V = 0$, Theorem 3 becomes $\sigma = U\sigma^{(n)}$, $\omega = U\sigma^{(n)}$. By hypothesis, σ and ω are relatively prime, so $U = 1$.

q.e.d.

Examples:

Take a code of block length $N = 15$ over $GF(2^2)$, whose syndrome gives S_1, S_2, S_3, and $S_4 = S_1^4$. We have

$q = 4$ symbols/alphabet letter

$N=15=4^2-1$ block length

$t = 2$ guaranteed error correction

$k = 9$ information digits

$r = 6$ check digits

Let ζ be a primitive element in $GF(2^2)$, which satisfies $\zeta^2+\zeta+1=0$. Let α be a primitive element of $GF(2^4)$, which satisfies

$\alpha^4 + \alpha + 1 = 0$ over $GF(2)$ or $\alpha^2 + \alpha + \zeta = 0$ over $GF(4)$. $\zeta = \alpha^5$.

All the elements of $GF(2^4)$ are represented as powers of α, as follows:

	α^3	α^2	α	1	$\alpha\zeta$	α	ζ	1	α	1
α^0	0	0	0	1	0	0	0	1	0	1
α^1	0	0	1	0	0	1	0	0	1	0
α^2	0	1	0	0	0	1	1	0	1	ζ
α^3	1	0	0	0	1	1	1	0	ζ^2	ζ
α^4	0	0	1	1	0	1	0	1	1	1
α^5	0	1	1	0	0	0	1	0	0	ζ
α^6	1	1	0	0	1	0	0	0	ζ	0
α^7	1	0	1	1	1	0	1	1	ζ	ζ^2
α^8	0	1	0	1	0	1	1	1	1	ζ^2
α^9	1	0	1	0	1	0	1	0	ζ	ζ
α^{10}	0	1	1	1	0	0	1	1	0	ζ^2
α^{11}	1	1	1	0	1	1	0	0	ζ^2	0
α^{12}	1	1	1	1	1	1	0	1	ζ^2	1
α^{13}	1	1	0	1	0	1	0	1	ζ	1
α^{14}	1	0	0	1	1	1	1	1	ζ^2	ζ^2
α^{15}	0	0	0	1	0	0	0	1	0	1

Example I:

$(1+S) = 1 + \alpha^3 z + \alpha^6 z^2 + \alpha^4 z^3 + \alpha^{12} z^4 + \ldots$

$\sigma^{(0)} = 1,\ \tau^{(0)} = 1,\ \omega^{(0)} = 1,\ \gamma^{(0)} = D,\ D(0) = 0,\ B(0) = 1$

$\Delta_1^{(1)} = \alpha^3$

$\sigma^{(1)} = 1 + \alpha^3 z,\ \tau^{(1)} = \alpha^{-3},\ \omega^{(1)} = 1,\ \gamma^{(1)} = \alpha^{-3},\ D(1) = 1,\ B(1) = 1$

$\Delta_1^{(1)} = 0$

$\sigma^{(2)} = 1 + \alpha^3 z,\quad \tau^{(2)} = \alpha^{-3} z,\quad \omega^{(2)} = 1,\quad \gamma^{(2)} = \alpha^{-3} z,\quad D(2)=1,\ B(2) = 1$

$\Delta_1^{(2)} = 1 \cdot \alpha^4 + \alpha^3 \cdot \alpha^6 = \alpha^{14} = \alpha^{-1}$

$\sigma^{(3)} = 1+\alpha^3 z + \alpha^{11} z^2,\ \tau^{(3)} = \alpha + \alpha^4 z,\ \omega^{(2)} = 1+\alpha^{11} z^2,\quad \gamma^{(3)} = \alpha,\ D(3)=2,\ B(3)=0$

$\Delta_1^{(3)} = \alpha^{12} + \alpha^3 \cdot \alpha^4 + \alpha^{11} \cdot \alpha^6 = 0$

$\sigma^{(4)} = 1 + \sigma^3 z + \alpha^{11} z^2,\ \omega^{(4)} = 1 + \alpha^{11} z^2,\quad D(4) = 2,\quad B(4) = 0$

The factorization of $\sigma^{(4)}$ is given by

$\sigma^{(4)} = (1+ \alpha^4 z)(1+ \alpha^7 z),\quad$ so $\quad X_1 = \alpha^4,\quad X_2 = \alpha^7$

$\tilde{\omega}^{(4)} = (z^2+\alpha^{11}).\quad Y_1 = \dfrac{(\alpha^4)^2 + \alpha^{11}}{\alpha^4(\alpha^4 + \alpha^7)} = \dfrac{\alpha^7}{\alpha^4 . \alpha^3} = 1$

$Y_2 = \dfrac{(\alpha^7)^2 + \alpha^{11}}{\alpha^7(\alpha^4 + \alpha^7)} = \dfrac{\alpha^{10}}{\alpha^7(\alpha^3)} = 1$

Example II:

$$(1+S) = 1 + \alpha^{11}z + \alpha^{10}z^2 + \alpha^5 z^3 + \alpha^{14}z^4 + \ldots$$

$\sigma^{(0)} = 1,\ \tau^{(0)} = 1,\ \omega^{(0)} = 1,\ \gamma^{(0)} = 0,\ D(0) = 0,\ B(0) = 0.$

$\Delta_1^{(0)} = \alpha^{11} = \alpha^{-4}$

$\sigma^{(1)} = 1+\alpha^{11}z,\ \tau^{(1)} = \alpha^4,\ \omega^{(1)} = 1,\ \gamma^{(1)} = \alpha^4,\ D(1)=1,\ B(1)=1$

$\Delta_1^{(1)} = 1 \cdot \alpha^{10} + \alpha^{11} \cdot \alpha^{11} = \alpha^6$

$\sigma^{(2)} = 1 +(\alpha^{11}+ \alpha^4 \cdot \alpha^6)z = 1 + \alpha^{14}z,\ \tau^{(2)} = \alpha^{-6} + \alpha^5 z,\ \omega^{(2)} = 1 + \alpha^{10}z,$

$\gamma^{(2)} = \alpha^{-6},\ D(2)=1, B(2)=0.$

$\Delta_1^{(2)} = 1 \cdot \alpha^5 + \alpha^{14} \cdot \alpha^{10} = \alpha^6$

$\sigma^{(3)} = 1+(\alpha^{14}+1)z + \alpha^{11}z^2 = 1 + \alpha^3 z + \alpha^{11}z^2,\ \tau^{(3)} = \alpha^{-6} + \alpha^8 z,$

$\omega^{(3)} = 1+(\alpha^{10}+1)z = 1+\alpha^5 z,\ \gamma^{(2)} = \alpha^{-12} + \alpha^{-1}z,\ D(3)=2,\ B(3)=1$

$\Delta_1^{(3)} = 1 \cdot \alpha^{14} + \alpha^{14} + \alpha^3 \cdot \alpha^5 + \alpha^{11} \cdot \alpha^{10} = 0$

$\sigma^{(4)} = 1+\alpha^3 z +\alpha^{11}z^2 = (1+\alpha^4 z)(1+\alpha^7 z),$ so $X_1=\alpha^4,\ X_2 = \alpha^7$

$\omega^{(4)} = 1+\alpha^5 z$

$$Y_1 = \frac{X_1 + \alpha^5}{X_1 + X_2} = \frac{\alpha^4 + \alpha^5}{\alpha^4 + \alpha^7} = \frac{\alpha^8}{\alpha^3} = \alpha^5 = \zeta$$

$$Y_2 = \frac{X_2 + \alpha^5}{X_1 + X_2} = \frac{\alpha^7 + \alpha^5}{\alpha^4 + \alpha^7} = \frac{\alpha^{13}}{\alpha^3} = \alpha^{10} = \zeta^2$$

