which approaches one uniformly on $\mathscr{M}_2$ as n increases for any a.o. ρ, since $H(Q) \geq 2$.

The same procedure is used for $\mathscr{A}_2$, but the sequence is

$$\alpha(j_1)\bar{\alpha}(k_1)\alpha(j_2)\bar{\alpha}(k_2) \cdots \to j_1,k_1,j_2,k_2 \cdots$$
$$\to h(j_1,j_2),h(k_1,k_2) \cdots \to \rho(h(j_1,j_2))\rho(h(k_1,k_2)) \cdots$$

so that the 2^n integers mapped by h^n are always drawn from the same distribution (Q_0 or Q_1). Using this encoding on the rows of the array in Fig. 1, sending representations of 2^n runs of zeros and 2^n runs of ones from each row, and using the appropriate marker-moving algorithm gives an a.o. sequence of universal codes for $\mathscr{A}_k$, $\mathscr{M}$, and $\mathscr{A}$.

References

[1] L. D. Davisson, "Universal noiseless coding," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 783–795, Nov. 1973.
[2] P. Elias, "Predictive coding," *IRE Trans. Inform. Theory*, vol. IT-1, pp. 16–33, esp. pp. 30–33, Mar. 1955.
[3] ——, "The efficient construction of an unbiased random sequence," *Ann. Math. Statist.*, vol. 43, pp. 865–870, 1972.
[4] ——, "Efficient storage and retrieval by content and address of static files," *J. Ass. Comput. Mach.*, vol. 21, pp. 246–260, 1974.
[5] ——, "Minimum times and memories needed to compute the values of a function," *J. Comput. Syst. Sci.*, Oct. 1974.
[6] R. A. Flower, "Computer updating of a data structure," Research Lab. Electron., M.I.T., Cambridge, Mass., Quart. Progress Rep. 110, pp. 147–154., July 15, 1973.
[7] R. W. Floyd, "Permuting information in idealized two-level storage," in *Complexity of Computer Computations*, Miller, Thatcher and Bohlinger, Eds. New York: Plenum, 1972, pp. 105–109.
[8] S. W. Golomb, "Run-length encodings," *IEEE Trans. Inform. Theory* (Corresp.), vol. IT-12, pp. 399–401, July 1966.
[9] ——, "A class of probability distributions on the integers," *J. Number Theory*, vol. 2, pp. 189–192, 1970.
[10] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.
[11] ——, personal communication.
[12] D. E. Knuth, *The Art of Computer Programming*, vol. 3. Reading, Mass.: Addison-Wesley, 1973, esp. pp. 181–218.
[13] A. Kohavi, *Switching and Finite Automata Theory*. New York: McGraw-Hill, 1970, esp. ch. 16.
[14] M. Minsky and S. Papert, *Perceptrons*. Cambridge, Mass.: M.I.T. Press, 1969, esp. pp. 215–226.
[15] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, Ill.: University of Illinois Press, 1949, esp. p. 64.
[16] T. Welch, "Bounds on information retrieval efficiency in static file structures," M.I.T., Cambridge, Mass., MAC TR-88, Project MAC, 1971.
[17] A. D. Wyner, "An upper bound on the entropy series," *Inform. Contr.*, vol. 20, pp. 176–181, 1972.
[18] R. Karp, personal communication.

The Algebraic Decoding of Goppa Codes

N. J. PATTERSON

***Abstract*—An interesting class of linear error-correcting codes has been found by Goppa [3], [4]. This paper presents algebraic decoding algorithms for the Goppa codes. These algorithms are only a little more complex than Berlekamp's well-known algorithm for BCH codes and, in fact, make essential use of his procedure. Hence the cost of decoding a Goppa code is similar to the cost of decoding a BCH code of comparable block length.**

I. Introduction

Let K be the finite field $GF(q^m)$. Let J be the finite field $GF(q)$. Let $g(x)$ be a polynomial of degree $n \geq 1$ with coefficients in K, and let L be a subset of K with the property that no element of L is a root of g. We define a *Goppa code* $\mathscr{G}$ with *Goppa polynomial* g and symbol field J as follows. It is convenient to index the coordinates of $\mathscr{G}$ by L. Then $\boldsymbol{C}$ is a codeword of $\mathscr{G}$, if and only if

$$\sum_{\gamma \in L} \frac{C_\gamma}{x - \gamma} \equiv 0 \bmod g(x). \tag{1}$$

Manuscript received January 22, 1974; revised October 20, 1974.
The author is with the Government Communications Headquarters, Cheltenham, England.