Example III:

$$(1+S) = 1 + \alpha^{11}z + \alpha^{2}z^{2} + \alpha^{8}z^{3} + \alpha^{14}z^{4} + \ldots$$

$$\sigma^{(o)} = 1, \quad \tau^{(o)} = 1, \quad \omega^{(o)} = 1, \quad \gamma^{(o)} = 0, \quad D(o) = 0, \quad B(o) = 0$$

$$\Delta_1^{(o)} = \alpha^{11}$$

$$\sigma^{(1)} = 1 + \alpha^{11}z, \quad \tau^{(1)} = \alpha^{4}, \quad \omega^{(1)} = 1, \quad \gamma^{(1)} = \alpha^{4}, \quad D(1) = 1, \quad B(1) = 1$$

$$\Delta_1^{(1)} = \alpha^{2} + \alpha^{22} = \alpha^{12}$$

$$\sigma^{(2)} = 1 + (\alpha^{11}+\alpha^{16})z = 1+\alpha^{6}z, \quad \tau^{(2)} = \alpha^{3} + \alpha^{-1}z, \quad \omega^{(2)} = 1+\alpha z, \gamma^{(2)} = \alpha^{3},$$

$$D(2) = 1, \quad B(2) = 0.$$

$$\Delta_1^{(2)} = 0, \quad \Delta_1^{(3)} = 0$$

$$\sigma^{(4)} = 1 + \alpha^{6}z, \quad \omega^{(4)} = 1 + \alpha z$$

$$X_1 = \alpha^{6}$$

$$Y_1 = \frac{\alpha^{6} + \alpha}{\alpha^{6}} = \frac{\alpha^{11}}{\alpha^{6}} = \alpha^{5} = \zeta$$

Problem:

If $(1+S) = 1 + 0z + \alpha^{14}z^{2} + \alpha z^{3} + 0z^{4} + \ldots$

show that $\sigma = 1 + \alpha^{2}z + \alpha^{4}z^{2}$, and that

$$X_1 = \alpha^{7}, \quad X_2 = \alpha^{12}, \quad Y_1 = \zeta, \quad Y_2 = 1.$$

Beloved Historical Dregs

Previous methods for calculating the σ's from the S's have relied on the direct solution of simultaneous linear equations. Instead of the key equation, $(1+S)\sigma \equiv \omega \bmod z^{2t+1}$, one has the equivalent matrix equation

$$\begin{bmatrix} 1 & 0 & 0 & 0 & & 0 \\ S_1 & 1 & 0 & 0 & & 0 \\ S_2 & S_1 & 1 & 0 & & \\ S_3 & S_2 & S_1 & 1 & & \\ \vdots & & & & & \\ S_t & \cdots & & & S_2 & S_1 & 1 \\ S_{t+1} & S_t & & & & S_1 \\ \vdots & & & & & \\ S_{2t} & & & & & S_t \end{bmatrix} \begin{bmatrix} 1 \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_t \end{bmatrix} = \begin{bmatrix} 1 \\ \omega_1 \\ \omega_2 \\ \vdots \\ \omega_t \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Given $S_1, S_2, \ldots, S_{2t}$, one might attempt to solve the above equations for $\sigma_1, \sigma_2, \ldots, \sigma_t$ and $\omega_1, \omega_2, \ldots, \omega_t$. First, one determines $\sigma_1, \sigma_2, \ldots, \sigma_t$ from the equations

$$\begin{bmatrix} S_{t+1} & S_t & \cdots & S_2 & S_1 \\ S_{t+2} & S_{t+1} & & & S_2 \\ & & & & \vdots \\ S_{2t} & & \cdots & S_{t+1} & S_t \end{bmatrix} \begin{bmatrix} 1 \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_{t-1} \\ \sigma_t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

or equivalently,

$$\begin{bmatrix} S_t & S_2 & S_1 \\ S_{t+1} & & S_2 \\ \vdots & & \\ S_{2t-1} & & S_t \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_t \end{bmatrix} = - \begin{bmatrix} S_{t+1} \\ S_{t+2} \\ \vdots \\ S_{2t} \end{bmatrix}$$

These equations are typically solved by means of a tedious Guass-Jordan reduction of the above matrix. This method requires many more computations than the generating function approach. The matrix method also requires more storage space. Furthermore, it is less elegant and harder to remember. In view of the iterative algorithm theorems, there is no longer any need to introduce these matrices at all. In short, the matrix method is now obsolete, and not used by those who think young.

Nevertheless, a considerable amount of work has been done with these matrices. Therefore, for historical reasons, we will show how the iterative algorithm relates to the matrix methods. If one attempts to solve the above equations for the n unknowns $\sigma_1, \sigma_2, \ldots, \sigma_n$, then the n x n coefficient matrix is given by

$$M_n = \begin{bmatrix} S_n & & S_2 & S_1 \\ & \diagdown & & S_2 \\ & & & \\ S_{2n-1} & & & S_n \end{bmatrix}$$

If the determinant of M_n is nonzero, then these equations have a unique solution for $\sigma_1, \sigma_2, \ldots, \sigma_n$. The traditional method for finding the σ's, given the S's, has been to select the largest n for which M_n is nonsingular, and then solve the corresponding equations.

The iterative algorithm may be used to determine which M_n's are singular and which are not, according to the following theorem:

$$\boxed{\begin{array}{l} \text{Nullity } M_n = |D(2n+1) - n| \\ \text{Rank } M_n = n - |D(2n+1) - n| \end{array}}$$

Proof: For $k = n, n+1, n+2, \ldots, 2n-1$, we may define the polynomial

$$\rho^{(k)} = \begin{cases} \tau^{(k)} & \text{iff} \quad k - D(k) \leq n-1 \\ z^{2n-k-1}\, \sigma^{(k)} & \text{iff} \quad k - D(k) \geq n \end{cases}$$

Since $k - D(k) \geq n$ iff $2n - k - 1 + D(k) \leq n-1$, we conclude that

$$\deg \rho^{(k)} \leq n-1 \quad \text{for } k = n, n+1, n+2, \ldots, 2n-1.$$

With each polynomial $\rho^{(k)}$, we associate an n-dinemsional column vector

$$\rho^{(k)} = \left[\rho_0^{(k)}, \rho_1^{(k)}, \ldots, \rho_{n-1}^{(k)} \right]^t$$

We then introduce the $n \times n$ matrix ρ by taking these $\rho^{(k)}$ for the successive columns:

$$\rho = \begin{bmatrix} \rho_0^{(n)} & \rho_0^{(n+1)} & \cdots & \rho_0^{(2n-1)} \\ \rho_1^{(n)} & \rho_1^{(n+1)} & & \rho_1^{(2n-1)} \\ \rho_2^{(n)} & \rho_2^{(n+1)} & & \rho_2^{(2n-1)} \\ \vdots & & & \\ \rho_{n-1}^{(n)} & \rho_{n-1}^{(n+1} & \cdots & \rho_{n-1}^{(2n-1)} \end{bmatrix}$$

We claim that the ρ matrix is nonsingular. For, if some linear combination of its columns were zero, then

$$\sum_{k=n}^{2n-1} c_k \underline{\rho}^k = 0$$

and

$$\sum_{k=n}^{2n-1} a_k z^{2n-k-1} \sigma^{(k)} + \sum_{k=n}^{2n-1} b_k \tau^{(k)} = 0$$

where

$$a_k = \begin{cases} c_k \text{ if } k - D(k) \geq n \\ 0 \text{ otherwise} \end{cases} \qquad b_k = \begin{cases} c_k \text{ if } k-D(k) \leq n-1 \\ 0 \text{ otherwise} \end{cases}$$

If $a_k \neq 0$ then $(1+S)\rho^{(k)} \equiv z^{2n-k-1} \omega^{(k)} \bmod z^{2n}$ and deg $(z^{2n-k-1} \omega^{(k)})$ $\leq 2n-k-1 + D(k) \leq n-1$. If $b_k \neq 0$, then $(1+S)\ \tau^{(k)} \equiv \gamma^{(k)} + z^k \bmod z^{k+1}$ and deg $\gamma^{(k)} \leq n-1$. Hence, if $b_n = b_{n+1} = \ldots = b_{n+j-1} = 0$, but $b_{n+j} \neq 0$, then

$$(1+S) \sum_{k=n}^{2n-1} c_k \underline{\rho}^{(k)} \equiv \xi + b_{n+j} z^{n+j} \bmod z^{n+j+1}$$

where deg $\xi \leq n-1$. We deduce that if $\sum_{k=n}^{2n-1} c_k \underline{\rho}^{(k)} = 0$, then $b_k = 0$ for $k = n, n+1, \ldots 2n-1$ and $\sum_{k=n}^{2n-1} a_k z^{2n-k-1} \sigma^{(k)} = 0$. But if $a_{n+j} \neq 0$ and

$a_{n+j+1} = a_{n+j+2} = \ldots = a_{2n-1} = 0$, then

$\sum_{k=n}^{2n-1} a_k z^{2n-k-1} \sigma^{(k)} \equiv a_{n+j} z^{n-j-1} \bmod z^{n-j}$. We conclude that $\sum_{k=n}^{2n-1} c_k \underline{\rho}^{(k)} =$ 0 iff every $c_k = 0$, and that the ρ matrix is nonsingular.

Since the matrix ρ is nonsingular, the rank of M_n is the same as the rank of the product matrix $P = M_n\rho$. The columns of P are $M_n\underline{\rho}^{(k)}$, $k = n, n+1, \ldots, 2n-1$. Due to the structure of the matrix M_n, the column $M_n \underline{\rho}^{(k)}$ is given by $\underline{P}^{(k)} = [P_n^{(k)}, P_{n+1}^{(k)}, \ldots, P_{2n-1}^{(k)}]^t$ where $\sum_i P_i^{(k)} z^i = (1+S)\, \rho^{(k)}$. If $\rho^{(k)} = \sigma^{(k)}$, then $k - D(k) \geq n$, $\deg(z^{2n-k-1}\sigma^{(k)}) \leq n-1$ and $(1+S)\rho^{(k)} \equiv z^{2n-k-1}\, \omega^{(k)} \bmod z^{2n}$ so that $\underline{P}^{(k)} = M_n\, \rho^{(k)} = \underline{0}$. On the other hand, if $\rho^{(k)} = \tau^{(k)}$, then $\deg \gamma^{(k)} \leq k - D(k) \leq n-1$ and $(1+S)\, \rho^{(k)} \equiv \gamma^{(k)} + z^k \bmod z^{k+1}$ so that $\underline{P}^{(k)} = M_n\underline{\rho}^{(k)} = [0,0,\ldots,0,1,\ldots\ ]^t$. (The first nonzero entry in $\underline{P}^{(k)}$ is a one in the $k - (n-1)$th coordinate.) From these arguments it follows that the P matrix is triangular, with zeroes above the main diagonal. Each entry on the main diagonal is either zero or one. If it is zero, the entire column containing it is zero. The null space of M_n is spanned by the columns $\rho^{(k)} = z^{2n-k-1}\, \sigma^{(k)}$; the range of M_n is spanned by the columns of P for which $P^{(k)} = (1+S)\, \tau^{(k)}$. The nullity of M_n is the same as the nullity of P, which is equal to the number of k's ($n \leq k \leq 2n-1$) for which $k - D(k) \geq n$, or equivalently, $D(k) \leq k - n$.

From the iterative algorithm,

$$D(k+1) = \begin{cases} D(k) & \text{if } D(k) > (k+1)/2 \text{ or if } \Delta_1^{(k)} = 0 \\ k + 1 - D(k) & \text{otherwise} \end{cases}$$

It is evident that D(k) is a monotonic nondecreasing function of k, and that $D(k) \leq k/2$ only if $D(k) = D(k-1)$.

We next claim that the set of k's for which $D(k) \leq k - n$ must occur consecutively. For if $D(k) \leq k - n < k + 1 - n \leq D(k+1)$, then $D(k+1) = k+1 - D(k) \geq n + 1$ and $D(i) = D(k+1) \geq n + 1 > i + 1 - n$ for $i = (k+1), k+2, \ldots, 2n-1$.

If $D(2n-1) = n-j$, $j > 0$, then

$$D(i) = \begin{cases} n-j \leq i-n \text{ for } i = 2n-j, \ldots, 2n-1 \\ \text{something} > i-n \text{ for } i = n, n+1, \ldots, 2n-j-1. \end{cases}$$

In this case, Nullity $M_n = j = n - D(2n-1)$.

On the other hand, if $D(2n-1) = n + j$, $j > 0$, then there exists a k such that

If $D(2n-1) = n - j$, $j > 0$, then

$$D(i) = \begin{cases} n + j > i - n \text{ for } i = k, k+1, \ldots, 2n-1 \\ k-j-n \leq i - n \text{ for } i = k-j, \ldots, k-1 \end{cases}$$

In this case, Nullity $M_n = j = D(2n-1) - n$. We conclude that

$$\boxed{\text{Nullity } M_n = \left|D(2n-1) - n\right|; \quad \text{Rank } M_n = n - \left|D(2n-1) - n\right|}$$

q.e.d.

Simplifications in the Binary Case

For binary BCH codes, the decoding procedure can be somewhat simplified. Since the only nonzero element in GF(2) is 1, every error value is 1. Thus, once the errors are located, they may be corrected immediately. Step IV of the general decoding procedure may be omitted. Additional simplifications result within the iterative algorithm as we shall now show.

Since every $Y_i = 1$, we may simplify the expression for ω:

$$\omega = \sigma + \sum_i X_i z \prod_{j \neq i} (1 - X_j z)$$

$$= \sigma + \sigma' = \sigma + \tilde{\sigma} = \hat{\sigma}$$

$$\boxed{\omega = \overset{\wedge}{\sigma}}$$

(Here we again use the superscripts "$\wedge$" and "$\smile$" to denote the even and odd parts respectively. "$\smile$" should not be confused with "$\sim$", which we have used to indicate the reciprocal function.)

If the S's are power-sum symmetric functions of any number of error locations, then $S_k = \sum_i X_i^k$, $S_{2k} = \sum_i X_i^{2k} = (\sum_i X_i^k)^2$, from which

$$\boxed{\hat{S} = S^2}\ .$$

This equation leads to a considerable simplification of the iterative algorithm. We begin with a lemma:

> Let $(1+S)$ and $(1+R)$ be reciprocal generating functions in a field of characteristic two: $(1+S)(1+R) = 1$. Then $\hat{R} = 0$ iff $\hat{S} = S^2$.

Proof: Separating $(1+S)(1+R) = 1$ into even and odd parts gives $(1+\hat{S})(1+\hat{R}) + \overset{\smile}{S}\overset{\smile}{R} = 1$ and $\overset{\smile}{S}(1+\hat{R}) + (1+\hat{S})\overset{\smile}{R} = 0$. Subtracting $\overset{\smile}{S}$ times the latter from $(1+\hat{S})$ times the former gives $((1+\hat{S})^2 - \overset{\smile}{S}^2)(1+\hat{R}) = (1+\hat{S})$, from which

$\hat{R} = \left(\dfrac{1+\hat{S}}{(1+\hat{S})^2 - \overset{\smile}{S}^2}\right) - 1$. In a field of characteristic two,

$(1+\hat{S})^2 - \overset{\smile}{S}^2 = 1 + \hat{S}^2 + \overset{\smile}{S}^2 = 1 + (\hat{S} + \overset{\smile}{S})^2 = 1 + S^2$ and $\hat{R} = 0$ iff

$1 + \hat{S} = 1 + S^2$ or $\hat{S} = S^2$

q.e.d.