Let $\boldsymbol{C}$ be a codeword and $\boldsymbol{R}$ the received word, so that the error vector $\boldsymbol{E}$ is given by

$$\boldsymbol{R} = \boldsymbol{C} + \boldsymbol{E}$$

so that

$$\sum_{\gamma \in L} \frac{R_\gamma}{x - \gamma} \equiv \sum_{\gamma \in L} \left(\frac{C_\gamma}{x - \gamma} + \frac{E_\gamma}{x - \gamma}\right) \bmod g(x)$$
$$\equiv \sum_{\gamma \in L} \frac{E_\gamma}{x - \gamma} \bmod g(x).$$

It is natural then to define the syndrome $S(x)$ as the polynomial of degree less than n such that

$$S(x) \equiv \sum_{\gamma \in L} \frac{R_\gamma}{x - \gamma} \bmod g(x). \tag{2}$$

We define

$$\sigma(x) = \prod_{\substack{\gamma \in L \\ E_\gamma \neq 0}} (x - \gamma)$$

(thus $\deg \sigma$ = number of errors), and we define $\eta(x)$ of degree less than n by

$$\eta(x) \equiv \sigma(x)S(x) \bmod g(x). \tag{3}$$

Now

$$\eta(x) = \sum_{\gamma \in L} \frac{E_\gamma}{x-\gamma} \prod_{\substack{\gamma \in L \\ E_\gamma \neq 0}} (x-\gamma)$$

$$= \sum_{\substack{\gamma \in L \\ E_\gamma \neq 0}} E_\gamma \prod_{\substack{\delta \in L \\ E_\delta \neq 0 \\ \delta \neq \gamma}} (x-\delta).$$

Now, if $E_\gamma \neq 0$, we get

$$\eta(\gamma) = E_\gamma \prod_{\substack{\delta \in L \\ E_\delta \neq 0 \\ \delta \neq \gamma}} (\gamma-\delta) = E_\gamma \sigma'(\gamma)$$

whence

$$E_\gamma = \frac{\eta(\gamma)}{\sigma'(\gamma)}. \tag{4}$$

So knowledge of σ and η determines $\boldsymbol{E}$. We are thus led to the following problem.

Given polynomials f and g over a finite field K with g having degree $n > 1$ and f not divisible by g, find a solution to $f\sigma \equiv \omega \bmod g$ with $\deg \sigma$ and $\deg \omega$ "small."

An algorithm to solve this problem implies a decoding algorithm for the Goppa codes.

Berlekamp [2] has given an elegant and economical solution for the case $g(x) = x^n$. We reduce our problem to this case so that we can use Berlekamp's procedure.

II. Preliminaries

Theorem 1: Let r be an integer, $0 \leq r < n$.

a) There exists a monic polynomial $\theta_r \neq 0$ such that
   i) $\deg \theta_r \leq r$
   ii) $f\theta_r \equiv \omega_r \bmod g$ and $\deg \omega_r \leq n - r - 1$.
b) If σ,η are coprime polynomials and $\deg \sigma \leq r$, $\deg \eta \leq n - r - 1$ with $f\sigma \equiv \eta \bmod g$, then σ divides θ_r.
c) Choose θ of smallest possible degree satisfying i) and ii), then θ divides θ_r.

Proof: Let $A(x) = a_0 + a_1x + \cdots + a_rx^r$ be a polynomial of degree r over $K[a_0,\cdots,a_r]$, the field K extended by $r + 1$ indeterminates $\{a_i\}_{0 \leq i \leq r}$. Requiring $\deg (A(x) \cdot f(x) \bmod g(x)) \leq n - r - 1$ imposes r linear constraints on the $r + 1$ indeterminates. Thus this system of r linear equations in $r + 1$ unknowns has a nonzero solution in K. This proves a).

Suppose $f\theta \equiv \omega \bmod g$ and $f\theta^* \equiv \omega^* \bmod g$, where θ,θ^* satisfy i) and ii). Then $\theta^*\omega \equiv \omega^*\theta \bmod g$. As $\deg \theta^*\omega$ and $\deg \omega^*\theta$ are less than $\deg g$, we get

$$\theta^*\omega = \omega^*\theta. \tag{5}$$

If θ^*,ω^* are coprime then θ^* divides θ proving b).