We next use this lemma to prove the following:

$$\omega^{(n)} = \overset{\wedge}{\sigma}(n)$$

$$\gamma^{(n)} = \begin{cases} \overset{\smile}{\tau}(n) & \text{if } n \text{ even} \\ \overset{\wedge}{\tau}(n) & \text{if } n \text{ odd} \end{cases}$$

$$\Delta_1^{(n)} = 0 \text{ if } n \text{ odd}$$

Proof: The proof is by induction on n. The theorem is true for n = 0. We assert that if

$$\omega^{(2k)} = \overset{\wedge}{\sigma}^{(2k)} \text{ and } \gamma^{(2k)} = \overset{\smile}{\tau}^{(2k)}$$

then

$$\omega^{(2k+1)} = \overset{\wedge}{\sigma}^{(2k+1)}, \gamma^{(2k+1)} = \overset{\wedge}{\tau}^{(2k+1)}, \Delta_1^{(2k+1)} = 0,$$

and therefore,

$$\sigma^{(2k+2)} = \sigma^{(2k+1)}, \omega^{(2k+2)} = \omega^{(2k+1)}, \tau^{(2k+2)} = z\,\tau^{(2k+1)},$$

$$\gamma^{(2k+2)} = z\,\gamma^{(2k+1)}, \omega^{(2k+2)} = \overset{\wedge}{\sigma}^{(2k+2)} \text{ and } \gamma^{(2k+2)} = \overset{\smile}{\tau}^{(2k+2)}$$

The only nonobvious claim of the previous sentence is that $\Delta_1^{(2k+1)} = 0$. To prove this, we write

$$(1+S)\,\sigma^{(2k+1)} \equiv \overset{\wedge}{\sigma}^{(2k+1)} + \Delta_1^{(2k+1)}\, z^{2k+2} \bmod z^{2k+3}$$

In view of the previous lemma, multiplying this by $(1+\overset{\frown}{R})$ gives

$$\sigma^{(2k+1)} \equiv \overset{\wedge}{\sigma}^{(2k+1)} + \overset{\frown}{R}\,\overset{\wedge}{\sigma}^{(2k+1)} + \Delta_1^{(2k+1)}\, z^{2k+2} \bmod z^{2k+3}$$

$$\overset{\smile}{\sigma}^{(2k+1)} + \overset{\frown}{R}\,\overset{\wedge}{\sigma}^{(2k+1)} \equiv \Delta_1^{(2k+1)}\, z^{2k+2} \bmod z^{2k+3}$$

Since the expression on the left is odd, $\Delta_1^{(2k+1)} = 0$.

q.e.d.

We further assert that

$$\boxed{\deg \sigma^{(n)} = D(n),\ \deg \tau^{(n)} = n - D(n)}$$

Proof: Suppose $\deg \sigma^{(n)} = D(n)$ and $\deg \tau^{(n)} = n - D(n)$. Then $\deg \sigma^{(n)} = 1 + \deg \tau^{(n)}$ only if n is odd, and in this case $\Delta_1^{(n)} = 0$. It follows that $\deg \sigma^{(n)} \neq \deg \Delta_1^{(n)}\, z\, \tau^{(n)}$. According to the recursive algorithm, $\sigma^{(n+1)} = \sigma^{(n)} - \Delta_1^{(n)}\, z\, \tau^{(n)}$. Since the two terms on the right have different degrees, $\deg \sigma^{(n+1)} = \max\{\deg \sigma^{(n)}, \deg \Delta_1^{(n)}\, z\, \tau^{(n)}\} = D(n+1)$. It is easy to verify that $\deg \tau^{(n+1)} = n+1 - D(n+1)$. Since $\deg \sigma^{(0)} = \deg \tau^{(0)} = 0 = D(0)$, the theorem is true by induction on n.

q.e.d.

In view of these results, we may compute the functions $\sigma^{(0)}$, $\tau^{(0)}$, $\sigma^{(2)}$, $\tau^{(2)}$, $\sigma^{(4)}$, $\tau^{(4)}$, ..., $\sigma^{(2t)}$, $\tau^{(2t)}$ by an abbreviated iterative algorithm, in which the ω and ν polynomials and the odd-indexed σ and τ polynomials never appear explicitly:

Abbreviated Iterative Algorithm for use in fields of characteristic two when $\hat{S} = S^2$.

Initially define $\sigma^{(0)} = 1$, $\tau^{(0)} = 1$

Proceed recursively as follows:

Define $\Delta_1^{(n)}$ as the coefficient of z^{n+1} in the product $(1+S)\,\sigma^{(n)}$

Let

$$\sigma^{(2k+2)} = \sigma^{(2k)} + \Delta_1^{(2k)}\, z\, \tau^{(2k)}$$

$$\tau^{(2k+2)} = \begin{cases} z^2\, \tau^{(2k)} & \text{if } \Delta_1^{(2k)} = 0 \text{ or if } \deg \sigma^{(2k)} > k \\ \dfrac{z\, \sigma^{(2k)}}{\Delta_1^{(2k)}} & \text{if } \Delta_1^{(2k)} \neq 0 \text{ and } \deg \sigma^{(2k)} \leq k \end{cases}$$

From this algorithm, it is immediately evident that

$$\tau_0^{(2k)} = 0 \text{ if } k > 0$$

Finally, we may also simplify the expression for the general solution of the equations $\sigma(0) = 1$, $(1+S)\,\sigma \equiv \hat{\sigma} \bmod z^{2t+1}$. According to the general iterative algorithm theorem 3,

$$\sigma = U\, \sigma^{(2t)} + V\, \tau^{(2t)}$$

Multiplying by (1+S) gives

$$(1+S)\,\sigma = U(1+S)\,\sigma^{(n)} + V(1+S)\,\tau^{(2t)}$$

$$(1+S)\,\sigma = U\,\overset{\wedge}{\sigma}^{(2t)} + V\,\overset{\smile}{\tau}^{(2t)}$$

which is an even function only if

$$\boxed{U = \hat{U} \text{ and } V = \overset{\smile}{V}}$$

Previous simplifications of the binary case have been based on the matrix

$$M_n = \begin{bmatrix} 1 & 0 & 0 & \cdots & & \\ S_2 & S_1 & 1 & \cdots & & \\ S_4 & S_3 & S_2 & S_1 & 1 & \cdots \\ \vdots & & & & & \\ S_{2n-4} & & & & & S_{n-1} \\ S_{2n-2} & \cdots & & & & \end{bmatrix}$$

It is a relatively tedious, but straightforward problem to show that

$$\boxed{\text{Nullity } M_n = \left| \left[\frac{n - \deg \sigma^{(2n)}}{2} \right] \right|}$$

Here the brackets denote the greatest integer less than or equal to the quantity inside. ($[5/2] = 2$; $[-1/2] = -1$). The proof of this theorem is directly parallel to the proof of the theorem in the section on "Beloved Historical Dregs", if one begins by defining

$$\rho^{(2k)} = \begin{cases} \tau^{(2k)} & \text{iff } \deg \tau^{(2k)} \leq n-1 \\ z^{2n-2k-1}\,\sigma^{(2k)} & \text{iff } \deg \sigma^{(2k)} \leq 2k-n \end{cases}$$

for $k = 0,1,2,\ldots,n-1$. We leave the remainder of the proof as an exercise for the reader.

This concludes the simplifications of the iterative algorithm and its properties that result in the binary case. Unfortunately, no comparable simplifications are known for BCH codes over GF(q) for any $q \neq 2$. Although it is true that $S_{qk} = S_k^q$, this apparently does not result in any Δ's being automatically zero.

Powers of α where $\alpha^5 + \alpha^2 + 1 = 0$

		α^4 ↓	α^3 ↓	α^2 ↓	α ↓	1 ↓	α^4 ↓	α^3 ↓	α^2 ↓	α ↓	1 ↓	
-31	0	0	0	0	0	1	0	0	0	0	0	∞
-30	1	0	0	0	1	0	0	0	0	0	1	0
-29	2	0	0	1	0	0	0	0	0	1	0	1
-28	3	0	1	0	0	0	0	0	0	1	1	18
-27	4	1	0	0	0	0	0	0	1	0	0	2
-26	5	0	0	1	0	1	0	0	1	0	1	5
-25	6	0	1	0	1	0	0	0	1	1	0	19
-24	7	1	0	1	0	0	0	0	1	1	1	11
-23	8	0	1	1	0	1	0	1	0	0	0	3
-22	9	1	1	0	1	0	0	1	0	0	1	29
-21	10	1	0	0	0	1	0	1	0	1	0	6
-20	11	0	0	1	1	1	0	1	0	1	1	27
-19	12	0	1	1	1	0	0	1	1	0	0	20
-18	13	1	1	1	0	0	0	1	1	0	1	8
-17	14	1	1	1	0	1	0	1	1	1	0	12
-16	15	1	1	1	1	1	0	1	1	1	1	23
-15	16	1	1	0	1	1	1	0	0	0	0	4
-14	17	1	0	0	1	1	1	0	0	0	1	10
-13	18	0	0	0	1	1	1	0	0	1	0	30
-12	19	0	0	1	1	0	1	0	0	1	1	17
-11	20	0	1	1	0	0	1	0	1	0	0	7
-10	21	1	1	0	0	0	1	0	1	0	1	22
- 9	22	1	0	1	0	1	1	0	1	1	0	28
- 8	23	0	1	1	1	1	1	0	1	1	1	26
- 7	24	1	1	1	1	0	1	1	0	0	0	21
- 6	25	1	1	0	0	1	1	1	0	0	1	25
- 5	26	1	0	1	1	1	1	1	0	1	0	9
- 4	27	0	1	0	1	1	1	1	0	1	1	16
- 3	28	1	0	1	1	0	1	1	1	0	0	13
- 2	29	0	1	0	0	1	1	1	1	0	1	14
- 1	30	1	0	0	1	0	1	1	1	1	0	24
0	31	0	0	0	0	1	1	1	1	1	1	15

Example: Consider the 3-error correcting binary BCH code of block length 31, with the field represented as polynomials of degree < 5 in α, where $\alpha^5 + \alpha^2 + 1 = 0$. Log and antilog tables for this field are given on the previous page. Suppose $S_1 = 11101 = \alpha^{14}$, $S_3 = 10000 = \alpha^4$, $S_5 = 00010 = \alpha^1$.

$$(1+S) = 1 + \alpha^{14}z + \alpha^{28}z^2 + \alpha^4 z^3 + \alpha^{25}z^4 + \alpha^1 z^5 + \alpha^8 z^6 + \ldots$$

$$\sigma^{(0)} = 1 \qquad \tau^{(0)} = 1$$

$$(1+S)\,\sigma^{(0)} = 1 + \alpha^{14}z \bmod z^2$$

$$\sigma^{(2)} = 1 + \alpha^{14}z \qquad \tau^{(2)} = \alpha^{17}z$$

$$(1+S)\sigma^{(2)} \equiv 1 + (\alpha^4 + \alpha^{11})z^3 \bmod z^4$$

$$\alpha^4 = 10000$$

$$\alpha^{11} = 00111$$
$$\underline{10111} = \alpha^{26} = \alpha^{-5}$$

$$\sigma^{(4)} = 1 + \alpha^{14} + \alpha^{12}z^2 \qquad \tau^{(4)} = \alpha^5 z + \alpha^{19}z^2$$

$$(1+S)\sigma^{(4)} = 1 + \alpha^{12}z^2 + (\alpha + \alpha^8 + \alpha^{16})z^5 \bmod z^6$$

$$\alpha = 00010$$

$$\alpha^8 = 01101$$

$$\alpha^{16} = 11011$$
$$\underline{10100} = \alpha^7$$

$$\sigma^{(6)} = 1 + \alpha^{14}z + 0z^2 + \alpha^{26}z^3, \qquad \tau^{(6)} = \alpha^{24}z + \alpha^7 z^2 + \alpha^5 z^3$$

Problems: (Binary Case)

1). Show that $\sigma^{(2k)}\,\tau^{(2k)} = \text{Odd} + z^{2k}$, where Odd is an odd function of z.

2). Consider the triple error correcting binary BCH code of block length 31. Suppose $S_1 = 01010 = \alpha^6$, $S_3 = 01111 = \alpha^{23}$, $S_5 = 0001 = \alpha^0$. Show that

$$\sigma^{(6)} = 1 + \alpha^{6}z + \alpha^{21}z^{2} + \alpha^{11}z^{3}.$$

Alternate BCH Codes

In the previous sections, we have assumed that the roots of the generating polynomial of a t-error correcting BCH code must include $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t}$. Although convenient, this definition is overly restrictive. More generally, for any m, one may define a t-error correcting BCH code as the cyclic code whose generator is the product of the distinct minimum functions of $\alpha^m, \alpha^{m+1}, \ldots, \alpha^{m+2t-1}$. If $m \neq 1$, this definition gives an alternate t-error correcting BCH code. There is no loss of generality in the assumption that $m > 0$, since the case $m = 0$ is identical to the case $m = N$.

The decoding procedure for alternate t-error correcting BCH codes may be derived as follows: In general, one has the identity

$$(1+S)\,\sigma = \omega, \text{ or equivalently,}$$

$$\left(1 + \sum_{k=1}^{m-1} S_k z^k + \sum_{k=m}^{\infty} S_k z^k\right)\sigma = \omega - \sigma$$

and

$$\left(\sum_{k=m}^{\infty} S_k z^k\right)\sigma = \omega - \left(1 + \sum_{k=1}^{m-1} S_k z^k\right)\sigma$$

Since the left side is divisible by z^m, so is the right side, and we may define the polynomial

$$\xi = \frac{\omega - \left(1 + \sum_{k=1}^{m-1} S_k z^k\right)}{z^{m-1}} + \sigma$$

Clearly $\deg \xi \leq \deg \sigma$ and $\left(\sum_{k=m}^{\infty} S_k z^{1+k-m}\right)\sigma = \xi - \sigma$ or

$$\boxed{\left(1 + \sum_{k=m}^{m+2t-1} S_k z^{1+k-m}\right)\sigma \equiv \xi \bmod z^{2t+1}}$$

Step II of the decoding procedure for an alternate t-error correcting BCH code consists of the solution of this equation for the polynomials σ and ξ, using the iterative algorithm. The error evaluator is given by the formula

$$\omega = z^{m-1}\,\xi + \left(1 + \sum_{k=1}^{m-1} S_k z^k - z^{m-1}\right)\sigma$$

The usual formula for the error values, namely,

$$Y_i = \frac{\omega(x_i^{-1})}{\prod_{j \neq i} (1 - x_j x_i^{-1})}$$

now becomes

$$\boxed{Y_i = \frac{X_i^{1-m}\,\xi(X_i^{-1})}{\prod_{j \neq i} (1 - X_j X_i^{-1})} = \frac{X_i^{-m}\,\tilde{\xi}(X_i)}{\prod_{j \neq i} (X_i - X_j)}}$$

Thus, the decoding procedure for an alternate BCH code differs from the decoding procedure for the BCH code with $m = 1$ only in these minor modifications in the equations to be solved at steps II and IV.