Now suppose c) is false. Choose g and r so that $\deg g$ is as small as possible while contravening c). Let $\theta \neq 0$ be of smallest degree satisfying i) and ii). Let $f\theta \equiv \omega \bmod g$. Since c) is false, there exists ψ so that $f\psi \equiv \phi \bmod g$; $\deg \theta$, $\deg \psi \leq r$; $\deg \omega$, $\deg \phi \leq n - r - 1$; while θ does not divide ψ. We choose ψ of smallest possible degree, subject to our choice of θ and g.

By b), θ and ω are not coprime. Let $p(x)$ be an irreducible polynomial dividing θ and ω. By minimality of $\deg \theta$ we get $p \mid g$. Suppose $p \mid \psi$, then $p \mid \theta$. Now θ/p is a polynomial λ of least degree satisfying

i) $\deg \lambda \leq r - \deg p$

ii) $\deg f\lambda \bmod g/p \leq \deg (g/p) - r - 1$.

So by our choice of g we get $\theta/p \mid \psi/p$ whence $\theta \mid \psi$. So we can assume that $\gcd(\theta,\omega) = \alpha$, say, is coprime to ψ. Let $\gcd(\alpha,g) = \alpha_0$, then $\alpha = \alpha_0\alpha_1$. Let $f\theta = \omega + \mu g$. We get $\alpha_1 \mid \mu$, and our choice of θ implies $\alpha_1 = 1$, whence $\alpha \mid g$.

Let $\theta = \alpha\theta'$, $\omega = \alpha\omega'$, and $g = \alpha g'$. By (5) $\theta\phi = \psi\omega$, or $\theta'\phi = \omega'\psi$ whence $\theta' \mid \psi$. Set $\psi = \beta\theta'$, hence, $\phi = \beta\omega'$. α and β are coprime. Now let $\beta_0 = \gcd(\beta,g)$, $\beta = \beta_0\beta_1$. Then $f\beta_0\beta_1\theta' = \beta_0\beta_1\omega' + \lambda g$, for some polynomial λ. Hence $f\beta_0\theta' \equiv \beta_0\omega' \bmod g$. $\alpha \nmid \beta_0$ so since ψ was chosen of smallest degree subject to $\theta \nmid \psi$, we get $\beta_0 = \beta$ or $\beta \mid g$. Let $g = \alpha\beta g''$. Now $f\theta' - \omega' \equiv 0 \bmod \beta g''$ and $\bmod\ \alpha g''$. Hence $f\theta' - \omega' \equiv 0 \bmod \alpha\beta g''$, or $f\theta' \equiv \omega' \bmod g$. This contradicts the choice of θ and proves c), completing the proof of Theorem 1.

The reader should not assume that the minimal θ shown to exist by c) has the property that $f\theta \bmod g$ is coprime to θ. Take for example $g(x) = x^4$, $f(x) = 1 + x^3$, $r = 1$. Then $\theta = x$, $\omega = x$ is obviously the choice we need for c).

The polynomials given by Theorem 1 can in fact be found by Euclid's algorithm [5].

We now take $r = [n/2]$, so $n - r - 1 = [(n-1)/2]$, and wish to compute θ_r satisfying i) and ii). By [1] this is the crucial step in giving an l-error correcting algorithm for a Goppa code with Goppa polynomial of degree $2l$.

III. The Case $g(x) = x^n$

In essence we use Berlekamp's well-known algorithm for decoding BCH codes [2, sec. 7.4]. Berlekamp in [2] assumes $f(0) = 1$, which suffices for his purposes; we cannot make this assumption, but the algorithm needs only minor changes. It seems worthwhile to give an exposition here, following Berlekamp very closely.

Let

$$f(x) = \sum_{i=0}^{n-1} a_ix^i.$$

Without loss of generality we take the first nonzero coefficient of f to be one. That is, $a_i = 0$, $0 \leq i \leq k - 1$, $a_k = 1$.

Algorithm 1 (Berlekamp): If $a_0 = 1$, define $\sigma_0 = 1$, $\tau_0 = 1$, $\omega_0 = 1$, $\gamma_0 = 0$, $D(0) = 0$, $B(0) = 0$. If $a_0 = 0$, define $\sigma_0 = 1$, $\tau_0 = 1$, $\omega_0 = 0$, $\gamma_0 = -1$, $D(0) = 0$, $B(0) = 1$. Thereafter proceed recursively, for $0 \leq k \leq n - 2$. Define Δ_k to be the coefficient of x^{k+1} in $f\sigma_k$. Set

$$\sigma_{k+1} = \sigma_k - \Delta_k x\tau_k.$$

$$\omega_{k+1} = \omega_k - \Delta_k x\gamma_k.$$

If $\Delta_k = 0$, or if $D(k) > (k + 1)/2$, or if both $D(k) = (k + 1)/2$ and $B(k) = 0$, set

$$D(k + 1) = D(k)$$

$$B(k + 1) = B(k)$$

and

$$\tau_{k+1} = x\tau_k$$
$$\gamma_{k+1} = x\gamma_k;$$

otherwise, set

$$D(k+1) = k+1-D(k)$$
$$B(k+1) = 1-B(k)$$
$$\tau_{k+1} = \frac{\sigma_k}{\Delta_k}$$
$$\gamma_{k+1} = \frac{\omega_k}{\Delta_k}.$$

Theorem 2 (Berlekamp [2]):

a) $\sigma_k(0) = 1$
b) $f\sigma_k \equiv \omega_k + \Delta_k x^{k+1} \bmod x^{k+2}$
c) $f\tau_k \equiv \gamma_k + x^k \bmod x^{k+1}$
d) $\deg \sigma_k \leq D(k)$
e) $\deg \tau_k \leq k - D(k)$
f) $\deg \omega_k \leq D(k) - B(k)$
g) $\deg \gamma_k \leq k - D(k) - (1 - B(k))$
h) $\omega_k \tau_k - \sigma_k \gamma_k = x^k$.

Proof: a)–g) are all readily proved by induction on k, noting that the initial conditions are chosen to make the theorem true at $k = 0$. To prove h), from b) and c) we find

$$\omega_k\tau_k - \sigma_k\gamma_k \equiv \sigma_k x^k \bmod x^{k+1}$$
$$\equiv x^k \bmod x^{k+1}$$

where the last congruence follows from a). Now $\deg \omega_k\tau_k \leq k$ by e), f); $\deg \sigma_k\gamma_k \leq k$ by d), g). Hence the result.

Theorem 3 (Berlekamp [2, sec. 7.43]): Let σ,ω be any pair of polynomials that satisfy $\sigma(0) = 1, f\sigma \equiv \omega \bmod x^{k+1}$. Let $D = \max(\deg \sigma, \deg \omega)$. Then there exist polynomials U and V such that

1) $U(0) = 1$
2) $V(0) = 0$
3) $\deg U \leq D - D(k)$
4) $\deg V \leq D - (k - D(k))$
5) $\sigma = U\sigma_k + V\tau_k$
6) $\omega = U\omega_k + V\gamma_k$.

Proof: $f\sigma \equiv \omega \bmod x^{k+1}$; $f\sigma_k \equiv \omega_k \bmod x^{k+1}$ by Theorem 2 b). So $\omega\sigma_k \equiv \omega_k\sigma \bmod x^{k+1}$ or

$$\sigma_k\omega - \omega_k\sigma = -x^kV(x), \qquad \text{where } V(0) = 0 \tag{6}$$

and $\deg V \leq D + \max(\deg \sigma_k, \deg \omega_k) - k \leq D + D(k) - k$ by Theorem 2. Similarly, $\tau_k\omega \equiv \sigma(\gamma_k + x^k) \bmod x^{k+1}$ or

$$\tau_k\omega - \gamma_k\sigma = x^kU(x), \qquad \text{where } U(0) = 1 \tag{7}$$

and $\deg U \leq D + \max(\deg \tau_k, \deg \gamma_k) - k \leq D - D(k)$ by Theorem 2. By (6), (7) $\sigma(\tau_k\omega_k - \sigma_k\gamma_k) = x^k(U\sigma_k + V\tau_k)$. By Theorem 2 h),

$$\sigma = U\sigma_k + V\tau_k. \tag{8}$$

Similarly

$$\omega = U\omega_k + V\gamma_k. \tag{9}$$

This completes the proof of Theorem 3.