Decoding Erasures as well as Errors

For many channels, it turns out to be wiser not to force the demodulator to make a choice between sufficiently close alternatives. The best strategy is to demodulate sufficiently weak or sufficiently ambiguous received signals not as any of the q letters in the input alphabet, but as an additional letter, "?", called an erasure. In addition to locating and correcting any errors which may be present, the decoder must then also attempt to determine the values of the symbols in the erased locations. Thus, the goal of the decoder is to correct all of the "errata", which consist of two types: erasures, whose locations are known but whose

values are unknown, and errors, whose locations and values are both unknown. It is helpful for the decoder to consider three different locator polynomials:

The erasure locator: $\Lambda = \pi (1-X_i z)$

X = erasures

The error locator: $\lambda = \pi (1-X_i z)$

X = errors

The errata locator: $\sigma = \pi (1-X_i z)$

X = errata

$$\boxed{\sigma = \Lambda \lambda}$$

The key equation, $(1+S)\sigma = \omega$, may be written as $(1+S)\ \Lambda\ \lambda = \omega$. More generally, for the alternate BCH codes discussed in the previous section, the key equation may be written as

$$(1 + \sum_{k=m}^{m+d-2} S_k z^{k+1-m})\ \Lambda\ \lambda \equiv \xi \bmod z^d$$

Here the code distance, d, is 2t+1 for the t-error-correcting code. The S's are the known weighted power-sum symmetric functions of the errata locations; Λ is the known erasure locator; λ is the unknown error locator; ξ is the unknown errata evaluator for the alternate BCH code, as discussed in the previous section.

To simplify the key equation, we combine the known S's with the known erasure locator polynomial to obtain Forney's (1964) T's. The T's are defined by the equation

$$(1 + \sum_{k=1}^{d-1} T_k z^k) = (1 + \sum_{k=m}^{m+d-2} S_k z^{k+1-m})\ \Lambda$$

The decoder must somehow find λ and ξ from the equation

$$(1 + \sum_{k=1}^{d-1} T_k z^k)\ \lambda \equiv \xi \bmod z^d$$

If there are s erasures and t errors, then deg λ = t and deg ξ = s+t.

We have

$$(1 + \sum_{k=1}^{s} T_k z^k + \sum_{k=s+1}^{d-1} T_k z^k)\ \lambda \equiv \xi \bmod z^d$$

$$(\sum_{k=s+1}^{d-1} T_k z^k)\ \lambda \equiv \xi - (1 + \sum_{k=1}^{s} T_k z^k)\ \lambda \bmod z^d$$

Since the left side is divisible by z^{s+1}, so is the right side. We define the polynomial η by the equation

$$\eta = \left(\frac{\xi - (1 + \sum_{k=1}^{s} T_k z^k)\ \lambda}{z^s} \right) + \lambda$$

Substituting this into the previous equation gives

$$(\sum_{k=s+1}^{d-1} T_k z^k)\ \lambda \equiv z^s(\eta - \lambda) \bmod z^d$$

$$(\sum_{k=s+1}^{d-1} T_k z^{k-s})\ \lambda \equiv \eta - \lambda \bmod z^d$$

$$(1 + \sum_{k=s+1}^{d-1} T_k z^{k-s})\ \lambda \equiv \eta \bmod z^{d-s}$$

If there are t errors, deg λ = t and deg $\eta \leq$ t. If t < (d-s)/2, these ... equations may be solved by the iterative algorithm.

The overall decoding procedure for correcting erasures as well as errors may be summarized as follows:

Step I: Compute the weighted power-sum symmetric functions of the errata locations, S_{m+1}, S_{m+2}, ..., S_{m+d-1}, and the erasure locator polynomial, $\Lambda = \prod_{X=\text{erasures}} (1-X_i z)$. Define $s = \deg \Lambda$.

Step II:

$$\text{Compute } (1 + \sum_{k=1}^{d-1} T_k z^k) = (1+ \sum_{k=m}^{m+d-2} S_k z^{k+1-m}) \Lambda$$

Step III: Use the iterative algorithm to find the polynomials λ and η such that

$$(1 + \sum_{k=s+1}^{d-1} T_k z^{k-s}) \lambda \equiv \eta \bmod z^{d-s}$$

λ is the error polynomial.

Step IV: Compute the errata locator,

$$\xi = (1 + \sum_{k=1}^{s} T_k z^k - z^s) \lambda + z^s \eta$$

Step V: Evaluate all errata values from the equation

$$Y_i = \frac{X_i^{-m} \tilde{\xi}(X_i)}{\prod_{j \neq i} (X_i - X_0)}$$

This procedure will correct any combination of s erasures and t errors if d is larger than s + 2t. In this sense, an error may be considered to be twice as harmful as an erasure. Heuristically, one can attribute this to the fact that there are two unknowns (a location and a value) associated with each error, but only one unknown (a value) associated with each erasure. One can not rely on this heuristic interpretation too much, however, because the criterion $s + 2t < d$ remains valid even in the binary case, where every

error has the known value 1.

DECODING MORE THAN t ERRORS

If there are no more than t errors, then the iterative algorithm will terminate with the correct error locator, $\sigma = \sigma^{(2t)}$ and the correct error evaluator, $\omega = \omega^{(2t)}$. In this case, the polynomial $\sigma^{(2t)}$ has $D(2t) = \deg \sigma^{(2t)}$ distinct reciprocal roots among the Nth roots of unity, corresponding to the error locations. The decoder will find these reciprocal roots at Step III of the decoding procedure. At step IV, the decoder will compute $Y_i = \frac{\omega(X_i)}{X_i \prod_{j \neq i} (X_i - X_j)}$. If there are no more than t errors, then each Y_i will be some nonzero element in GF(q), equal to the value of the error at location X_i.

If there are more than t errors, then almost anything can happen. It is possible (though very unlikely) that the iterative algorithm terminates with the correct error locator and the correct error evaluator, even though $D(2t) = \deg \sigma^{(2t)} > t$. It is also possible that the recursive algorithm terminates with a legitimate error locator and a legitimate error evaluator corresponding to some error pattern of weight $\leq t$. In that case the decoder completes steps III and IV of the decoding procedure without difficulty, and incorrectly "corrects" what it incorrectly believes to be the error pattern. Although wrong, the decoder cannot be blamed for such a mistake, since, on the basis of the received codeword, the error pattern which it incorrectly corrected is more probable than the actual error pattern, which has greater weight.

Far more likely than either of these events, however, are two other possibilities: failure at step III or failure at step IV. The recursive

algorithm may terminate with an illegitimate polynomial, $\sigma^{(2t)}$, which does not have deg $\sigma^{(2t)}$ distinct roots among the Nth roots of unity. In this case, step III ends in failure. Even if all roots of $\sigma^{(2t)}$ are distinct Nth roots of unity, the polynomial $\omega^{(2t)}$ may be illegitimate. Although the Y's which the decoder computes at step IV must lie in the field which contains the Nth roots of unity over GF(q), they may not lie in the subfield consisting of GF(q) itself. In the event of such a failure at step III or IV, the decoder has detected that more than t errors have occurred.

As a prelude to formulating certain procedures which can be used to correct many patterns of t+1, t+2, ..., errors in certain codes, we insert an alternate step into the decoding algorithm:

Step II 1/2: Compute the generating function $S^{(2t)}$, defined by

$$(1+S^{(2t)}) = \frac{\omega^{(2t)}}{\sigma^{(2t)}}$$

For the moment, we avoid specifying the number of coefficients of $S^{(2t)}$ which are to be computed. Since $\omega^{(2t)}$ and $\sigma^{(2t)}$ are given, it is clear that for $i > D(2t)$, the decoder may compute

$$S_i^{(2t)} = - \sum_{j+1}^{D(2t)} S_{i-j}^{(2t)} \sigma_j^{(2t)}$$

If the polynomial $\sigma^{(2t)}$ is legitimate, then $\sigma^{(2t)} = \prod_i (1-X_i z)$, where the X_i are distinct Nth roots of unity, and $\sigma^{(2t)}$ divides $1 - z^N$. In this case we may write $\sigma^{(2t)} = (1-z^N)/\xi^{(2t)}$ and $(1+S^{(2t)}) = \frac{\xi^{(2t)}\omega^{(2t)}}{1-z^N}$

$$= \xi^{(2t)}\omega^{(2t)} \sum_{j=0}^{\infty} z^{Nj}.$$

If deg $\omega^{(2t)} \leq$ deg $\sigma^{(2t)} = D(2t)$, then deg $(\xi^{(2t)}\omega^{(2t)}) \leq N$ and we have $S_{N+k}^{(2t)} = S_k^{(2t)}$ for $k = 1,2,3,\ldots$ Conversely, if $S_{N+k}^{(2t)} = S_k^{(2t)}$ for $k + 1,2,\ldots, D(2t)$, then $S_{N+k}^{(2t)} = S_k^{(2t)}$ for all k, because

$$S_{N+D(2t)+i}^{(2t)} = -\sum_{j=1}^{D(2t)} S_{N+D(2t)+i-j}^{(2t)} \sigma_j^{(2t)} = \sum_{j=1}^{D(2t)} S_{D(2t)+i-j}^{(2t)} \sigma_j^{(2t)}$$

$$= S_{D(2t)+i}^{(2t)}$$

(by induction on i). Since $S_{N+k}^{(2t)} = S_k^{(2t)}$ for all $k > 0$, $1 + S^{(2t)} = \frac{\Phi^{(2t)}}{1-z^N}$, where deg $\Phi^{(2t)} \leq N$. We then have $\frac{\omega^{(2t)}}{\sigma^{(2t)}} = \frac{\Phi^{(2t)}}{1-z^N}$, or $\sigma^{(2t)} \Phi^{(2t)} = \omega^{(2t)}(1-z^N)$. This shows that $\sigma^{(2t)}$ divides $\omega^{(2t)}(1-z^N)$. According to theorems 2 and 1a of the iterative algorithm, $\sigma^{(2t)}$ and $\omega^{(2t)}$ are relatively prime. Therefore, $\sigma^{(2t)}$ divides $(1-z^N)$. We have proved that

> The reciprocal roots of $\sigma^{(2t)}$ are distinct Nth roots of unity iff $S_{N+k}^{(2t)} = S_k^{(2t)}$ for $k = 1,2,\ldots, D(2t)$.

Thus, the success or failure of Step III may be anticipated by an inspection of certain coefficients of $S^{(2t)}$ at Step II 1/2. By further investigation of these coefficients, we may anticipate the success or failure of Step IV:

> If $X_1, X_2,\ldots,X_{D(2t)}$ are distinct and $\sigma^{(2t)} = \prod_{i=1}^{D(2t)} (1-X_i z)$, then the quantities $\frac{\omega(X_i)}{X_i \prod_{j\neq i} (X_i-X_j)} \in GF(q)$ iff $S_{qk}^{(2t)} = S_k^{(2t)}$ for $k = 1,2,\ldots, D(2t)$.

Proof: If $\sigma^{(2t)} = \prod_{i=1}^{D} (1-X_i z)$, then a partial fraction expansion of $\frac{\omega^{(2t)}}{\sigma^{(2t)}}$ gives $\frac{\omega^{(2t)}}{\sigma^{(2t)}} = \sum_i \frac{Y_i X_i z}{1-X_i z} = \sum_{k=1}^{\infty} (\sum_i Y_i X_i^k) z^k$, where $Y_i = \frac{\tilde{\omega}^{(2t)}(X_i)}{X_i \prod_{j \neq i} (X_i - X_j)}$

and $S_k^{(2t)} = \sum_{i=1}^{D} Y_i X_i^k$, for all k. In particular, $S_{kq}^{(2t)} = \sum_{i=1}^{D} Y_i X_i^{kq}$. In a field including GF(q), we may also write $(S_k^{(2t)})^q = (\sum_{i=1}^{D} Y_i X_i^k)^q =$

$(S_k^{(2t)})^q = \sum_{i=1}^{D} Y_i^q X_i^{qk}$. If $Y_i = Y_i^q$, then we have $(S_{kq}^{(2t)})^q = \sum_{i=1}^{D} Y_i X_i^{kq} =$

$\sum_{i=1}^{D} Y_i^q X_i^{kq} = (S_k^{(2t)})^q$ for all k. Conversely, if $(S_{kq}^{(2t)}) = (S_k^{(2t)})^q$, then we have both $\sum_{i=1}^{D} Y_i X_i^{qk} = (S_k^{(2t)})^q$ for k = 1,2,..., D and $\sum_{i=1}^{D} Y_i^q X_i^{qk} = (S_k^{(2t)})^q$

for k = 1,2,..., D. The unique solution of the first set of equations is given by $Y_i = \frac{\tilde{\Omega}(X_i^{-1})}{X_i \prod_{j \neq i} (X_i - X_j)}$, where Ω is determined by deg $\Omega \leq D$, $\Omega(0) = 1$,

and $(1 + \sum_{k=1}^{D} (S_k^{(2t)})^q z^k)(\prod_{i=1}^{D} (1-X_i^q z)) \equiv \Omega \bmod z^{D+1}$

This same expression furnishes the unique solution to the (identical) second set of equations, whence $Y_i^q = Y_i$ and $Y_i \in GF(q)$, for i = 1,2,...,D.

q.e.d.

These results provide a method of anticipating the success or failure of steps III and IV by computing $S^{(2t)}$ at step II 1/2. Since the success of step III cannot be assured until one computes $S_{2t+1}^{(2t)}, \ldots, S_{N+D(2t)}^{(2t)}$,

this computation may appear prohibitively time-consuming. It is true that the computation of $N + D(2t) - 2t$ successive coefficients of $S^{(2t)}$ is approximately as much work as a Chien search over $N + D(2t) - 2t$ successive Nth roots of unity, and that with only $2t - D(2t)$ more steps, step III might be completed via a Chien search. Nevertheless, there are two major advantages to Step II 1/2:

1) The vast majority of illegitimate polynomial $\sigma^{(2t)}$ and $\omega^{(2t)}$ may be detected by computing only a few coefficients of $S^{(2t)}$.

2) Sometimes corrective steps may be taken.

For example, although the decoder does not know S_{2t+1}, he often knows S_{2t+K} for various (small) values of K, due to conjugate constraints $S_{qk} = S_k^q$ and cyclic constraints $S_{k+N} = S_k$. After computing $S^{(2t)}$, the decoder then knows $\Delta_K^{(2t)} = S_{2t+K} - S_{2t+K}^{(2t)}$ for various values of K. If $\Delta_K^{(2t)} \neq 0$, then $\sigma^{(2t)}$ and $\omega^{(2t)}$ are not legitimate. However, these known $\Delta_K^{(2t)}$ may enable the decoder to compute the polynomials U and V, and thereby determine the true error polynomial from theorem 3 of the recursive algorithm:

$$\sigma = U\,\sigma^{(2t)} + V\,\tau^{(2t)}$$

In general, we define

$$\boxed{\Delta^{(2t)} = (S - S^{(2t)})/z^{2t}}$$

(Notice that the first coefficient, $\Delta_1^{(2t)}$, coincides with the $\Delta_1^{(2t)}$ defined in the iterative algorithm.)