Theorem 4 (Berlekamp [2]): Suppose σ,ω are coprime polynomials and $\sigma(0) = 1, f\sigma \equiv \omega \bmod x^k$, $\deg \sigma \leq [k/2]$, and $\deg \omega \leq [(k-1)/2]$. Then $\sigma = \sigma_{k-1}$, $\omega = \omega_{k-1}$.

Proof: We use Theorem 3. By (7) we find

$$\begin{aligned} k - 1 &\leq \deg(\tau_{k-1}\omega - \sigma\gamma_{k-1}) \\ &\leq \max\left(\frac{k-1}{2} + k - 1 - D(k-1),\right. \\ &\qquad\qquad \left.\frac{k}{2} + k - 1 - D(k-1)\right) \\ &= k - 1 + \frac{k}{2} - D(k-1). \end{aligned}$$

Therefore

$$D(k-1) \leq \frac{k}{2}. \tag{10}$$

Also $D(k-1) = k/2$ implies $\deg \gamma_{k-1} = k - 1 - D(k-1)$ whence $B(k-1) = 1$ by Theorem 2 g). Thus we obtain $\deg \sigma_{k-1} \leq k/2$ and $\deg \omega_{k-1} \leq (k-1)/2$. Now it follows that $\deg(\sigma_{k-1}\omega - \omega_{k-1}\sigma) < k$, and, using (6), that $V = 0$. So $\sigma = U\sigma_{k-1}$, $\omega = U\omega_{k-1}$. As σ,ω are coprime, this proves Theorem 4.

Theorem 5 (Berlekamp [2]):

a) if $B(k) = 0$, then $\deg \tau_k = k - D(k)$
$\deg \omega_k = D(k)$;
b) if $B(k) = 1$, then $\deg \gamma_k = k - D(k)$
$\deg \sigma_k = D(k)$.

Proof: By Theorem 2 h) $\omega_k\tau_k - \sigma_k\gamma_k = x^k$. If $B(k) = 0$, then $\deg \sigma_k\gamma_k \leq k - 1$ so $\deg \omega_k\tau_k = k$. Now, using Theorem 2 e), f), we get part a). If $B(k) = 1$, then $\deg \tau_k\omega_k \leq k - 1$ so $\deg \sigma_k\gamma_k = k$. Now, using Theorem 2 d), g), we get part b).

Theorem 6: Let σ be a polynomial of least degree such that $f\sigma \equiv \omega \bmod x^n$, where $\deg \sigma \leq n/2$, $\deg \omega \leq (n-1)/2$. Suppose $\sigma = x^a\sigma^*$, where $\sigma^*(0) = 1$. So $\omega = x^a\omega^*$. Then

1) $\sigma^* = \sigma_{n-a-1}$;
2) $\omega^* = \omega_{n-a-1}$;
3) if $B(n-a-1) = 1$, then $D(n-a-1) \leq [n/2] - a$, while if $B(n-a-1) = 0$, then $D(n-a-1) \leq [(n-1)/2] - a$;
4) if $a > 0$, then $\sigma_{n-a-1} \neq \sigma_{n-a}$ and $f\sigma \not\equiv \omega \bmod x^{n+1}$.

Proof: Clearly σ^* and ω^* are coprime and

$$f\sigma^* \equiv \omega^* \bmod x^{n-a}$$

and

$$\deg \sigma^* \leq \left[\frac{n}{2}\right] - a \leq \frac{n-a}{2}$$
$$\deg \omega^* \leq \left[\frac{n-1}{2}\right] - a \leq \frac{n-a-1}{2}. \tag{11}$$

By Theorem 4, $\sigma^* = \sigma_{n-a-1}$ and $\omega^* = \omega_{n-a-1}$, proving 1) and 2). 3) follows at once from (11) and Theorem 5. Finally, if $a > 0$, then $fx^{a-1}\sigma^* \not\equiv x^{a-1}\omega^* \bmod x^n$, or

$f\sigma^* \not\equiv \omega^* \bmod x^{n-a+1}$. This shows $\sigma_{n-a-1} \neq \sigma_{n-a}$, proving 4).

This yields an algorithm to determine $\sigma \neq 0$ of least degree such that $f\sigma \equiv \omega \bmod x^n$, $\deg \sigma \leq [n/2]$, and $\deg \omega \leq [(n-1)/2]$. σ is unique (up to multiplication by a field element) by Theorem 1. We suppose $\sigma = x^a\sigma^*$, where $\sigma^*(0) = 1$.