Setting $(1+S) = \dfrac{\omega}{\sigma} = \dfrac{U\omega^{(2t)} + V\gamma^{(2t)}}{U\,\sigma^{(2t)} + V\,\tau^{(2t)}}$

and $(1 + S^{(2t)}) = \frac{\omega^{(2t)}}{\sigma^{(2t)}}$, we get

$$\Delta^{(2t)} = \frac{1}{z^n}\left(\frac{\omega}{\sigma} - \frac{\omega^{(2t)}}{\sigma^{(2t)}}\right) = \frac{-V(\omega^{(2t)}\,\tau^{(2t)} - \sigma^{(2t)}\,\gamma^{(2t)})}{z^n\,\sigma^{(2t)}\,(U\,\sigma^{(2t)} + V\,\tau^{(2t)})}$$

$$\Delta^{(2t)} = \frac{-V}{\sigma^{(2t)}(U\,\sigma^{(2t)} + V\,\tau^{(2t)})}$$

$$\Delta^{(2t)} = -V \sum_{k=0}^{\infty} (1 - \sigma^{(2t)}(U\,\sigma^{(2t)} + V\,\tau^{(2t)}))^k$$

$$\Delta_1^{(2t)} = -V_1$$

$$\Delta_2^{(2t)} = -V_2 + V_1(U_1 + \sigma_1^{(2t)} + V_1\,\tau_0^{(2t)} + \sigma_1^{(2t)})$$

$$\Delta_3^{(2t)} = -V_3 + V_2(U_1 + \sigma_1^{(2t)} + V_1\,\tau_0^{(2t)} + \sigma_1^{(2t)})$$

$$-V_1\{(U_1 + \sigma_1^{(2t)} + V_1\,\tau_0^{(2t)} + \sigma_1^{(2t)})^2$$

$$-[\sigma_2^{(2t)} + \sigma_1^{(2t)}(V_1 + \sigma_1^{(2t)} + V_1\,\tau_0^{(2t)})$$

$$+U_2 + U_1\,\sigma_1^{(2t)} + \sigma_2^{(2t)} + V_2\,\tau_0^{(2t)} + V_1\,\tau_1^{(2t)}]\}$$

In a field of characteristic p,

$$\sum_{k=0}^{\infty} \xi^k = \left(\prod_{m=0}^{\infty}(1 - \xi^{p^m})\right)^{p-1}$$

Using this identity gives

$$\Delta^{(2t)} = -V\cdot\left(\prod_{m=0}^{\infty}(\sigma^{(2t)})^{p^m}\right)^{p-1}\left(\prod_{m=0}^{\infty}(U^{p^m}(\sigma^{(2t)})^{p^m} + V^{p^m}(\tau^{(2t)})^{p^m}\right)^{p-1}$$

In the special case when $U = 1$ and $V = V_1 z$ and $p = 2$, this becomes

$$\Delta^{(2t)} = V \prod_{m=0}^{\infty} \left((\sigma^{(2t)})^2 + V \sigma^{(2t)} \tau^{(2t)} \right)^{2^m}$$

and

$$(\sigma^{(2t)})^2 + V \sigma^{(2t)} \tau^{(2t)}$$

$$= 1 + (\sigma_1^{(2t)})^2 z^2 + (\sigma_2^{(2t)})^2 z^4 + \ldots$$

$$+ V_1 z \{\tau_1^{(2t)} z + (\tau_2^{(2t)} + \tau_1^{(2t)} \sigma_1^{(2t)}) z^2 + \ldots \}$$

So

$$\Delta^{(2t)} = V_1 z + \{V_1^2 \tau_1^{(2t)} + V_1 (\sigma_1^{(2t)})^2\} z^3 + \ldots$$

Example: We continue the example of the previous section, using the 3-error correcting binary BCH code of block length 31, with the field represented as polynomials of degree < 5 in α, where $\alpha^5 + \alpha^2 + 1 = 0$. The power-sum symmetric functions of the error locations were $S_1 = 11101 = \alpha^{14}$, $S_3 = 10000 = \alpha^4$, and $S_5 = 00010 = \alpha^1$. Using the recursive algorithm as in the previous section, we find $\sigma^{(6)} = \alpha^{26} z^3 + 0\, z^2 + \alpha^{14} z + 1$ and $\tau^{(6)} = \alpha^{24} z + \alpha^7 z^2 + \alpha^5 z^3$. We have $(1+S) = 1 + \alpha^{14} z + \alpha^{28} z^2 + \alpha^4 z^3 + \alpha^{25} z^4 + \alpha^1 z^5 + \alpha^8 z^6 + \ldots$

Using the formula

$$S_i^{(6)} = \begin{cases} S_i & \text{for } i \leq 6 \\ \sum_{j=1}^{3} S_{i-j}^{(6)} \sigma_j^{(6)} = \alpha^{14} S_{i-1}^{(6)} + \alpha^{26} S_{i-3}^{(6)} & \text{for } i > 6 \end{cases}$$

$$(1+S^{(6)}) = 1 + \alpha^{14} z + \alpha^{28} z^2 + \alpha^4 z^3 + \alpha^{25} z^4 + \alpha^1 z^5 + \alpha^8 z^6 + \alpha^{25} z^7 + \alpha^{19} z^8 + \alpha^{20} z^9 + \alpha^2 z^{10} + \ldots$$

However, in view of the cyclic and conjugate constraints, it is known that

$S_9 = S_{31+9} = S_{40} = S_5^8 = \alpha^8 \neq \alpha^{20} = S_9^{(6)}$. We conclude that the polynomial $\sigma^{(6)}$ is illegitimate; it does not have three distinct reciprocal roots in $GF(2^5)$, because $\Delta_3^{(6)} = S_9 - S_9^{(6)} = \alpha^8 - \alpha^{20} = 1 \neq 0$. Corrective measures are in order. If we assume that no more than four errors occurred, then general solution $\sigma = U\,\sigma^{(6)} + V\,\tau^{(6)}$ becomes $\sigma = \sigma^{(6)} + V_1 z\,\tau^{(6)}$. (Recall that, for binary codes, U must be even and V odd, and if either U or V had degree ≥ 2, then σ would have degree > 4.) According to the calculations on the previous page,

$$\Delta_3^{(2t)} = + V_1^2\,\tau_1^{(2t)} + V_1(\sigma_1^{(2t)})^2$$

In the present example this becomes

$$1 = V_1^2\,\alpha^{24} + V_1\,\alpha^{28}$$

or

$$(V_1\,\alpha^{-4})^2 + (V_1\alpha^{-4}) = \alpha^{-1}$$

The solutions of this quadratic equation are given by

$$V_1\,\alpha^{-4} = \alpha^{11} \text{ or } \alpha^{11} + 1 = \alpha^{19}$$

$$V_1 = \alpha^{15} \text{ or } \alpha^{23}$$

If $V_1 = \alpha^{15}$, then $\sigma = 1 + \alpha^{14}\,z + \alpha^8\,z^2 + \alpha\,z^3 + \alpha^{20}\,z^4$

If $V_1 = \alpha^{23}$, then $\sigma = 1 + \alpha^{14}z + \alpha^{16}z^2 + \alpha^5 z^3 + \alpha^{-3}\,z^4$

It happens that both of these polynomials are legitimate, and each has four distinct reciprocal roots in $GF(2^5)$.

In general, if one assumes that there are t+u errors, then one can use the expressions for the generating function $\Delta^{(2t)}$ to obtain simultaneous equations for u unknown coefficients of the polynomials U and V. These equations may have no solutions, as may happen if the received word lies in a coset all of whose words have weight $> t+u$. Or, there may be several

legitimate solutions, as in the above example. Or, there may be a unique solution.

Unfortunately, the simultaneous nonlinear equations appear to be quite complicated, and little is known about the conditions for any solutions, for a unique solution, or how to go about finding the solution(s) if it (they) exist. If one assumes that there are only t+1 errors, then this extra error can be located by solving a single algebraic equation in the single unknown, V_1. However, if there are more than t+1 errors, then the situation gets very complicated very quickly.