Let $N = \max(\deg \sigma, (\deg \omega) + 1)$.

Algorithm 2: We proceed exactly as in Algorithm 1, except that at each iteration, if $\Delta_k \neq 0$ and

$$\left(\left(\text{if } B(k-1) = 0 \text{ and } D(k-1) \leq k - \left[\frac{n}{2}\right] - 1\right) \quad \text{or} \quad \left(\text{if } B(k-1) = 1 \text{ and } D(k-1) \leq k - \left[\frac{n-1}{2}\right] - 1\right)\right)$$

then set $\tilde{\sigma} = x^{n-k}\sigma_{k-1}$, $\tilde{\omega} = x^{n-k}\omega_{k-1}$, $\tilde{N} = n - k + D(k-1) + 1 - B(k-1)$, and terminate. If we compute $\sigma_0,\sigma_1,\cdots,\sigma_{n-1}$ without terminating, then set $\tilde{\sigma} = \sigma_{n-1}$, $\tilde{\omega} = \omega_{n-1}$, $\tilde{N} = D(n-1) + 1 - B(n-1)$.

Theorem 7:

1) $\sigma = \tilde{\sigma}$
2) $\omega = \tilde{\omega}$
3) $\tilde{N} = \max(\deg \sigma, (\deg \omega) + 1)$.

Proof: Suppose our algorithm has computed $\sigma_0, \sigma_1,\cdots,\sigma_{n-1}$ without terminating. Then by Theorem 6 (setting $a = n - k$), we find $\sigma = \sigma_{n-1} = \tilde{\sigma}$ and 1), 2) follow. So we may assume $\tilde{\sigma} = x^{n-k}\sigma_{k-1}$, for some $k < n$. Then $\deg \sigma_{k-1} \leq [k/2]$, $\deg \omega_{k-1} \leq [(k-1)/2]$. Hence by Theorem 4, σ_{k-1} and ω_{k-1} are coprime. By Theorem 1 $\sigma \mid \tilde{\sigma}$. Let $\tilde{\sigma}(x) = \sigma(x)\psi(x)$, $\tilde{\omega}(x) = \omega(x)\psi(x)$, hence, $\psi(x) = x^c$, for some $c \leq n - k$. $c \geq 1$ implies

$$\frac{f(x)\sigma(x)\psi(x)}{x} \equiv \frac{\omega(x)\psi(x)}{x} \bmod x^n$$

or $f(x)\tilde{\sigma}(x) \equiv \tilde{\omega}(x) \bmod x^{n+1}$, whence

$$f(x)\sigma_{k-1}(x) \equiv \omega_{k-1}(x) \bmod x^{k+1}.$$

This implies $\Delta_k = 0$, contradicting the algorithm. So $c = 0$, whence $\sigma = \tilde{\sigma}$. This proves 1) and 2). By Theorem 5, $\max(\deg \sigma_r, (\deg \omega_r) + 1) = D(r) + 1 - B(r)$; 3) is an immediate consequence.

IV. General g

We now make no assumptions on g.

We wish to solve the equation $f(x)\sigma(x) \equiv \omega(x) \bmod g(x)$. The idea is to load f into a feedback shift register wired to multiply by $x \bmod g(x)$. We next compute a polynomial f' whose coefficients are given by the successive values of a particular cell of the register. Algorithm 2 is now used to solve $f'(y)\sigma'(y) \equiv \omega'(y) \bmod y^n$. It then turns out that for a suitable choice of i, $\sigma(x) = x^i\sigma'(x^{-1})$ gives us the answer we require.

Here then is an algorithm to compute the σ of least degree such that $f\sigma \equiv \omega \bmod g$, $\deg \sigma \leq [n/2]$, $\deg \omega \leq [(n-1)/2]$.

Algorithm 3:

1) For $0 \leq i \leq n-1$, let a_i be the coefficient of x^{n-1} in $x^if(x) \bmod g$.

2) Let $h(y) = a_0 + a_1y + \cdots + a_{n-1}y^{n-1}$.

Case A—n even: Use Algorithm 2 to find σ',ω',N such that $\deg \sigma' \leq n/2$, $\deg \omega' \leq [(n-1)/2] = n/2 - 1$, and $h(y)\sigma'(y) \equiv \omega'(y) \bmod y^n$, where

$$N = \max(\deg \sigma', (\deg \omega') + 1).$$

Case B—n odd: Use Algorithm 2 to find σ',ω',N such that $\deg \sigma' \leq (n-1)/2$, $\deg \omega' \leq [(n-2)/2] = (n-1)/2 - 1$, and $h(y)\sigma'(y) \equiv \omega'(y) \bmod y^{n-1}$, where

$$N = \max(\deg \sigma', (\deg \omega') + 1).$$

3) Suppose $\sigma' = c_0 + c_1y + \cdots + c_ry^r$, $r \leq N$. Then set $\sigma = c_0x^N + \cdots + c_rx^{N-r}$ and $\omega = f\sigma \bmod g$.

Proof of Algorithm 3: It is convenient here to introduce the ring $R[x]$ of all formal power series $\sum_{i=-\infty}^{\infty} a_ix^i$ (where a_i are coefficients in our field K) in which $a_i = 0$, for every $i > 0$ with at most a finite number of exceptions. Addition and multiplication in $R[x]$ are defined in the obvious way. Observe that $K[x]$, the ring of polynomials in x, is embedded in $R[x]$ in a natural manner.

Theorem 8: Let $\alpha, \beta \in R[x]$ be polynomials in x. Let $\deg \beta = n$. Let $\alpha = q\beta + r$, where r is a polynomial of degree $<n$. Suppose $r \neq 0$. Let

$$\alpha(x) = \left(\tilde{q}(x) + \sum_{i=1}^{\infty} b_ix^{-i}\right)\beta(x)$$

where $\tilde{q}(x)$ is a polynomial in x. If $b_1 = b_2 = \cdots = b_s = 0$ and $b_{s+1} \neq 0$, then $\deg r = n - s - 1$.

Proof: For $a \geq 0$, let $x^a\alpha(x) = q_a(x)\beta(x) + r_a(x)$, where $\deg r_a < n$. Then

$$x^a\alpha(x) = \left(\tilde{q}(x)x^a + \sum_{j=1}^{a} b_jx^{a-j} + \sum_{i=1}^{\infty} b_{i+a}x^{-i}\right)\beta(x).$$

Let

$$q^*(x) = \tilde{q}(x)x^a + \sum_{j=1}^{a} b_jx^{a-j}$$

for $a \geq 1$. q^* is a polynomial in x. Then

$$(q_a - q^*)\beta + \left(r_a - \beta\sum_{i=1}^{\infty} b_{i+a}x^{-i}\right) = 0.$$

Now comparing coefficients of x^k for $k \geq n$ we get $q_a - q^* = 0$, or $r_a = \beta\sum_{i=1}^{\infty} b_{i+a}x^{-i}$. So comparing the coefficient of x^{n-1}, we get that b_{a+1} is the coefficient of x^{n-1} in $r_a = x^a\alpha \bmod \beta$. Theorem 8 is now obvious.

We return to Algorithm 3

$$f(x) = ((a_0x^{-1} + a_1x^{-2} + \cdots + a_{n-1}x^{-n}) + b_nx^{-(n+1)} + \cdots)g(x)$$

where $b_n,b_{n+1},\cdots$ are elements of K. Substituting x^{-1} for y in 2) of Algorithm 3 yields

$$(a_0 + a_1x^{-1} + \cdots + a_{n-1}x^{-(n-1)})\sigma'(x^{-1}) \equiv \omega'(x^{-1}) \bmod x^{-M}$$

where $M = n$ if n is even, $M = n - 1$ if n is odd. Hence

$$\begin{aligned} f(x)\sigma'(x^{-1}) &= (x^{-1}\sigma'(x^{-1})(a_0 + a_1x^{-1} + \cdots \\ &\quad + a_{n-1}x^{-(n-1)}) + b_nx^{-(n+1)} + \cdots)g(x) \\ &= (x^{-1}\omega'(x^{-1}) \\ &\quad + x^{-(M+1)}(d_0 + d_1x^{-1} + \cdots))g(x) \end{aligned}$$

where the d_i are some elements of K whose value is not important. Multiplying by x^N we get, setting $\sigma(x) = x^N\sigma'(x^{-1})$ and noting that $N = \max(\deg \sigma', (\deg \omega') + 1)$,

$$f(x)\sigma(x) = (\lambda(x) + x^{-(M+1-N)}(d_0 + d_1x^{-1} + \cdots))g(x)$$

where $\lambda(x)$ is a polynomial in x. Hence by Theorem 8, $\omega = f\sigma \bmod g$ has degree $\leq n - 1 - (M - N) = N - 1$, if n is even and N, if n is odd. Hence we get $\deg \sigma \leq N$, and $\deg \omega \leq N - 1$, if n is even and $\deg \omega \leq N$, if n is odd. By Algorithm 3, $N \leq n/2$, if n is even and $N \leq (n - 1)/2$, if n is odd. Hence $\deg \sigma \leq [n/2]$, $\deg \omega \leq [(n - 1)/2]$, as asserted.

Reversing the steps of the preceding proof, it may be shown that our algorithm yields the unique polynomial (up to multiplication by a field element) with the required properties.

Remarks:

1) Implementation of this algorithm is hardly more difficult than for a BCH decoder using Berlekamp's algorithm.

2) Berlekamp's Algorithm 1 requires $a_0, a_1, \cdots$ sequentially, so if convenient each a_i can be fed immediately upon calculation to a decoder carrying out Algorithm 1.

3) In practice there is no need to compute the ω_i and γ_i of Algorithm 1.

Algorithm 3 together with [1] gives a simple algebraic t-error-correcting procedure for a Goppa code with Goppa polynomial of degree $2t$.

V. Binary Goppa Codes

Let $K = GF(2^m)$. In this case, the key equation for the Goppa code over $GF(2)$ with location field K and Goppa polynomial g becomes

$$f\sigma \equiv \sigma' \bmod g \tag{12}$$

where σ' is the formal derivative of σ. For simplicity, we assume g is irreducible. (12) has a unique solution with $\deg \sigma < \deg g$ and σ, σ' coprime (or σ square free).

Now $\sigma = \alpha^2 + x\beta^2$, where $\deg \alpha \leq (n - 1)/2$, $\deg \beta \leq n/2$. Since g is irreducible, f is coprime to g, whence there exists h such that $f(x)h(x) = 1 \bmod g(x)$. So $f(\alpha^2 + x\beta^2) \equiv \beta^2 \bmod g$ whence

$$(h + x)\beta^2 \equiv \alpha^2 \bmod g. \tag{13}$$

If $h(x) = x$, then $fx \equiv 1 \bmod g$, whence $\sigma = x$ is the solution. Otherwise, there exists a unique nonzero polynomial (mod g), d say, such that $d^2 = (h + x) \bmod g$. Now from (13), $d^2\beta^2 \equiv \alpha^2 \bmod g$. So $d\beta \equiv \alpha \bmod g$. This gives us an algorithm for σ.

Algorithm 4:

1) Find h such that $fh = 1 \bmod g$ (see, for example, [2, sec. 2.3]). If $h(x) = x$, set $\sigma = x$ and terminate.

2) Calculate d such that $d^2 = (h + x) \bmod g$. (Note that $d \to d^2 \bmod g$ is a linear transformation, T say. If we are going to carry out this procedure many times, it is perhaps best to store T^{-1} in matrix form since $d = T^{-1}(h + x)$.)

3) Using Algorithm 3, find α and β with β of least degree such that $d\beta \equiv \alpha \bmod g$ with $\deg \beta \leq n/2$, $\deg \alpha \leq (n - 1)/2$.

4) Set $\sigma = x\beta^2 + \alpha^2$.

Algorithm 4, with [1], yields a t-error-correcting algorithm for the binary Goppa code with Goppa polynomial of degree t.

Acknowledgment

I should like to thank J. C. Cock and Dr. G. H. Toulmin for very helpful suggestions.

References

[1] E. R. Berlekamp, "Goppa codes," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 590–592, Sept. 1973.
[2] ——, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.
[3] V. D. Goppa, "A new class of linear error-correcting codes," *Probl. Peredach. Inform.*, vol. 6, pp. 24–30, Sept. 1970.
[4] ——, "Rational representation of codes and (L,g) codes," *Probl. Peredach. Inform.*, vol. 7, pp. 41–49, Sept. 1971.
[5] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A method for solving the key equation for decoding Goppa codes," presented at the IEEE Int. Symp. Information Theory, Notre Dame, Ind., Oct. 27–31, 1974.